

The Analog Simulators

1. Introduction

Today we are witnessing an increasing spread of digital equipments in our lives. The word "microprocessor" has entered the vocabulary of ordinary people for several years, while in the 70's it was known only by a handful of initiates. Since the discovery of the transistor (1947), analog electronics have lost ground to this new digital world that claims to be better. A question naturally arises: what else is the analog good at the moment? The answer is not difficult to give, if we look carefully and realize that the world around us is not only black and white, 0 and 1, but it contains a lot of perfumes, colors, mysteries, life and death. These things are impossible to be represented with completely digital equipment. Here the analog could help, indeed, to a limited extent (and this is because nature is much more complex than any machinery invented by man). Because it does not work with discrete values, analog is closer to nature than digital, and therefore can be used as an interface between the surrounding world and a digital system.

With the discovery of semiconductor devices, microelectronic systems have become more and more complex over time. In order to verify the functionality of a more complex electronic circuit, most of the time it had to be realized physically, and then some measurements of its parameters were performed. All these things took time, in a fast-paced world. A tool was needed to make the designers' work easier and to give a quick answer to their questions. After the first microprocessor was produced in 1971, in **1973**, Laurence Nagel (from the Electronics Research Laboratory of the University of California, Berkeley) with direction from his scientific advisor professor Donald Pederson, wrote in FORTRAN programming language the first analog simulator. He named it **S P I C E 1** ("**Simulation Program with Integrated Circuit Emphasis**"). Since then, a lot of analog simulation programs have been written. Among them, it is worth mentioning:

- SPICE2(1975, Fortran language, Berkeley) with improved features. It was able to do the following analysis:
 - o AC analysis (linear small-signal frequency domain analysis)
 - o DC analysis (nonlinear quiescent point calculation)
 - o DC transfer curve analysis (a sequence of nonlinear operating points calculated while sweeping an input voltage or current, or a circuit parameter)
 - o Noise analysis (a small signal analysis done using an adjoint matrix technique which sums uncorrelated noise currents at a chosen output point)
 - o Transfer function analysis (a small-signal input/output gain and impedance calculation)
 - o Transient analysis (time-domain large-signal solution of nonlinear differential algebraic equations)
- SPICE3(1989, C language, Thomas Quarles, Berkeley) with added more sophisticated MOSFET (Metal-Oxide-Semiconductor Field Effect Transistor) models, which were needed because new devices were produced by the industry.
- XSPICE (Georgia Institute of Technology, open-source project)
- Ngspice (open-source project), based on SPICE3
- Multisim (National Instruments), based on SPICE3
- Oregano (Richard Hult & others), based on SPICE, Ngspice and Gnuccaps engines
- Proteus (LabCenter)
- LTSpice (Linear Technology Corporation, now part of Analog Devices, Inc.)
- TINA (Texas Instruments, Inc.)
- Pspice (Cadence)
- Tanner T-Spice, Analog FastSPICE (AFS) and Eldo platforms from Mentor (now part of Siemens company).

It should be mentioned that there is no perfect analog simulator. There may be situations when a circuit simulation works, but when the same circuit is physically manufactured, it may not work as in the simulation or it may not work at all. The reciprocal of the previous sentence can also be true, sometimes.

Depending on the destination intended by the designer, analog simulators can be classified in:

- a. Home-made simulators (for a specific device or a class of devices). These simulators are often written by researchers or teachers for testing their research or for educational purposes;
- b. Free simulators;
- c. paid simulators;
- d. paid simulators with limited free options.

2. The lab analog simulator

In the laboratory that takes place this semester, we will first learn to use the [PSpice Student simulator version 9.1](#) (released in 2000), offered free of charge by Cadence company (with some limited options). This program was written for Windows OS and it is a mixed signal simulator (it contains both analog and digital parts). During the software installation of the program, both *Capture* and *Schematics* editors should be selected (Fig.1).

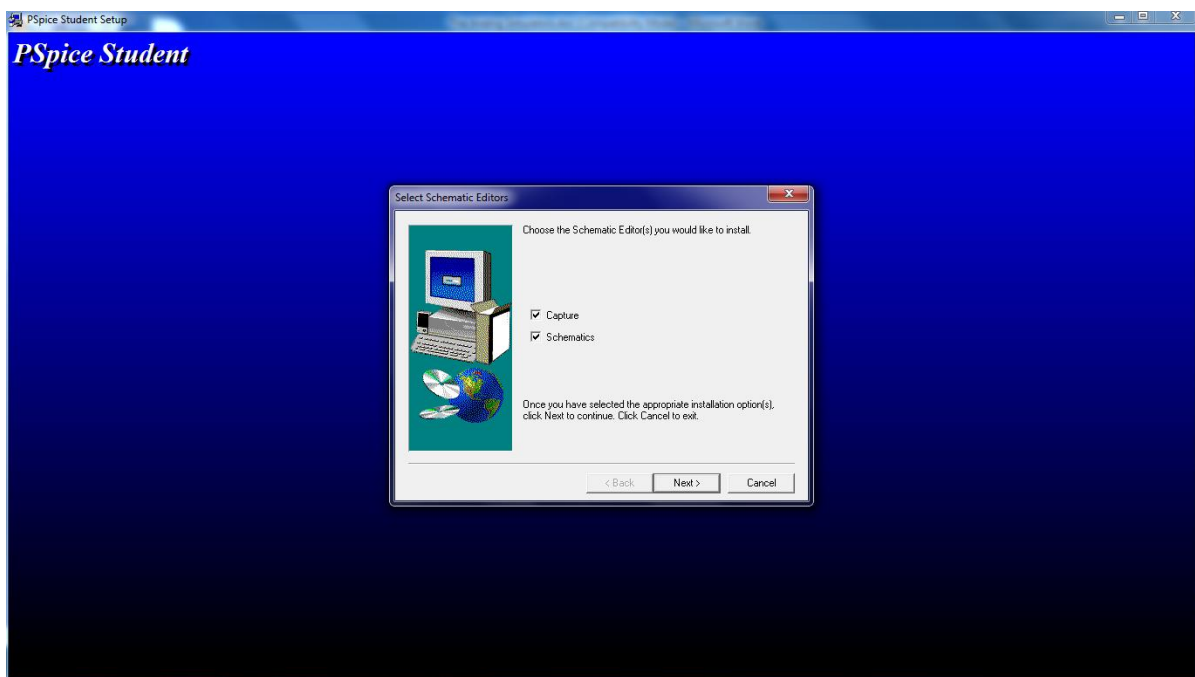


Fig.1. The selection of schematic editors during the setup of PSpice Student 9.1

After the installation has finished, the Schematics application must be run (Fig.2). This interface allows drawing a circuit on size A sheet (11" by 8.5"). The maximum number of parts which may be placed is limited to 50. Circuit simulation with PSpice Student 9.1 is limited to circuits with up to:

- 64 nodes;
- 10 transistors;
- 65 digital primitive devices;
- 10 transmission lines in total (ideal or non-ideal).
- 4 pairwise coupled transmission lines.

There is also a sample library which includes 39 analog parts and 134 digital parts.

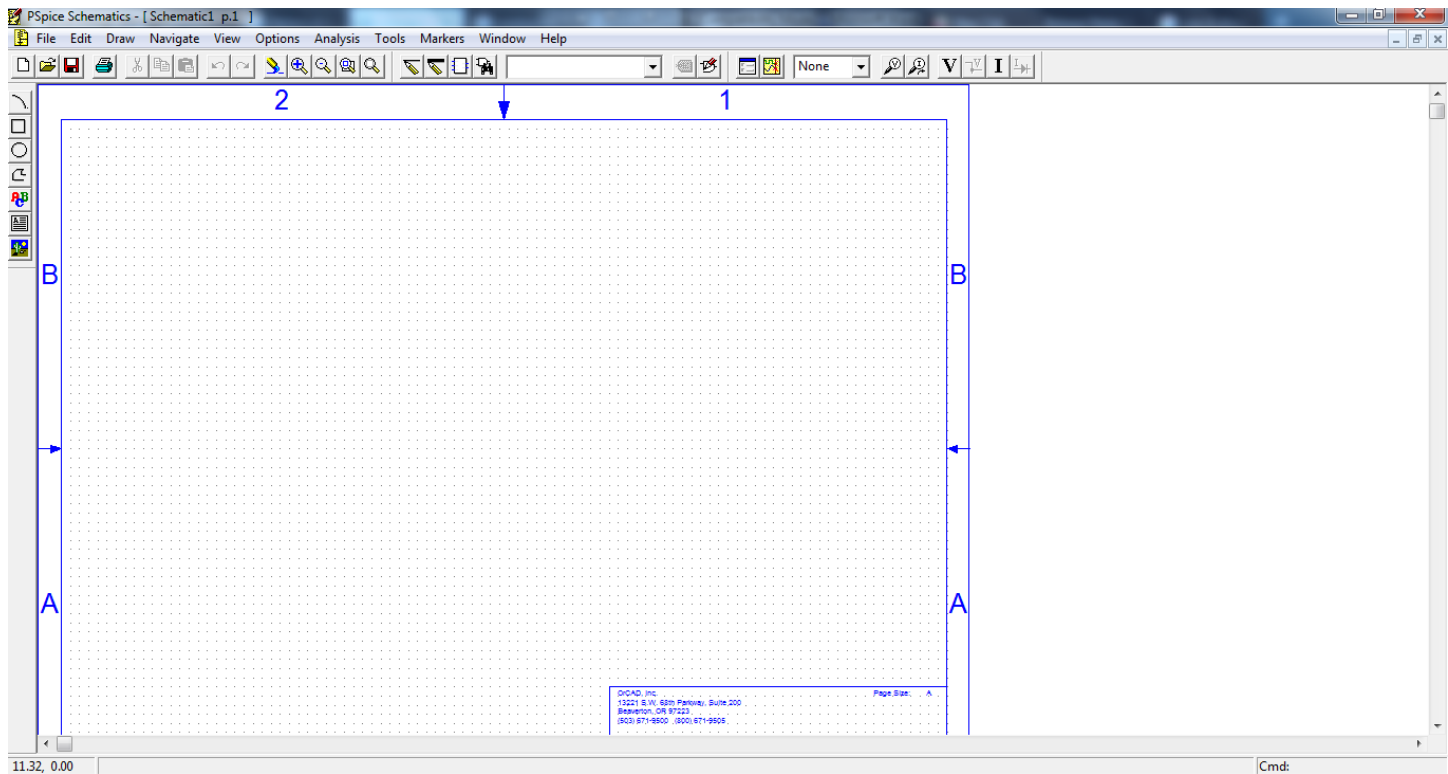


Fig.2. The PSpice Schematics editor interface

Example

Consider a simple voltage divider formed by one DC voltage source and two resistors (Fig.3). Observe that the negative terminal of the battery and the bottom terminal of R2 are grounded. In any spice simulator there must be a reference node with zero voltage potential. Otherwise, if the ground reference (with zero voltage potential) is not included in the diagram, the simulator may generate errors. The DC current (I) through the DC voltage source (V) is found through each of the two resistors because the components are mounted in series.

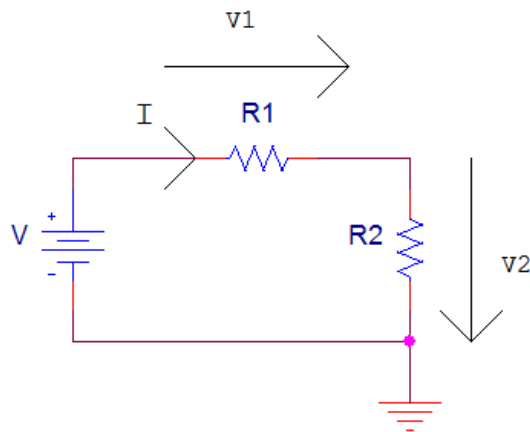


Fig.3. A simple voltage divider biased from a DC voltage source

The following system of linear equations can be written for the circuit:

$$\begin{cases} V = I \cdot R_1 + I \cdot R_2 \\ V_1 = I \cdot R_1 \\ V_2 = I \cdot R_2 \end{cases} \quad (1)$$

To draw the circuit in the Schematics editor, we will add some parts by using the mouse command **Draw-Get New Part...** (or the keyboard shortcut **CTRL+G**) (Fig.3).

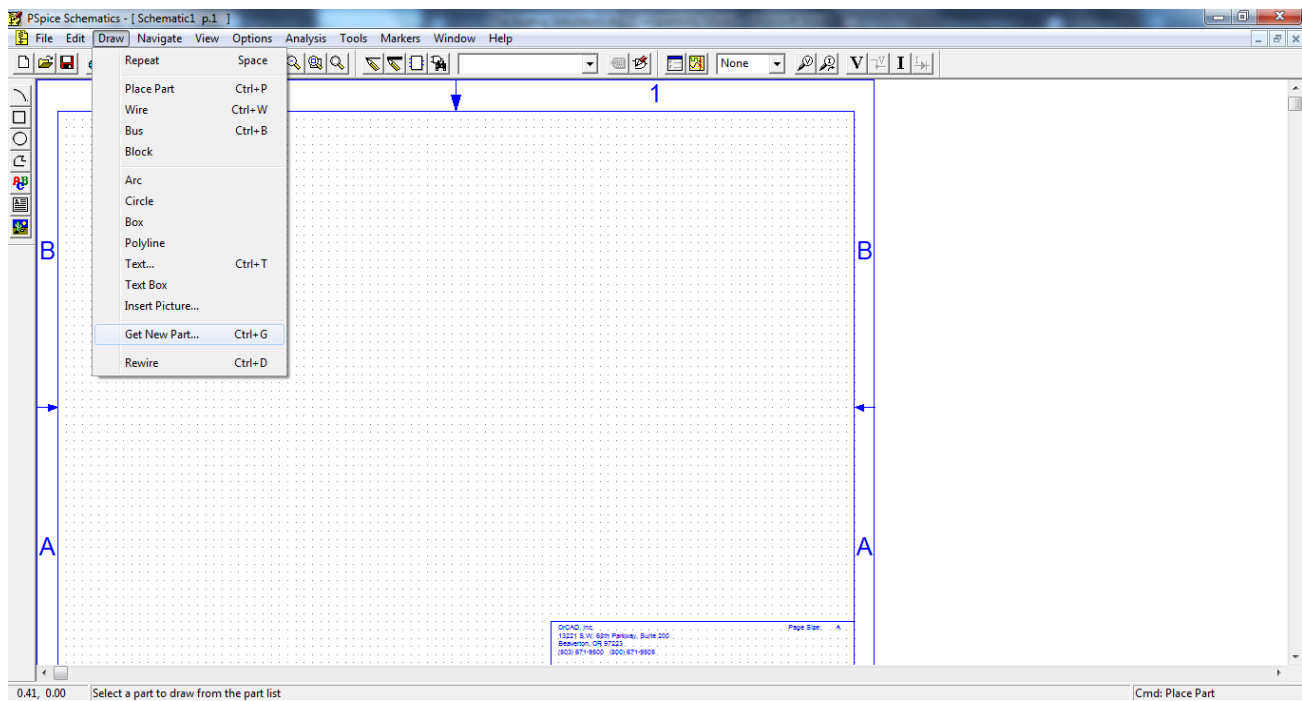


Fig.4. *Get New Part...* command

To see the symbols for the parts, the **Advanced >>** button should be selected(Fig.5).

