

SPICE-Simulation using LTspice IV



Tutorial for successful simulation of electronic circuits with the free full version of LTspice IV (before named “SwitcherCAD”), available at Linear Technologies (www.linear.com).

Version 1.3

Copyright by
Gunthard Kraus, Elektronischule Tettang, Germany,
Guest Lecturer at the DHBW (Duale Hochschule Baden-Wuerttemberg),
Friedrichshafen, Germany

Email: krausg@elektronischule.de or Gunthard.Kraus@gmx.de

Homepage: www.elektronischule.de/~krausg

December 15th, 2010

| Contents | Page |
|---|-------------|
| 1. Introduction | 5 |
| 2. Installation | 5 |
| 3. Exercises with the included Example „Astable Multivibrator“ | 5 |
| 3.1. Using Schematic and Simulation | 5 |
| 3.2. Presentation of Simulation Results | 7 |
| 3.3. Deleting Curves | 8 |
| 3.4. Changing Curve Colours | 9 |
| 3.5. Changing the Simulation Time | 10 |
| 3.6. Changing Current- or Voltage Ranges | 12 |
| 3.7. Using Cursors | 14 |
| 3.7.1. Working with one Cursor | 14 |
| 3.7.2. Working with two Cursors | 14 |
| 3.7. Differential Measurements | 15 |
| 3.8. Current Measurements | 16 |
| 3.9. Changing Part Values | 17 |
| 4. Project 1: RC Lowpass Filter | 18 |
| 4.1. Drawing the Circuit Diagram with the Editor | 18 |
| 4.2. Changing Part Values | 19 |
| 4.3. Time Domain Simulation: non-repetitive Signals at the Input | 20 |
| 4.3.1. Step Response | 20 |
| 4.3.2. Switching ON and OFF | 22 |
| 4.3.3. Impulse Response | 23 |
| 4.4. Time Domain Simulation: using Periodic Signals at the Input | 26 |
| 4.4.1. Sine Wave (f = 1591 Hz) | 26 |
| 4.4.2. Square Wave (f = 1591 Hz) | 27 |
| 4.4.3. Triangle Wave (f = 1591 Hz) | 28 |
| 4.5. Frequency Domain simulation: AC Sweep | 29 |
| 5. FFT (= Fast Fourier Transform) | 31 |
| 6. Project 2: Rectifiers | 34 |
| 6.1. Simple Rectifier without Transformer | 34 |
| 6.2. Important: creating SPICE Model and Symbol for a Transformer | 35 |
| 6.2.1. The easiest Solution: a simple ideal Transformer | 35 |
| 6.2.2. Creation of the SPICE Model for a real Two Windings Transformer | 35 |
| 6.2.3. Creation of the Symbol for a Two Windings Transformer | 35 |
| 6.3. One Pulse Rectifier with Transformer | 38 |
| 6.4. Rectifier with Diode „1N4007“ | 40 |
| 6.5. Two Pulse Rectifier with Transformer | 42 |
| 7. Project 3: Three Phase AC-system | 44 |
| 7.1. Schematic and Simulation | 44 |
| 7.2. Rectifier Bridge for Three Phase AC Systems (= dynamo in a modern car or motorcycle) | 45 |

| | |
|--|-----------|
| 8. Project 4: U-I-Curves of Parts | 47 |
| 8.1. Resistor | 47 |
| 8.2. Diode | 48 |
| 8.3. NPN Transistor | 49 |
| 8.4. N Channel junction FET | 51 |
| | |
| 9. Project 5: Transistor Circuits | 52 |
| 9.1. One Stage Amplifier | 52 |
| 9.1.1. Time Domain Simulation | 52 |
| 9.1.2. Frequency Domain Simulation: AC Sweep | 54 |
| 9.2. Two Stage Broadband Amplifier with Feedback | 55 |
| 9.2.1. Task | 55 |
| 9.2.2. Circuit Diagram for Simulation | 55 |
| 9.2.3. Time Domain Simulation | 56 |
| 9.2.4. DC Bias | 57 |
| 9.2.5. Frequency Domain Simulation: AC Sweep | 59 |
| 9.3. Parametric Sweep | 60 |
| | |
| 10. Project 6: OPA Circuits | 62 |
| 10.1. Starting with an Inverting Amplifier | 62 |
| 10.2. Preparing a SPICE Model from the Internet | 62 |
| 10.2.1. Gainblock for 1kHz to 30 MHz with OPA355 | 64 |
| 10.2.2. Simulation using the selfmade OPA355 Subcircuit Model | 64 |
| 10.3. Usage of Labels | 67 |
| | |
| 11. Project 7: DC-DC Converters | 69 |
| 11.1. Model for the Power-MOSFET „IRFZ44N“ | 69 |
| 11.2. The Step Up Converter | 71 |
| 11.3. The Flyback Converter | 73 |
| 11.4. The Step Down Converter | 75 |
| | |
| 12. Project 8: Thyristor Circuits | 77 |
| 12.1. Thyristor Model | 77 |
| 12.2. Switching Resistive Loads | 78 |
| 12.3. Switching Inductive Loads | 79 |
| 12.4. Circuit with Gate-Transformer | 80 |
| | |
| 13. Project 9: Echos on Transmission Lines | 81 |
| 13.1. Transmission Lines -- only two Wires? | 81 |
| 13.2. Echoes | 83 |
| 13.3. Simulation of the Example with LTspice | 85 |
| 13.4. Open or Short Circuit at Cable's End | 88 |
| 13.5. Lossy Cables (e. g. RG58 / 50Ω) | 90 |
| 13.5.1. How can I simulate a RG58 Coaxial Cable? | 90 |
| 13.5.2. Simulation of Cable Loss at 100MHz | 91 |
| 13.5.3. Feeding the RG58 Cable Entry with a Pulse Voltage | 94 |
| 13.5.4. A Short Circuit at RG58 Cables's End | 95 |
| | |
| 14. Project 10: S-Parameters | 96 |
| 14.1. Echoes once again, but with more System (= S parameters) | 96 |
| 14.2. Example: 110MHz Tchebyshev Lowpass Filter (LPF) | 99 |

| | |
|--|------------|
| 15. Project 11: Double Balanced Mixer | 103 |
| 15.1. Fundamentals and Informations | 103 |
| 15.2. The Ring Modulator | 104 |
| 15.3. The necessary Transformers | 105 |
| 15.4. DBM Simulation with ideal Transformers | 106 |
| | |
| 16. Project 12: Simulations with Digital Circuits | 108 |
| 16.1. What you should know before | 108 |
| 16.2. Simple start: the inverter (= NOT) | 109 |
| 16.3. AND- and NAND-Gate | 110 |
| 16.4. D Flipflop | 111 |
| 16.5. Three Stage Frequency Divider with D Flipflops | 112 |
| | |
| 17. Project 13: Noise Simulation | 113 |
| 17.1. Fundamentals | 113 |
| 16.1.1. „Noise“ -- where does it come from? | 113 |
| 16.1.2. Other Sources of Noise | 115 |
| 16.1.3. Noise Temperature and Noise Factor for a Twoport System | 116 |
| 17.2. Simulation of the Spectral Noise Density | 116 |
| 17.3. Simulation of the Noise Figure NF (in dB) | 119 |
| | |
| 18. Simulation of a Sine Oscillator | 123 |
| | |
| 19. Signals and Harmonics | 128 |
| 19.1. Fundamentals | 128 |
| 19.2. Simulation of a Single Pulse Spectrum | 131 |
| 19.3. Simulation of a Periodic Pulse Spectrum | 133 |
| 19.4. An Ideal Sine Wave | 134 |
| 19.5. The „One Side Clipped“ Sine Wave | 135 |
| 19.6. The Symmetrically Clipped Sine Wave | 136 |
| | |
| 20. The Secret of the Impulse Response | 137 |
| 20.1. First example: Dirac Pulse applied to a 160Hz RC Lowpass Filter | 138 |
| 20.2. Second Example: Dirac Test of a 110MHz Lowpass Filter (see Chapter 14.2) | 142 |
| 20.2.1. Simulating S21 (= Forward Transmission) | 142 |
| 20.2.2. Simulation of S11 (= input reflection) or S22 (= output reflection) | 147 |
| | |
| 21. Modulation | 149 |
| 21.1. Principle of Amplitude Modulation (AM) | 149 |
| 21.2. Amplitude Shift Keying (ASK) | 151 |
| 21.3 Frequency modulation (FM) | 152 |
| 21.3.1. Generating an FM signal using the SFFM voltage supply | 152 |
| 21.3.2. Frequency Shift Keying (FSK) | 154 |

1. Introduction

Modern electronics needs circuit simulation -- only in this manner you can save time, cost and effort when designing new or modifying existing circuits. Every new idea can be tested without a real printed circuit board or a soldering iron. Therefore the "SPICE" program was developed before 1980 at the Berkeley University, running on FORTRAN machines. But the more important version for the „normal“ user is **PSPICE** (= SPICE for the PC).

Around the „SPICE Kernel“ lot of people have programmed shells and programs for simple and intuitive usage and so you can find lot of software offers on the market. Most of the available programs are excellent, but huge and expensive and so we have to say „Thank You“ to Linear Technologies. They offer a free full SPICE-program named „LTspice“ without any restrictions. It was foreseen to simulate switching power supplies using the semiconductors of the enterprise.....but can also be used for nearly other electronic purpose. It can be downloaded from the web without any problems or fees but the usage is a little tricky -- a mixture of command lines, menus and mouse clicks. So it needs a lot of effort before the first own simulation. And this was the reason for an old professor like me to write this tutorial for other people.

Warning for the new user:

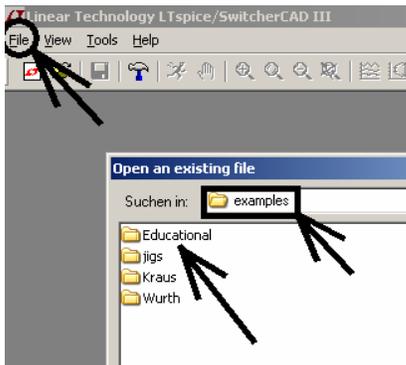
There are sometimes several ways to get to the same command (hot keys, menu bar symbols, menu selection, menus that appear with a right mouse click, etc.). So find out the way which you personally prefer and don't be afraid of or confused by the different kinds of run commands.

2. Installation

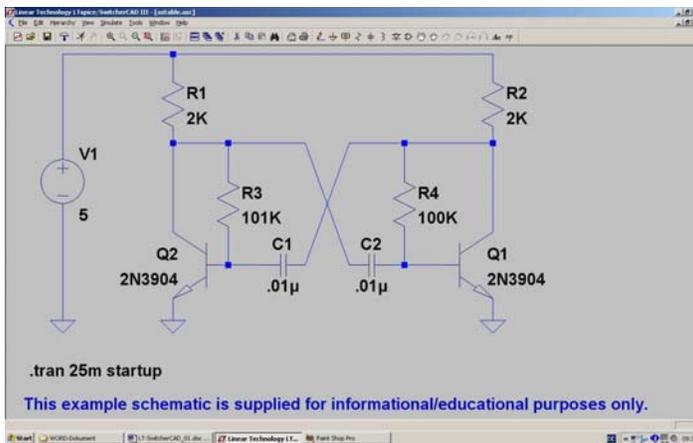
No problem: download the zipped software from the homepage (www.linear.com), click on the „.exe“-file and you will be guided by the program. An icon is created at last on your Windows-screen.

3. Exercises with the included Example „Astable Multivibrator“

3.1. Using Circuit and Simulation



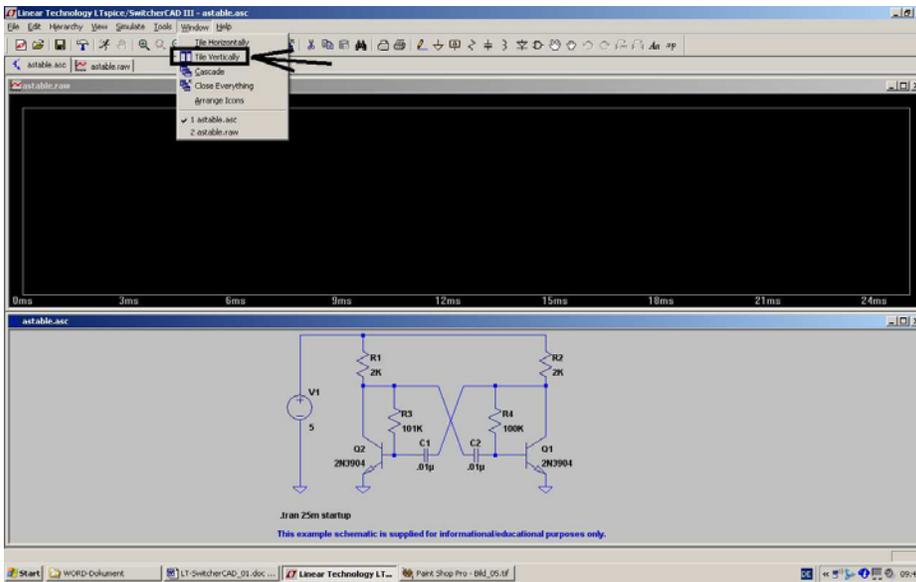
Click on „File“ in the upper left corner and then open „examples“. In the folder „educational“ you find the file „astable.asc“. Open it.



So the screen with the schematic should now look like.

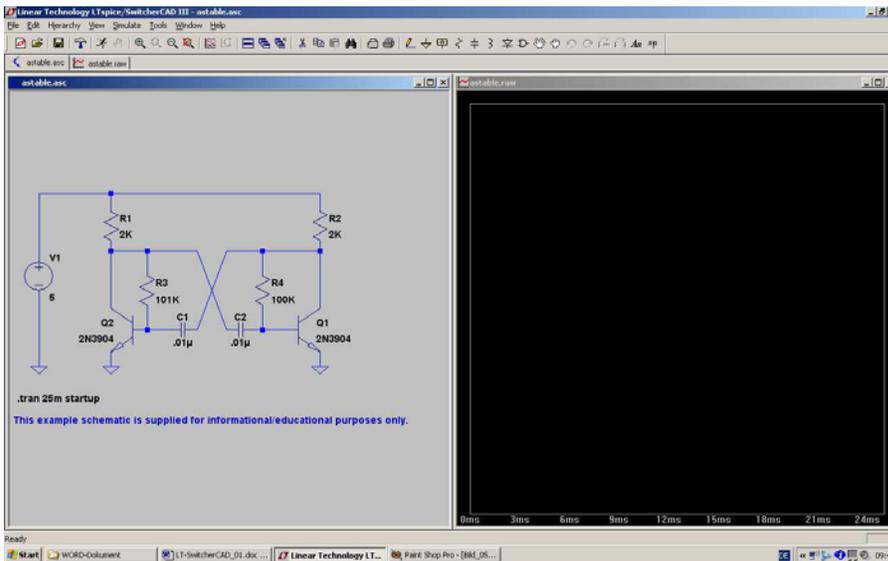


Now look for the button with the „running guy“. This starts the simulation.



The simulation is done, but where are the results? (The diagram in the upper half of the screen is still empty....)

If you prefer another manner of presentation, open the menu „Window“ and click on „Tile vertically“.



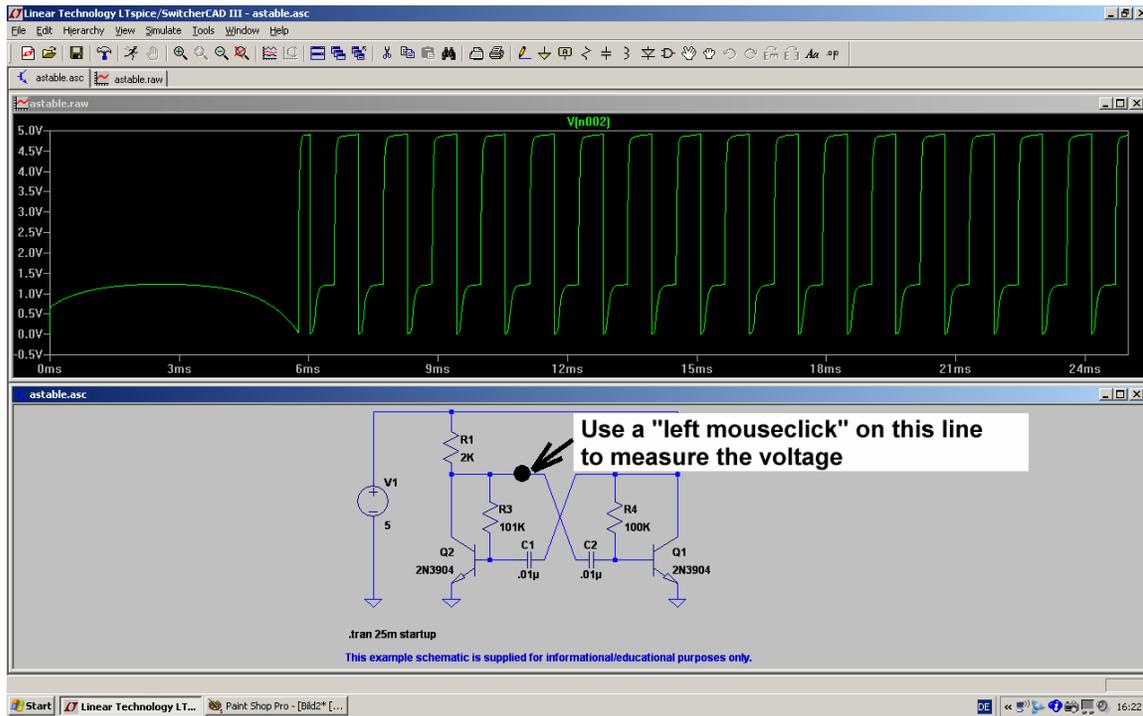
Better now for you?

Information:

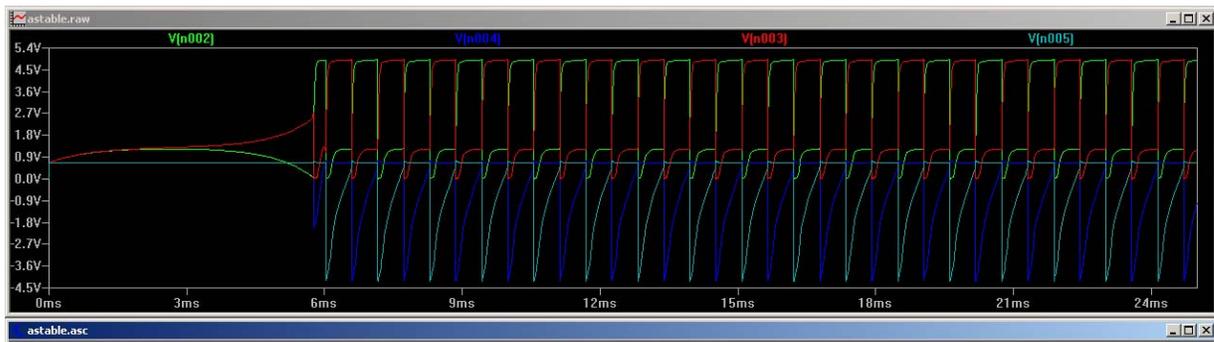
Who likes a result plot with thick lines for the curves must now click on the “button with the hammer”. There he finds a menu “Waveforms” and in it “Plot data with thick lines”.

3.2. Presentation of Simulation Results

Move the cursor to the point or line in the schematic of which you want to know the voltage. When arriving the cursor changes its form to a „testclip“. Now a simple mouseclick activates the presentation of the desired curve:



You can continue in this manner, but remember: you have only one diagram. And how is it possible to separate the different curves of the different points? Where can you get the informations from? For instance so the presentation of all base- and collector-voltages looks like:



In principle you have a choice of 3 possibilities:

```

SPICE Netlist: C:\Programme\LTC\SwcADIII\examples\Educational\astable.net
* C:\Programme\LTC\SwcADIII\examples\Educational\astable.asc
R1 N001 N002 2K
R2 N001 N003 2K
R3 N002 N004 101K
R4 N003 N005 100K
C1 N003 N004 .01µ
C2 N005 N002 .01µ
V1 N001 0 5
Q1 N003 N005 0 0 2N3904
Q2 N002 N004 0 0 2N3904
.model NPN NPN
.model PNP PNP
.lib C:\Programme\LTC\SwcADIII\lib\cmp\standard.bjt
.tran 25m startup
* This example schematic is supplied for informational/educational purposes onl
.backanno
.end
    
```

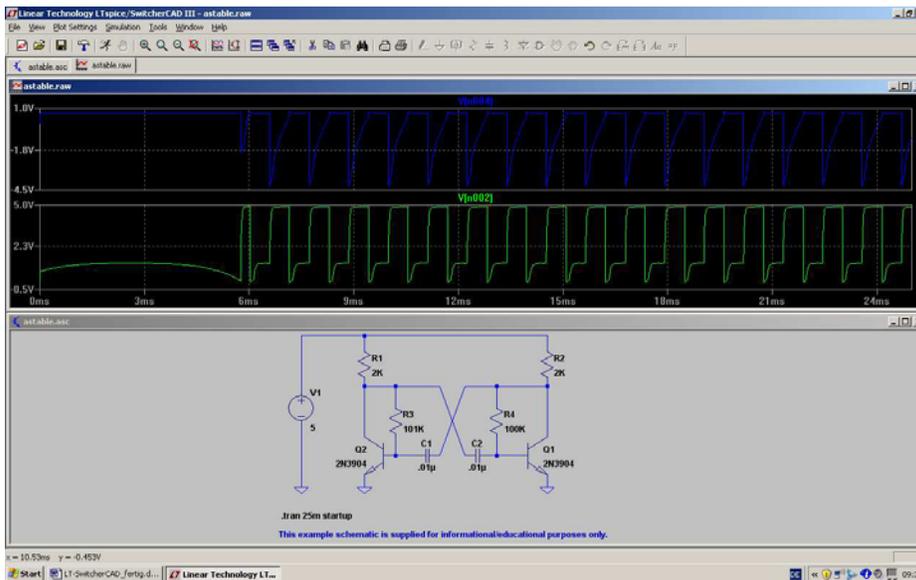
a) Look for the node's name in the „**Netlist**“, because the different curves are named „**V(node)**“. You find this important list in the menu „**VIEW**“ and „**SPICE Netlist**“.

This is the netlist for our example and easy to read. Every line belongs to a part in the schematic and begins with the part's name, followed by the nodes between the part is inserted. At last you find the part's value.



b) Much easier is the cursor method. Move the cursor to the desired point and look for the cursor's change to a testclip.

In this moment you get a message in the lower left corner of the screen with the node's name



c) Much easier is to use separate diagrams for the different curves.

Right click right on the presentation window and choose

„Add Plot Plane“.

Now you get an additional diagram. Left click on it with the mouse to activate it. Afterwards click on the point in the schematic (f. i. the base of the left transistor) to present the desired

voltage
If necessary, repeat this procedure to get more diagrams.

But how to delete undesired presented curves: see the following chapter!

3.3. Deleting Curves

This is important to know if you present too much curves in only one diagram and you are loosing the overview. So

- a) Left click on the diagram to activate it -- the frame colour should change to a dark blue.
- b) Press **F5** and the cursor will change to scissors.
- c) Move the scissors to the name of the undesired curve -- e. g. "V(n001)" above the diagram and click on it. The curve will disappear at once.

Important:
To measure another voltage in the schematic, please press at first **„Escape“** on the keyboard to leave the delete function. Then click left on the schematic and watch that its frame changes to dark blue. If this is true, move the cursor to the desired point in the schematic until the form changes to a test clip...and so on....

3.4. Changing Curve Colours

A very simple exercise.....

The image shows a screenshot of a circuit simulation software interface. The top part of the screen displays a waveform plot with two signals: a green square wave and a blue sine wave. The x-axis is labeled with time values 9ms and 12ms. Below the plot is a circuit diagram showing a voltage source V1, a resistor R1 (2K), and another resistor (2K). A red circle highlights a context menu that appears when the signal name 'V(n005)' is right-clicked. The menu includes a 'Default Color' dropdown set to blue, an 'Attached cursor' field set to '1s', an 'Enter an algebraic expression to plot' field containing 'V(n005)', and a color selection palette with various color options. A red arrow points from the text '.....then choose Your desired colour!' to the color palette. Another white circle highlights the signal name 'V(n005)' in the plot area, with a white arrow pointing to the text 'At first click right on the Signal's name.....'.

At first click right on the Signal's name.....

.....then choose Your desired colour!

Expression Editor - F(V(n005),...)

Default Color: [blue] Attached cursor: 1s

Enter an algebraic expression to plot: [default]

V(n005)

Delete this Trace

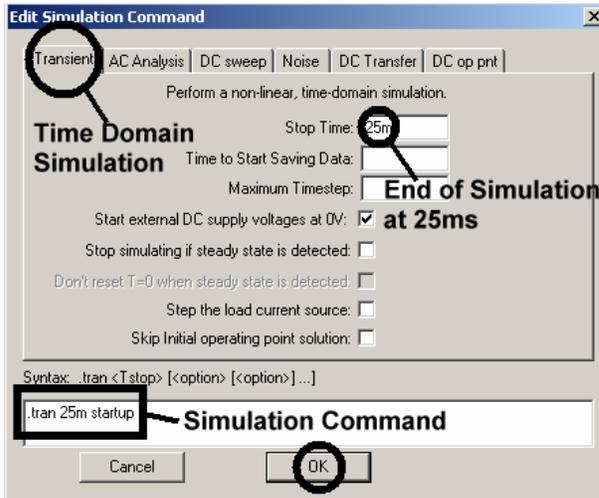
3.5. Changing the Simulation Time

.tran 25m startup

This simple line is the simulation command! It means::

Simulate all signal curves from 0 to 25ms.

(The simulation will always start at zero).

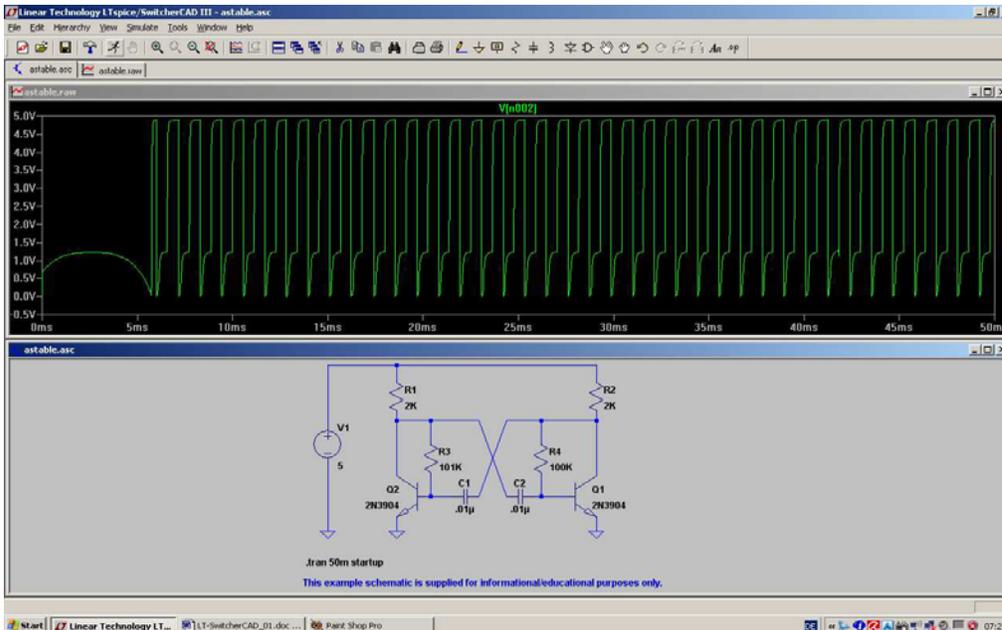


If you now move the cursor to the text of this line and right click, then you'll get this menu.