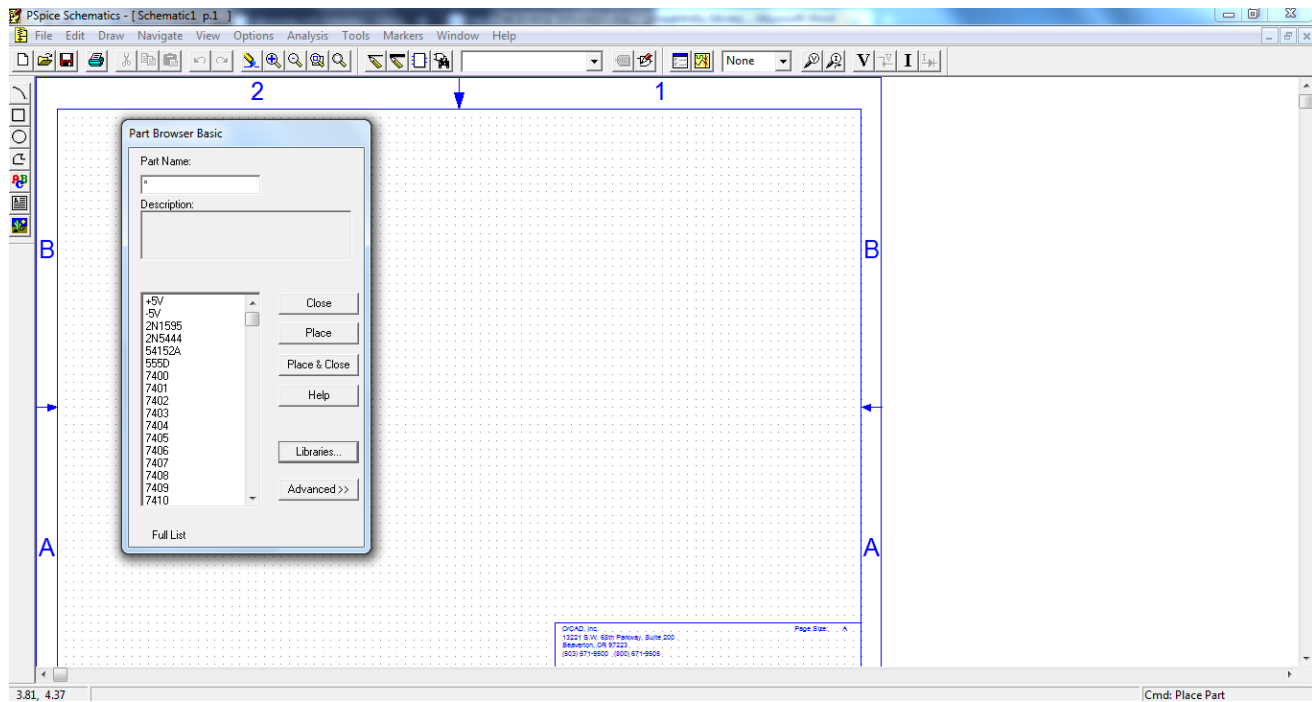


Fig.5.Selecting the **Advanced >>** button to see the parts symbols

In the **Part Name** tab press **R** for resistor symbol (Fig. 6):

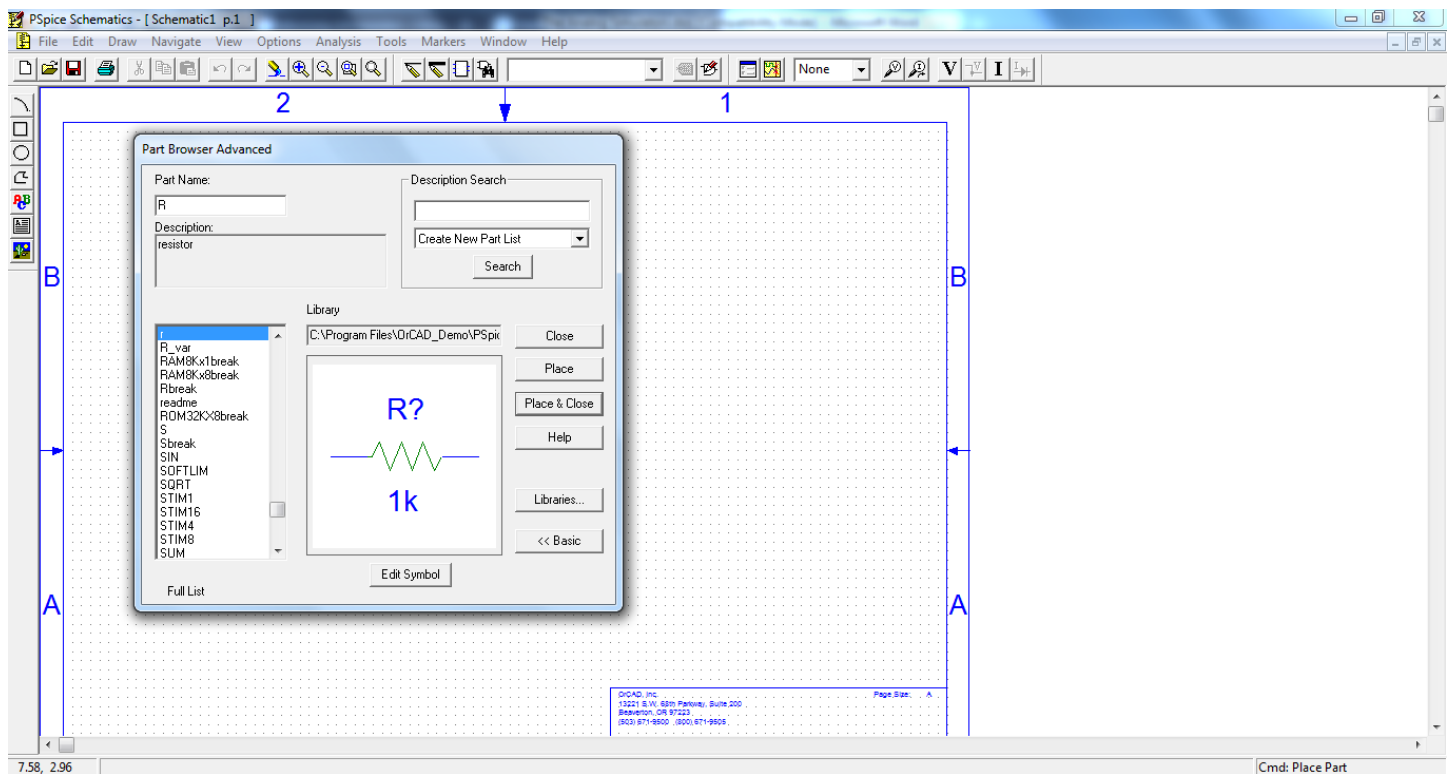


Fig.6. Choosing a resistor from the parts list

The resistor may be selected to be placed on the schematic page, by pressing the **Place** button. Then, in the desired position on the schematic, the part may be dropped on the page with the mouse by using the *left click*(Fig.7). After placement, the resistor is highlighted in red.

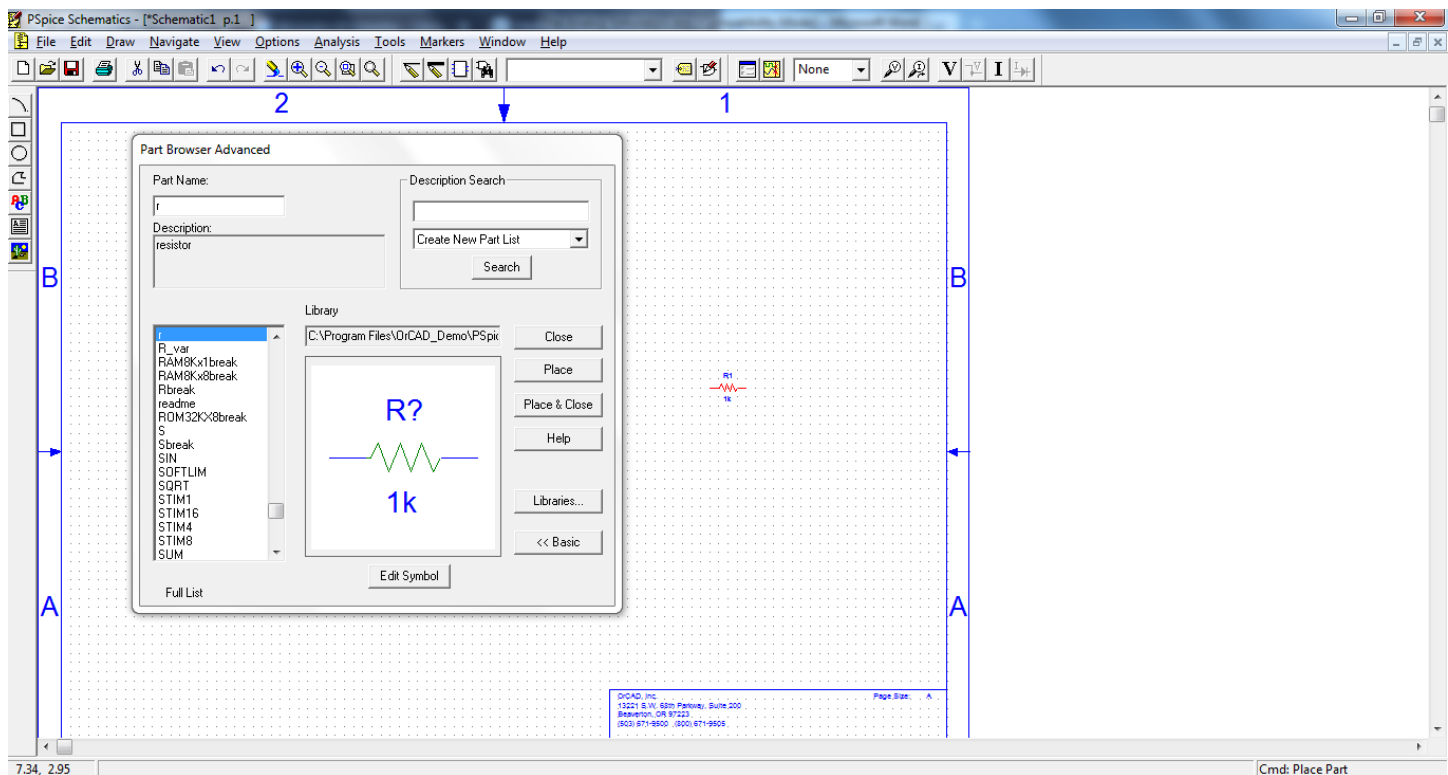


Fig.7. Placing a resistor on the schematics page

If you selected the wrong component, you may cancel the placement of the part with the *right click* of the mouse or *Esc* key. A second resistor may be selected to be placed on the schematic by pressing again **Place**

button. The resistor may be rotated (vertical position) by using the **CTRL+R** keyboard shortcut, and then, dropped on the page with the *left click* of the mouse (Fig.8). The mirror function is accomplished by using the **CTRL+F** keyboard shortcut.

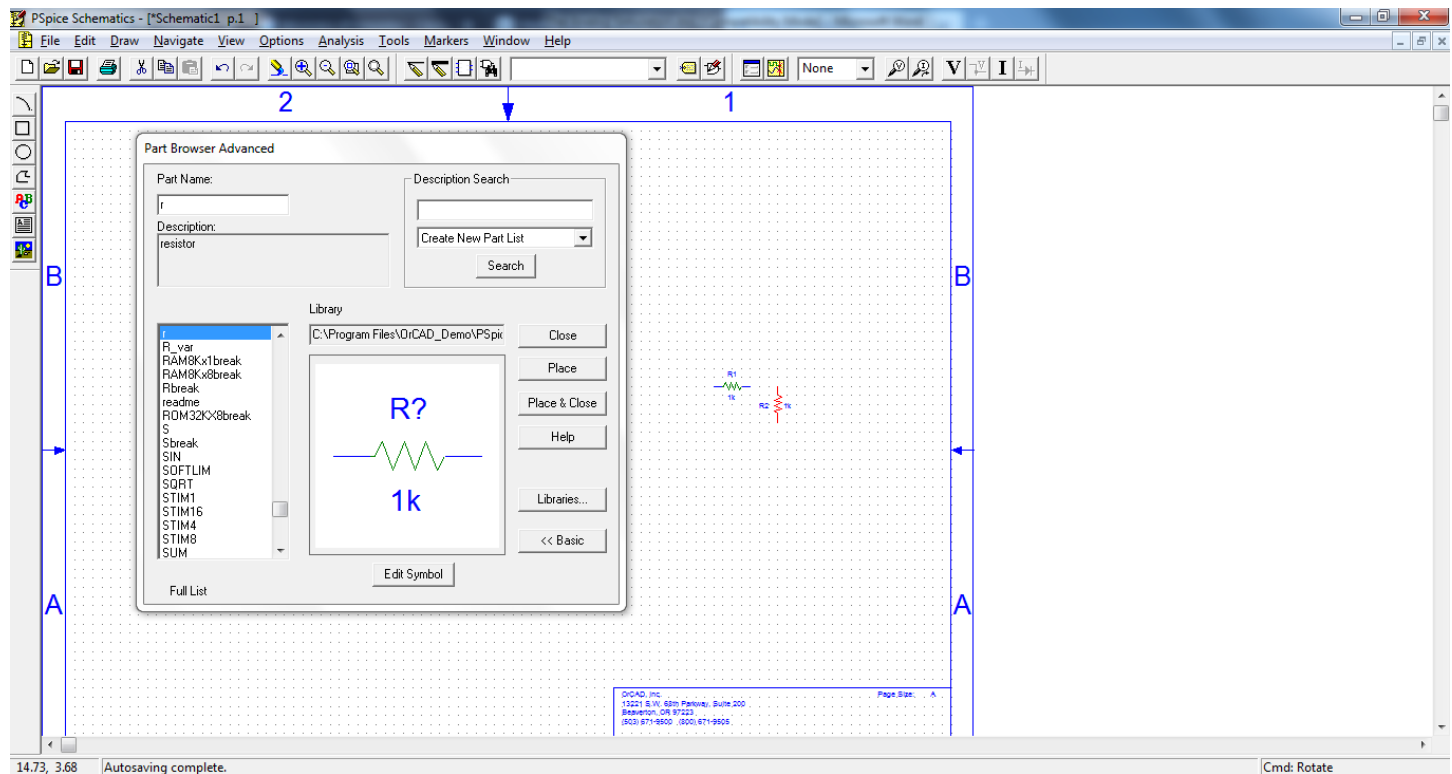


Fig.8. Placing a second resistor on the schematics page

The voltage source may be put on the schematic by typing VDC in the *Part Name:* tab and by placing it with the mouse in the desired section(*left click*) (Fig.9). After the placement of the DC voltage source, if a new voltage source remains selected, you may cancel a new undesired placement by using *right click* of the mouse or *Esc* key.

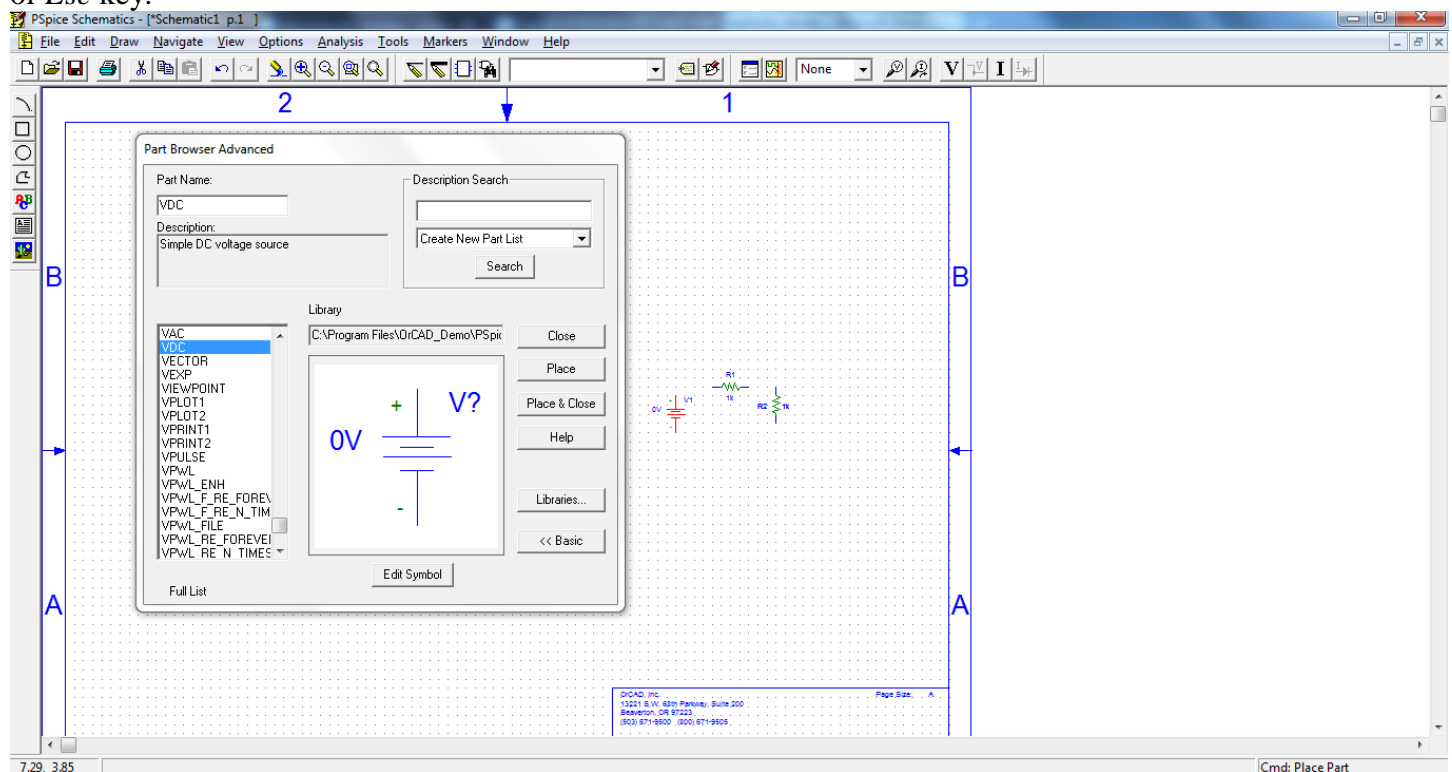


Fig.9. Placing the DC voltage source on the schematics page

The ground of the circuit may be selected by typing *GND* in the *Part Name:* tab, then selecting the *GND_ANALOG* symbol. Finally, it may be placed with the mouse in the desired section(*left click*) (Fig.10).

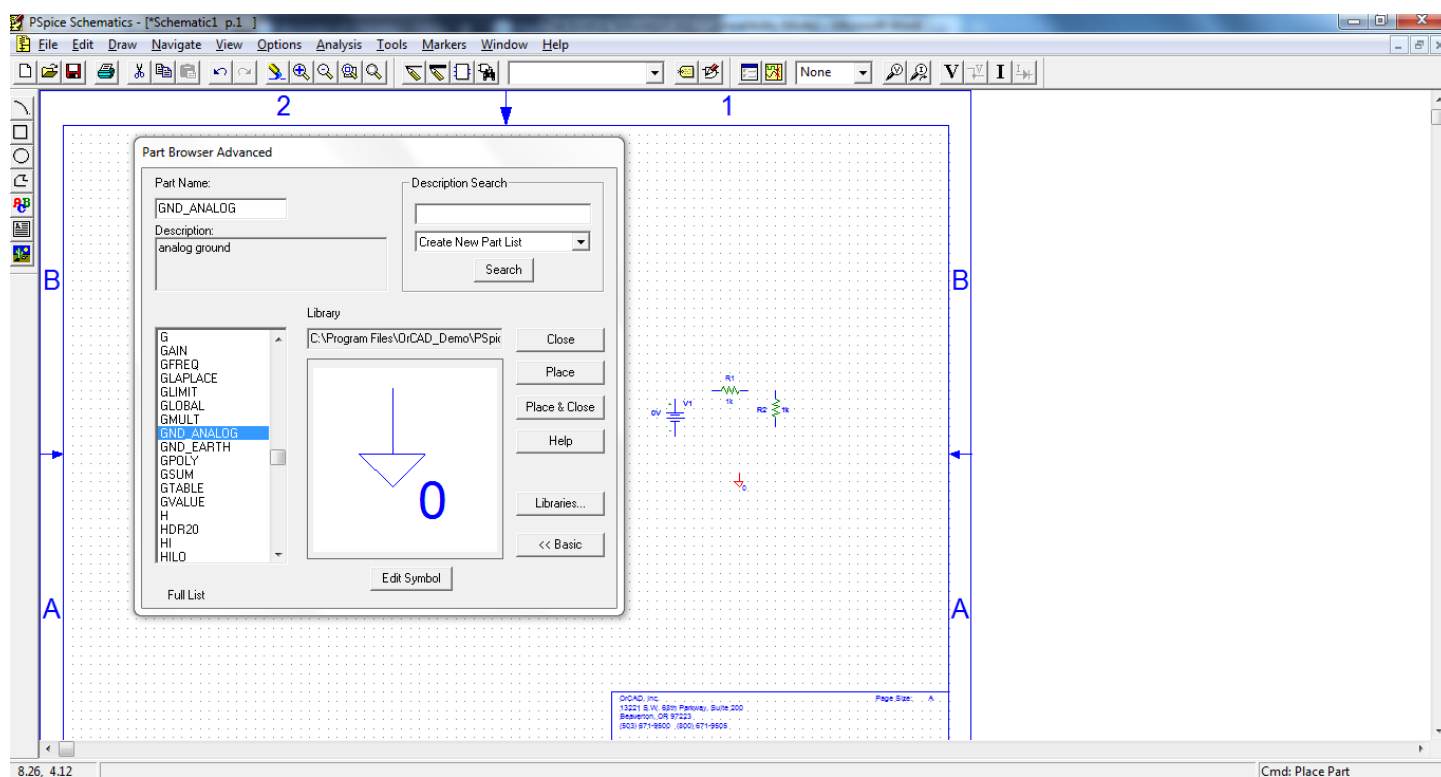


Fig.10. Placing the ground on the schematics page

After all the part symbols are placed on the schematic, the *Part Browser* may be closed by selecting the *Close* button. Next step is to wire the parts by using the mouse. To enlarge the schematic, you may use the ***Zoom to fit page*** feature by selecting it with the mouse. Then, you may select the wiring pen (*draw wire*) which is near it to connect the parts (or ***CTRL+W*** keyboard shortcut) (Fig.11).

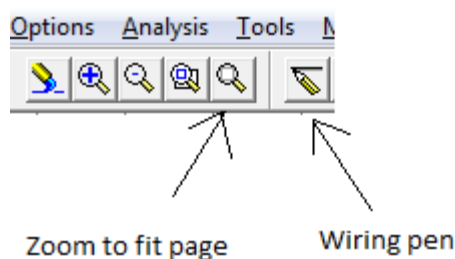


Fig.11. Selecting the wiring pen (*draw wire*) to wire the parts

The wire can be drawn by positioning the pen with the mouse in the starting position (from the terminal where the connection starts) and then by pressing once the *left click* of the mouse. Then, you can walk on the diagram with the wire until you reach the end position (at the terminal where the connection ends), and pressing again the *left click* of the mouse (Fig 12). In this way, the connections between the parts can be made (Fig.13). If you want to give up the pen, press once the *left click* of the mouse or the *Esc* key.