Please enter a new value for "Stop Time" if you want a longer or shorter simulation time, press OK and finally the simulation button (with "the running guy" on it) to start PSPICE.

Task:

Please choose a stop time of 50ms and simulate the collector voltage V(n002) at the left transistor



This should be the result.

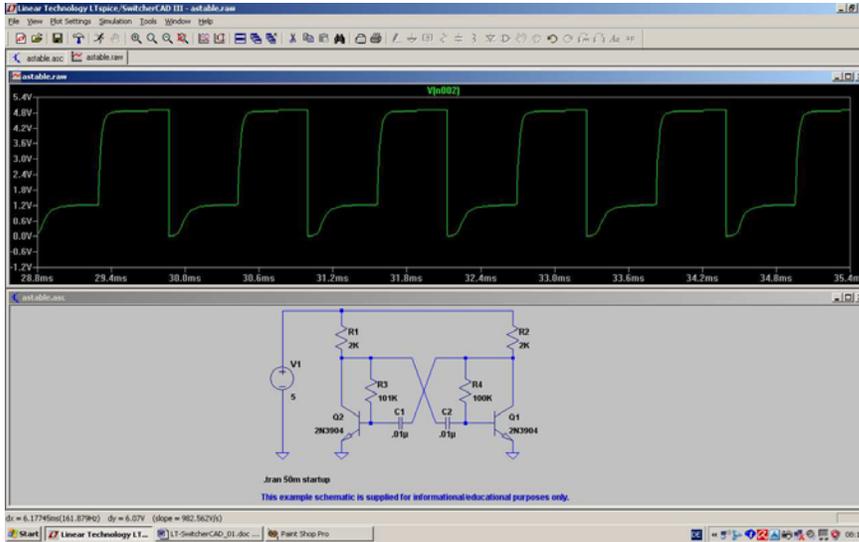
You can see that the simulation of the circuit behaviour after the start is a very important point -- e. g. for the analysis of switching power supplies (...and the parts, delivered by Linear Technologies...). But if you are only interested in the „steady state“, then you should use the zoom function.

Task:
Make the voltage curve visible between 30 and 35ms.

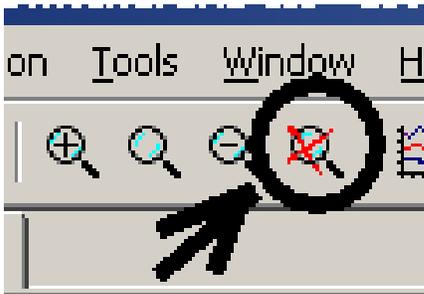
Solution:



a) Press this „enlarge button” and afterwards drag the region between 30 and 35ms with the pressed left mouse button.



b) When releasing the mouse button you should see this diagram.



c) If you want to go back to the full screen presentation press this button.

3.6. Changing Current or Voltage Ranges

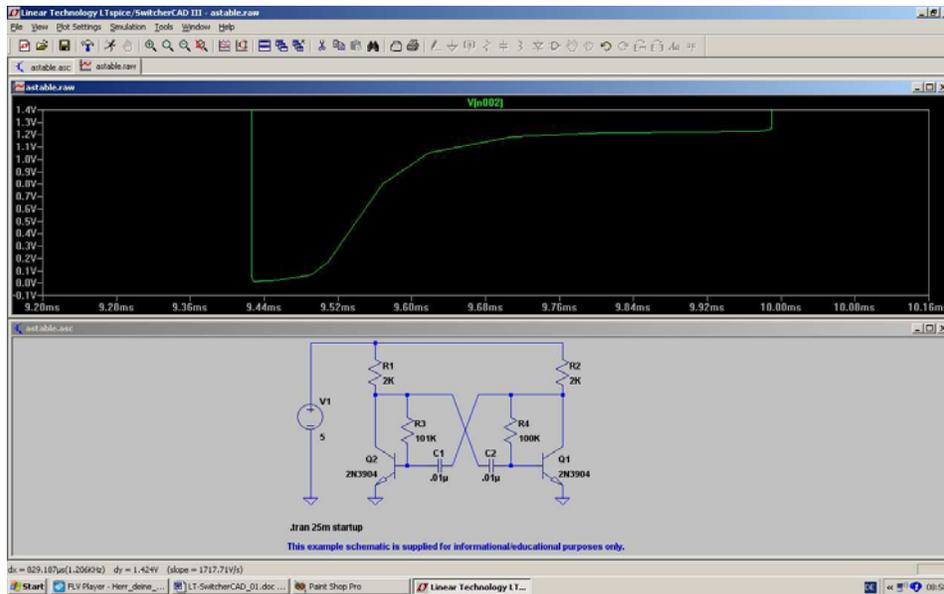
Very often you want to examine little details of a curve. So you have 2 possibilities:

- a) Use the discussed zoom function from chapter 3.4.
- b) Switch from "Autorange" to „Manual Limits“.

Example for solution a):

Task:

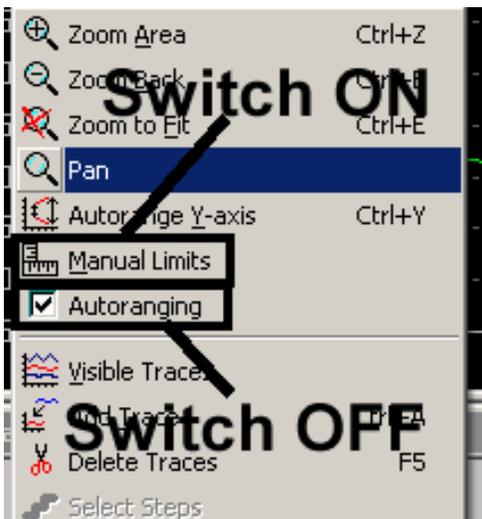
Show only the short negative „peak“ of the collector voltage by zooming:



Example for Solution b):

Task:

Set the values for the vertical axis to the range 0.....+1.5V.



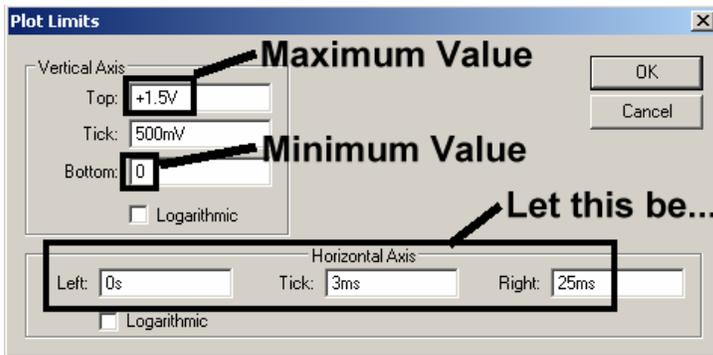
Step 1:

RIGHT click on the calibration of the vertical axis in the diagram and you get this menu for **Manual Setting on BOTH AXIES**.

(Attention:

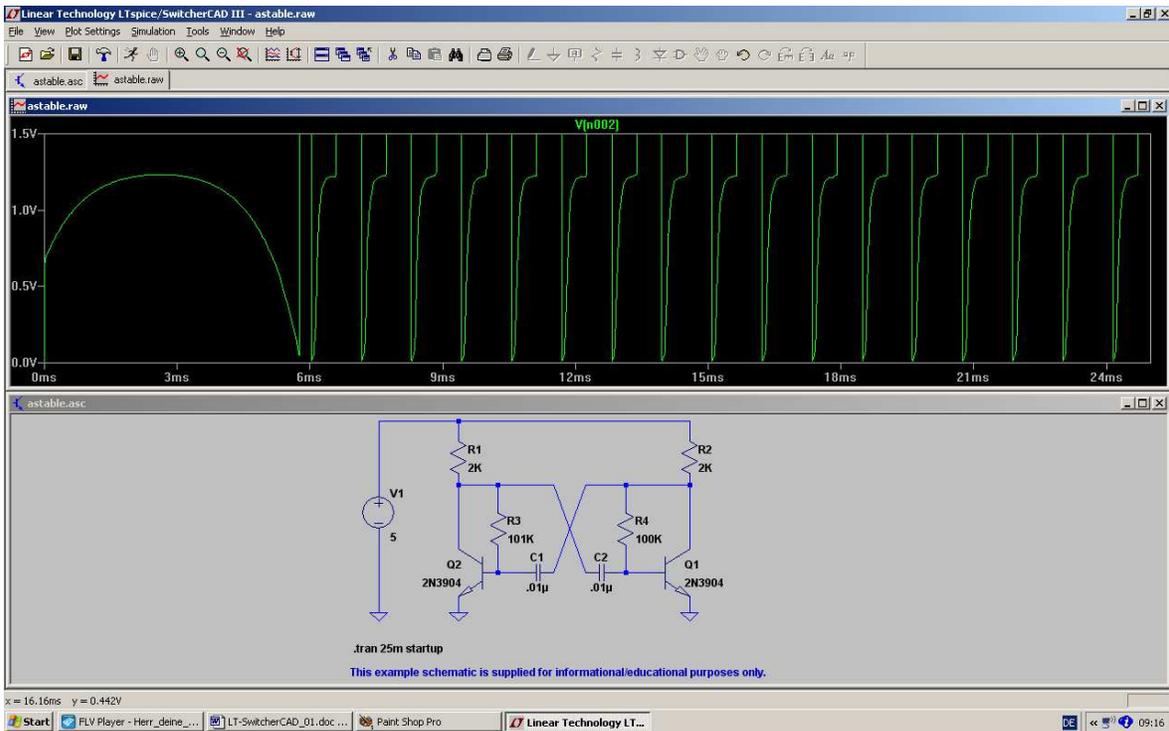
If you left click on the axis scaling, you'll only see the „quick solution menu“. Please test this, but we want to have a look on the „big solution“ and continue with the case of the right mouseclick).

Now switch off "Autoranging" and activate „Manual Limits“ .



Step 2:
Enter the new border values for the vertical axis.

If you want you can also modify the horizontal time axis. But at this moment: let it be...

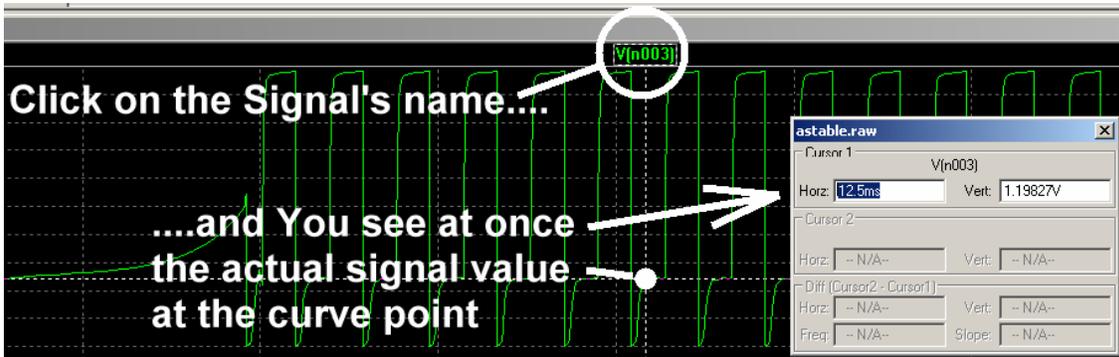


This is the goal when returning to the full screen presentation including the schematic, where you now -- if you want! -- can modify part values.

3.7. Using Cursors

3.7.1. Working with one Cursor

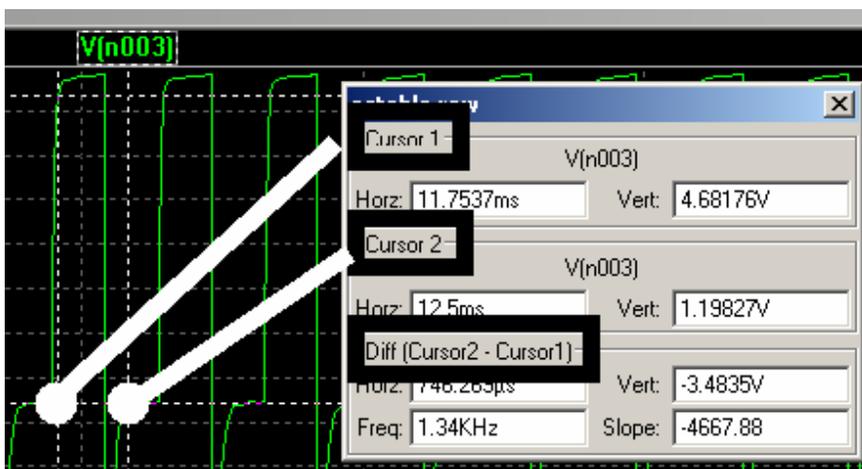
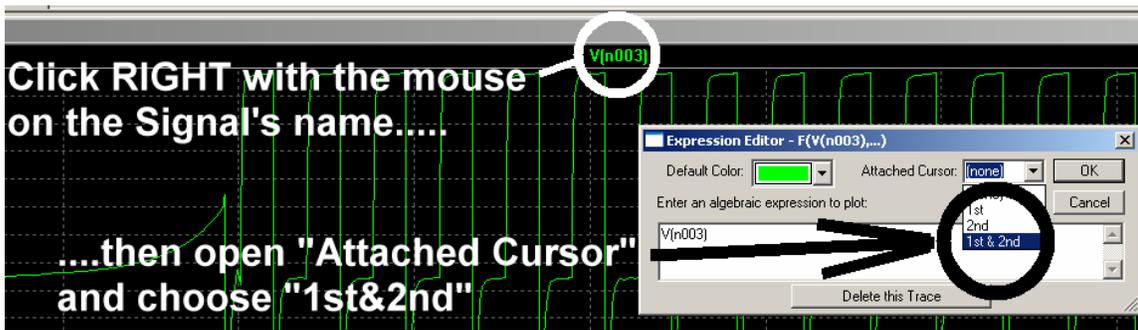
Measuring the signal value at special points is easy with a cursor:



If you now move the mouse to the marked point, you suddenly see the number of the cursor (here: „1“) hanging on your mouse. By “dragging” (= pressing the left mouse button + moving the mouse) it is possible to shift the cursor along the signal's curve. The actual curve value is then indicated in the window.

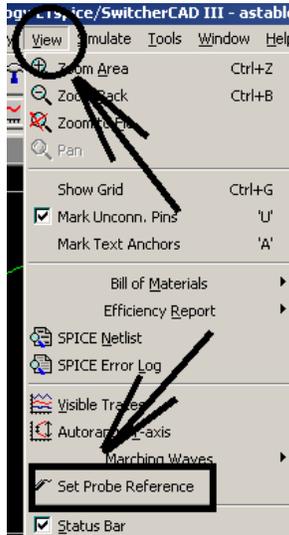
3.7.2. Working with two Cursors

RIGHT click on the signal's name and open „Attached Cursor“ in the menu:



3.7. Differential Measurements

Every measured and presented voltage is referred to a REFERENCE POINT (normally: ground). But how can we get the voltage between two points within the schematic? In this case you need a „differential voltmeter“, but where can you find that?

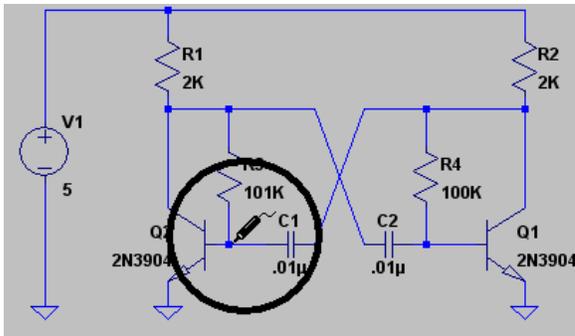


Sorry:
LTspice does not use a differential voltmeter like other SPICE programs!

You must change the REFERENCE POINT for this purpose. That means that you at first must set a new reference point at one node and then click onto another node!

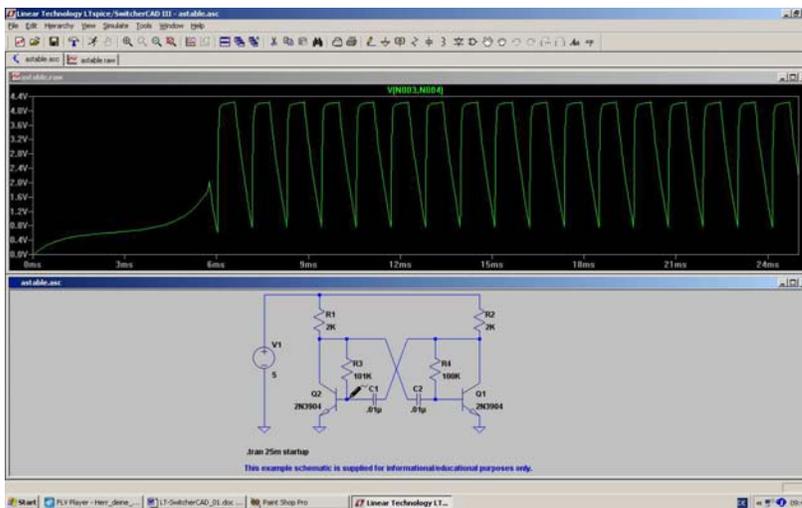
So please open the menu „View“ and activate „Set Probe Reference“.

Now we want to see the capacitor C1's voltage (= base capacitor of the left transistor).



After OK you now have a “testclip” as cursor. Click with it on the base of the left transistor.

Then click on the collector of the right transistor (= other side of capacitor C1). That is all!



Now use -- if you want! -- the zoom function for interesting details.

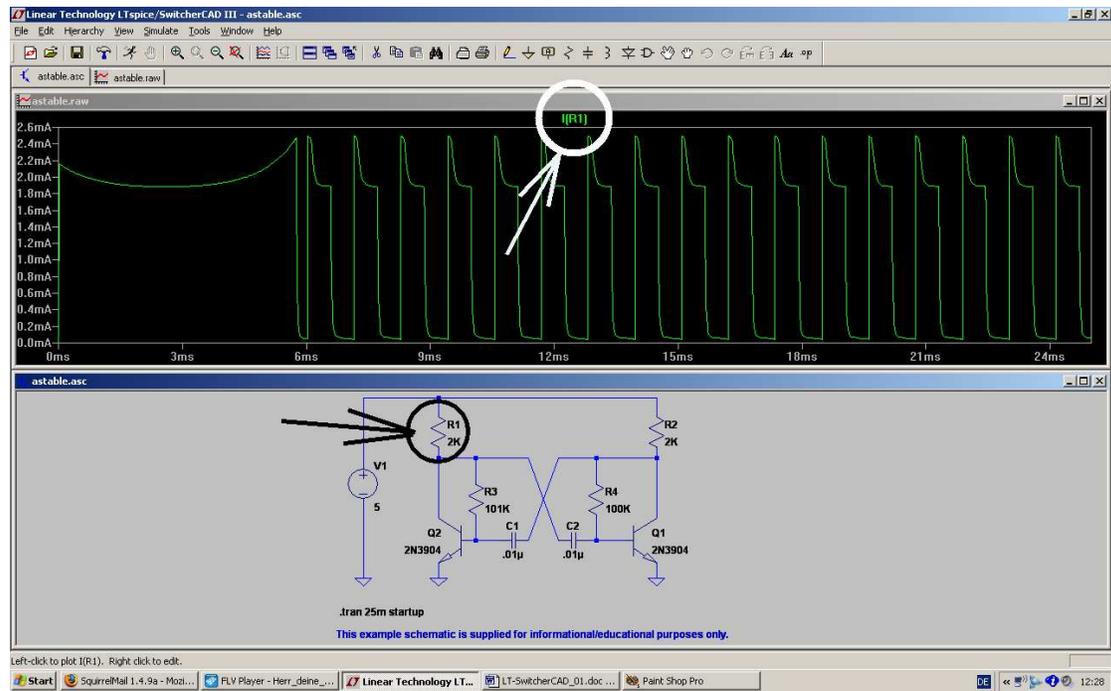
To leave this “differential measurement option” simply click on ESCAPE.

3.8. Current Measurements

Make the circuit diagram window active (...by a left mouseclick on it....) and move the cursor exactly **over the part** of which you want to know the current. The cursor changes to a “current sensor” and with a left mouseclick on the part you see the current curve.

Example:

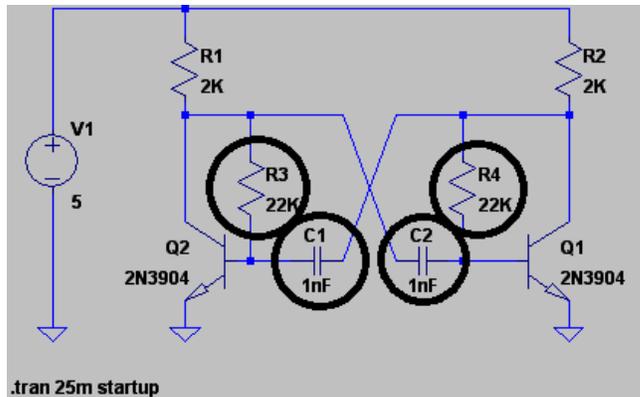
Current in resistor R1 (= Collector resistor of the left transistor):



3.9. Changing Part Values

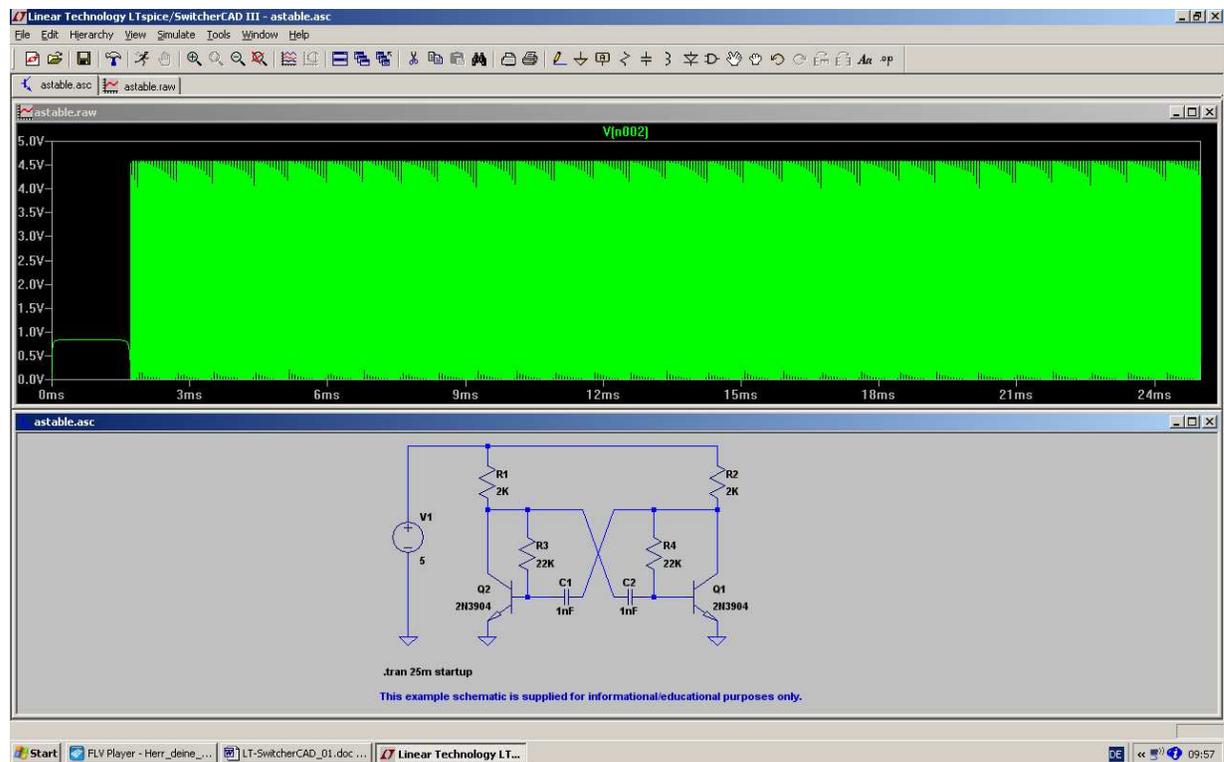
Now we want to repeat the simulation with different part values. Let every capacitor be $C = 1000\text{pF}$ and the both base resistors of the transistors $R3 = R4 = 22\text{k}\Omega$.

Move at every part the cursor exactly on its **value indication** and **right click with the mouse**. So you get the property menu and you can modify the value. Close with OK.



This is the new circuit to be simulated....

....and this should be the simulation result:

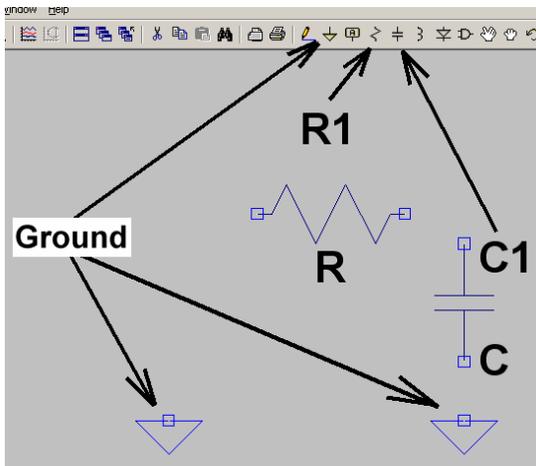


Please use the zoom function or modify the simulation time to see the details or to measure the new oscillation frequency.

4. Project 1: RC Lowpass Filter

4.1. Drawing the Circuit Diagram with the Editor

Open a new file and save it in a new folder (e. g. named „RC-LPF_01“).



Then press the „resistor button“ to get the resistor, turn the symbol left with

<CTRL> + <R>

for 90 degrees and place it on your sheet. Leave this function either by a right mouseclick or by pressing „ESCAPE“ on the keyboard.

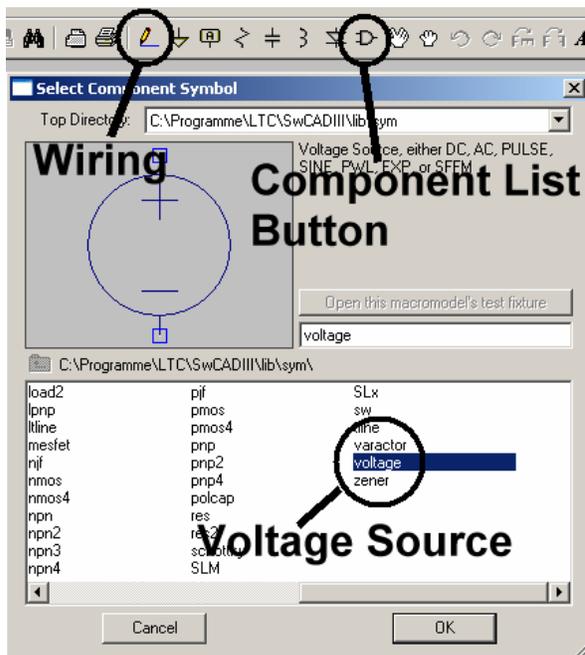
Repeat this procedure for the capacitor and two ground symbols.

Two tips:

- To delete a part or a connection in the circuit, press **F5**. At once you get scissors as cursor. Move these scissors over the symbol (or the wiring connection) and left click: the part or the connection disappears at once.
- To move a part, press **F7** and the cursor changes to a hand. Move this hand over the symbol of the part and left click. Now the part is connected to the cursor and can be moved with the mouse. A new left mouseclick will fix it in the new position.

Do never forget:

Leaving these special functions is done by pressing the ESCAPE KEY on the keyboard!



We still need a voltage source and you must therefore open the „component list“ by pressing the special button.

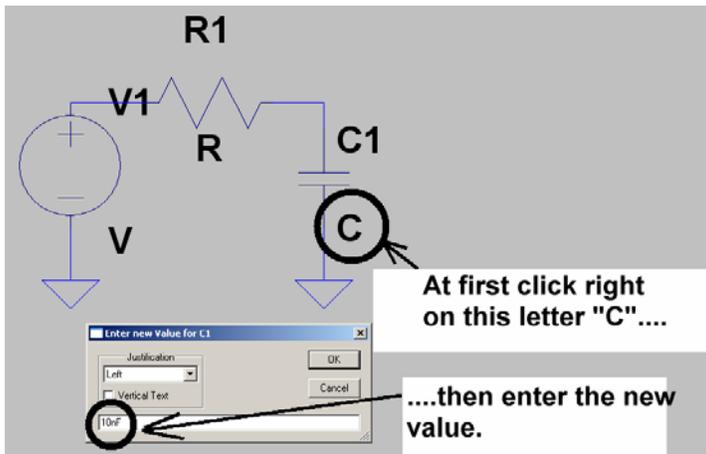
In the appearing list please look for „voltage“, mark it with a mouseclick and press OK. The symbol of the voltage source is then hanging on the cursor and can be placed within the schematic.

Now we can start with wiring. Press the button with the pencil, roll the cursor to the start of the connection, click left and roll him to the other point. One more left mouseclick and the connection is perfect!

Important:

For an „edged“ wiring start as before, roll to the edging point and left click. Then you can do the rest of wiring in another direction.

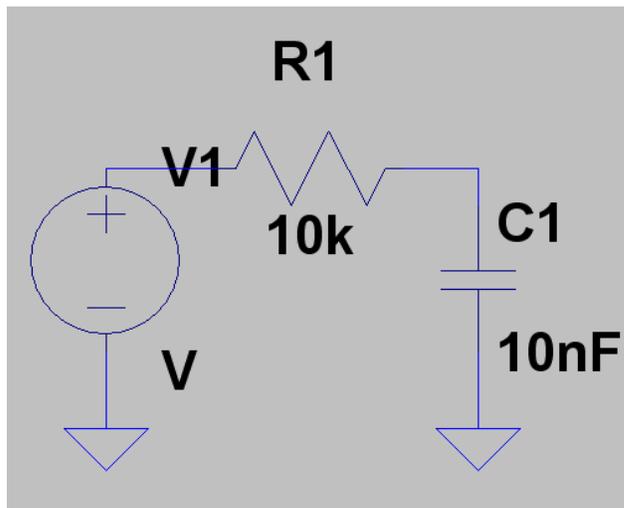
4.2. Changing Part Values



This is demonstrated by capacitor C1:

Move the cursor on the letter "C" and right click. Then enter the new value and close with OK.

Please change the value of the resistor R1 to 10k in the same manner.



So the circuit should now look like.

But: before programming the voltage source we must decide wether we want a

time domain

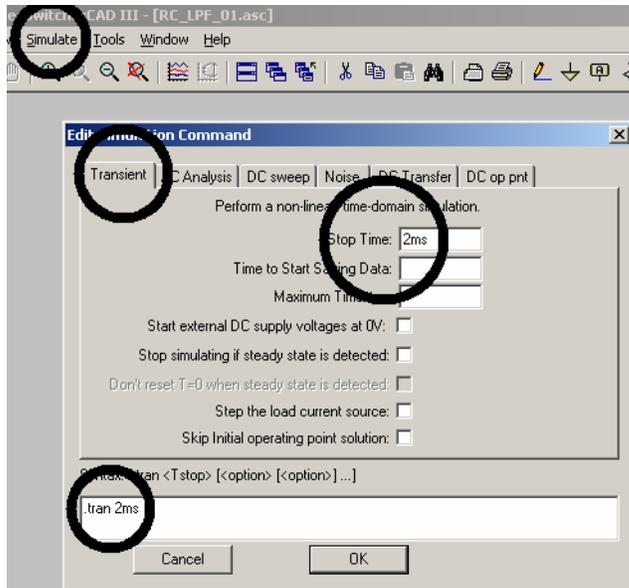
or a

**frequency domain-
simulation!**

4.3. Time Domain Simulation: Non Repetitive Signals at the Input

4.3.1. Step Response

Please use a **transient simulation in the time domain** and let the input voltage jump in a very short time from 0 to +1V.

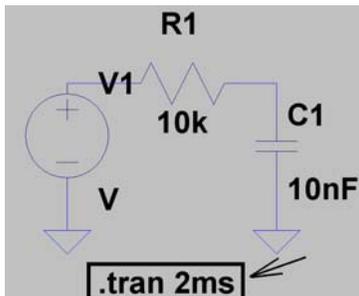


Click on the button „**Simulate**“, then „**Edit Simulation Command**“ and „**Transient**“

Enter a **stop time of 2ms** and check that this was accepted by filling the lower empty field with

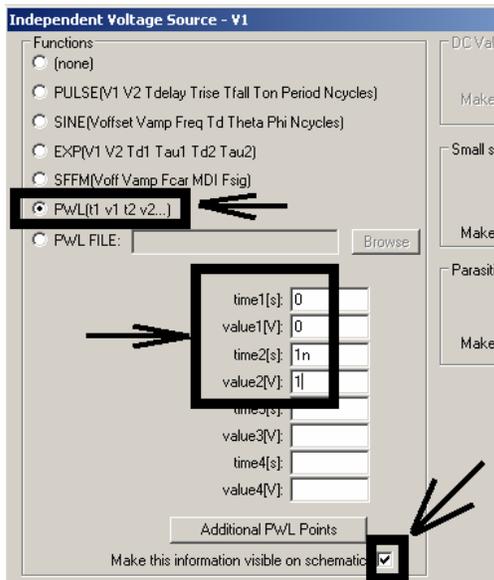
.tran 2ms

Then press OK.



This command line hangs now on the cursor and must be placed by a mouse click somewhere in the Editor screen.

To get the correct input voltage, right click on the symbol of the voltage source to open the property window. There open the „**ADVANCED**“ – menu.



The input voltage curve is generated by a series of linear lines (= „pieces“) between fixed points. This is done by the

PWL = piecewise linear

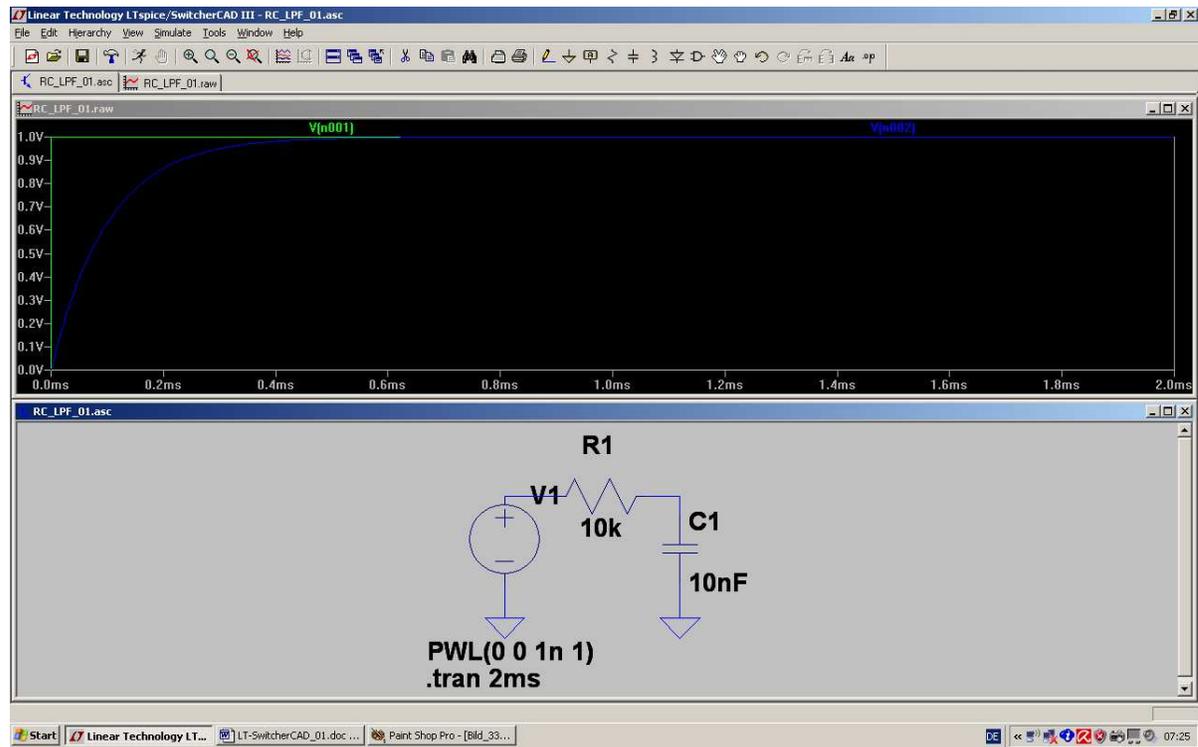
option, which now must be activated.

At time1 = 0 the voltage is also 0V.
At 1 ns the voltage value has risen to +1V and this value must not change.

Please enter these two pairs of values in the list of the menu, make the information visible on the screen and close with OK.

Place this information on the screen. Move it -- if necessary! -- by pressing F7, clicking on it and dragging with the mouse. Leave the F7 function by pressing the ESCAPE key on the keyboard.

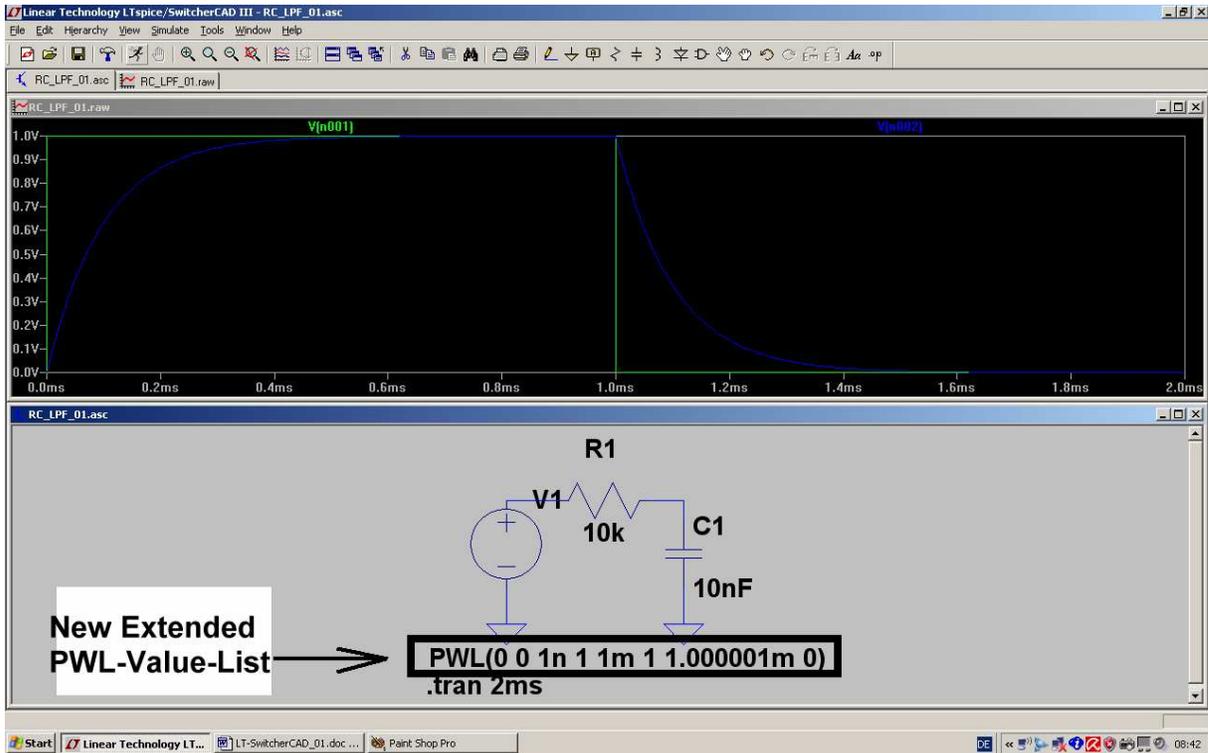
Now start the simulation, click on the input resp. output point in the schematic and admire the result:



4.3.2. Switching ON and OFF

No problem because you have only to extend the value-pair list of the PWL source:

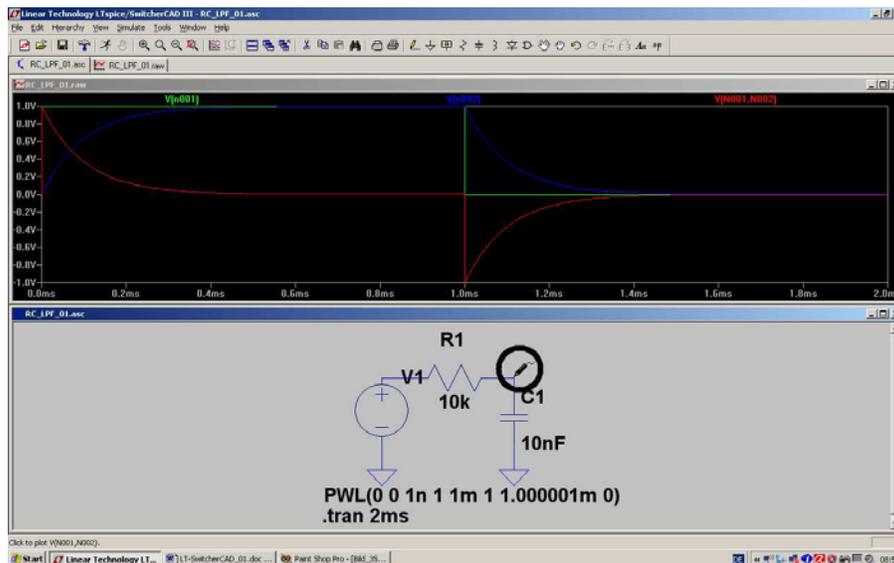
| | | |
|-----------|---|--------------------------------------|
| 0 | 0 | (Zero Volt at start point) |
| 1n | 1 | (+1 Volt after 1 nanosecond) |
| 1m | 1 | (still +1 Volt during 1 millisecond) |
| 1.000001m | 0 | (Within 1 nanosecond back to 0V) |



Task:

Add now the voltage at the resistor to this diagram!

Solution: add a reference point to the output (= connection between the resistor and capacitor). Touch the connection between voltage source and resistor with the cursor and click. At once the desired resistor voltage is presented in the diagram.



4.3.3. The Impulse Response

Fourier, Fast Fourier and Laplace Transformations are important tools for modern Communication or Control Techniques. With these tools you can get all informations about the system's reaction to any input voltage form -- if you know the **Transfer Function**.

This transfer function is the ratio of the output sine voltage to an applied input sine voltage, but presented in a complex "magnitude + phase" – form.

So let us have a look at the **impulse response**:

If you apply a „Dirac - Impulse“ to the input of an unknown system, then the answer (= Impulse Response) at the output will contain every information about this system -- because this answer is a "time domain form" of the transfer function.

And with the transfer function you can -- by "convolution" -- calculate the output signal for every possible input signal.

The Dirac Impulse is a little crazy: the amplitude is infinite, the pulse length is infinitely short...but the "pulse area" (= magnitude x pulse length) is "1".

This is impossible to generate but you can use a definite limited amplitude and a definite short pulse length -- only be shure that you have again a pulse area of "1" and the pulse length is much shorter than the system's time constant (...a factor of 100 to 1000 will do the job). And in real linear systems it is allowed to reduce the pulse magnitude to a value which is not dangerous for the system itself. Afterwards simply find out the necessary correction coefficient to get again the "ideal" values.

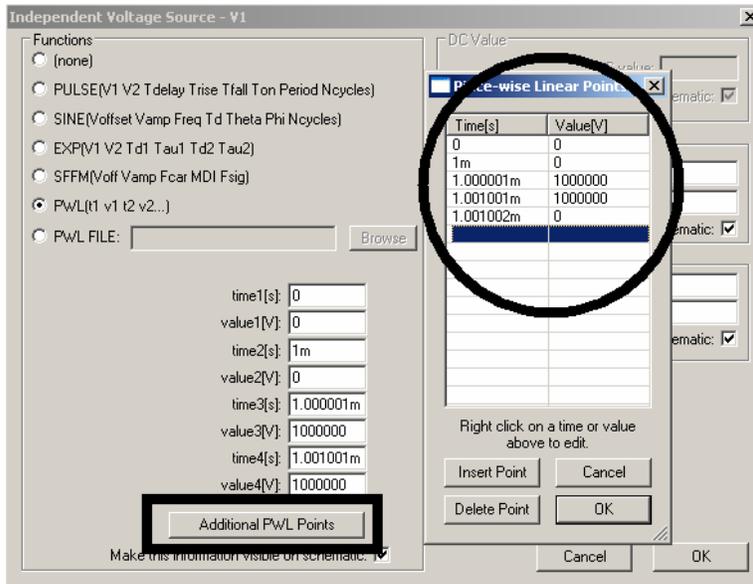
Let us have a look at our lowpass filter.

Task 1): Apply at the input of the LPF (R = 10kΩ, C = 10nF) a Dirac impulse with a peak value of V_{ampl} = 1 Megavolt. Start with a delay time of 1 ms at an amplitude of 0V. Then the pulse rises to 1 Megavolt within 1 Nanosecond, stays on this value for 1 Microsecond and decreases to 0V within 1 Nanosecond.
So we get a pulse area of

$$1 \text{ Megavolt} \times 1 \text{ Microsecond} = 1 \text{ Voltsecond}$$

Simulate the impulse response for a stop time of 2ms

Task 2): Reduce the pulse amplitude to 100 Kilovolt but use a pulse length of 10 Microseconds to get again a pulse area of 1 Voltsecond. Simulate and compare the result with task 1.



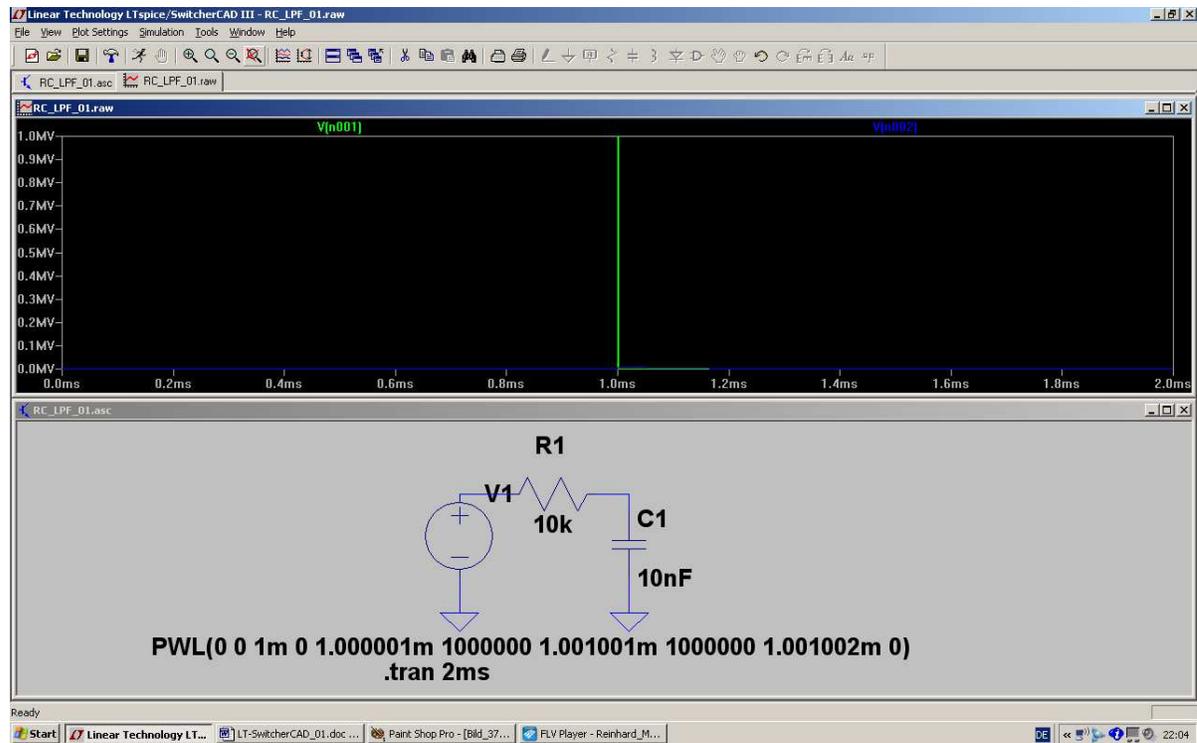
Solution for Task 1:

First right click on the symbol of the voltage source to open the property menu. Then fill the PWL-list with the informations about the additional points and value pairs (use the button "Additional PWL Points"):

```

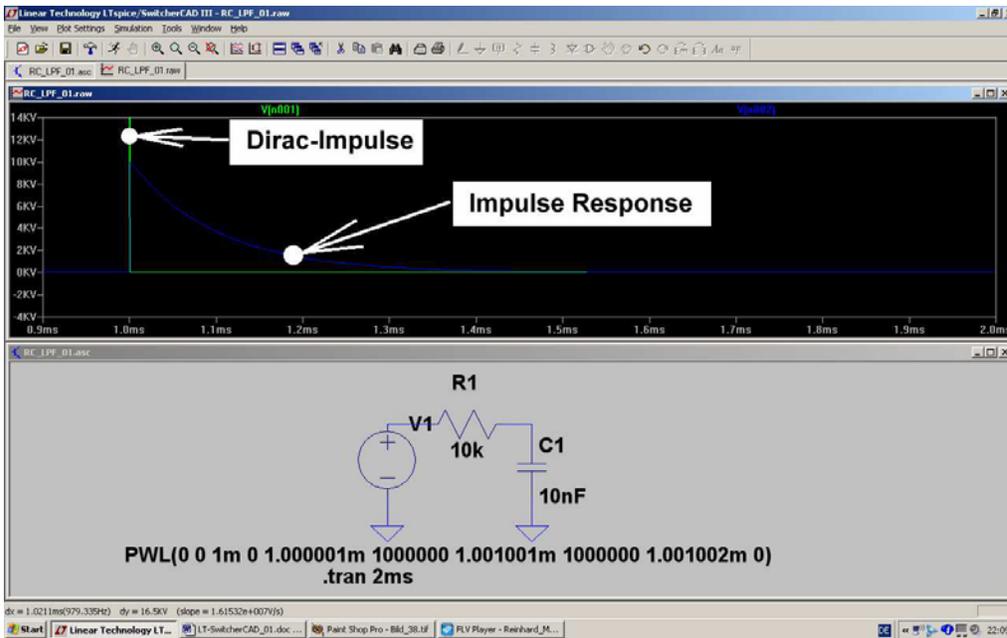
0           0
1m          0
1,000001m  1000000
1,001001m  1000000
1,001002m  0
  
```

When simulating input and output voltage you get this screen where only the Dirac impulse with its amplitude of 1 Megavolt can be seen:



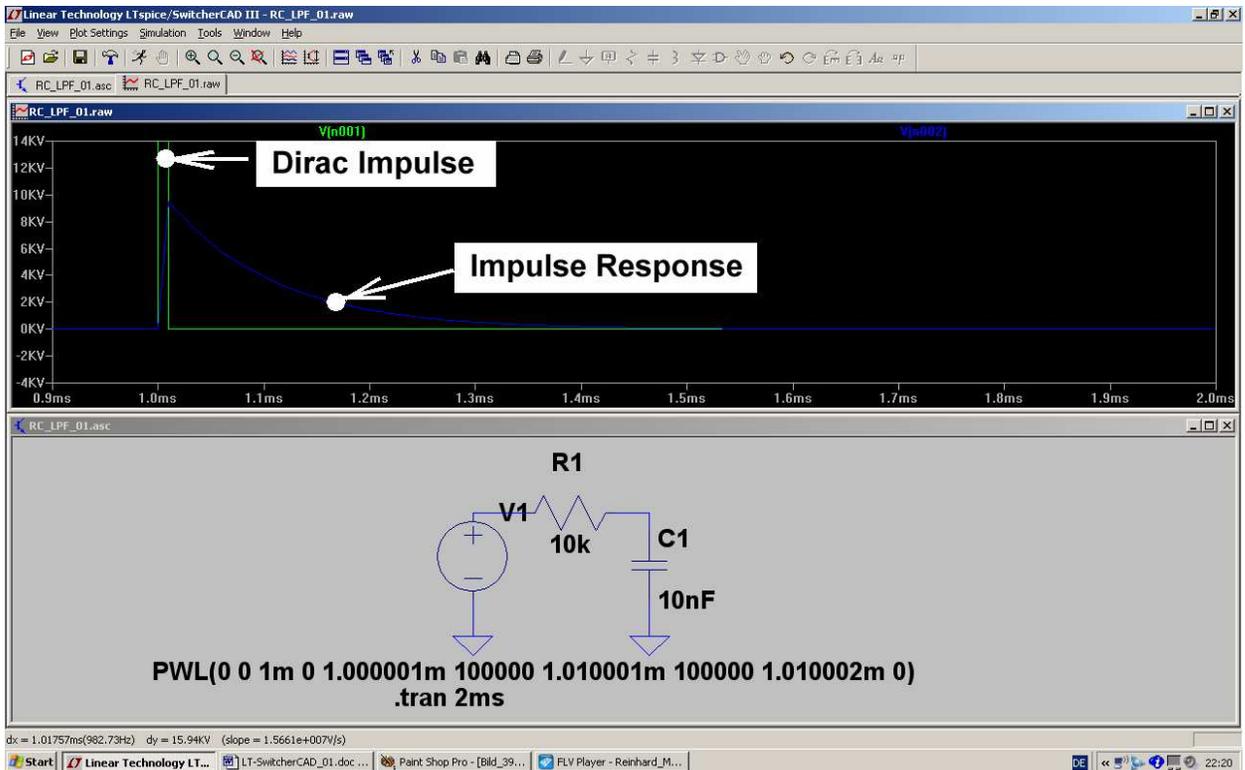
Use the zoom function for other details, because the system's response has only an amplitude of 10KV. So change the scaling of the two axis to see the response curve -- and afterward have a look at the solution on the following page.

This is the correct presentation:



Solution for Task 2:

Please be aware of the new pulse length (10 microseconds) and amplitude (100KV) and the modified PWL list in the property menu of the voltage source!