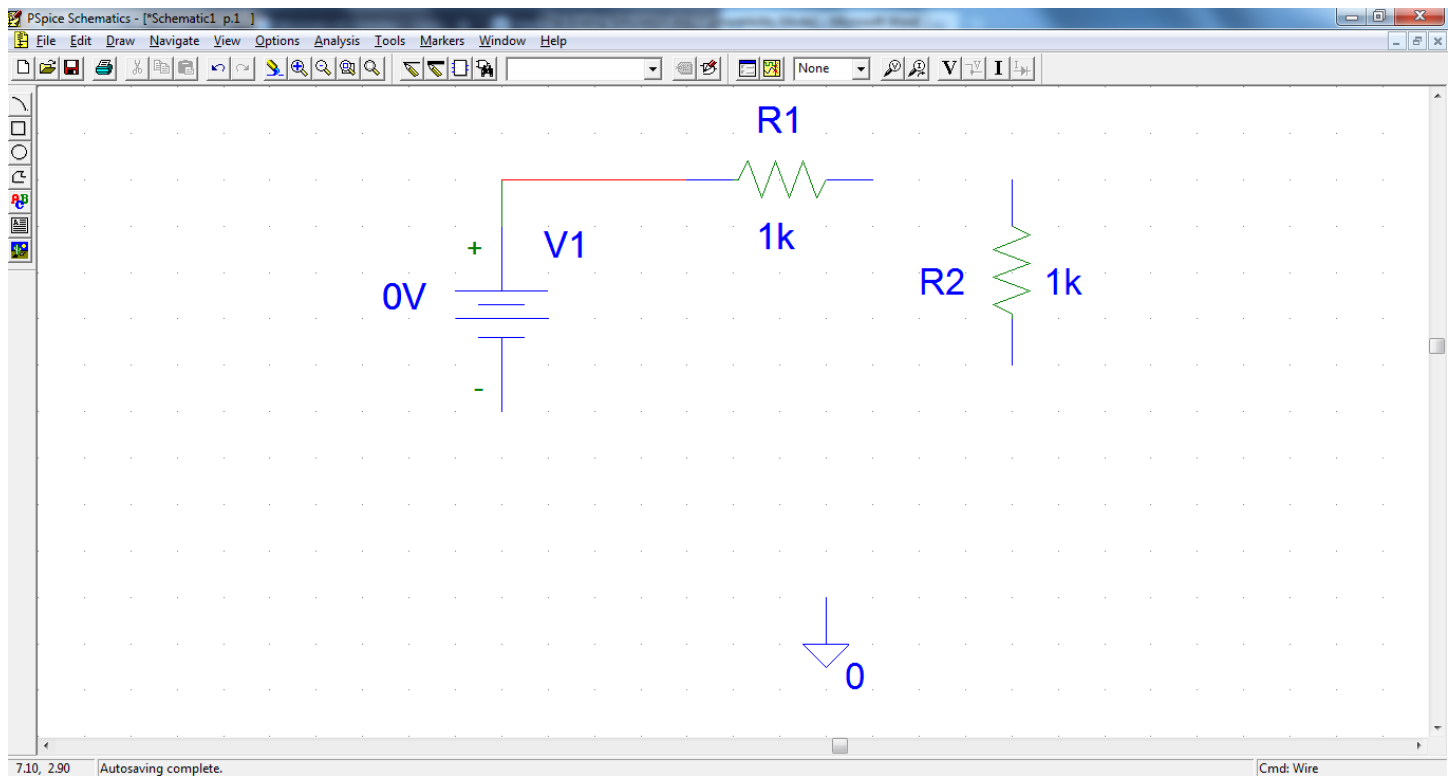


Fig.12. Wiring two components

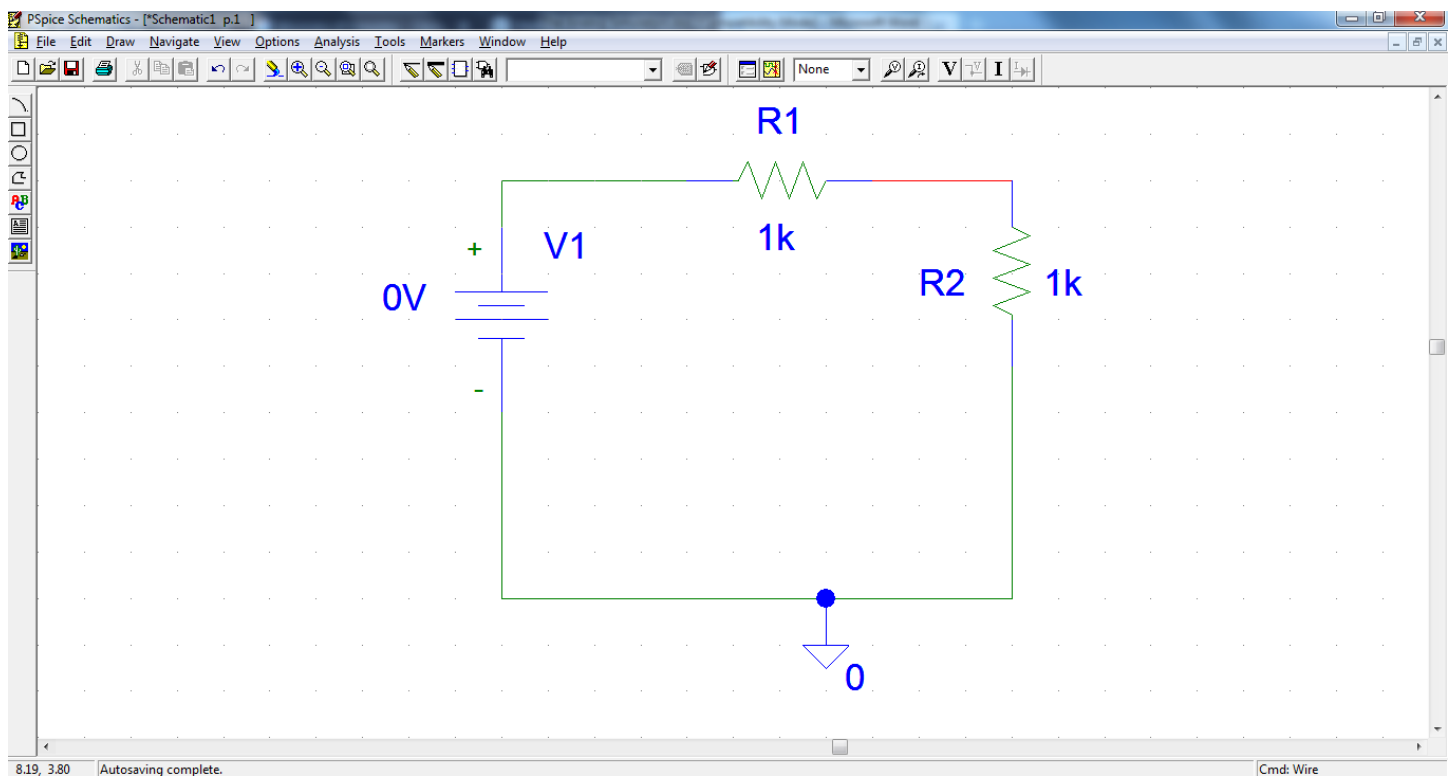


Fig.13. Wiring the circuit

In the process of drawing the schematic, extra care must be taken, so that no additional nodes are drawn compared to the schematic from the laboratory platform. If these nodes appear (this is the most common mistake), it means that you doubled the wire by another wire (this occurs especially near the component terminal). To repair the mistake, you should select the short wire with the mouse (it will change in red color) and delete it (by pressing the *DELETE* key) (Fig.14).

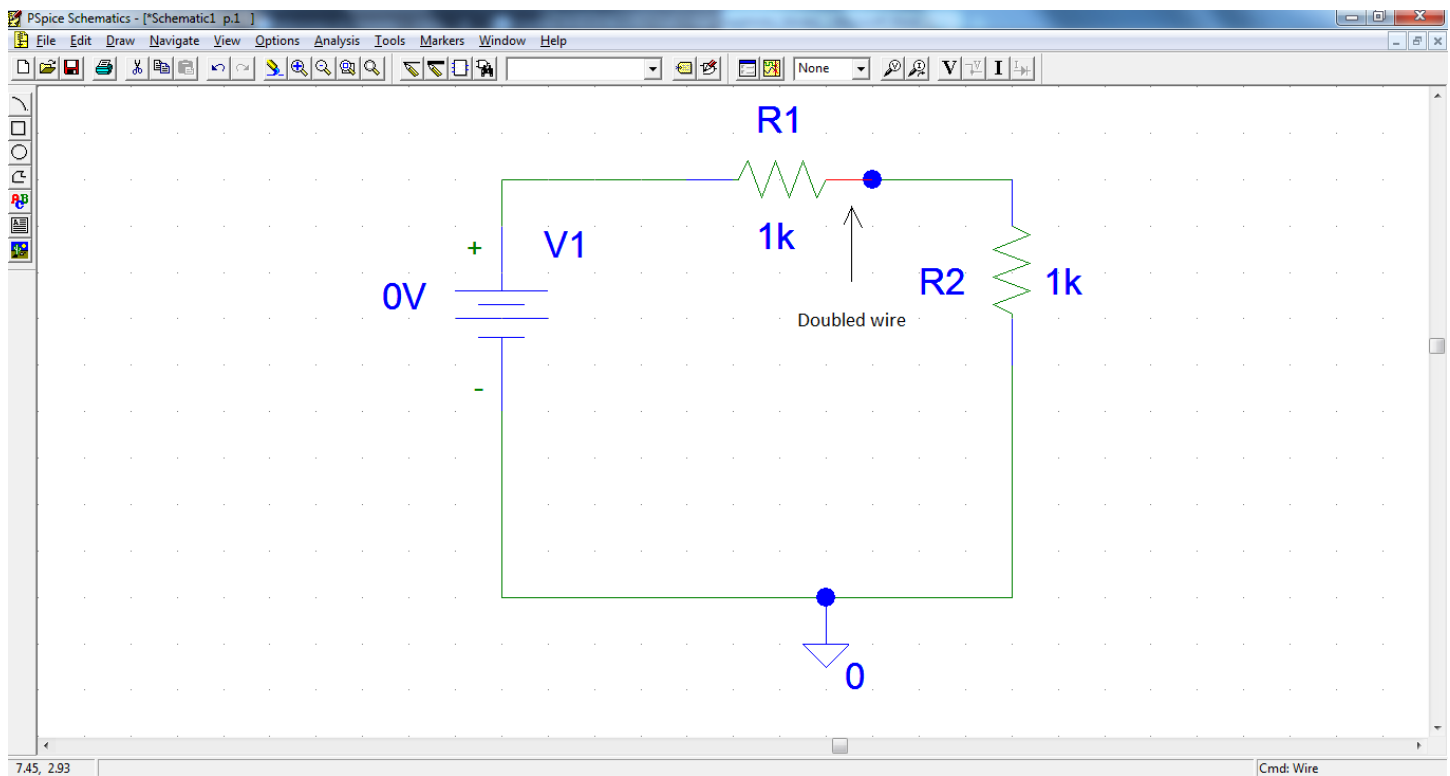


Fig.14. Drawing error

In the next step, the components are named and their numerical values are assigned according to the schematic diagram (Figs.15-16). To accomplish these things, you should double click the names and the values and change them. The names and the values may be dragged and dropped with the mouse on the schematic to create extra space or to give a neat look to the circuit.

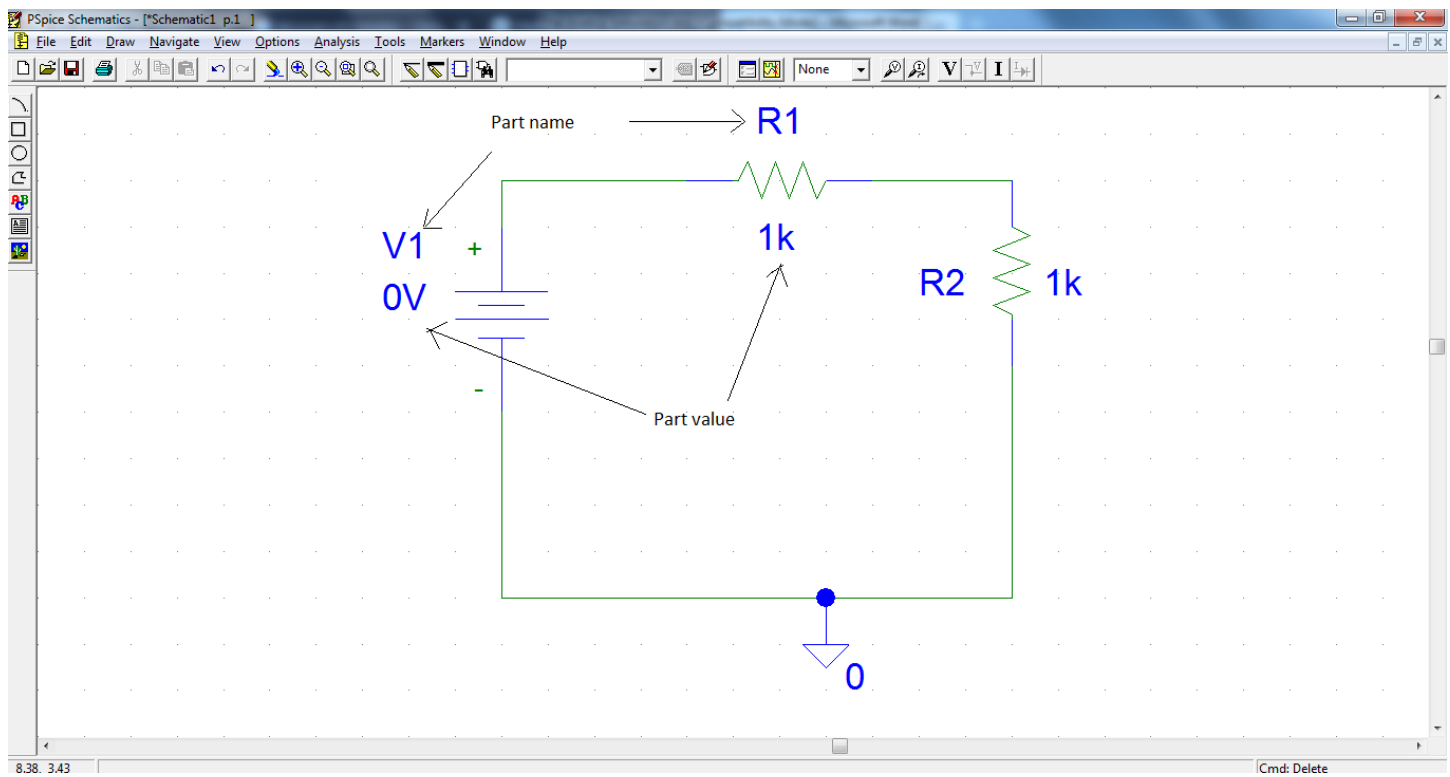


Fig.15. The names and the values positions for the parts

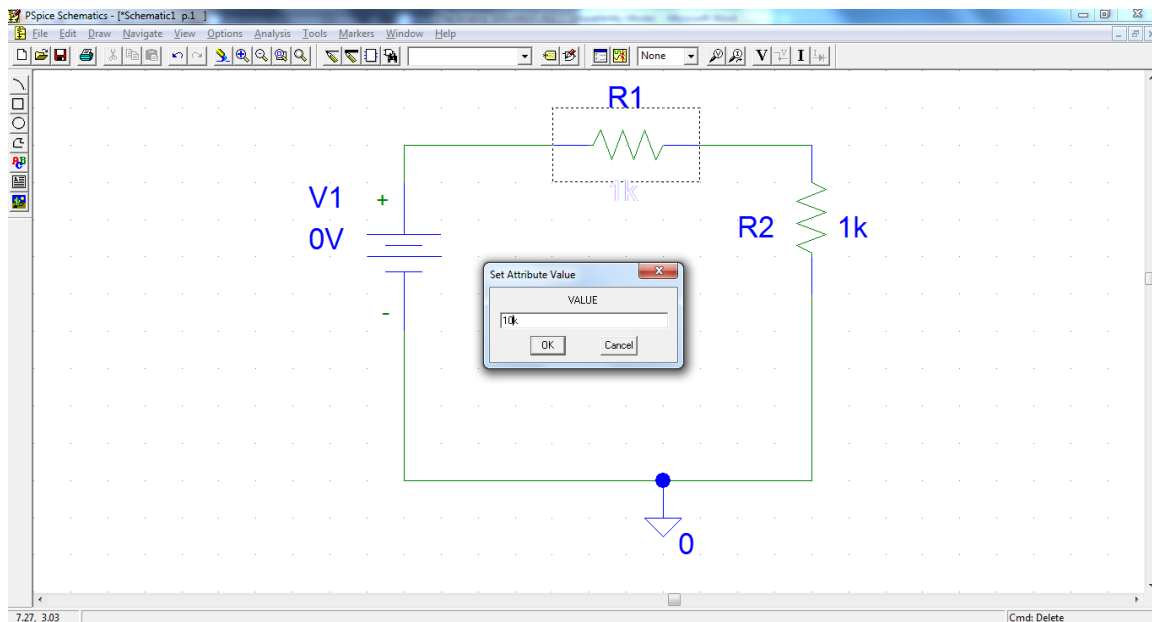


Fig.16. Changing the part value

You can draw the voltage divider circuit (Fig.17) by using the described methods.

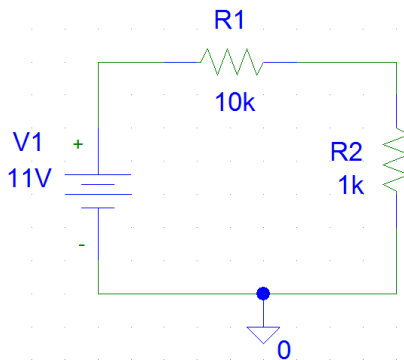


Fig.17. The voltage divider circuit to be simulated

After the circuit is drawn, you should save it (*File-Save as...*) by giving it a suitable name (Fig.18).