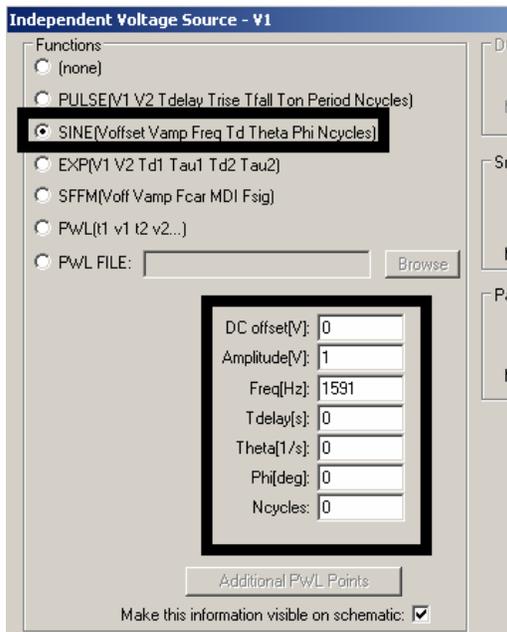


4.4. Time Domain Simulation using Periodic Signals at the Input

4.4.1. Sine Wave (f = 1591 Hz)

We use a sine wave with a peak value of 1V at the corner frequency of this LPF. The corner frequency can be calculated in this manner:

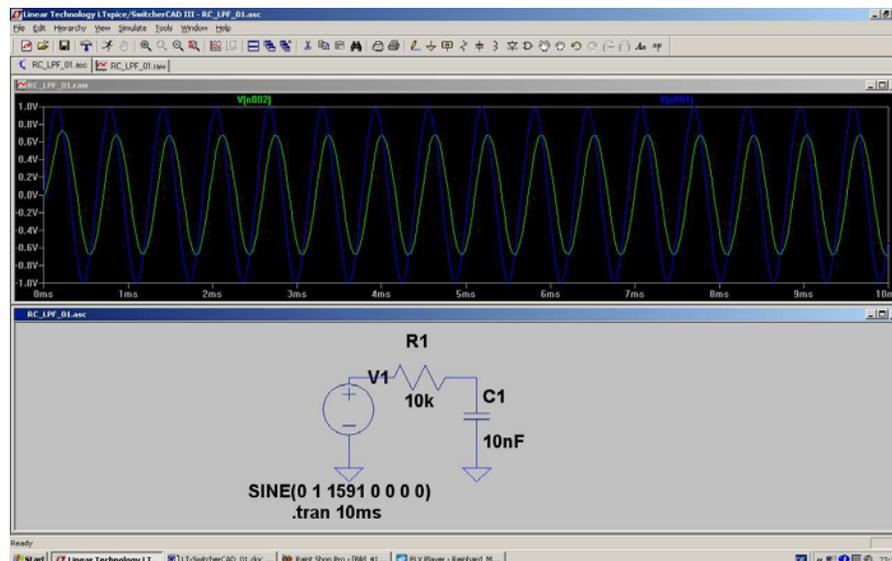
$$f_{\text{Corner}} = \frac{1}{2 \cdot \pi \cdot R \cdot C} = \frac{1}{2 \cdot \pi \cdot 0.1\text{ms}} = 1591\text{Hz}$$



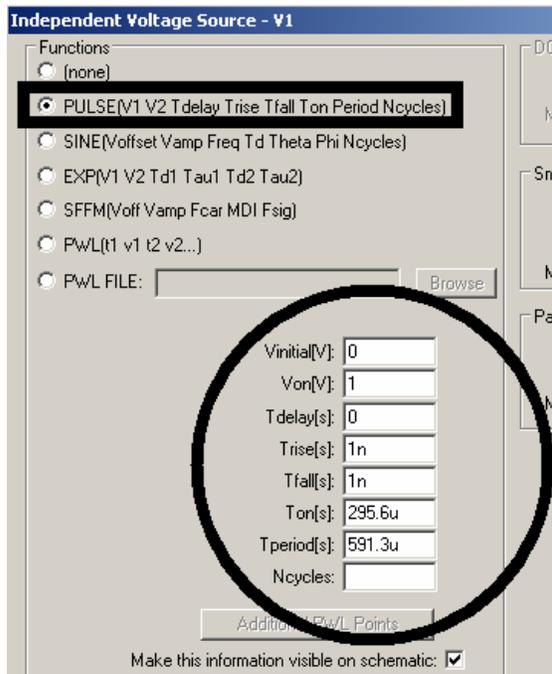
Open the property menu of the voltage source by clicking on the source symbol and choose "SINE". Then enter the following properties:

| | |
|------------------|---|
| DC Offset | = 0 |
| Amplitude | = 1 Volt (= peak value) |
| Frequency | = 1591 Hz |
| Tdelay | = 0 (= delay time after start) |
| Theta | = 0 (= damping factor. Enter zero for a constant sine wave) |
| Phi | = 0 (= start value of phase) |
| Ncycles | = 0 or empty field (this entry is important if you want to create a „Burst Signal“) |

Now open the „Simulate“ menu and „Edit Simulation Command“. Enter a stop time of 10 ms and then start the simulation. Check the values of 45 degrees of phase shift and an amplitude degradation to 70% at the corner frequency on the screen.



4.4.2. Square Wave (f = 1691 Hz)

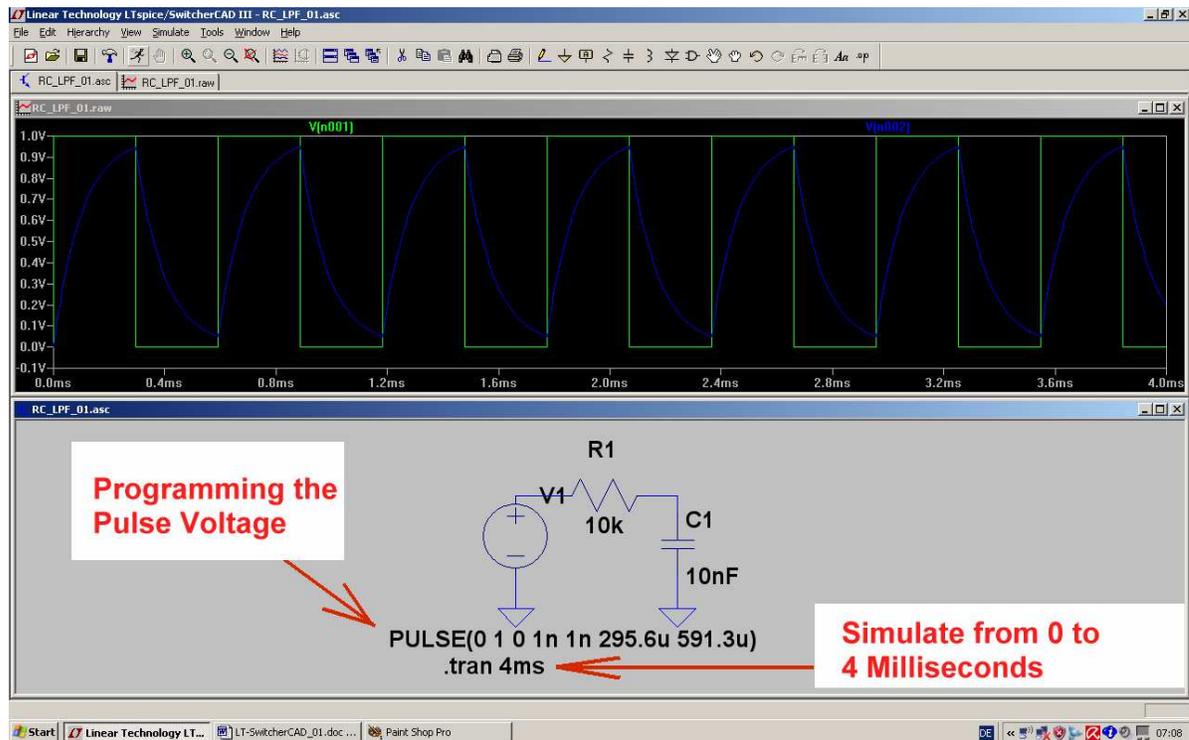


Start with the properties of the voltage source.

For „PULSE“ we need the following entries:

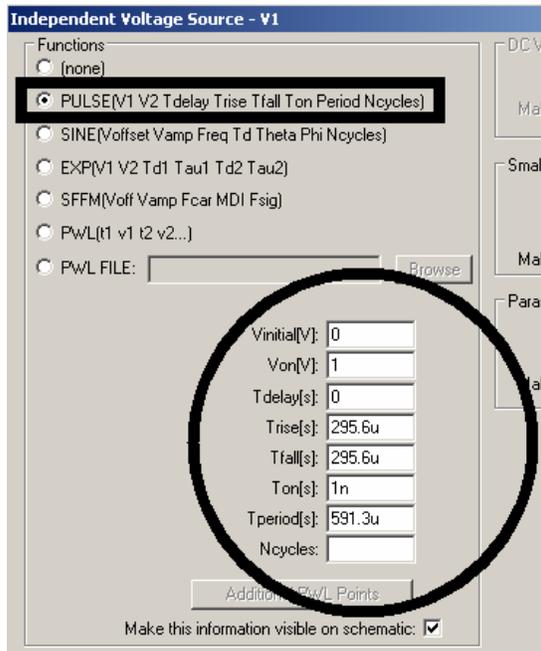
- Vinitial** = 0 (= minimum voltage value)
- Von** = 1 Volt (= maximum voltage value)
- Tdelay** = 0 (= delay after start)
- Trise** = 1 Nanosecond (= rise time)
- Tfall** = 1 Nanosecond (= fall time)
- Ton** = 295,6 Microseconds (= pulse length)
- Tperiod** = 591,3 Microseconds(= Period Time)
- Ncycles** = 0 or empty field (= number of cycles when programming a burst signal)

With a stop time of 4ms you get this result:



4.4.3. Triangle Wave (f = 1691 Hz)

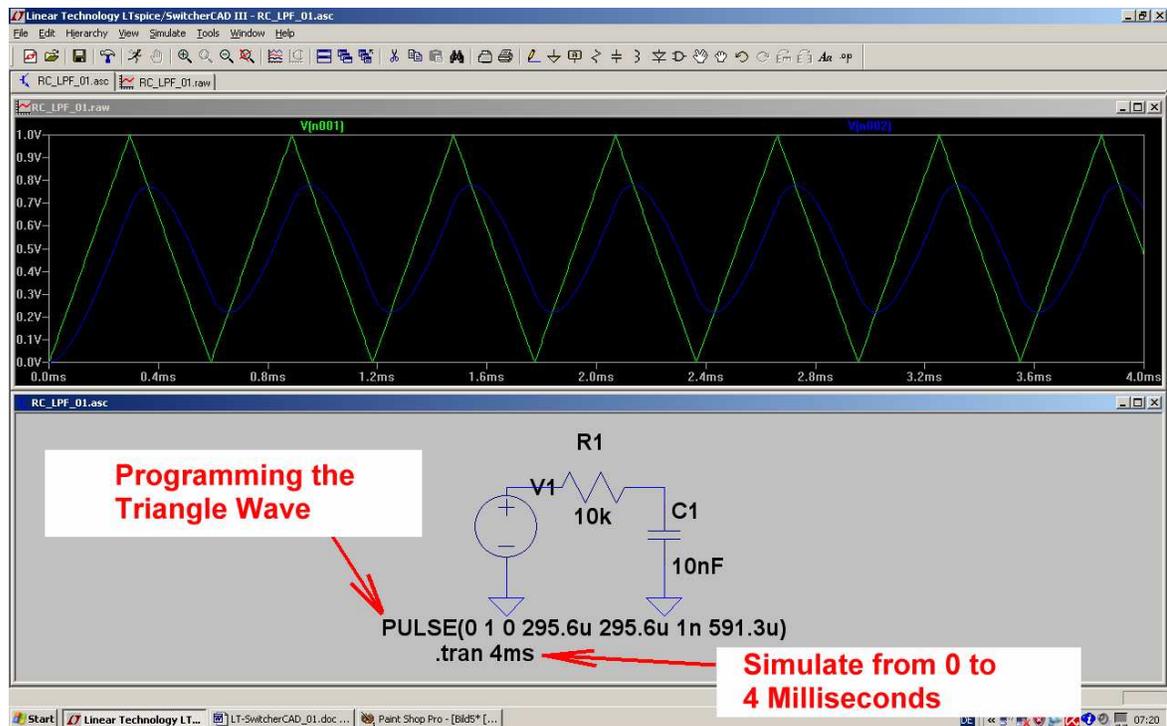
This is a little bit difficult, because we have to „strangulate“ the Pulse Source:
If we choose a very long rise and fall time but a short pulse length then we get exactly that what we need: a triangle wave.



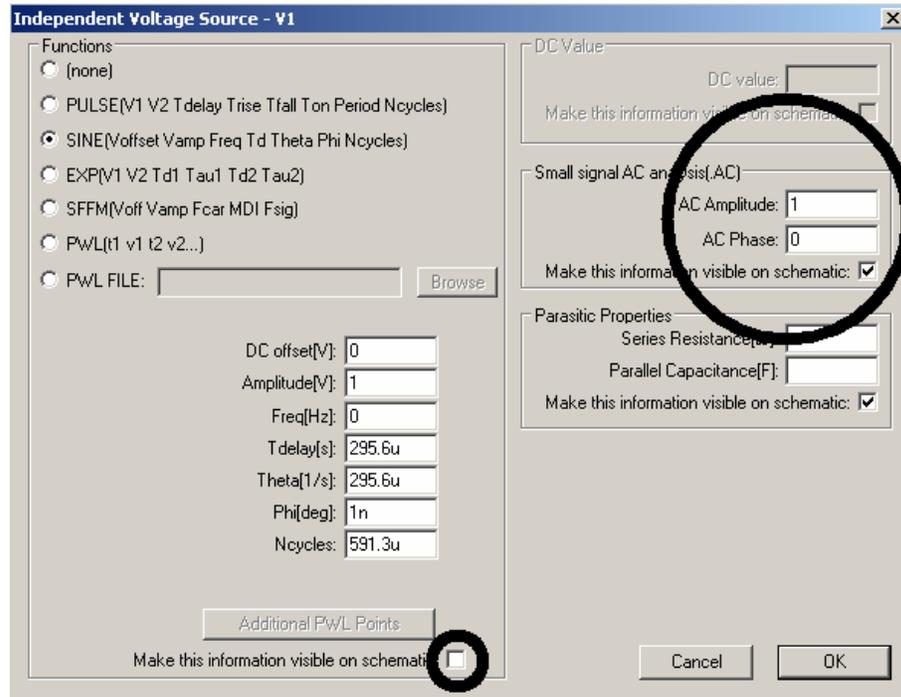
So please enter the following properties:

- Vinitial** = 0 (= minimum voltage value)
- Von** = 1 Volt (= maximum voltage value)
- Tdelay** = 0 (= delay after start)
- Trise** = 295,6 Microseconds (= rise time)
- Tfall** = 295,6 Microseconds (= fall time)
- Ton** = 1 Nanosecond (= pulse length)
- Tperiod** = 591,3 Microseconds(= period time)
- Ncycles** = 0 or empty field (= number of cycles for a burst signal)

This is the result for a stop time of 4ms:



4.5. Frequency Domain Simulation: AC Sweep

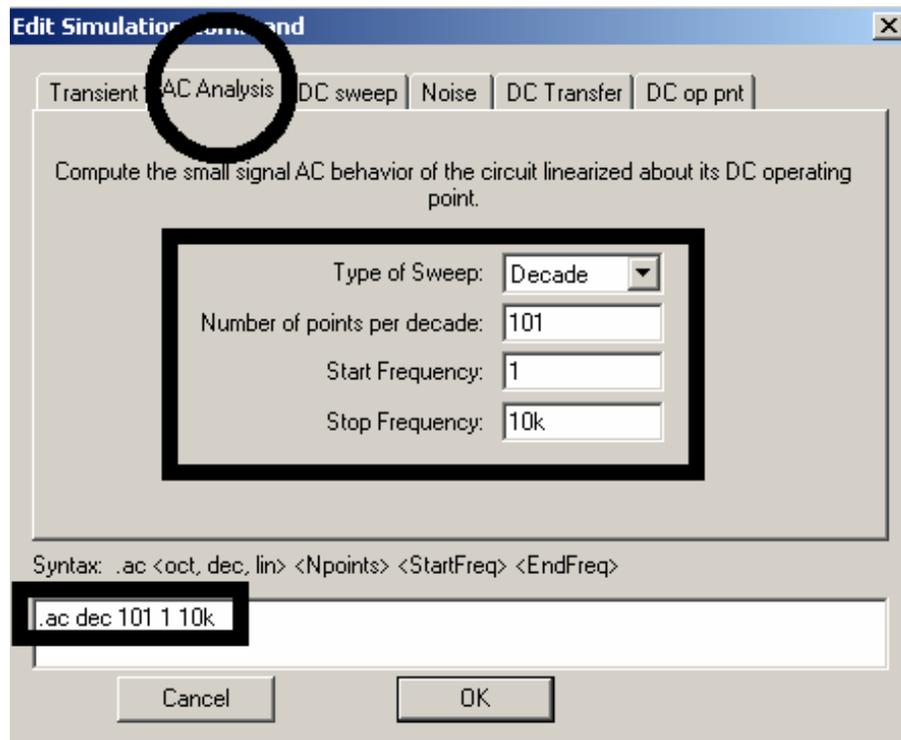


Step 1:
Switch the voltage source to „**AC Sweep**“ by opening the property menu and the two entries:

AC Amplitude = 1 Volt

AC Start Phase = 0

Ignore the programming for the time domain of the last example and delete the mark in the field „Make this information visible on schematic“.



Step 2:
Open „**Simulate**“ and the line „**Edit Simulation Command**“.

Change to the „**AC Analysis**“ menu and enter:

Decade sweep

101 points per decade

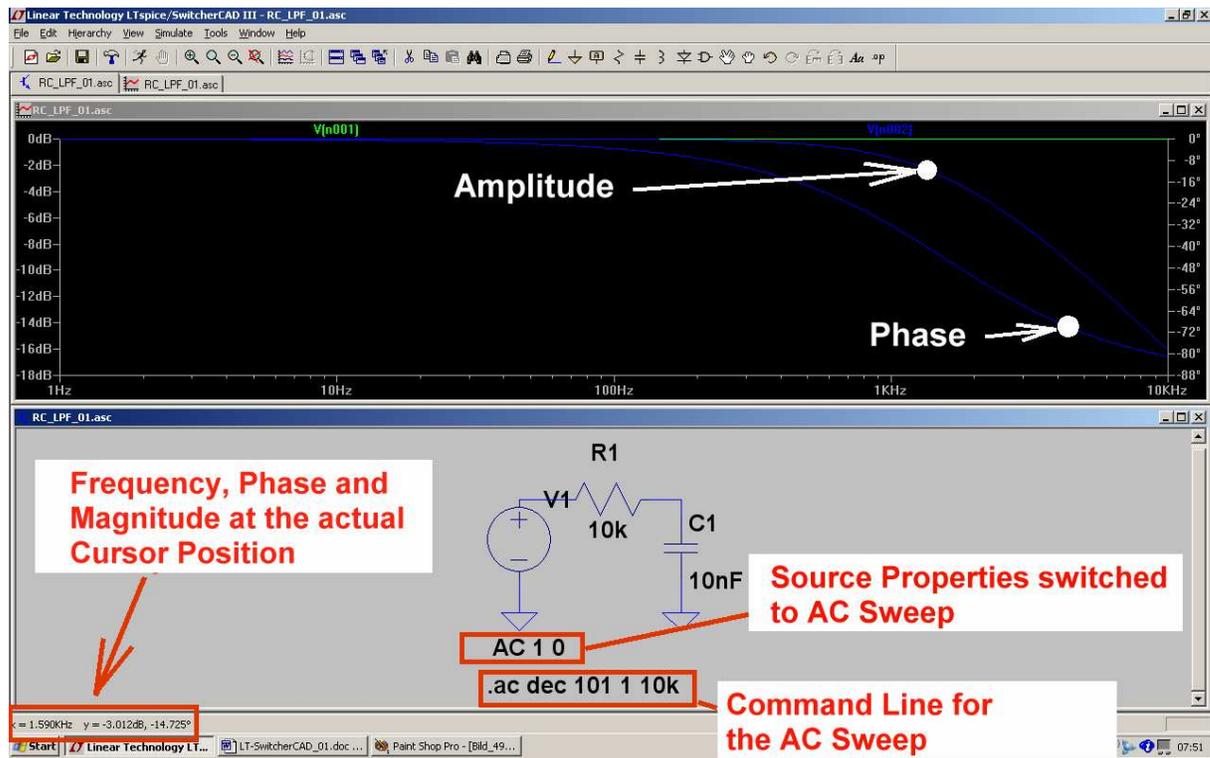
Start Frequency = 1Hz

Stop Frequency = 10kHz

Please check now the generated simulation command:

.ac dec 101 1 10k

Then start the simulation and test the result. Use the cursor to see the actual frequency, phase and magnitude.



Congratulations, the first own example is now successfully terminated!

5. FFT (= Fast Fourier Transformation)

Very often we need information about all frequencies which can be found in a signal. It is a fact that only a pure sine wave consists of only one frequency. So, every deviation from the ideal sine wave curve produces new frequencies (harmonics) with frequency values of 2x, 3x, 4x...of the fundamental frequency. This information can be obtained by using the FFT.

But first several things are needed to get the FFT working and to get good results.

=====
The start frequency AND the minimum width of a spectral line AND the smallest frequency step are all determined by the

**Minimum frequency step = start frequency = minimum line width =
= 1 / simulation time**

Also for periodic signals the simulation time must be an integer multiple of the period time. If you ignore that you get "ghost lines" and / or an additional noise floor.

(An additional note for specialists only: you should reduce the used simulation time by one timestep. Otherwise you are already starting a new period which will not be complete. But this error is only noticeable for very "coarse time steps" or for a very small number of periods).

=====
Necessary properties of the **maximum time step**:

- a) Do not make it too coarse. This avoids „corners“ in the simulated time domain signal.
- b) The related „**minimum sample frequency = 1 / maximum time step**“ must have a minimum frequency value twice of the maximum frequency line in the simulated signal. Otherwise the „Nyquist - Shannon – Law“ is violated and you get undesired aliasing effects.
- c) The maximum frequency AND the dynamic range of a FFT are increased by the number of used samples in the FFT. But: these samples must be true and you must not use more samples for the FFT than you get from the time domain simulation. The number of samples can easily be calculated by

True number of samples = simulation time / maximum time step

=====
LTspice uses data compression for the results file of the time domain simulation. This is undesirable because it loses informations and also a lot of the mentioned „true samples“. So switch off the data compression by the following Spice directive:

.options plotwinsize=0

=====
For a FFT the number of used samples must always be a multiple of „2“.

No problem because LTspice will always default to those in the choice menu.

Note::

More „true“ samples give a higher stop frequency AND an extended amplitude range in the simulated spectrum. But the necessary calculation time AND the resultant data file also increases.

=====
For specialists only.

Because only the maximum time step is fixed, the sample rate applied by SPICE can vary or be increased automatically when simulating "the difficult parts of a curve". But this cannot be foreseen, equates to an additional FM and causes additional but unexpected spectral lines and an increased "noise floor" at high frequencies.

This effect can be reduced by choosing the minimum sample rate as high as possible (= maximum time step as low as possible) and choosing as less samples for the FFT as possible.

Example:

Let us use the RC-LPF of the last chapter ($R = 10\text{k}\Omega / C = 10\text{nF}$). The input is fed by a symmetric square wave voltage (minimum amplitude value = 0V , maximum amplitude value = 1V , frequency = 1kHz).

If we use a simulation time of 20ms with a maximum time step of 100ns we meet the above recommendations as follows:

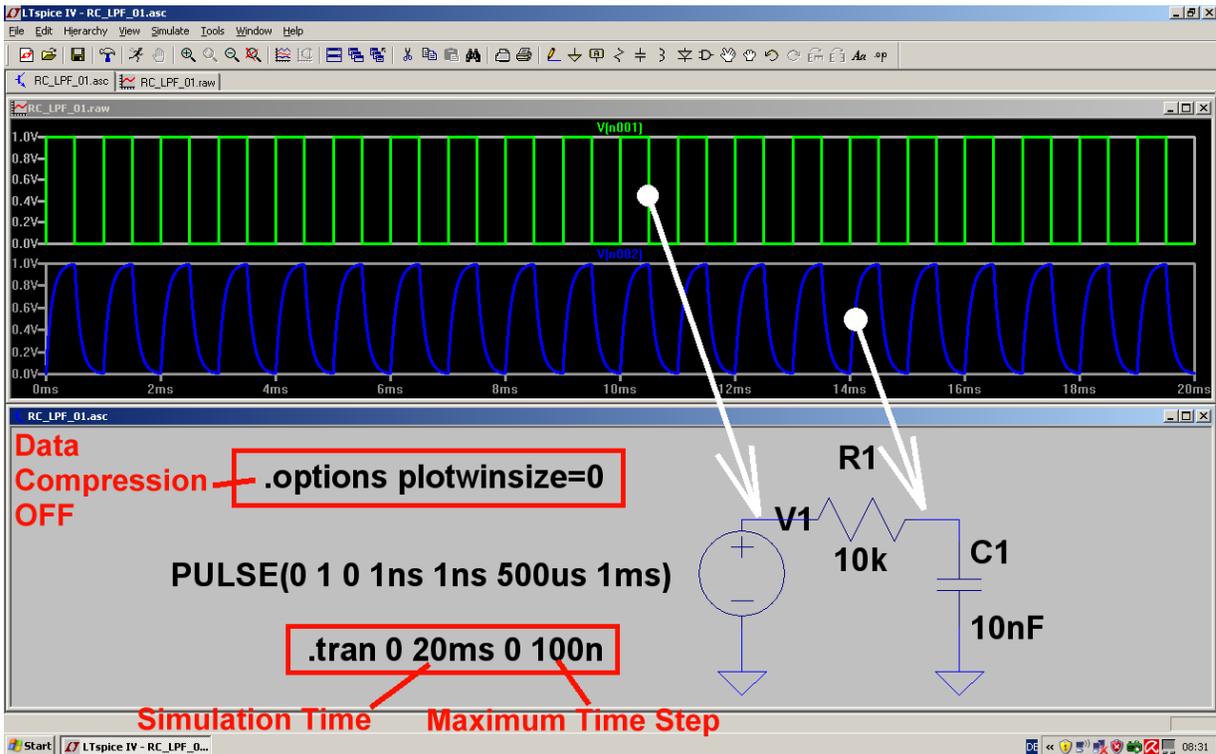
- The simulation time of 20ms is an integer multiple of the signal's period time (1ms).
- A maximum time step of 100ns is corresponding to a minimum sample frequency of $1/100\text{ns} = 10\text{MHz}$. This gives a Shannon corner frequency of $0.5 \times \text{sample frequency} = 5\text{MHz}$.
- The number of true samples is $20\text{ms} / 100\text{ns} = 200\,000$. So we enter

„131 072 sampled data points in time“

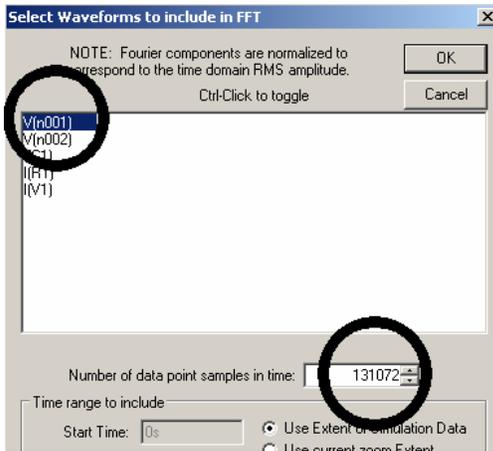
in the FFT menu.

Now draw the circuit, set the properties of the voltage source, choose simulation time and time step and write the directive to switch off the data compression.

The time domain simulation result (= input and the output voltage) should be presented in two plot panes. Also “thick lines” are used for the curves (= button with the hammer / menu “waveforms”).



Now right click on the input voltage and choose “View” and “FFT”



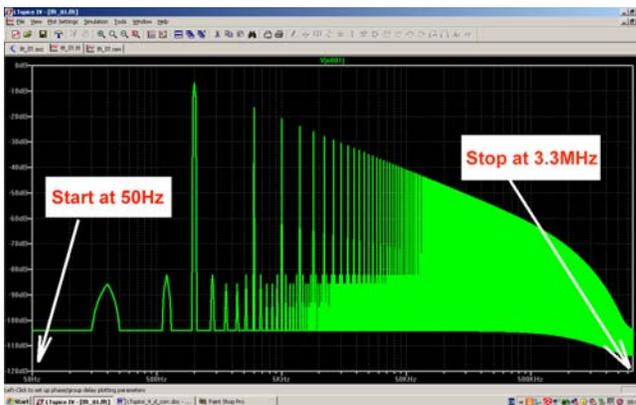
Select **Input Voltage V(n001)** by a mouseclick

As discussed we have 200 000 real samples of this curve. So we enter a

“number of 131 072 data point samples in time”

and start the FFT.

This is the successful result:



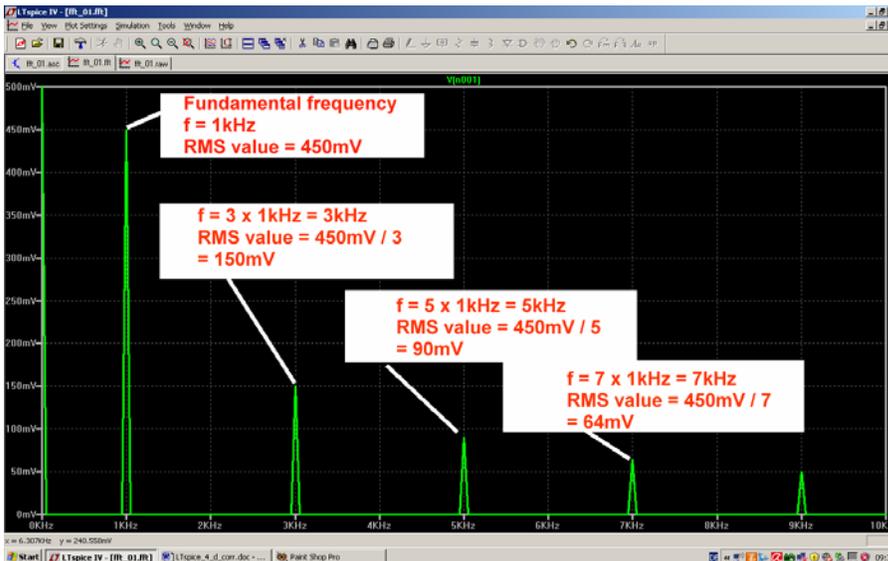
Start frequency = frequency resolution =

$$50\text{Hz} = 1 / 20\text{ms.}$$

The stop frequency is 3.3MHz and can only be increased by using a higher number of samples.

Now change the diagram to linear vertical and horizontal result presentation to confirm theory which says:

In a symmetric square wave you can only find the fundamental frequency and its odd harmonics. The amplitudes of the harmonics decrease exactly with their order:



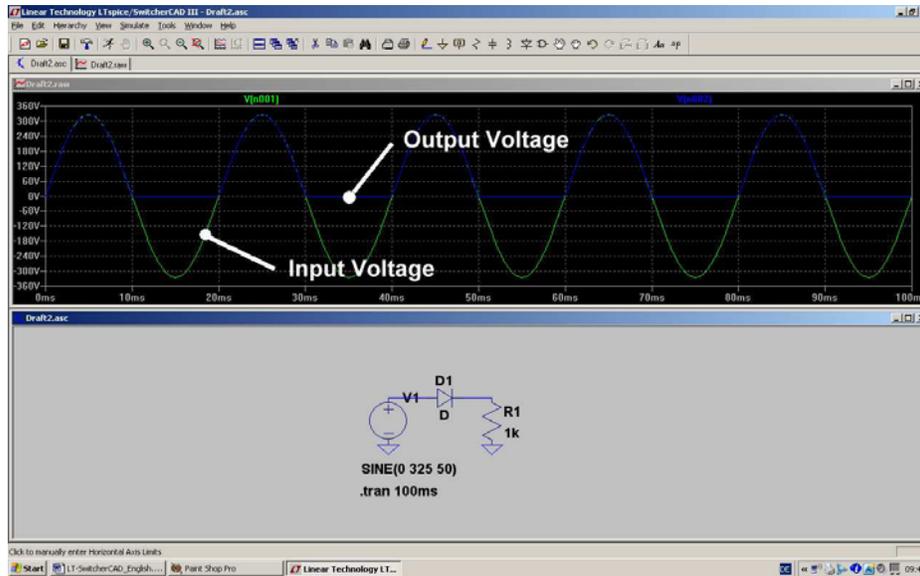
Note:

If anybody is additionally interested in the phase information: click on the scaling of the right vertical diagram axis and press “Phase” in the menu.

6. Project 2: Rectifiers

6.1. Simple Rectifier without Transformer

No problem: draw a circuit consisting of a Voltage Source, a diode and a resistor (1k). Now apply a sine wave (50 Hz / peak value = 325 V) to the input and simulate from 0 to 100ms.

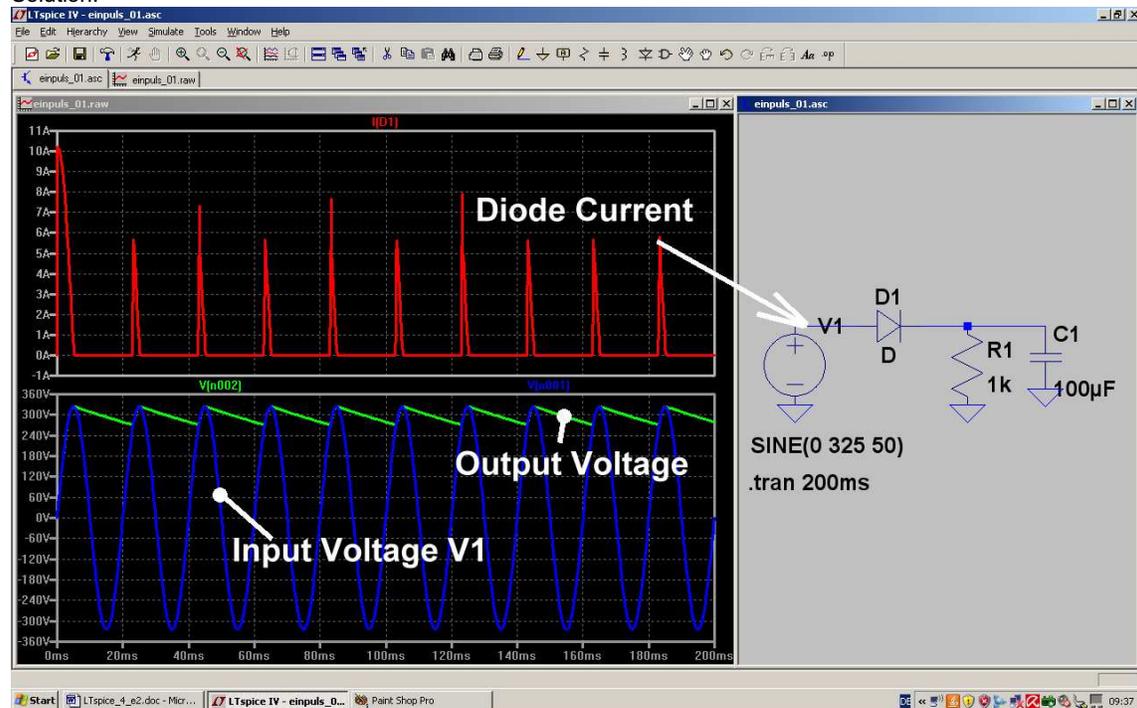


Task:

Add a parallel capacitor of 100 Mikrofard to the output.

Show also the diode current (by moving the cursor over the diode's symbol to change the cursor into a current sensor).

Solution:

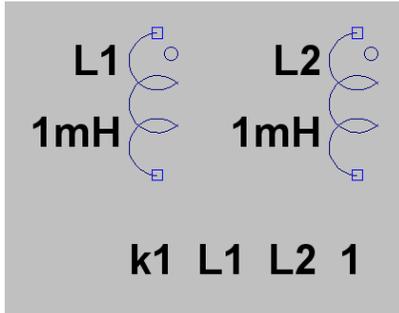


6.2. Important: creating SPICE Model and Symbol for a Transformer

6.2.1. The easiest Solution: a simple ideal Transformer

Sorry, but in the LTSpice part library you can't find any transformer! So let us start with a simple but good solution found in the original SPICE manual:

Place the necessary windings (with their Inductances) in the schematic and add a SPICE Directive for the magnetic coupling "kn" between these windings!



Example for a two winding transformer and voltage transfer ratio = 1:1

a) Place "Ind2" from the symbol library two times on the screen, because only at this symbol the start of the winding procedure is marked by a circle.

b) The line "k1 L1 L2 1" says that the **magnetic coupling factor between the windings L1 and L2 is "1" = 100%**.

c) Attention: **It is not possible to enter a given voltage transfer ratio "(N1 / N2)" into PSPICE!!**

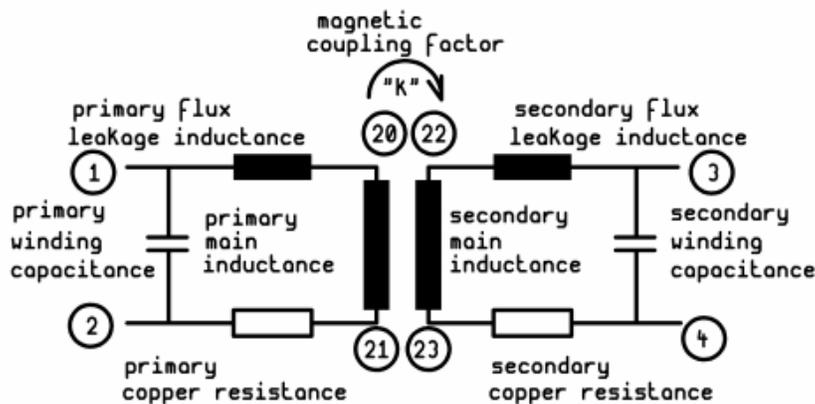
The reason is not complicated: PSPICE works always and only with parts and their properties (here: inductances and magnetic coupling factor). So we must use the inductances and the following relation:

$$\frac{L_{\text{Primary}}}{L_{\text{Secondary}}} = \left(\frac{N1}{N2} \right)^2$$

So choose a realistic value for L1 and then calculate L2 by using this formula.

6.2.2. Creation of the SPICE Model for a real Two Windings Transformer

We use the following transformer model for our work, which includes the „magnetic coupling“ between the primary and secondary winding, the flux leakage inductances, the winding capacitances and the copper resistors of the windings.



Node 1 and 2 are the primary winding connections, node 3 and 4 the connections for the secondary winding of the model for the simulation. For the "inner nodes" you should choose much higher numbers to avoid collisions when "blowing up the circuit diagram" with more windings or parts.

Now have a look at the following SPICE – Model, using the **subcircuit „xformer_01“ with its connections 1 / 2 / 3 / 4:**

(**Lleak** is the name for the **flux leakage inductances** which represent the part of the flux who does not run through both main inductances leaded by the magnetic core of the transformer. All other circuit elements correspond to the above diagram):

```

*
* [1] ---. ||
*      ) || .--- [3]
*      ) || (
*      ) || (
* [2] ---. || .--- [4]
*
.SUBCKT xformer_01 1 2 3 4
*
** Primary
Lleak1 1 20 1mH
Lpri1 20 21 1H
Rpri1 21 2 1
Cpri1 1 2 20pF
*
**Secondary
Lleak3 3 22 1mH
Lsec1 22 23 1H
Rsec1 23 4 1
Csec1 3 4 20pF

K Lpri1 Lsec1 0.999
.ENDS

```

Explanation:

Every line starts with the name of the part, followed by the numbers of the nodes between this part is inserted in the circuit. And at last you find the parts value.

In praxis the magnetic coupling is always below „1“ (due to the flux leakage). So always insert a value smaller than 1 (here: k = 0.999). Some SPICE versions start a protest action if you don't follow this advice.

Primary and secondary winding use the same number of turns. So you have to enter the same inductance value to get a 1:1 - transformer.

Note:

The description of the circuit starts always with **.SUBCKT** and ends with **.ENDS**

=====

Very important:

a) The finished model is named **xformer_01.lib** an should be saved in the folder

Programs / LTC / LTspiceIV / lib / sub

where all other LTC-SPICE-models are already located. Please pay attention to the correct path and the correct extension when saving this file.....

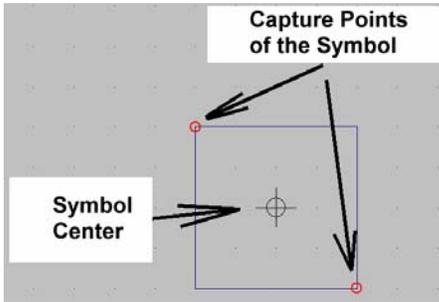
b) It is **impossible** to enter the turn ratio! The only possibility is to enter different primary and secondary main inductances and we'll handle this in the next project.

=====

6.2.2. Creation of the Symbol for a Two Windings Transformer

Step 1:

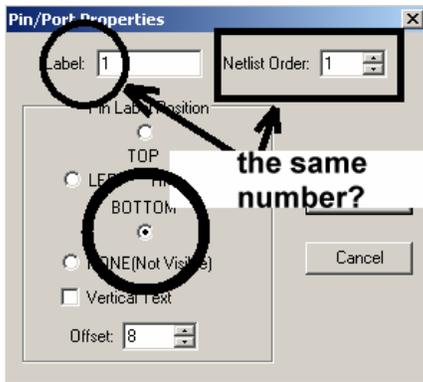
Open „File“ and click on „New Symbol“. Now you see an empty screen with “cross and circle” in its center.



Step 2:

Open „DRAW“, choose „Rect“ (= rectangle) and draw it with the dimensions 20mm x 20mm around the symbol's center point (...as a help: You find a fine point grid with 5mm distance on the screen).

(The red circles at two corners of the rectangle are „capture points“ for the user when placing this symbol in a circuit).

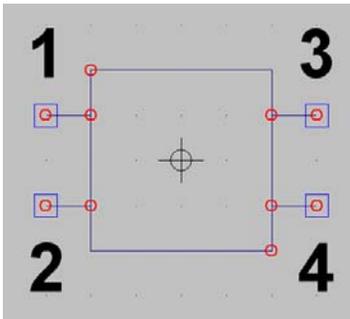


Step 3:

Go to „Edit“ and „Add Pin / Port“ to generate the connection points. But in the menu you have at every connection point not only to enter the pin Number BUT ALSO the relative position between this point and the Point's Label.

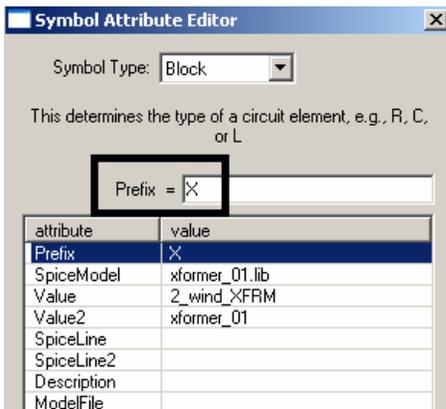
And always examine carefully whether the point's label and netlist order are identical!

Then click OK and you have the pin at your cursor. Now you can place it in a distance of 5mm from the symbol (See next figure).



Step 4:

Generate all 4 Pins and afterwards open once more „DRAW“ to connect the pins to the symbol by wires. So the terminated symbol should look like.



Step 5:

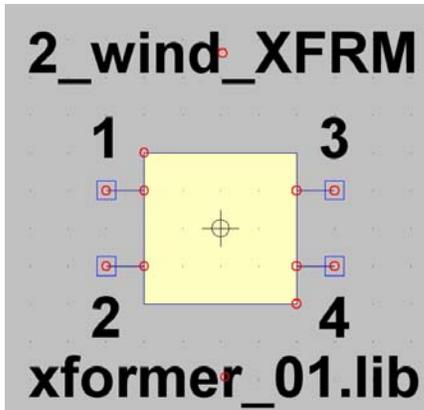
Let us complete the real transformer for the simulation. Please open „EDIT / Attributes / Edit Attributes“ to arrive at this menu.

Now enter all entries as given in the figure.

(Remark: Prefix „X“ means a subcircuit).

Warning:

The line for „Value2“ MUST NEVER BE EMPTY --- otherwise you can get a cryptic error message! Normally use the same entry as in the line for the spice model, but without extension.



Step 6:

At last make visible all important informations for the user. Start with „Edit“ and „Attributes“, but now choose „Attribut Window“. When clicking on „value“, then you get the description „2_wind_XFRM“ at your cursor and you can place it over the symbol. Then repeat the procedure but now choose „SpiceModel“ = xformer_01.lib and place it below the symbol. The work is done!

Step 7:

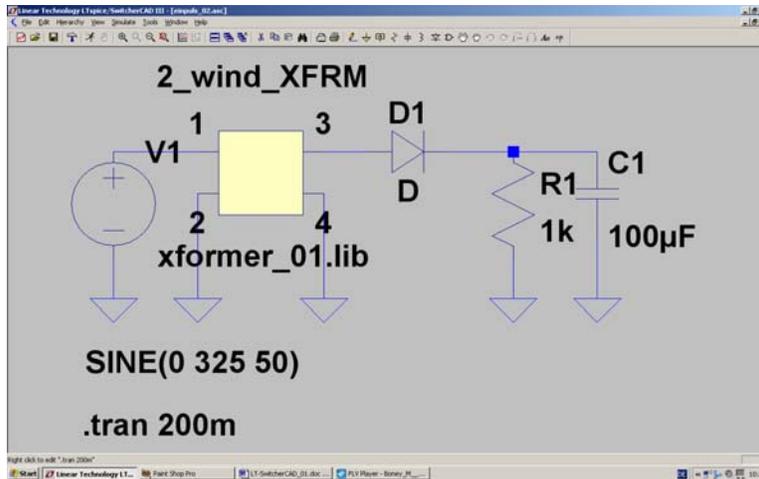
Save the complete symbol. Open „File“ and „Save as“ and the correct path starting at „SwCADIII“ and „lib“ just down to the folder „sym“ (= symbols). There create a new folder named „Xformers“ and save this new part as „xformer_01.asy“.

6.3. One Pulse Rectifier with Transformer

Now let us test a circuit with line transformer, one pulse-rectifier and output capacitor.

Task:

Test a power supply consisting of a line transformer, rectifier, load resistor R1 = 1kΩ and load capacitor C1 = 100μF. Line input voltage is 230Vrms / 50 Hz, the turn ratio of the transformer is 20:1. Simulate in the time domain from 0...200ms.



Solution:

Most of the necessary parts are already known from the last example. Fetch and place them, wire and enter the correct part values including the properties of the voltage source and add the command for the simulation from 0 to 200ms in the time domain.

But now we need the line transformer of the last chapter. Fetch and place the part.

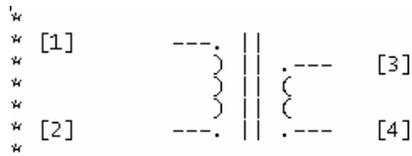
Then we look for the necessary transformer property entries.

You cannot enter the turn ratio N1/N2 in a direct form, so you need the following equation for the inductances:

$$\frac{L_{\text{primary}}}{L_{\text{secondary}}} = \left(\frac{N_1}{N_2} \right)^2$$

Let us apply a value of 1Henry for the primary main inductance (...ideal transformer...) and calculate the secondary main inductance for a turn ratio of 20:1 in the following manner:

$$L_{\text{secondary}} = \frac{L_{\text{primary}}}{(20)^2} = \frac{1\text{H}}{400} = 2.5\text{mH}$$



For an ideal transformer you can reduce the values for the flux leakage inductances, the copper resistors and the winding capacitances.

But always you must never set them to zero”.....

```

.SUBCKT xformer_01 1 2 3 4
*
* Primary
*
Lleak1 1      20      1uH
Lpri1  20     21      1H
Rpri1  21     2       1
Cpri1  1      2       1pF

* Secondary
*
Lleak3 3      22      1uH
Lsec1  22     23      2.5mH
Rsec1  23     4       1
Csec1  3      4       1pF

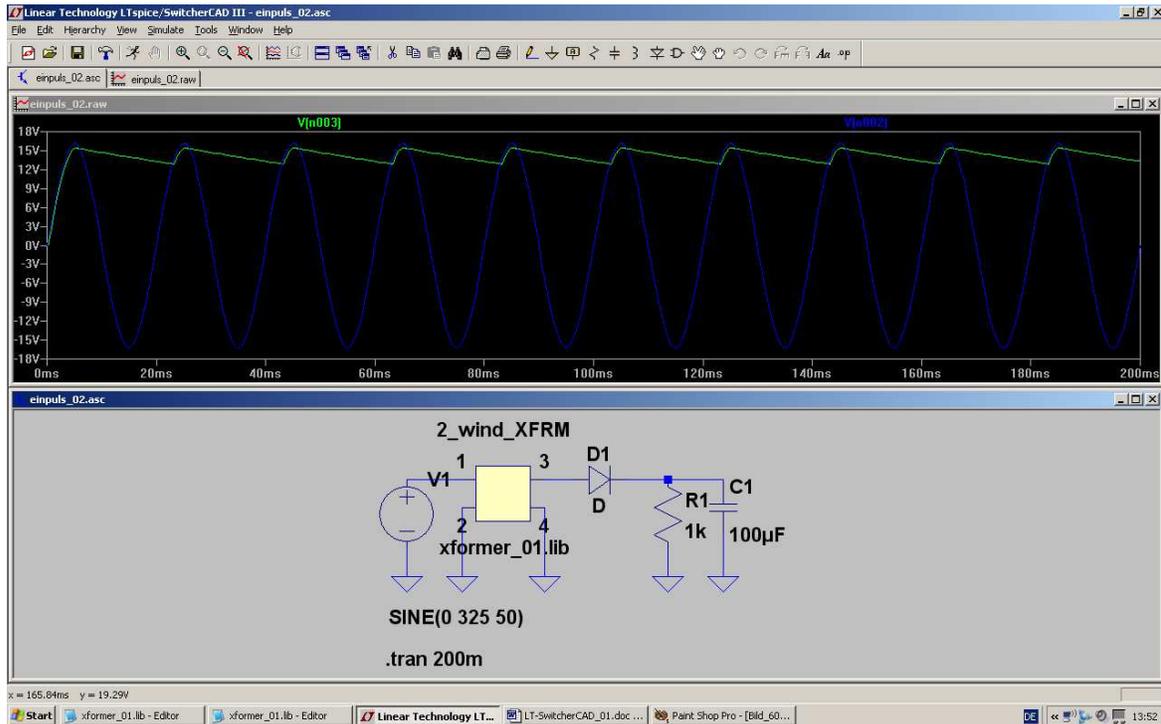
K Lpri1 Lsec1          0.999
.ENDS

```

Especially a resistor of zero Ohms in series to an inductance causes in a “normal SPICE version” at once an error message or a simulation abort!

So this model file of the transformer is a good template for your work. Open the „xformer_01.lib“ – File, change the entries and save it.

Now the simulation runs and here you have the result screen with the voltages at the left and the right side of the diode:



6.4. Rectifier with Diode 1N4007

We want to use the well known universal line rectifier diode **1N4007** instead of the ideal diode. Sorry; but the SPICE-model of this type cannot be found in the LTC-library... So let us tackle the import of foreign models (from the WEB.....).

Step 1:

Use Google and look for **diode.lib**
(This is an ORCAD-library with a lot of rectifiers).

But be carefully:

You get this library as a HTML-file and you must not use it in this form!!!!

So mark the content by **<STRG> + <a>**, copy to the clipboard by **<STRG> + <c>** and insert it into a new file of a text editor (like Notepad).

Now save this file as „**diode.lib**“ in the folder

LTC / LTspiceIV / lib / sub

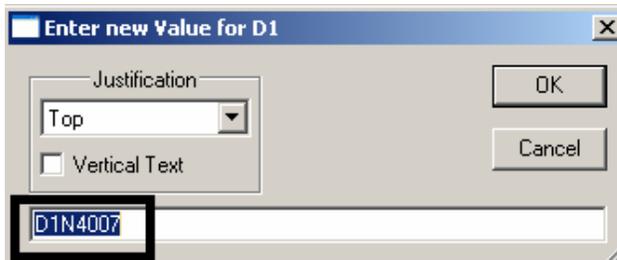
(But pay attention to get the correct extension „*.lib“ by choosing the type **“all files”**. Otherwise the editor will add a further extension **“.txt”** and the library cannot be found by our program...)

Step 2:

```
*$  
.model D1N4004 ako:D1N4001 D(Bv=600) ; use  
*$  
.model D1N4005 ako:D1N4001 D(Bv=900) ; use  
*$  
.model D1N4006 ako:D1N4001 D(Bv=1200) ; use  
*$  
.model D1N4007 ako:D1N4001 D(Bv=1500) ; use  
*$  
.model D1N4009 D(Is=544.7E-21 N=1 Rs=.1 Ikf=  
+ Vj=.75 Fc=.5 Isr=30.77n Nr=2  
*$
```

Now open „diode.lib“ with the text editor and search for „**1N4007**“. You will then recognize that the correct diode model name is

D1N4007



Step 3:

Move the cursor on the letter „D“ at the symbol of the diode and right click with your mouse.