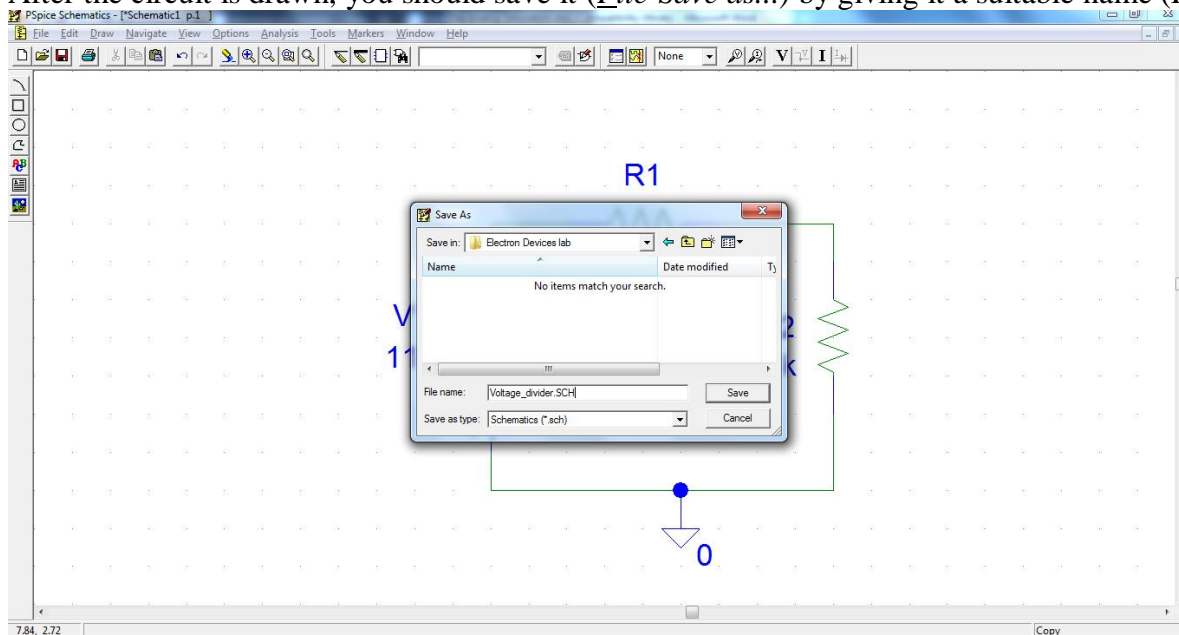


Fig.18. Saving the file

After this step, the DC simulation can be started (by selecting *Analysis-Simulate* or by pressing *F11* button). Then, if there is no error, the Pspice A/D program (*Probe*) should automatically start (Fig.19). Because the circuit is very simple and it has only a DC voltage source and two resistors, there is nothing to be displayed in this window (for the moment).

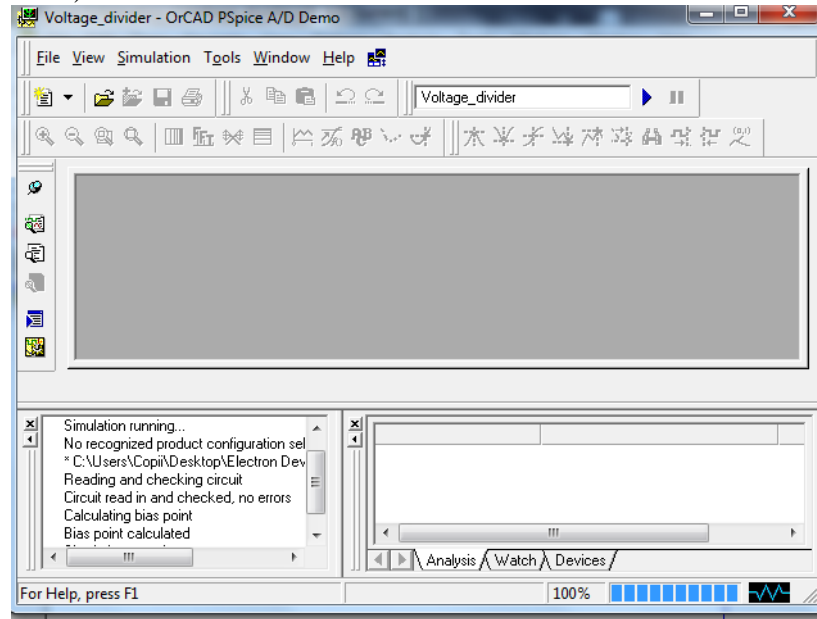


Fig.19. The PSpice A/D software

By selecting *Enable Bias Voltage Display (V)* and *Enable Bias Current Display (I)* buttons (Fig.20) in Pspice Schematics, the voltages and the currents are represented on the circuit page (Fig.21).

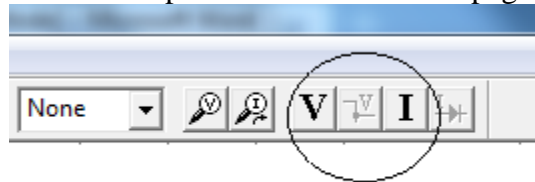


Fig.20. Enabling voltage and current display on the schematic

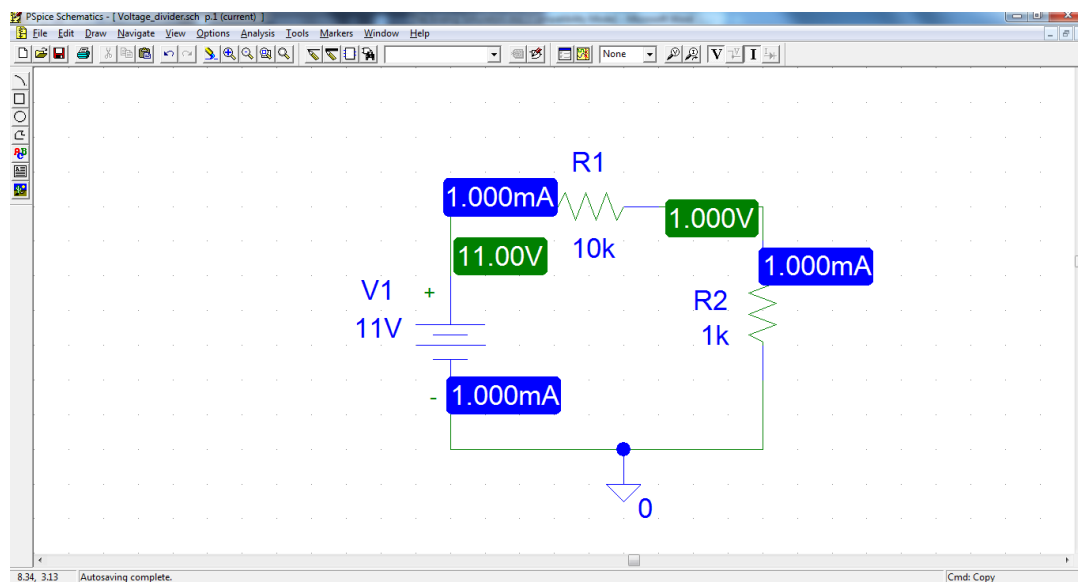


Fig.21. The circuit DC voltages and the DC currents after the simulation

You can manually calculate the current and voltage at the terminals of the resistors, and then compare with the values resulting from the simulation.

The simulator is able to perform 12 types of analyzes by selecting the *Analysis-Setup* interface (Fig.22):

-**AC Sweep(1)**;

-Load Bias Point;

-Save bias Point;

-**DC Sweep(2)**;

-Monte Carlo/Worst Case;

-**Bias Point Detail(3)**;

-**Parametric(4)**;

-**Sensitivity(5)**;

-**Temperature(6)**;

- **Digital Setup(7)**;

-Transfer Function;

-**Transient(8)**.

The bolded analyzes will be further discussed.

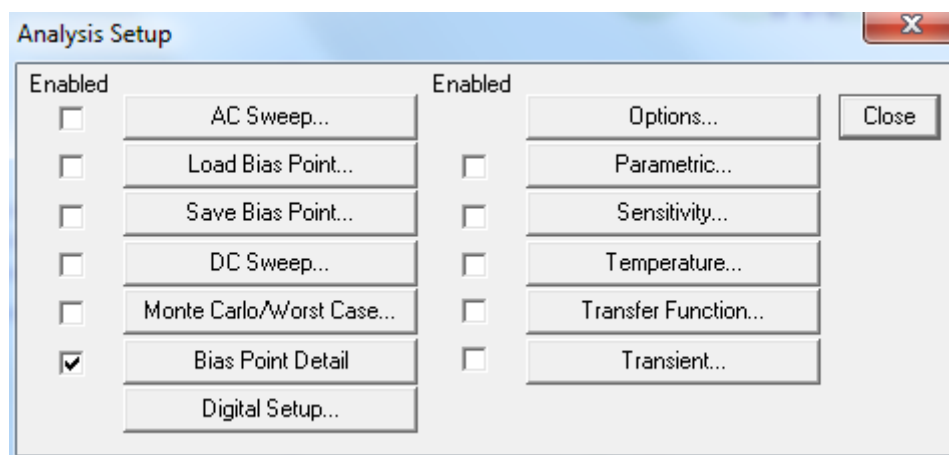


Fig. 22 The Analysis Setup interface

1. AC Sweep

The AC sweep is used for frequency domain analysis. It permits to plot magnitude of the voltage/current versus frequency for different inputs in the circuit. In the AC sweep menu there are three types of analysis: *Linear*, *Octave* and *Decade*, which describe the X-axis scaling (it shows how the frequency that is represented on the x-axis will be scaled.). The option *Noise enable* will permit to simulate noise either on the output or the input of the circuit. These noise calculations are performed at each frequency step and can be plotted in probe. The two types of noise are: V(ONoise) for noise on the outputs and V(INoise) for noise on the input source.

- To use input noise you need to tell PSpice where you consider the 'input' in your circuit to be, for example, if your voltage source is labeled 'V1'.
- Finally you need to specify in what interval you want the noise to be calculated (note: the default interval for spice is zero, i.e.: no noise will be calculated).

2. DC Sweep

The DC sweep allows doing various different sweeps of the circuit to see how it responds to various conditions. For all the possible sweeps (voltage source, temperature, current source, model parameter and global) you need to specify a start value, an end value, and the number of points you wish to calculate. For example you can sweep your circuit over a voltage range from 0 to 12 volts. The main two sweeps that will be most important to us at this stage are the voltage sweep and the current sweep. For these two, you need to indicate to PSpice what component you wish to sweep, for example V1 or V2. Another excellent feature of the DC sweep in PSpice, is the ability to do a **nested sweep**. A nested sweep allows you to run two simultaneous sweeps to see how changes in two different DC sources will affect your circuit. Once you've filled in the main sweep menu, click

on the nested sweep button and choose the second type of source to sweep and name it, also specifying the start and end values. (Note: you should click on **enable nested sweep**). Again you can choose Linear, Octave or Decade, but also you can indicate your own list of values, example: 1V 10V 20V. Do not separate the values with commas.

3. Bias Point Detail

By clicking on **enable bias current display** or **enable bias voltage display**, it displays the voltage and current at certain points within the circuit.

4. Parametric

Parametric analysis allows you to run another type of analysis (transient, sweeps) while using a range of component values using the **global parameter** setting. The best way to demonstrate this is with an example, we will use a resistor, but any other standard part would work just as well (capacitor, inductor). First, double-click the value label of the resistor that is to be varied. This will open a "Set Attribute Value" dialog box. Enter the name **{RVAL}** (including the curly braces) in place of the component value. This indicates to PSpice that the value of the resistor is a global parameter called RVAL. In order to define the RVAL parameter it is necessary to place a global parameter list somewhere on the schematic page. To do this, choose "**Get New Part**" from the menu and select the part named **param**. Place the box anywhere on the schematic page. Now double-click on the word **PARAMETERS** in the box title to bring up the parameter dialog box. Set the NAME1= value to **RVAL** (no curly braces) and the VALUE1= value to the nominal resistance value. This nominal value is required, but it is only used if the DC bias point detail is computed. Otherwise, the value is ignored by PSpice. Finally, go to the "Analysis Setup" menu and enable "Parametric" analysis. Open the Parametric setup dialog box and enter the sweep parameters: Name: **RVAL** Swept variable type: Global Parameter. Make sure the other analysis type(s) are selected in the analysis setup menu (transient, sweeps). PSpice will now automatically perform the simulation over and over, using a new value for **RVAL** during each run.

5. Sensitivity

Sensitivity causes a DC sensitivity analysis to be performed in which one or more output variables may be specified. Device sensitivities are provided for the following device types only: resistors, independent voltage and current sources, voltage and current-controlled switches, diodes, and bipolar transistors. You would use the sensitivity setting for discovering the maximum range of circuit performance and the causes of extreme operation. These techniques are used to identify effective changes to improve the quality of circuit operation (for example, which components need to have tight tolerance and which can be lower quality and less expensive).

6. Temperature

The temperature option allows to specify a temperature, or a list of temperatures (do not include commas between temperature values) for the circuit components. For a list of temperatures that simulation is done for each specified temperature.

7. Digital setup

This paragraph will only indicate the features of the digital setup on the analysis menu, see below for a more complete description. In addition to letting you simulate analog circuits, PSpice provides a number of digital parts that can be used in a homogeneous digital circuit, or a heterogeneous analog/digital combination. The digital analysis option allows you to specify the timing of your circuit, by running the gates at their minimum, maximum and typical values. A superb feature allows you to test the worst case timing of your circuit to see how it will operate under these extreme conditions. You also have the option of setting the value of any flip flops you have in your circuit to predefined states which is good to simulate any startup conditions for finite state machines that you are simulating.

8. Transient

The transient analysis is probably the most important analysis you can run in PSpice, and it computes various values of your circuit over time. Two very important parameters in the transient analysis are: *print step* and *final time*. The ratio of *final time/print step* determines how many calculations PSpice must do to plot a wave form. PSpice always defaults the start time to zero seconds and going until it reaches the user defined final time. It is incredibly important that you think about what print step you should use before running the simulation, if you make the print step too small the probe screen will be cluttered with unnecessary points making it hard to read, and taking extreme amounts of time for PSpice to calculate. However, at the opposite side of that coin is the problem that if you set the print step too high you might miss important phenomenon that are occurring over very short periods of time in the circuit. Therefore play with step time to see what works best for your circuit. You can set a step ceiling which will limit the size of each interval, thus increasing calculation speed. Another handy feature is the Fourier analysis, which allows you to specify your fundamental frequency and the number of harmonics you wish to see on the plot. PSpice defaults to the 9th harmonic unless you specify otherwise, but this still will allow you to decompose a square wave to see its components with sufficient detail.