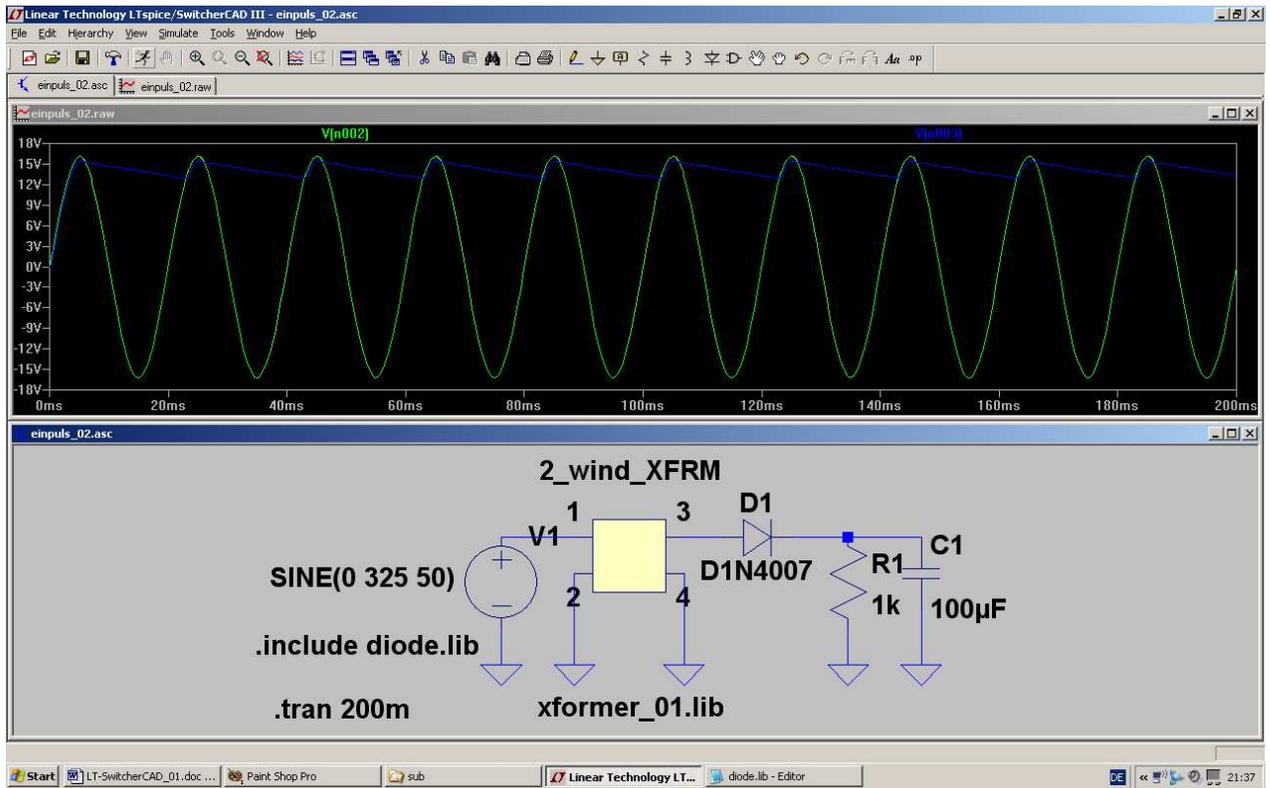
Then enter the new correct value „**D1N4007**“ and close by pressing OK.

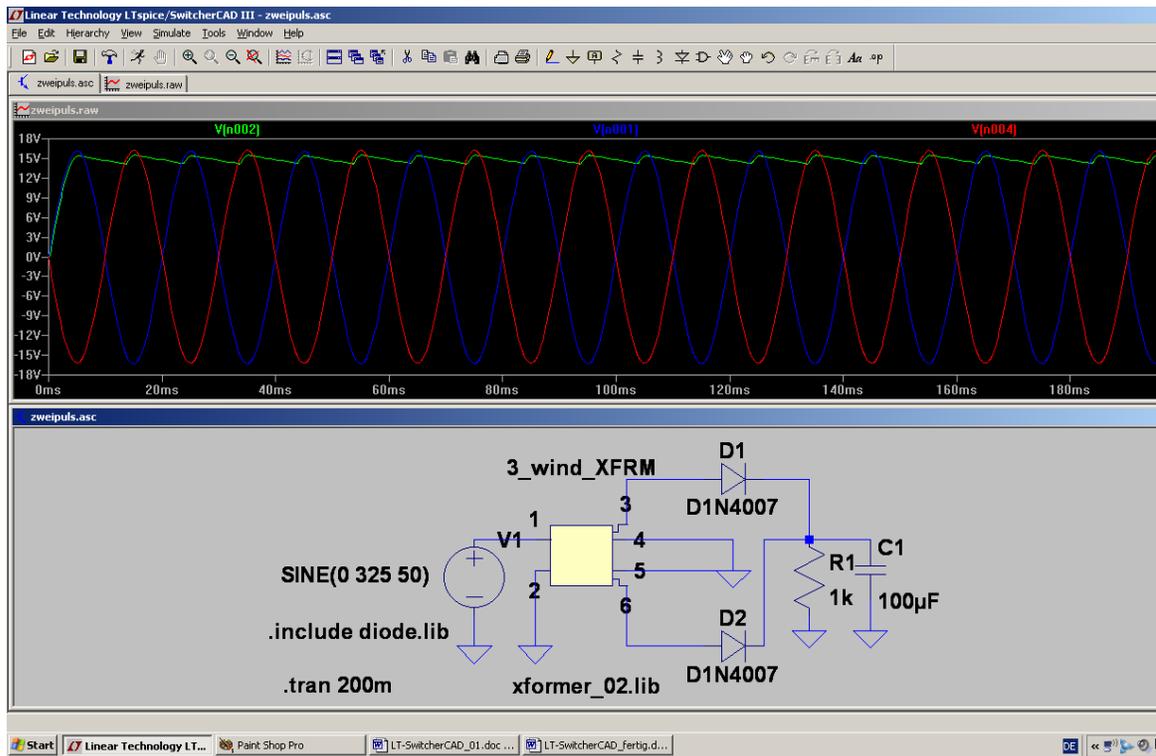
Step 4:

Enable the program to find out the correction location of this D1N4007 model and add the following Spice directive on the screen:

.include diode.lib

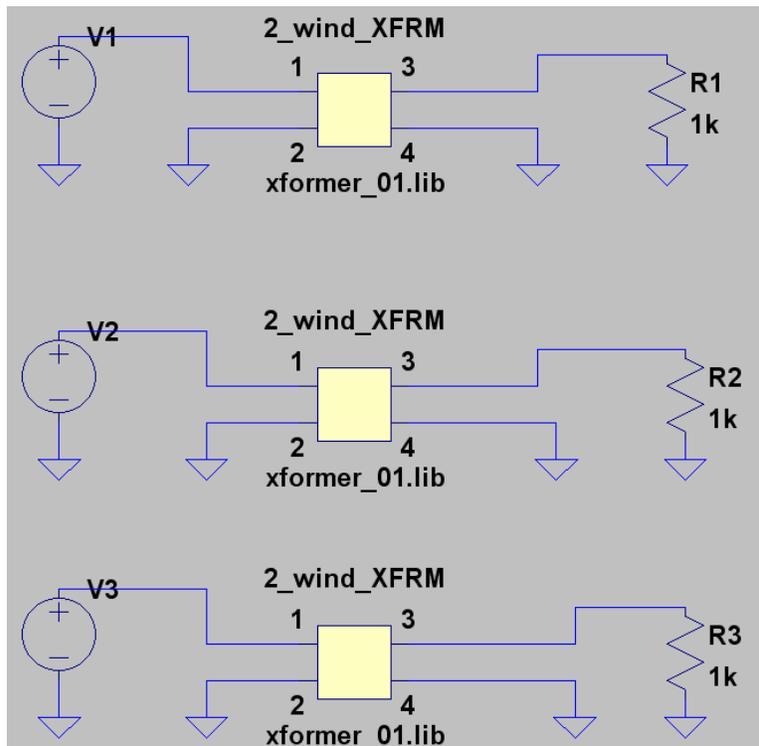
Step 5: self-explaining....





7. Project 3: Three Phase AC System

7.1. Schematic and Simulation

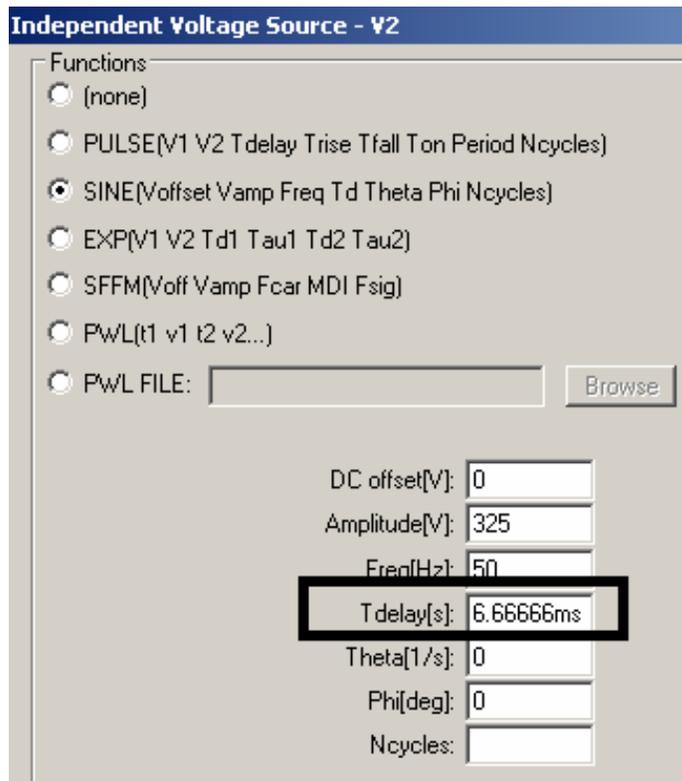


We need 3 voltage sources, 3 transformers and 3 load resistors to review this technique. For the transformers we will use our well-known file

„xformer_01.lib“,

but in this case **choose identical main inductances on the primary and secondary side to get a turns ratio of 1:1.**

Fetch all these parts from the library and draw the circuit diagram.



Then move the cursor over the symbol of every voltage source and right click. All entries for the 3 sources are identical (except one....):

DC offset = 0 Volt

Amplitude = 325 V (peak value for RMS = 230V)

Frequency = 50 Hz

Phi = 0 degrees (start phase of the sine curve)

Important:

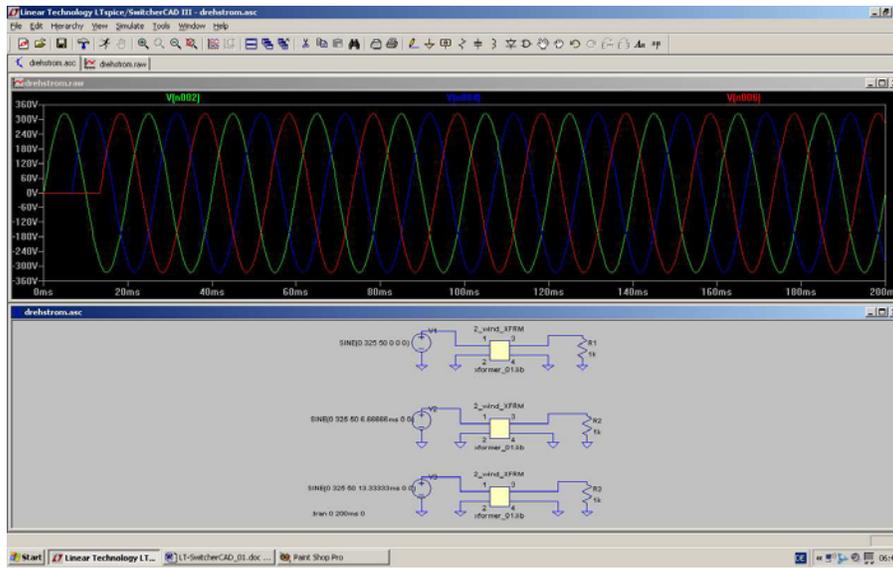
The 3 voltages have the same amplitude and the same frequency, but differ in the phase shifts. We cannot enter the necessary phase difference of 120 degrees in the property menu -- so simply enter different start delays:

for V1 the delay time is 0ms

for V2 the delay time is 6,66666 ms (see figure)

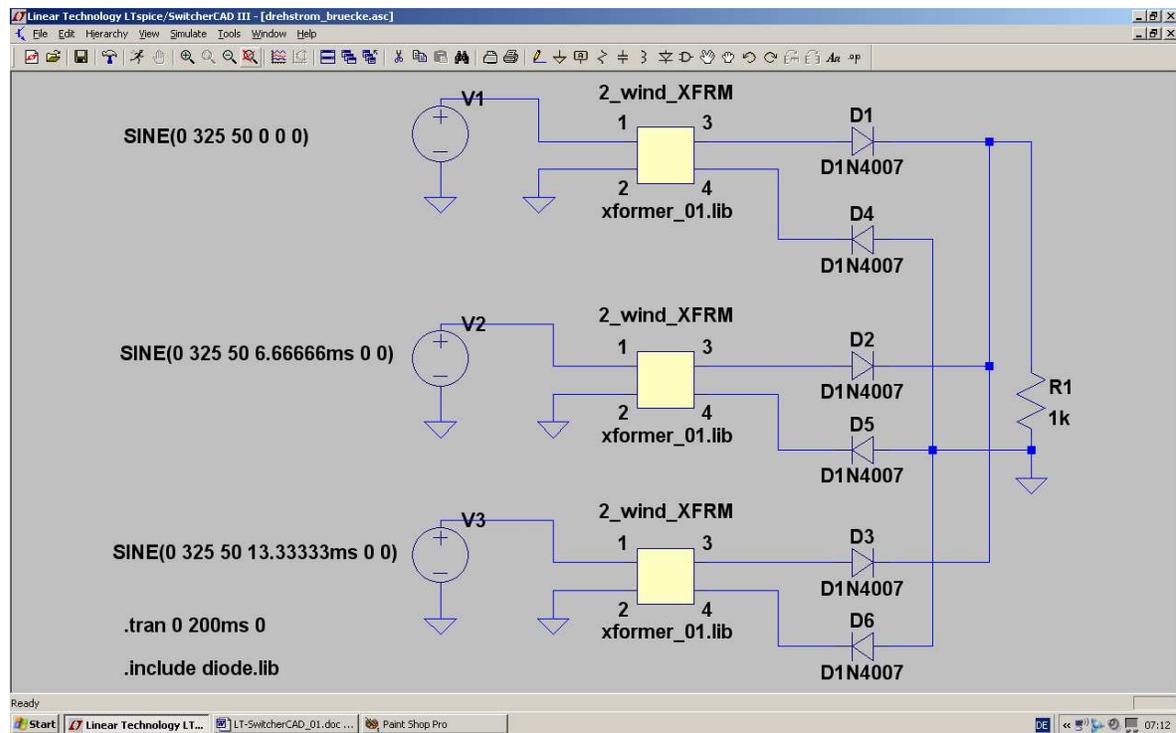
for V3 the delay time is 13,33333ms

When simulating for 200ms, you will get the following result:

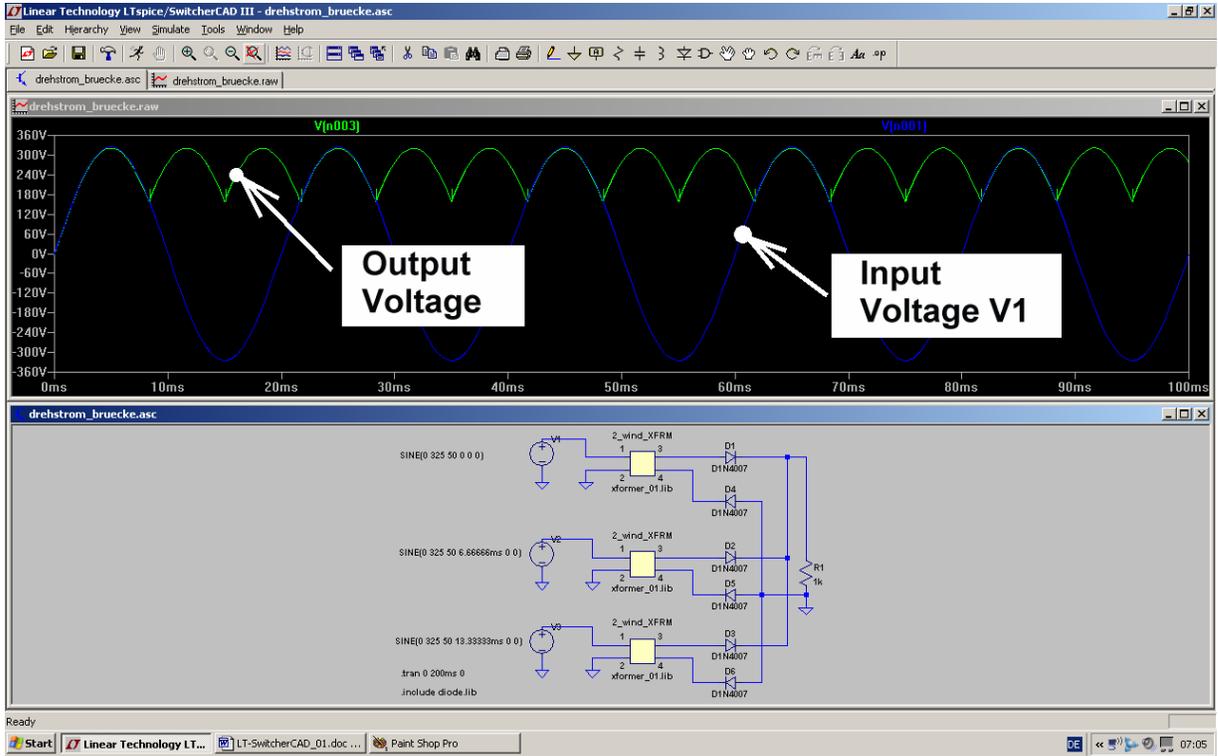


7.2. Rectifier Bridge for Three Phase AC Systems (e. g. Dynamo in a modern Car or Motorcycle)

Save this circuit in a new project folder and give it an appropriate name. Then modify the circuit by adding 6 diodes (1N4007). Delete two of the three resistors and wire the circuit as shown:



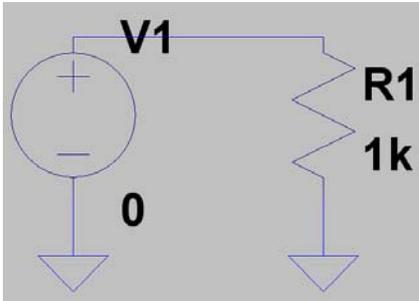
If you now simulate, you will get the following result for the input voltage V1 and the output voltage at R1:



8. Project 4: V-I-Curves of Parts

8.1. Resistor

A resistor is the best object to make visible the V-I curve of a part.

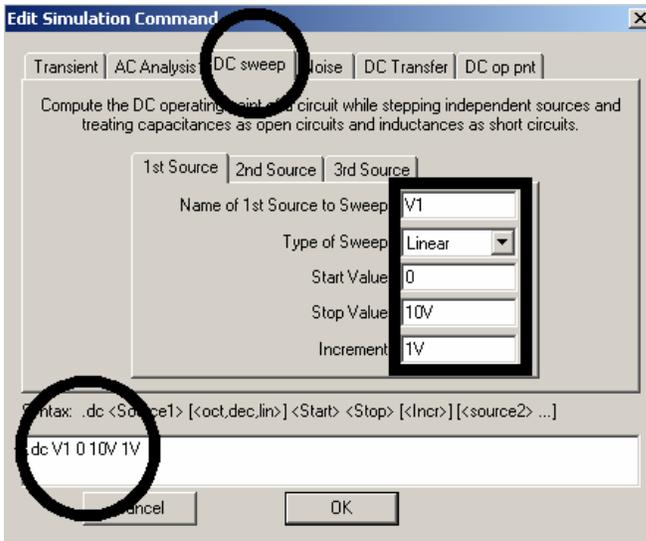


So open a new project and draw the schematic.

For the resistor R1 choose a value of 1kΩ.

Then open the property menu of the voltage source (by a right mouseclick on the "+"-sign in the symbol) and enter in the right upper corner of the menu "0" (= Zero Volts) as DC-Value.

Check the schematic for its correctness.



In the simulation menu open

„Edit Simulation Command“

and fill in the following DC-Sweep-entries for the voltage source V1:

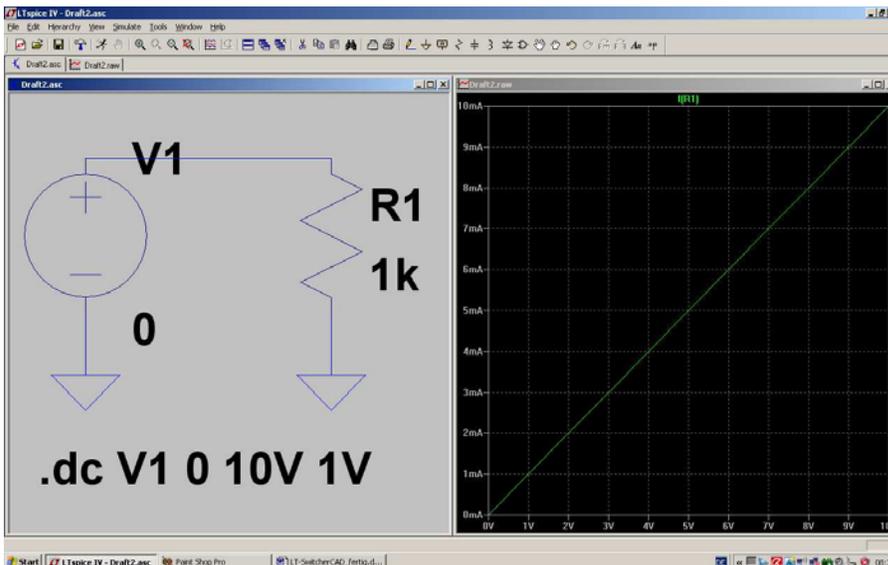
Start Value = 0 Volts

Stop Value = 10 Volts

Increment = 1Volt

Linear Sweep

Please check the so generated simulation command **„.dc V1 0 10V 1V“** in the lower part of the window.



Then close with OK and start the simulation. Move the cursor to the upper connection of the resistor R1. When the cursor suddenly looks like a "current sensor": left click and you get this screen.

8.2. Diode

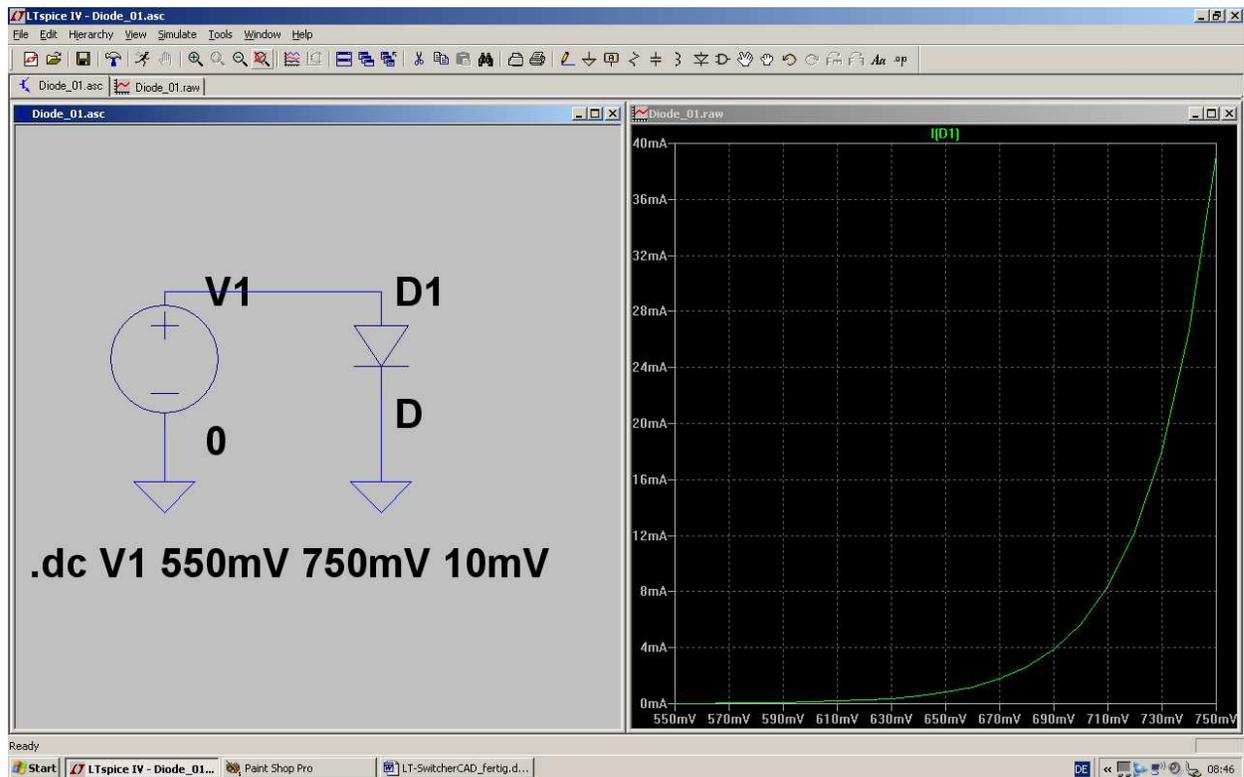
This is now a very simple exercise:

First save the project under a new name. Then delete the resistor (key F5) and replace it by a diode (= part „diode“ in the Library).

At last modify the simulation command to

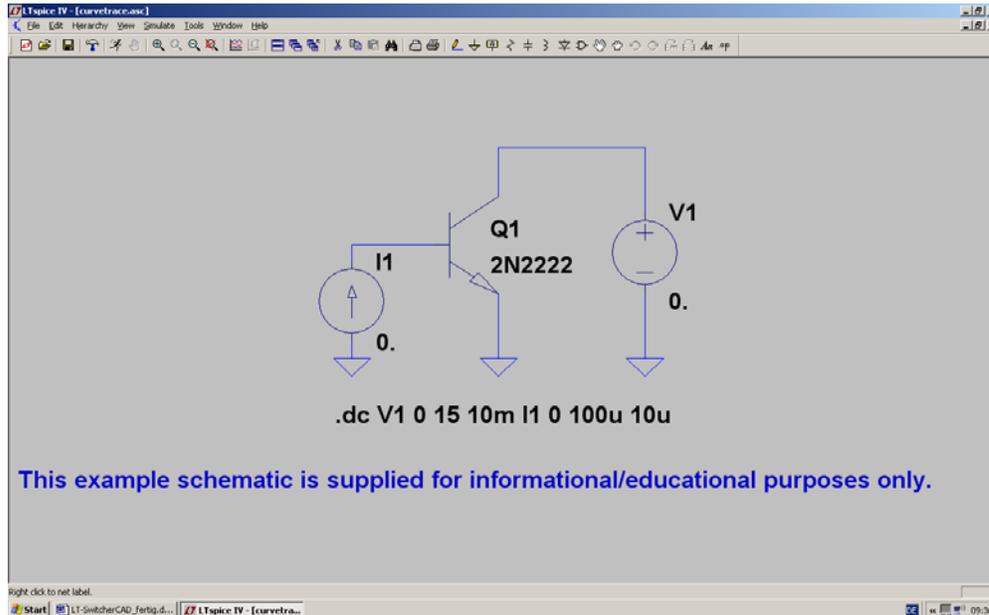
```
.dc V1 550mV 750mV 10mV
```

Now the voltage range is 550mV to 750mV with an increment of 10mV. This is the optimum for a silicon diode.



8.3. NPN Transistor

In the „example“ folder of the program you will find a very nice example „**curvetrace.asc**“ for this purpose:



You can see:

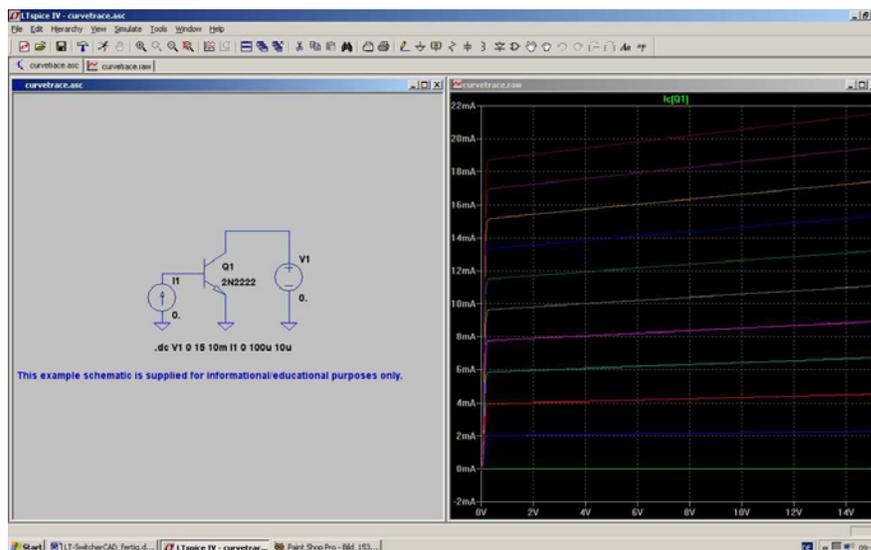
- a) A DC voltage source V1 is connected to the collector of the transistor. The voltage start value is “0”.
- b) A current source I1 with start value „0“ feeds the base of the transistor.

The simulation command

.dc V1 0 15 10mV I1 0 100u 10u

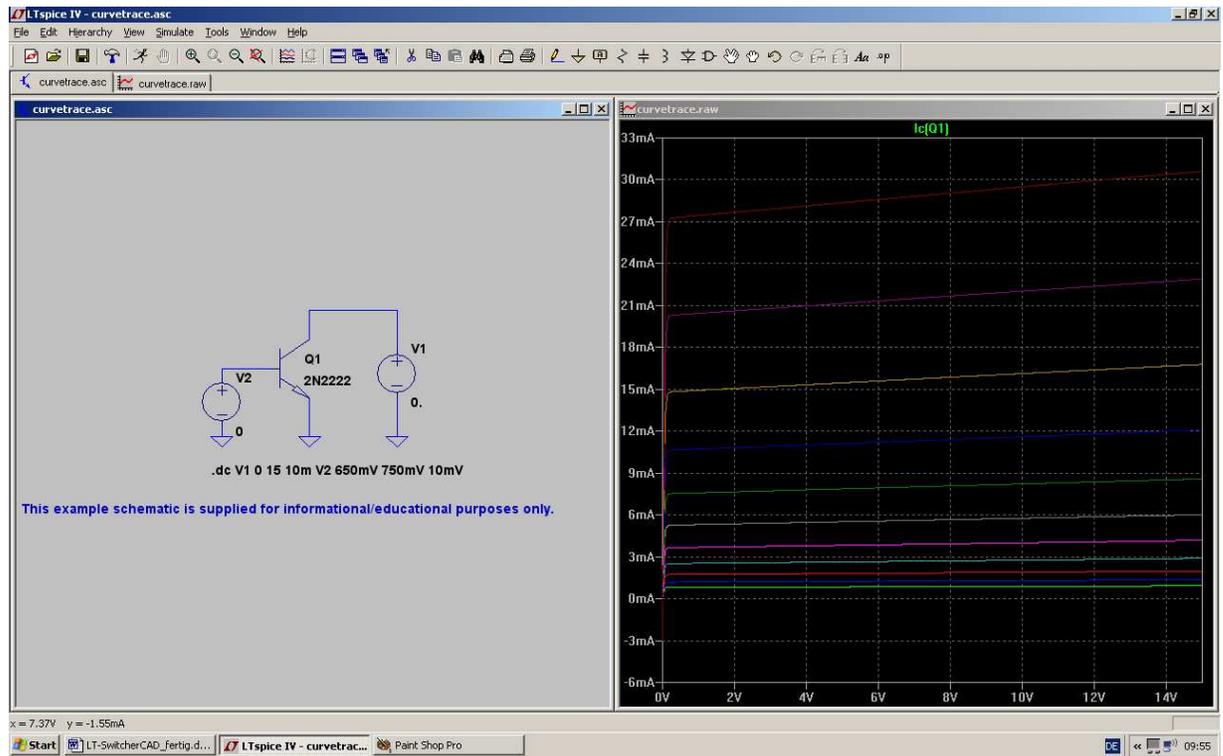
means:

**Increment voltage V1 from 0 Volts to +15 Volts in steps of 10mV. Use this voltage for the horizontal axis in the waveform viewer .
Then apply a base current between 0 and 100 Microamperes (incremented by steps of 10 Microamperes) and write the simulated curves into the waveform viewer.**



Move the cursor to the collector of transistor Q1. When the cursor suddenly looks like a “current sensor”, left click and you will get this screen.

If you are interested in the relation between collector current and base voltage, then you need this schematic:



- Replace the base current source with a voltage source. Start value is "0".
- Modify the simulation command to:

.dc V1 0 15V 10mV V2 650mV 750mV 10mV

The voltage source V1 stays the same. The base voltage V2 is stepped from 650mV to 750mV by increments of 10mV.

That is all. See the result in the waveform viewer on the top right of this page....

8.4. N-Channel Junction FET

First we have to replace the NPN transistor of the last example by the FET (= part „njf“ in the part library of the program). To show the example to its best we will use the well known

BF245B

First search on the internet with the Google entry

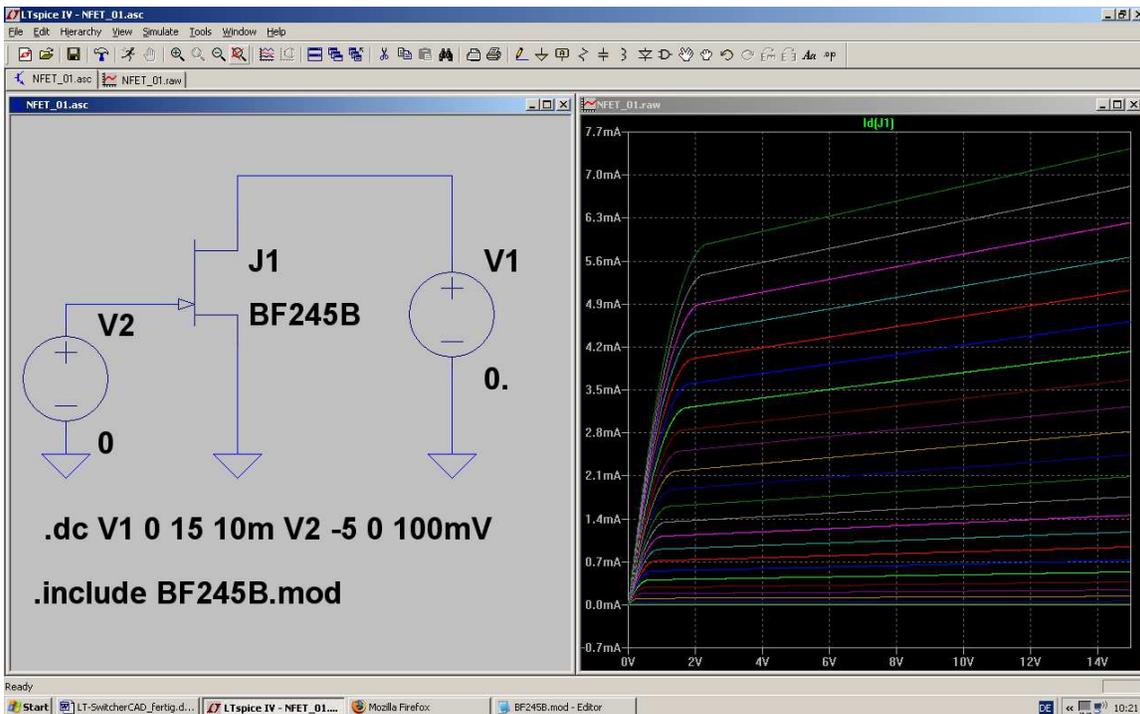
bf245b spice model

Copy the content of the result to the clipboard and then into a new editor sheet. Save this sheet as

BF245B.mod

in the folder „sub“ of the program's part library.

And then do the following:



First select the N-Channel Junction FET symbol from the library (= „njf“) and place it on the schematic. Then right click on the model description „NJF“ beside the symbol. Change it to **BF245B** and finally add a SPICE directive to the schematic:

.include BF245B.mod

Now the software will use the downloaded BF245B model for the simulation.

Lastly the gate voltage range must be altered. For a normal junction FET it is the best to use a range from

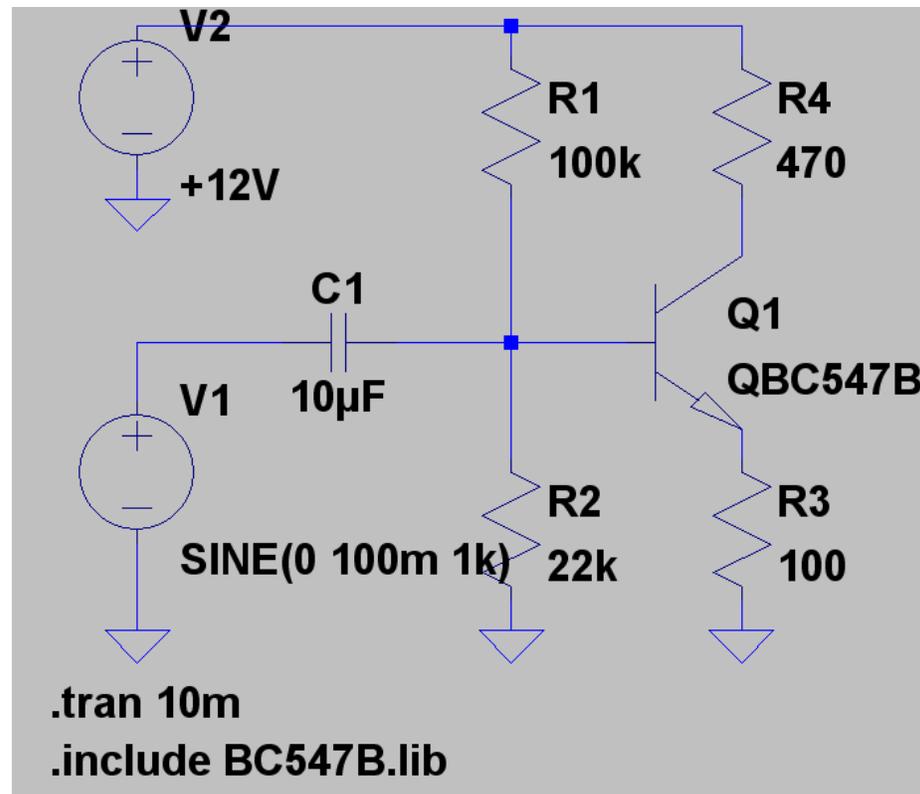
-5V to 0V in Steps of 100mV

The simulation should now run without any problems.

9. Project 5: Transistor circuits

9.1. One Stage Amplifier

9.1.1. Time Domain Simulation



We need a power supply, a sine voltage source, one capacitor, four resistors and an npn-transistor.

Let's start!

Procedure:

Step 1:

Place 4 resistors, 1 capacitor, 2 voltage sources, 1 npn-transistor and wire.

Step 2:

Enter the correct values for the resistors and the capacitor.

In the upper voltage source V2 choose „DC“ with an amplitude of +12V.

The lower voltage source V1 must be programmed as a sine wave with Vpeak = 100mV and 1kHz.

Step 3:

Simulate from 0 to 10ms.

Step 4:

Start the search engine and look for **BC547B.lib** on the internet. The content of the found html-file must be marked, copied to the clipboard and inserted into a new sheet of a text editor. Save the file as „**BC547B.lib**“ in the folder

LTC / LTspiceIV / lib / sub

Step 5:

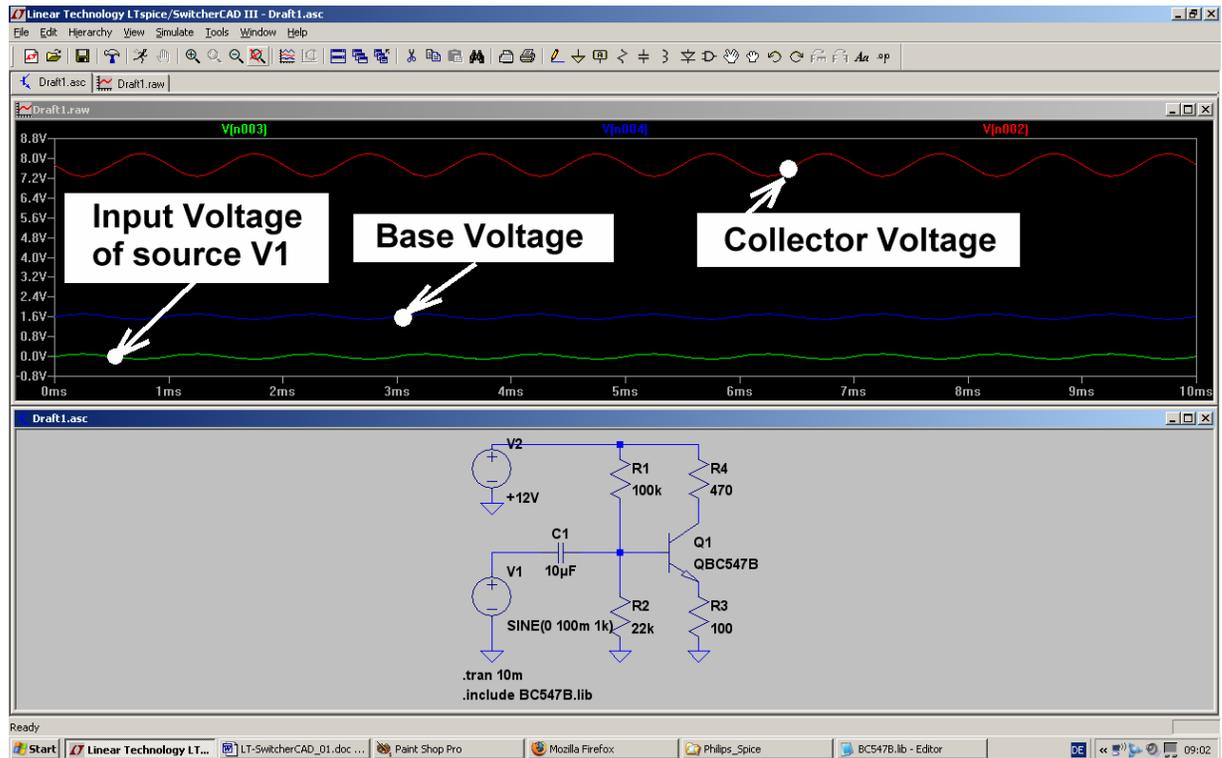
Move the cursor to the information „**NPN**“ beside the transistor's symbol, click right and enter the correct information „**QBC547B**“ (...because this is the same as the name of the Transistor's model file in „**BC547.lib**“) and

add the spice directive **.include BC547.B** to your schematic.

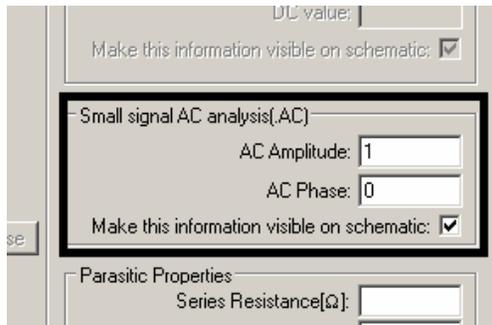
In the simulation we would like to see the input voltage, the base and the collector voltage of the transistor. Please verify the phase shift of 180 degrees caused by the transistor. Also we can test the "predicted gain" for such a stage, given by the simple equation

$$VU = RC / RE = 470 / 100 = 4,7$$

(to check this please measure the amplitude of the output voltage and calculate the ratio with the input voltage's amplitude of 100mV).

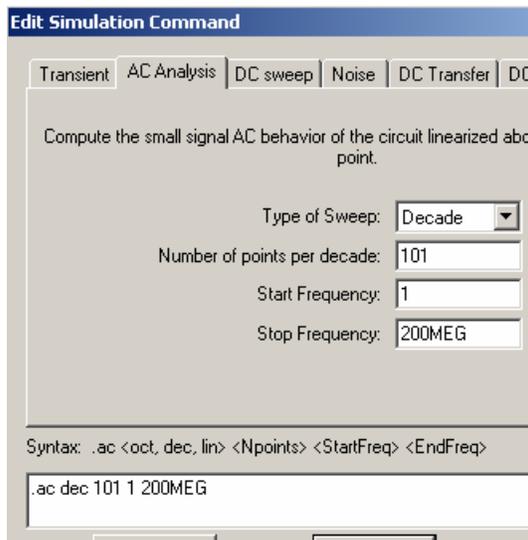


9.1.2. Frequency Domain Simulation (“AC Sweep”)



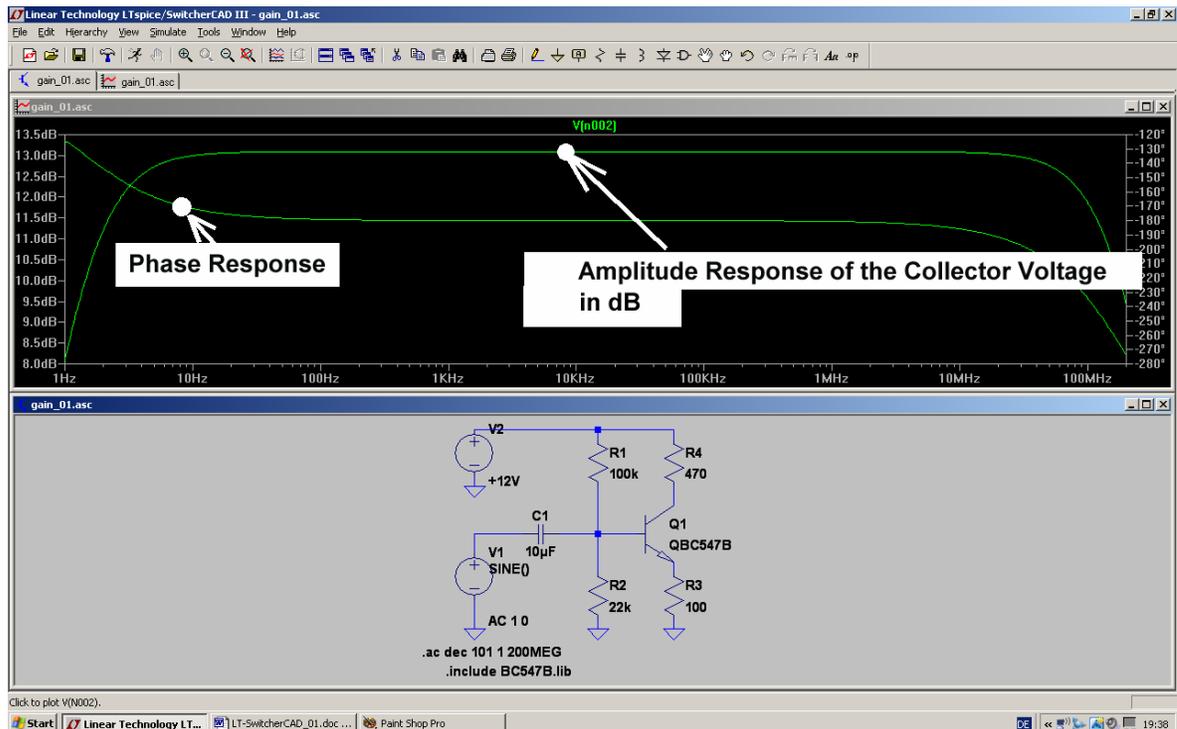
Step 1:
Change the properties of voltage source V1 to “AC-Sweep” by entering the values “1” for the AC Amplitude and “0” for the AC Phase.

Make the information visible on the schematic.



Step 2:
Open „Simulate” and „Edit Simulation Command” to get this menu. In „AC Analysis” please enter the values for a decade sweep with 101 points per decade from 1 Hz to 200MHz.

So the simulation result should look like this:



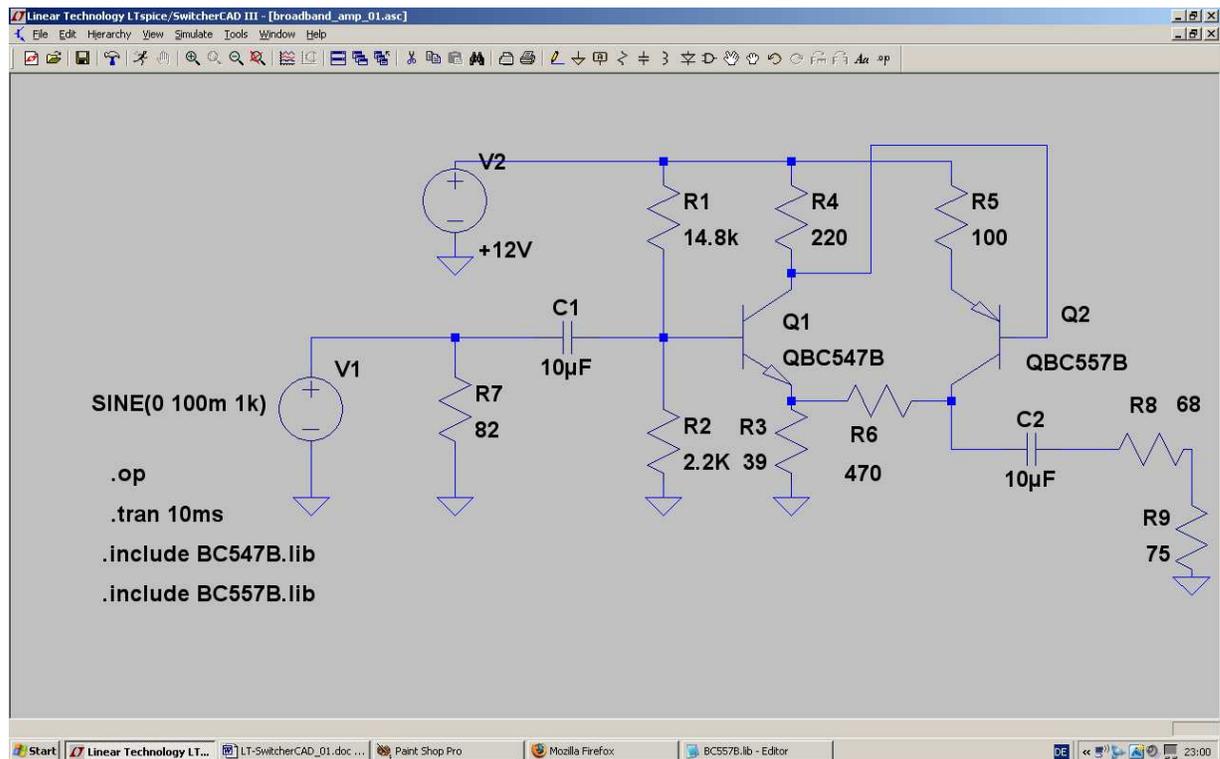
9.2. Two Stage Broadband Amplifier with Feedback

9.2.1. Task

Design a gainblock for a 75Ω communication system with the following specifications:

- Input and output resistance are both 75Ω . So a lot of 75Ω blocks can be connected in series using the RG59 cable type.
- The gain should be 6dB when loading the output by 75Ω .
- Use a power supply with +12V DC.
- The gain must be constant in the frequency range from 1kHz to 10MHz.
- Use an npn and a pnp transistor (BC547B and BC557B) with feedback.
- Use a trimmer resistor in the base circuit of the first transistor to compensate all part tolerances.

9.2.2. Circuit Diagram for Simulation



Draw this circuit and take note:

The values of the resistors R1 and R2 must always be written as

14.8k and **2.2k**

Never write 14,8k or 2,2k -- you will get wrong results without any warnings!!

The **npn transistor BC557B** is not included in the LT's library, so you need the internet again. When you have found the model file, copy it into an editor's new file (do you remember? You receive a HTML file) and save it in the library folder.

Note: When you have found the model files the transistors are named **QBC547B** and **QBC557B**. Use these names in the spice directive and in the schematic.

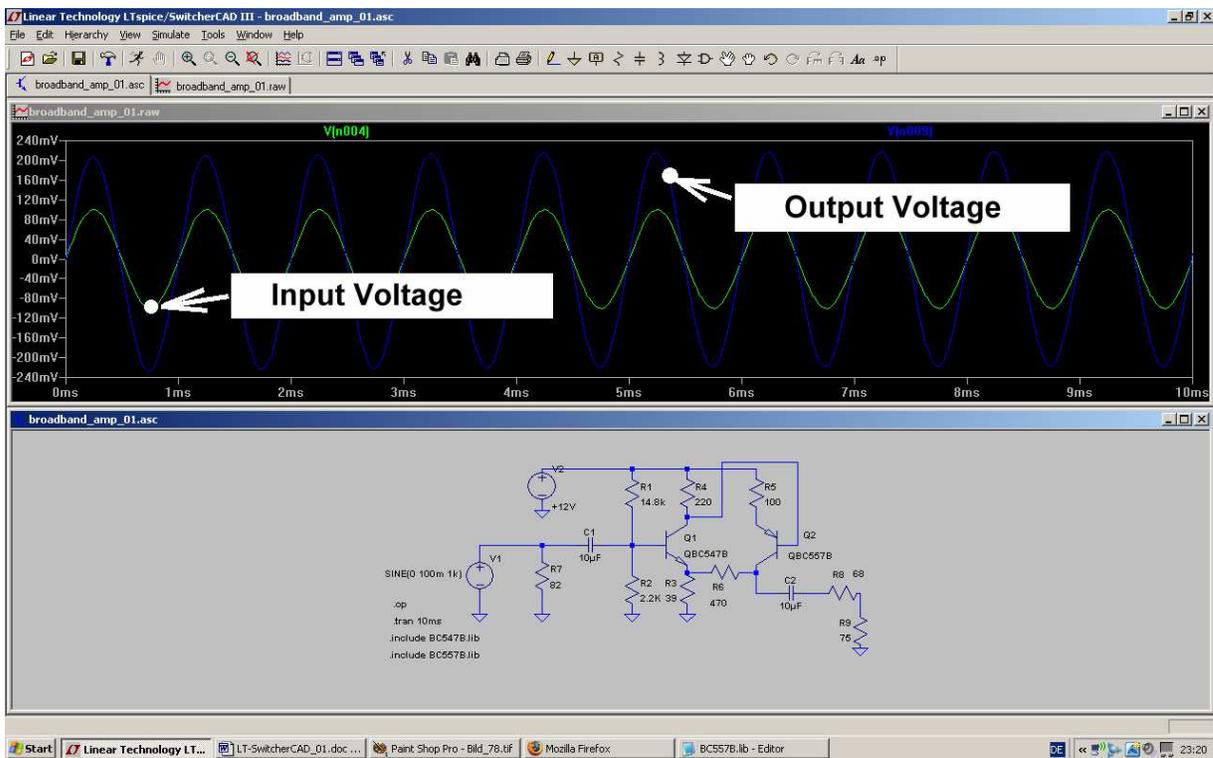
The input voltage is a sine wave with a peak value of 100mV and a frequency of 1kHz. Simulation time is 0....10 Milliseconds.

Something new is the command **„.op“**

With this command you get information about the DC-values in the circuit (= „DC operating point“). The information is presented in form of a list.

The test points in the schematic are the voltage source V1 and the termination resistor R9 (75Ω) at the output.