Laboratory exercise

Consider the circuit given in Fig.23. The components are described in *Table 1* from Annex 1. It is required to draw and to simulate the circuit. V1 is a DC voltage supply (1V). V2 is a sine-wave voltage source for time domain analysis (VSIN) with the following parameters: VOFF=0; VAMPL=1V; FREQ=5k. V3 is a sine-wave voltage generator for frequency domain analysis (VAC) with the following parameters: DC=0V; ACMAG=1V.

Homework: manually calculate the amplitudes of the electric parameters (I,V) on the components, the circuit impedance, power factor, and draw the phasor diagram.

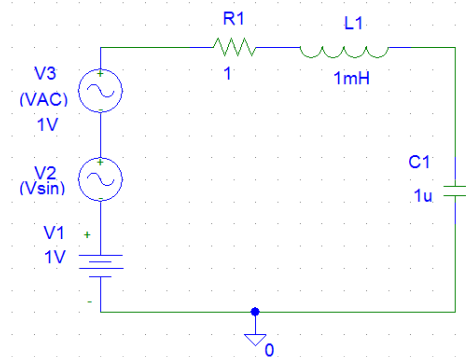


Fig. 23 The series RLC circuit

After drawing the circuit, a simulation setup should be made. We'll start with AC analysis by selecting: *Analysis-Setup...-AC Sweep...*. The AC simulation will be in the 1KHz÷100KHz range and the frequency will be displayed in decades (Fig.24):

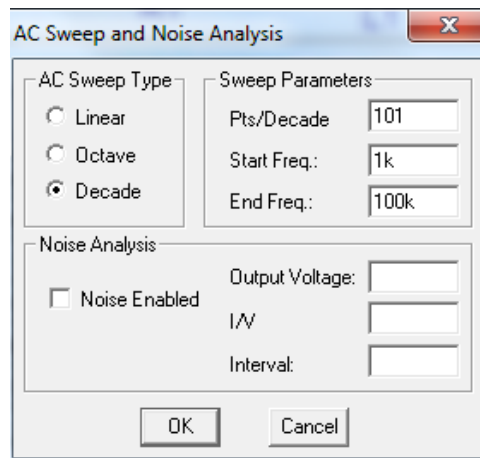


Fig. 24 The selection of AC setup parameters

Next step is the time domain analysis by selecting *Transient...* button. The simulation will be made for a time of 20ms with a resolution of 1 μ s (Fig.25).

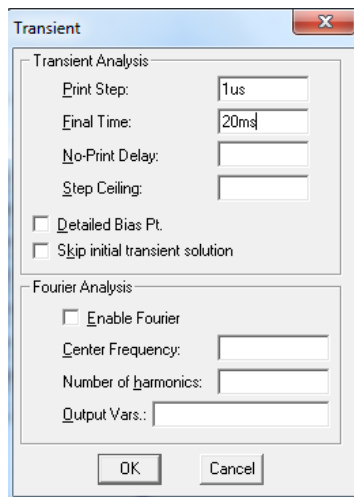


Fig. 25 The selection of Transient parameters

After this step, the circuit may be simulated (*Analysis-Simulate* or *F11*). The *PSpice A/D Demo* starts and a *Analysis Type* window will ask you to choose between frequency domain (AC) and time domain (*Transient*) analysis (Fig.26).

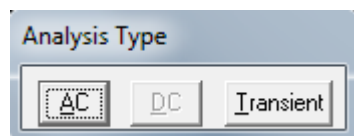


Fig.26. The selection between frequency domain or time domain analysis

a. AC Analysis

We'll start with the AC (freq.) analysis. To measure the simulated current through the RLC circuit, the *Current Marker* in the *PSpice Schematics* is selected (Fig.27), and then it should be dropped with the mouse into a terminal of one of R, L or C components (near the component symbol, in the place where the wire color is changing from green to blue, Fig.28).

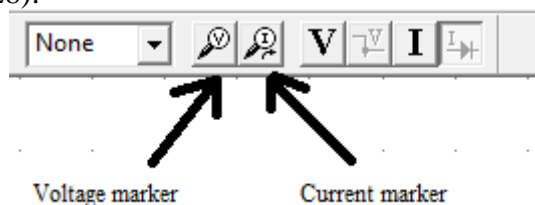


Fig.27. The voltage and the current markers

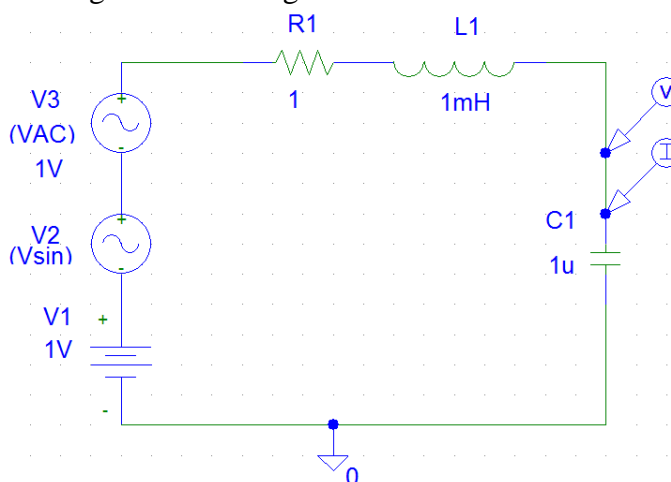


Fig.28. Placing the voltage (V) and the current markers (I) on the schematic

The simulation results may be studied in *Orcad PSpice A/D Demo* window (Fig.29).

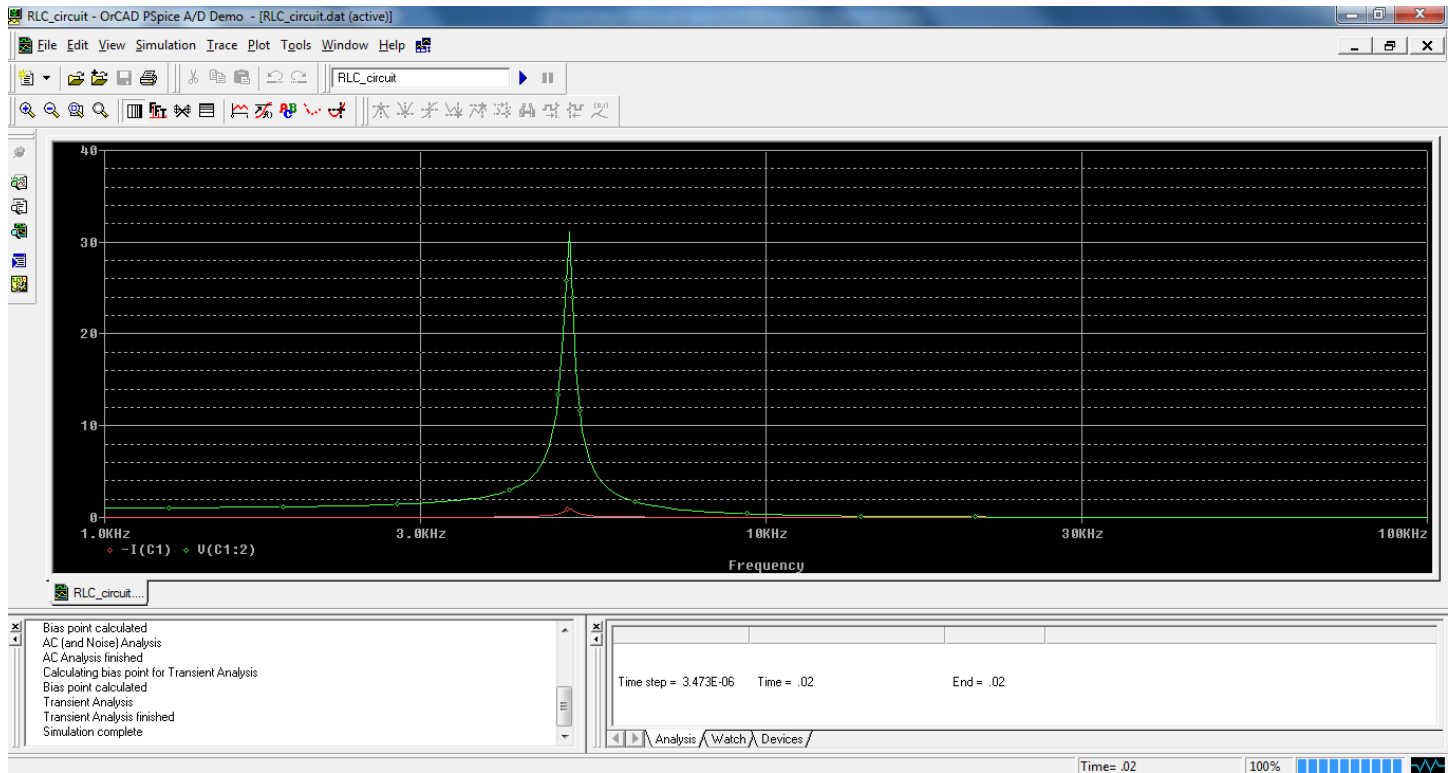


Fig.29. The current (red color) through RLC circuit and the capacitor dropout voltage (green color) in the frequency domain

The *toggle cursor* should be selected in *Pspice A/D Demo* (Fig.30) to see the numerical values on the graph (Fig.31). The cursor may be moved on the graph by using the mouse (coarse positioning) or by using the keyboard *left* (<-) and *right* (->) arrows (fine positioning).

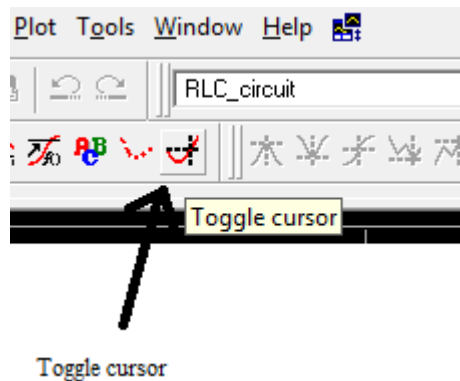


Fig.30. The selection of the *toggle cursor*

When several graphs are displayed, the measurement of the data associated with a graph using the toggle cursor can be done by selecting with the mouse the colored rhombus next to the desired electrical quantity (Fig.31).

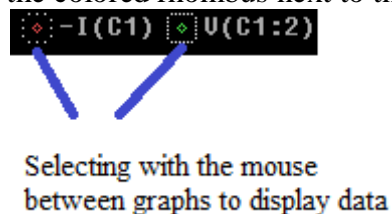


Fig.31. Measurement of the data associated with a graph when multiple graphs are displayed

The data displayed in the Probe Cursor box is taken from X and Y axis (Fig.32). In the first column there is data collected from X axis, and in the second column data is taken from Y axis. The first row contains the coordinates (X, Y) where the cursor is positioned. The second row contains the coordinates (X_0, Y_0) of the first simulated point of the graph. In the third row (*dif*), the difference between the cursor coordinates and the initial coordinates ($X-X_0, Y-Y_0$) is calculated.

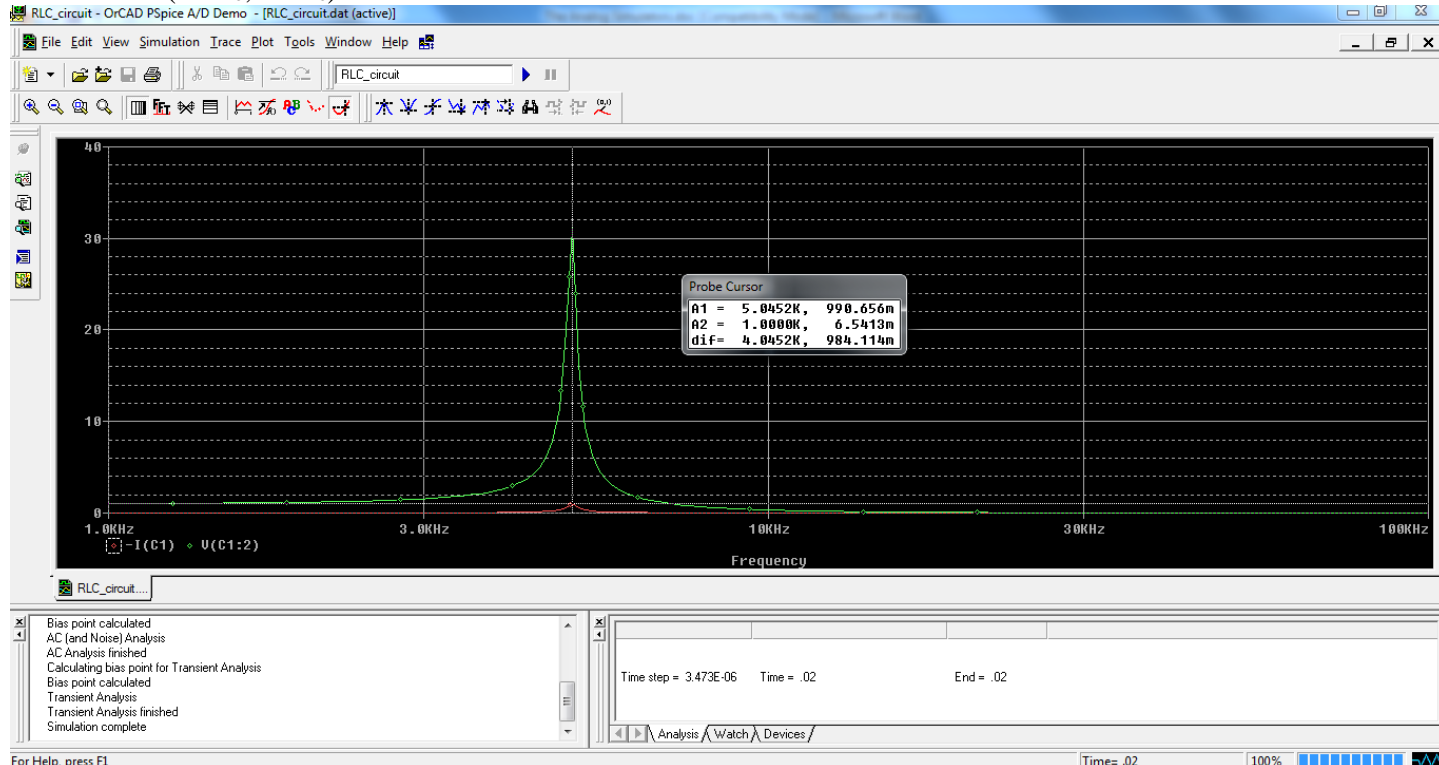


Fig.32. Data measurement by using the *toggle cursor (Probe Cursor)*

To display V-I characteristic on the component (e.g. the capacitor), change the X-axis parameter to the *current (I)* parameter: *Plot-Axis Settings...-X Axis – Axis Variable...-I(C1)* (Fig.33).

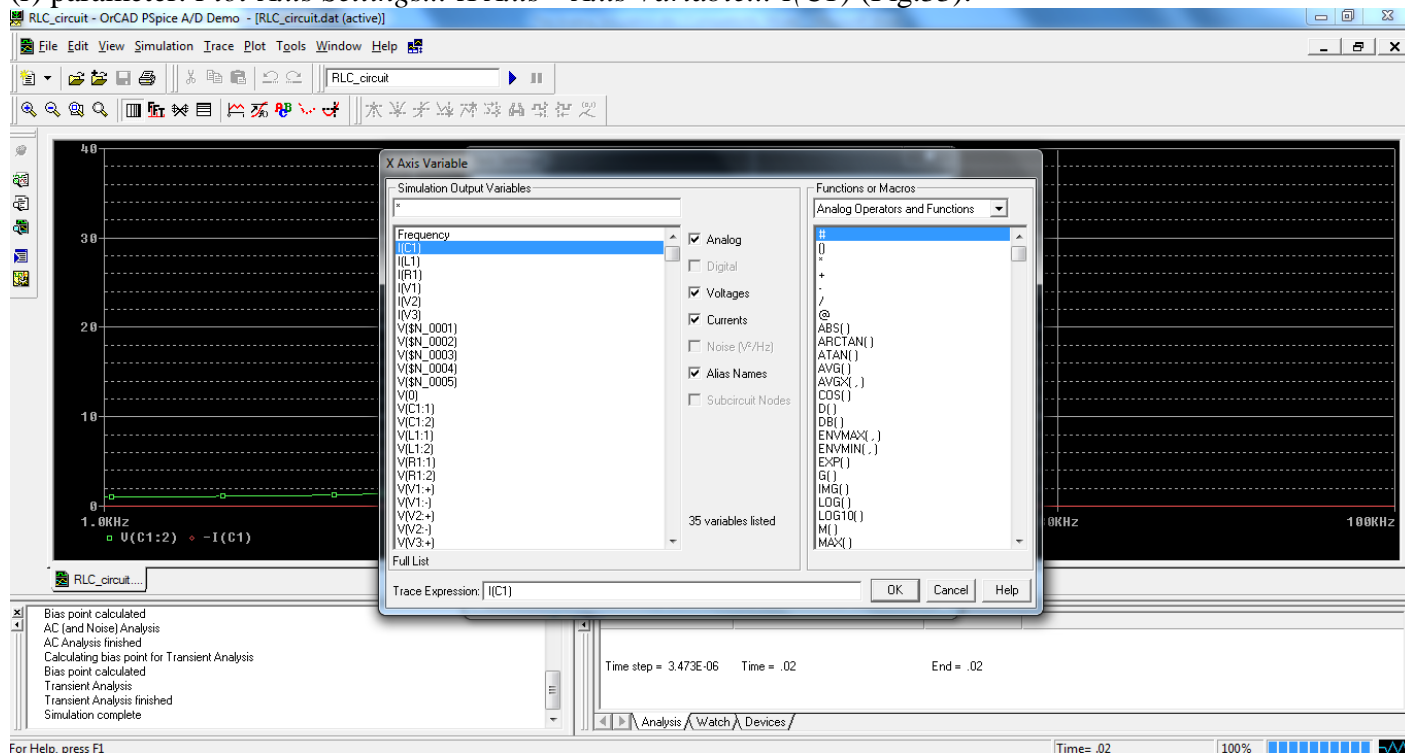


Fig.33. Changing the X-axis parameter to display the V-I characteristic

The unnecessary data (e.g. the $-I(C1)$ graph) may be removed by selecting it with the mouse and by pressing the *DELETE* key. In this way, only the V-I curve is kept on the simulation page(Fig.34).

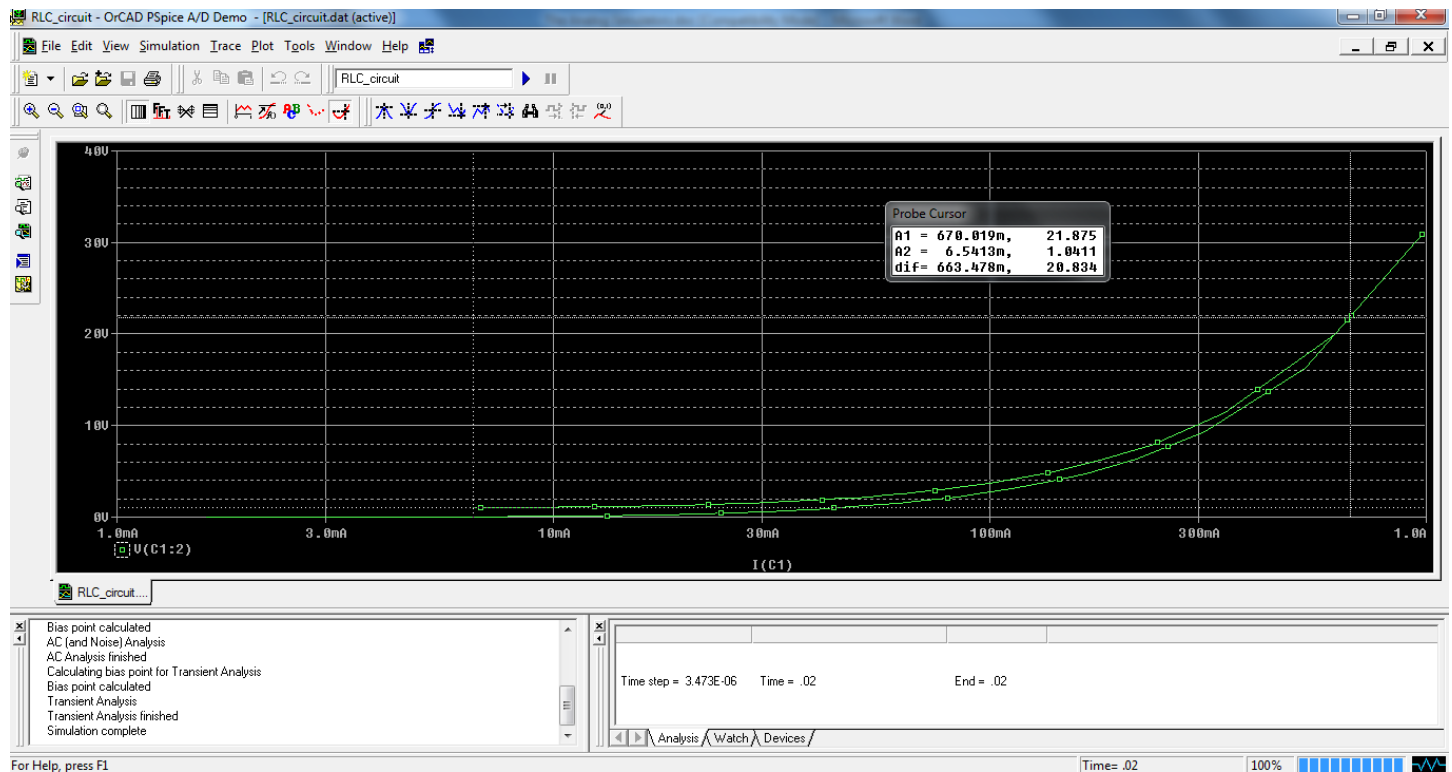


Fig.34.The V-I characteristic on the capacitor in the frequency domain

The potential measured with the voltage marker is relative to the ground. To measure the voltage between the terminals of a component which has no terminal connected to the ground (e.g. the resistor R, or the inductor L), you should select in the *PSpice A/D Demo* menu: *Trace-Add Trace...* the desired *Simulation Output Variable*. For example, to evaluate the voltage across the inductor, select $V1(L1)$ from the list of parameters (Fig.35) (the voltage marker has been previously deleted in the schematic by selecting it with the mouse and then by pressing the *DELETE* key).

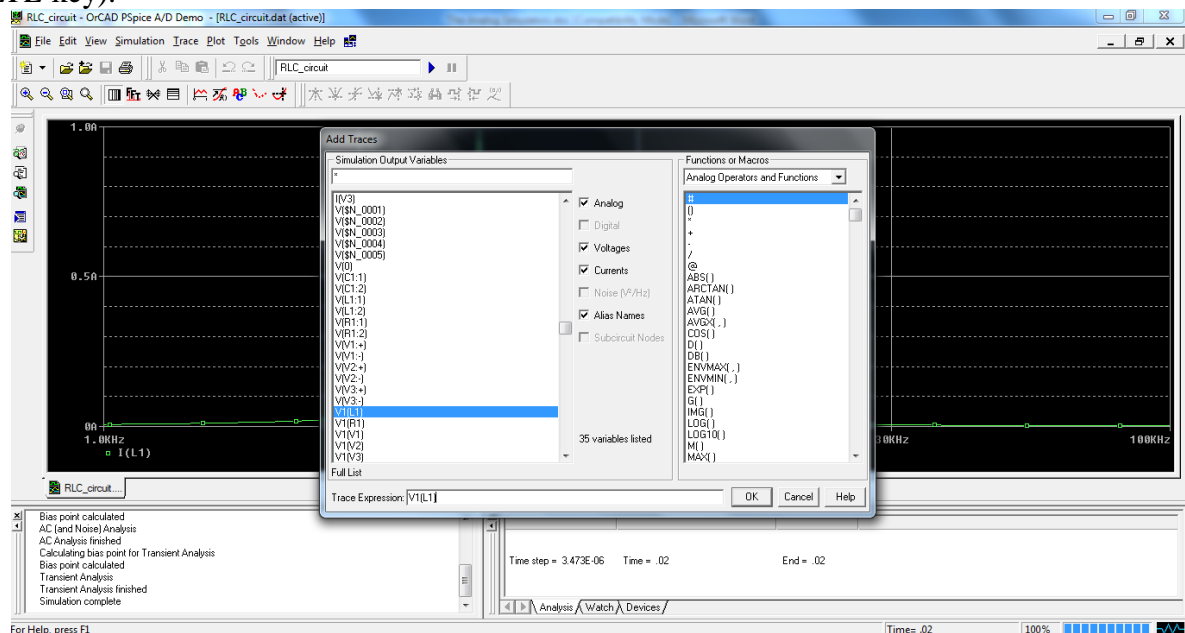


Fig.35.The selection of the dropout voltage on the inductor (L)

Finally, the voltage amplitude on the inductor (L) is available by selecting *toggle cursor* feature and by choosing the voltage graph (Fig.36).

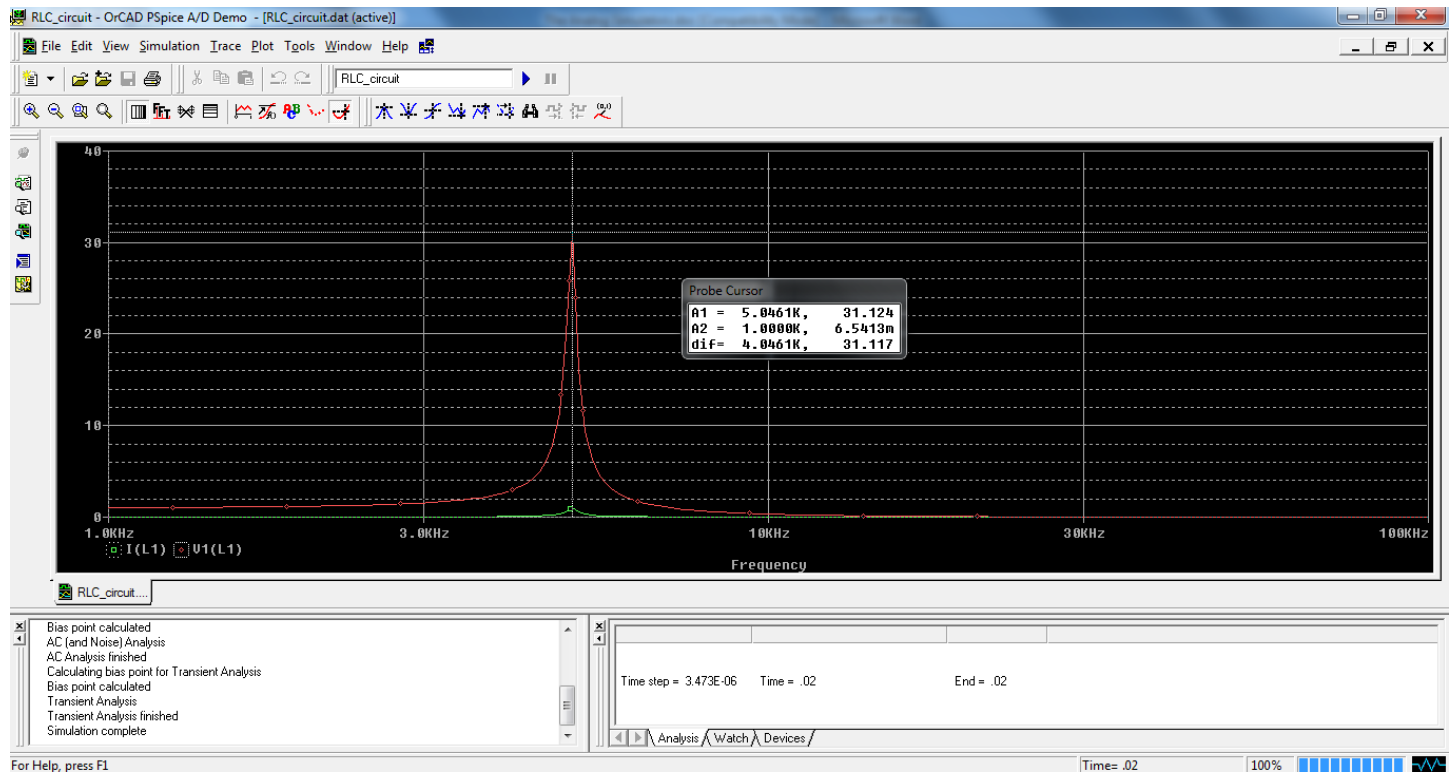


Fig.36. The measurement of the dropout voltage on the inductor (L) in the frequency domain

b. Transient Analysis

The next step is to go back in *Schematics*, to simulate again the circuit (*F11*) for choosing the time domain analysis (*Transient*) in *PSpice A/D Demo* (to display the data, select *toggle cursor*, Fig.37).

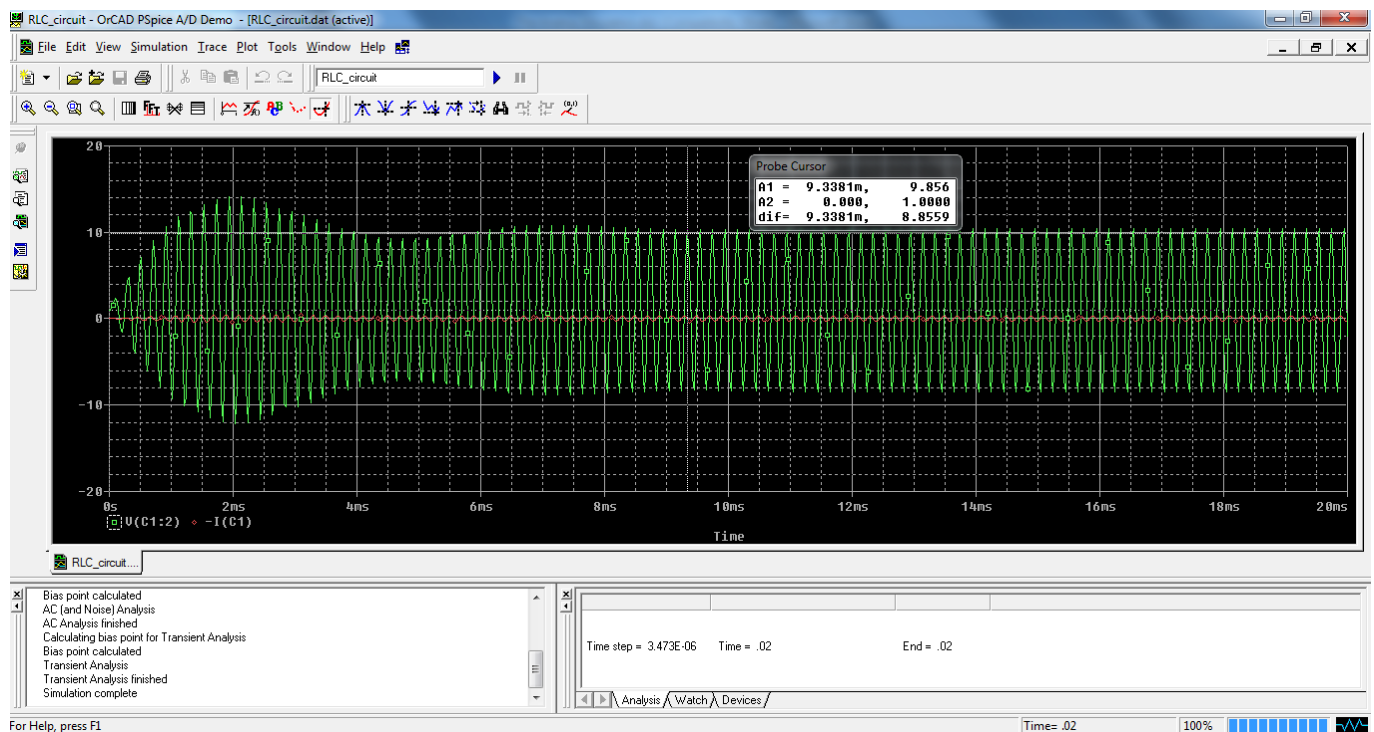


Fig.37 The current and the dropout voltage on the capacitor in the time domain

To display the V-I characteristic in the time domain (Fig.38), perform the same procedure as for the frequency domain measurement.

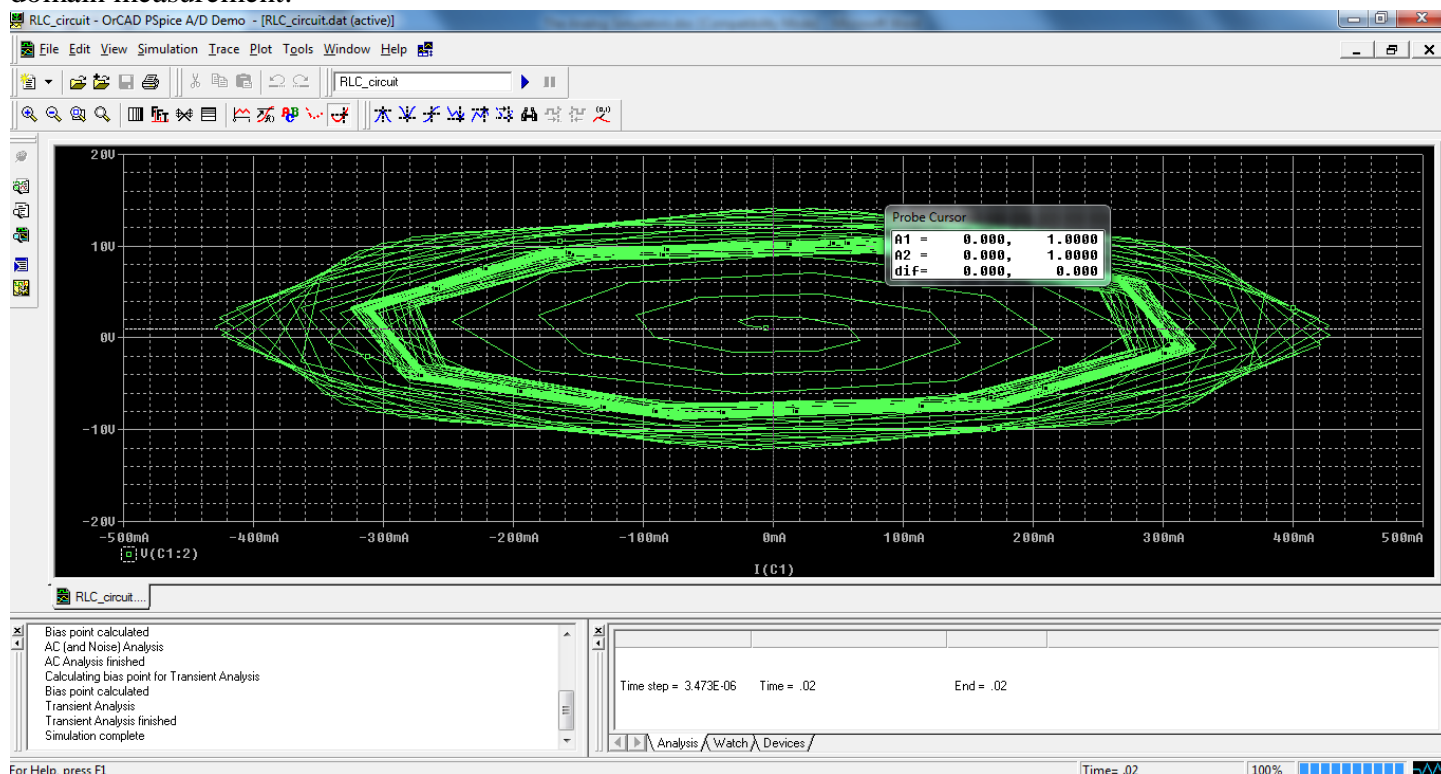


Fig.38 The V-I characteristic on the capacitor in the time domain

If the DC currents and voltages are necessary, you may select *Enable Bias Voltage Display* **V** and *Enable Bias Current Display* **I** in Schematics.

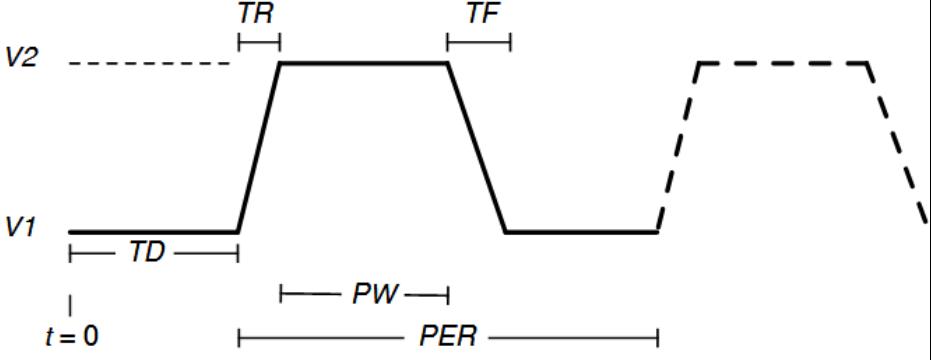
Exercises

1. Do the V , I and $V=f(I)$ simulations for the R, L, C and the supply voltage in the frequency domain
2. Do the V , I and $V=f(I)$ simulations for the R, L, C and the supply voltage in the time domain
3. Measure the period of the signal and calculate the frequency in the time domain
4. Measure the phase shift between V and I in the time domain.
5. Do a DC sweep for V1 DC source voltage in the range 0-1V with an increment of 0.1V. Measure the voltage on the capacitor by using the *PSpice A/D Demo* interface. What kind of function is the $V(C)=f(VI)$ characteristic?

Annex 1

Table 1

Po s.	Component type	Value	Library
1.	R (resistor)	Numerical value is taken from the laboratory platform. <ul style="list-style-type: none"> - Mili-ohms if <i>m</i> is written after the numerical value - Ohms if nothing is written after the numerical value - Kilo-ohms if <i>k</i> is written immediately after the numerical value - Mega-ohms if <i>meg</i> is written immediately after the numerical value 	Analog.slb
2.	C (capacitor)	Numerical value is taken from the laboratory platform. <ul style="list-style-type: none"> - Pico-farads if <i>p</i> is written after the numerical value - nano-farads if <i>n</i> is written after the numerical value 	Analog.slb

		<ul style="list-style-type: none"> - micro-farads if μ is written after the numerical value - mili-farads if m is written after the numerical value - farads if nothing is written after the numerical value 	
3.	L (inductor)	Numerical value is taken from the laboratory platform. <ul style="list-style-type: none"> - nano-Henry if n is written after the numerical value - micro-Henry if μ is written after the numerical value - mili-Henry if m is written after the numerical value - Henry if nothing is written after the numerical value 	Analog.slb
4.	Sw_tclose (normal open switch, closing after a set time)	Numerical value for the time is taken from the laboratory platform. <ul style="list-style-type: none"> - Pico-seconds if p is written after the numerical value - nano-seconds if n is written after the numerical value - micro-seconds if μ is written after the numerical value - mili-seconds if m is written after the numerical value - seconds if nothing or s is written after the numerical value 	Eval.slb
5.	Sw_topen(normal closed switch, opening after a set time)	Numerical value for the time is taken from the laboratory platform. <ul style="list-style-type: none"> - Pico-seconds if p is written after the numerical value - nano-seconds if n is written after the numerical value - micro-seconds if μ is written after the numerical value - mili-seconds if m is written after the numerical value - seconds if nothing or s is written after the numerical value 	Eval.slb
6.	VPULSE (used to function as a signal generator)	 <ul style="list-style-type: none"> - DC : DC component of the wave - AC: AC component of the wave - V1: initial voltage level. It is the value when the pulse is not "on." So for a square wave, the value when the wave is 'low'. This can be zero or negative as required. - V2: the value when the pulse is fully turned 'on'. This can also be zero or negative. (Obviously, V1 and V2 should not be equal.) - TD: the time delay. The default units are seconds. The time delay may be zero, but not negative. - TR: the rise time of the pulse. PSpice allows this value to be zero, but zero rise time may cause convergence problems in some transient analysis simulations. The default units are seconds. - TF: the fall time in seconds of the pulse. - PW: the pulse width. This is the time in seconds that the pulse is fully on. - PER: the period and is the total time in seconds of the pulse. 	Source.slb
7.	VAC (sine-wave voltage source for frequency)	<ul style="list-style-type: none"> - ACMAG: the RMS value of the voltage. - DC: the DC offset voltage - ACPHASE: the phase angle of the voltage (if left unspecified 	Source.slb

	domain analysis)	will be set by default to 0°).	
8.	VSIN (used to function as a signal generator, sine-wave voltage source for time domain analysis)	<ul style="list-style-type: none"> - DC: the DC component of the sine wave - AC: the AC value of the sine wave. - VOFF: the DC offset value (set to zero if you need a pure sinusoid). - VAMPL: the undamped amplitude of the sinusoid; i.e., the peak value measured from zero if there were no DC offset value. - FREQ: the frequency in Hz of the sinusoid. - TD: the time delay in seconds (set to zero for the normal sinusoid). - DF: damping factor (set to zero for the normal sinusoid). - PHASE: phase advance in degrees (set to 90 if you need a cosine wave form). <p>Note: the normal usage of this source type is to set VOFF, TD and DF to zero as this will give you a 'nice' sine wave.</p>	Source.slb
9.	VDC (simple DC voltage source)	<ul style="list-style-type: none"> - Value in volts. 	Source.slb
10.	GND_ANALOG	<ul style="list-style-type: none"> - Ground (node potential is 0 volts). - It is mandatory to be used in any PSpice schematic! 	Port.slb

Bibliography

1. Cadence Design Systems, PSpice User's guide, Second Edition, 31 May 2000, Portland, Oregon, USA
2. <http://bwrce.eecs.berkeley.edu/Courses/IcBook/SPICE/>
3. <https://en.wikipedia.org/wiki/SPICE>
4. http://denethor.wlu.ca/PSpice/pspice_tutorial.html
5. http://tuttle.merc.iastate.edu/ee201/spice/pspice_transient.pdf