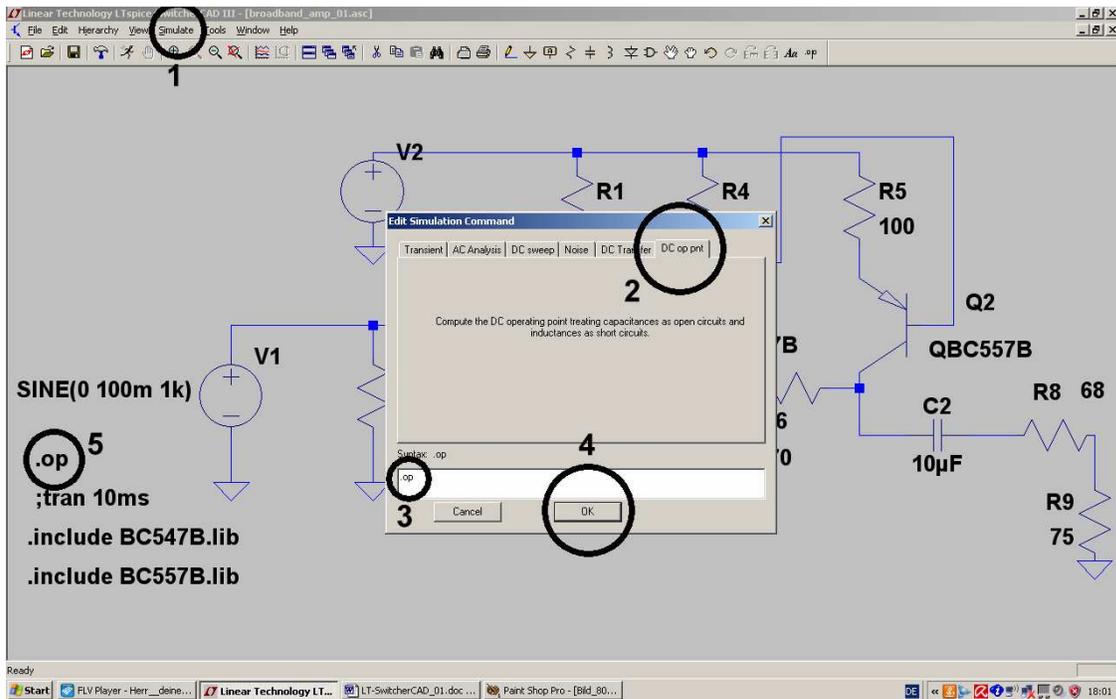
9.2.3. Time Domain Simulation



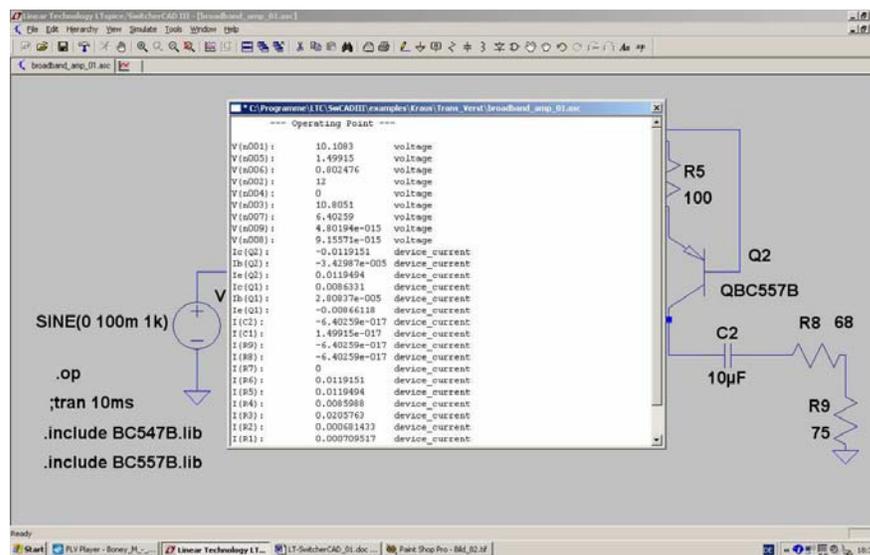
As required the output voltage's peak value is twice that of the input voltage and that gives a gain of 6 dB..... Two common emitter stages produce a phase shift of $2 \times 180 = 360$ degrees. So the output voltage is again in phase with the input voltage.

9.2.4. DC Bias

If you want to know the DC-Currents and DC-Voltages for the operating point, then you should do this:

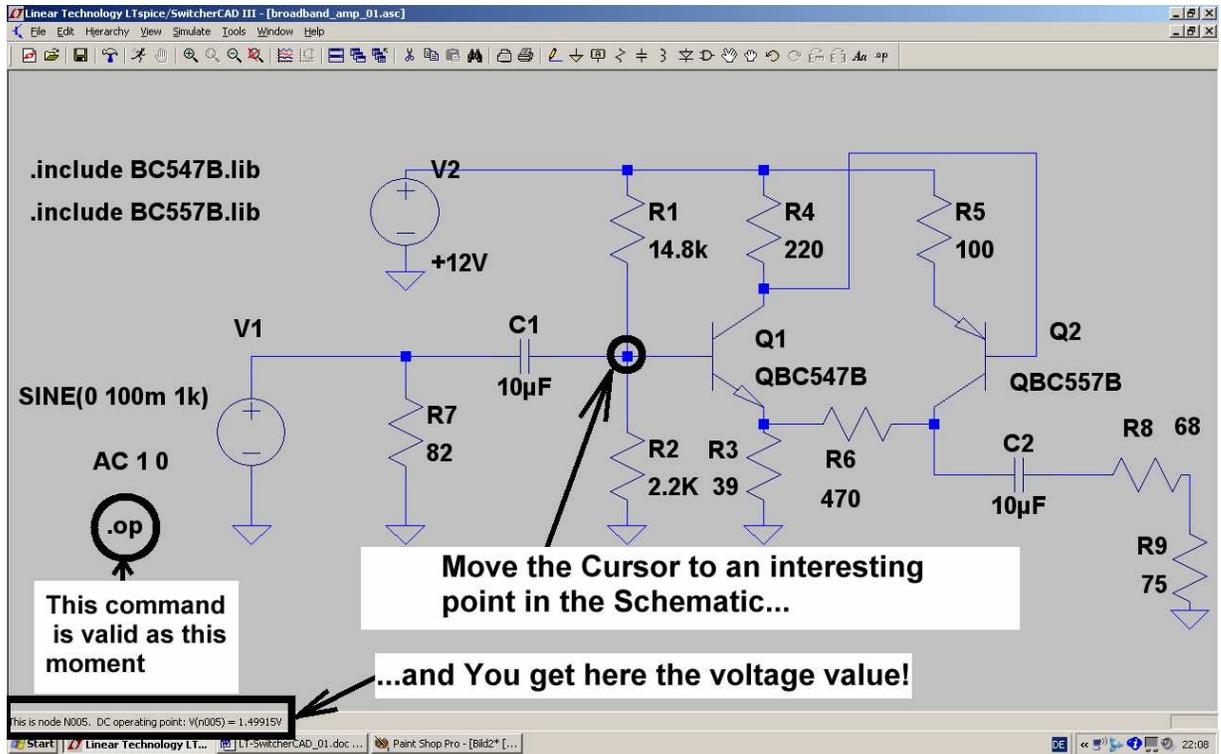


- 1) Open „Simulation“ and „Edit Simulation Command“.
- 2) Change to „DC op point“
- 3) In the window you should now find **„.op“**
- 4) Confirm this by pressing OK and test, whether
- 5) the command **„.op“** is now connected to the cursor. if it is then it can be placed on the schematic. (You can check this as the „Tran“-Command will now start with a semicolon (**;tran 10ms**)?)



Now press the simulation button (with the running guy) and wait for the result. This is what it looks like and you can examine the current / voltage list.

Now close the list because there is an easier way to get every current- or voltage value (See next page):



Note:

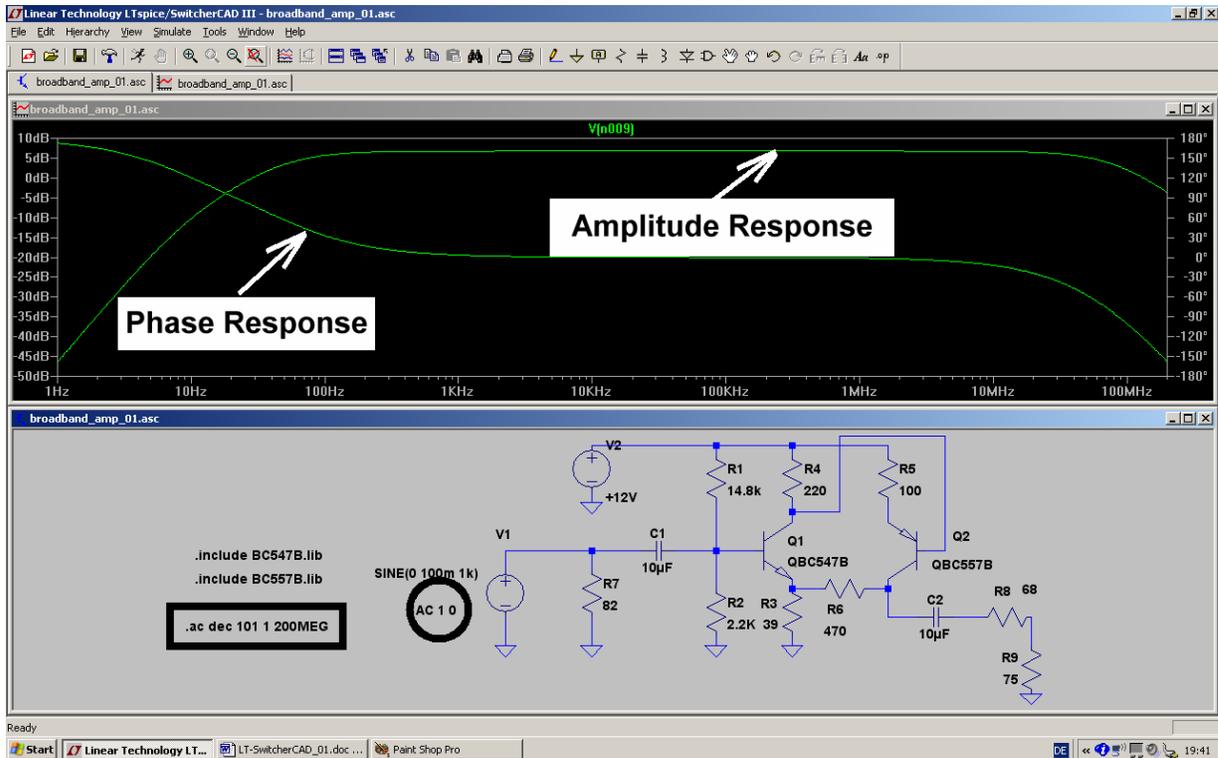
To return to the transient simulation, please replace the point in the `.op` simulation command by a semicolon. Also replace the semicolon in the `;tran` command by a point.

But the fastest way is using the path „Simulation / Edit Simulation Command / Transient”. There click OK and this operation will be automatically done by the software.

9.2.5. Frequency Domain Simulation: AC Sweep

No problem:

Open the property menu of the voltage source V1, enter „1“ for the AC-amplitude, „0“ for the AC-phase and a new simulation command for an AC-sweep from 1Hz to 200MHz with 101 points per decade:



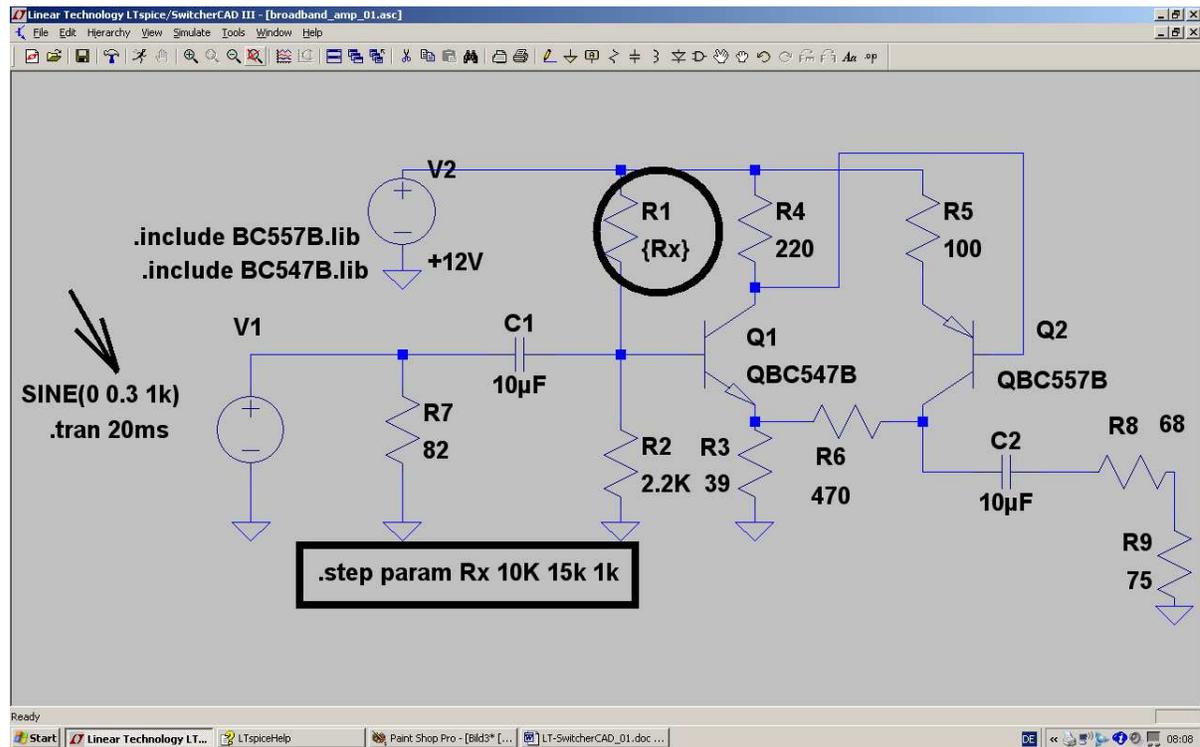
9.3. Parametric Sweep

If you have a look at “LTspice / Dot Commands” in the online help, then you will find a lot of commands and capabilities.

Let us test the SPICE-command **.step**

This is important because with „step“ you can automatically vary the value of a part or a voltage and repeat the simulation.

Let's take the circuit of the last section and vary the upper resistor R1 in the base voltage divider in 5 steps from 10k to 15k. This will enable you to get the optimum operating point.



Note:

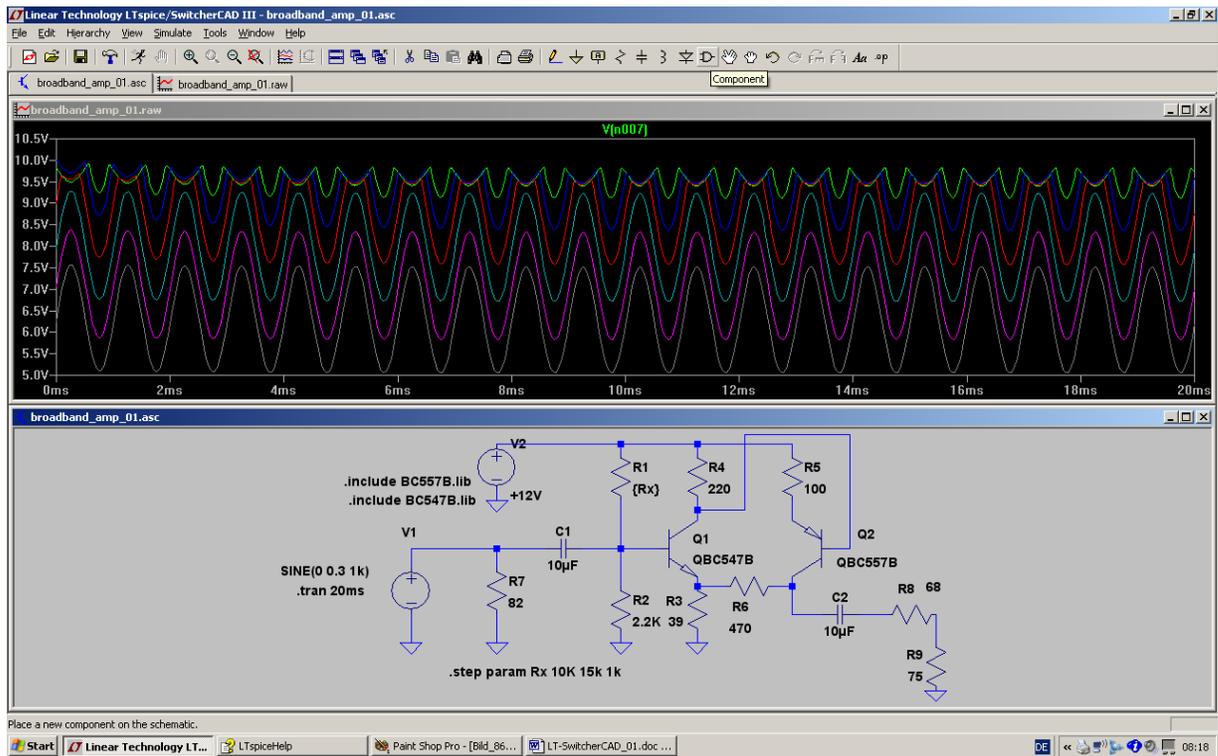
a) The peak value of the input voltage is now 0.3V

b) The value of R1 is replaced by the variable **{Rx}**

c) With the spice directive **.step param Rx 10k 15k 1k** the value of Rx is increased from 10k to 15k in steps of 1k.

Now press the simulation button (the “running guy”). The simulation will run, but you don't see any results. So move your cursor to the collector of the transistor Q2 and click.

You will see this result and that a value of approx. 15k for R1 (= lowest curve) will do the job perfectly (and give a $V_{DC} = +6.5V$ at the collector of Q2)



10. Project 6: OP Amp Circuits

10.1. Let us start with an Inverting Amplifier

Let us have a look at a simple OP Amp circuit using one of the LT models which come with the software.

Task:

To simulate an inverting amplifier with the „LT1001“ to have the following specifications:

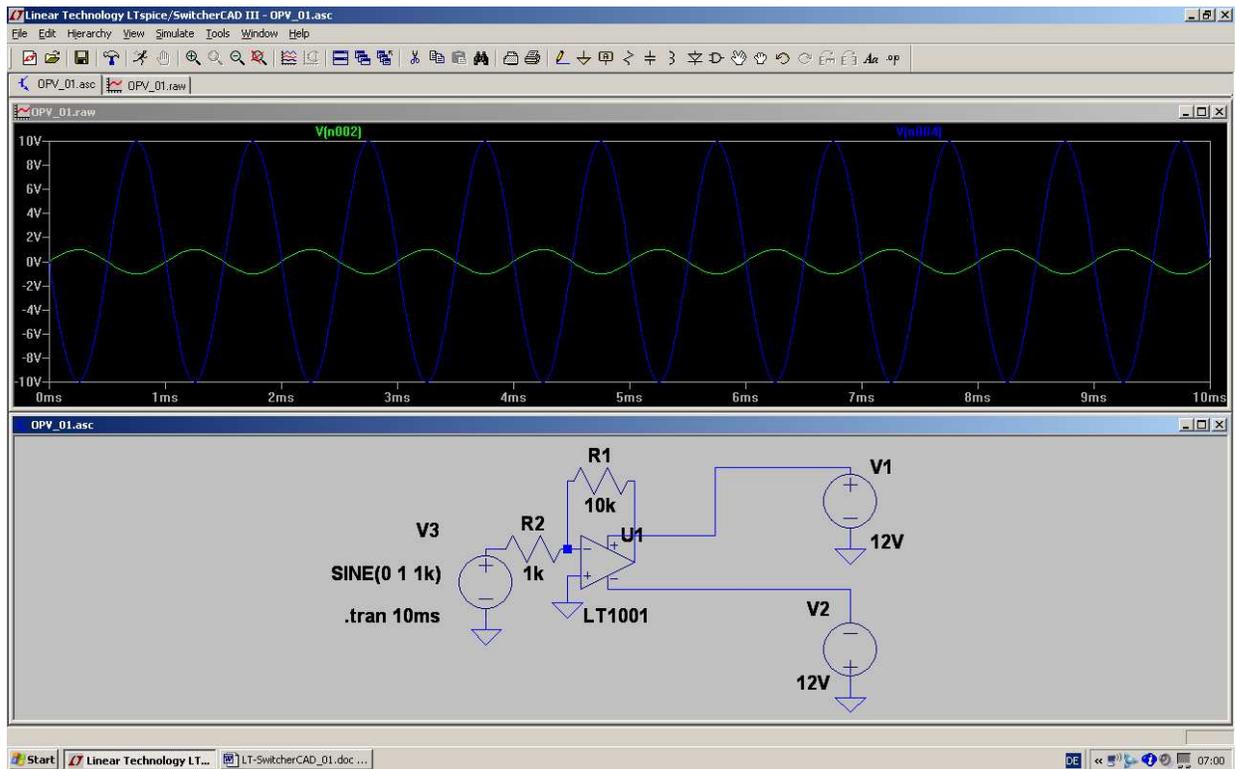
Gain = 10 (= 20dB)

Input Resistance = 1 k Ω

Use two power supplies with +15V and -15V

The input signal is a sine wave with a peak value of 1V and a frequency of 1kHz. Simulate the input and output voltage from 0...10ms.

Solution:



You see, as predicted by theory, that there is a phase shift of 180 degrees between input and output.

Additional task:

Use an AC sweep to determine the frequency response and the upper cutoff frequency. Simulate up to 1MHz.

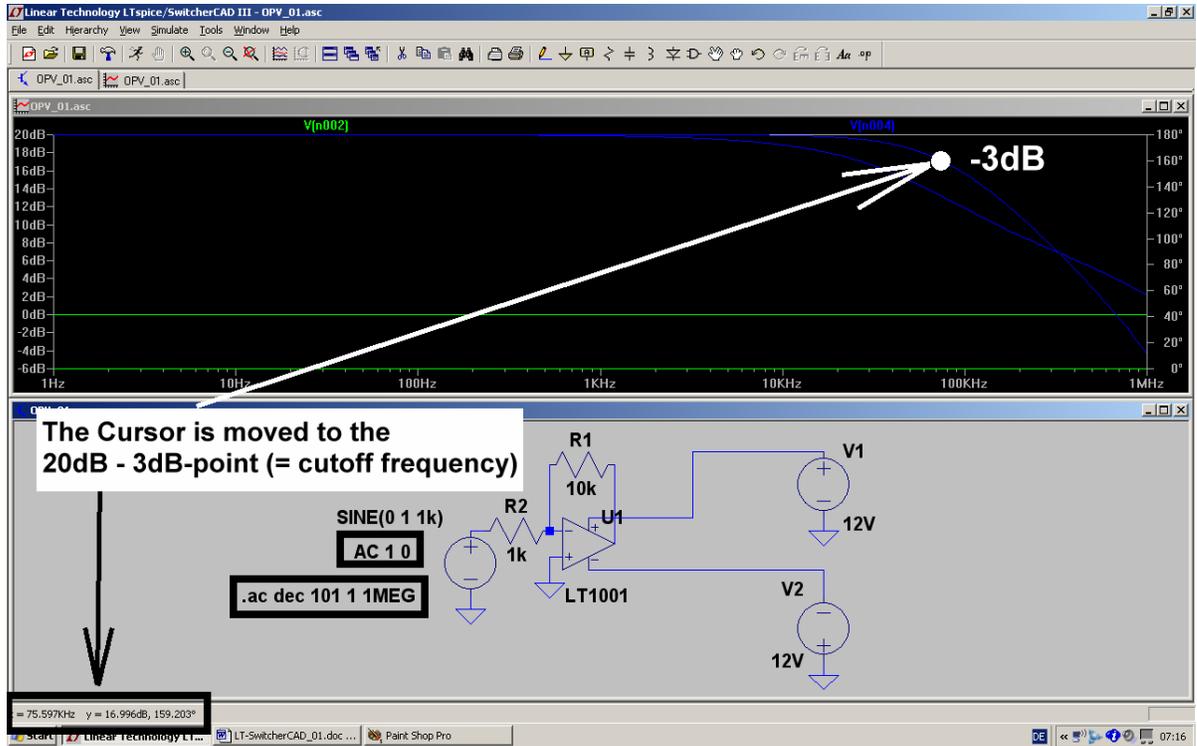
Solution:

In the properties of the voltage source V3 set **AC amplitude = 1** and **AC phase = 0**

Then write the simulation command

.ac dec 101 1 1MEG

for a decadic sweep with 101 points per decade from 1 Hz to 1MHz.



Using the cursor you will find the corner (= cutoff) frequency at 75.6kHz. Here the gain has decreased by -3dB.

10.2. Preparing a SPICE Model from the Internet

When doing your own development work you often need a special circuit from a manufacturer. Most companies offer their SPICE model collection free of charge on the Internet. So search for and use them!
Modern OP Amps can work up to 500MHz and more - even the circuit's noise can be simulated. Let us use such a modern part in a simulation and learn how to create a symbol for a downloaded complex circuit.

10.2.1. Gainblock for 1kHz to 30 MHz with OPA355

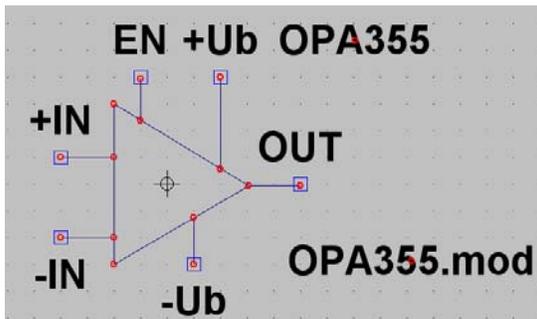
The OPA355 is a modern low noise OP Amp with a cutoff frequency of 200MHz (there the gain has decreased to 1). We want to create a „gainblock“ for a 75Ω – system with it.

Specifications:

- Only one power supply with +5 V (= maximum Vcc value). Therefore the mean output DC voltage must be set to +2.5V.
- Gain = 4 for an unloaded output, but gain = 2 with a terminated output pin.
- Input resistance = output resistance = 75Ω .
- Lower corner frequency = 100 Hz. Upper corner frequency = 50 MHz for a gain reduction of 0.5dB.
- Use a non inverting circuit.

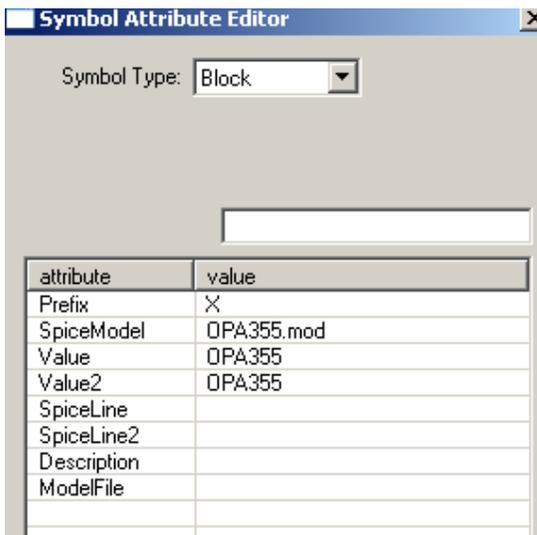
10.2.2. Simulation using the selfmade OPA355 Subcircuit Model

The necessary SPICE model can be found as „sbom143.zip“ in the Texas Instruments homepage. After de-zipping save it as „OPA355.mod“ in the well-known library folder „lib / sub“.



Once more you have to create a new symbol (See chapter 6.2.3).
Be aware of the additional **ENABLE** pin. In the schematic this pin must be connected to the supply voltage (+5V):

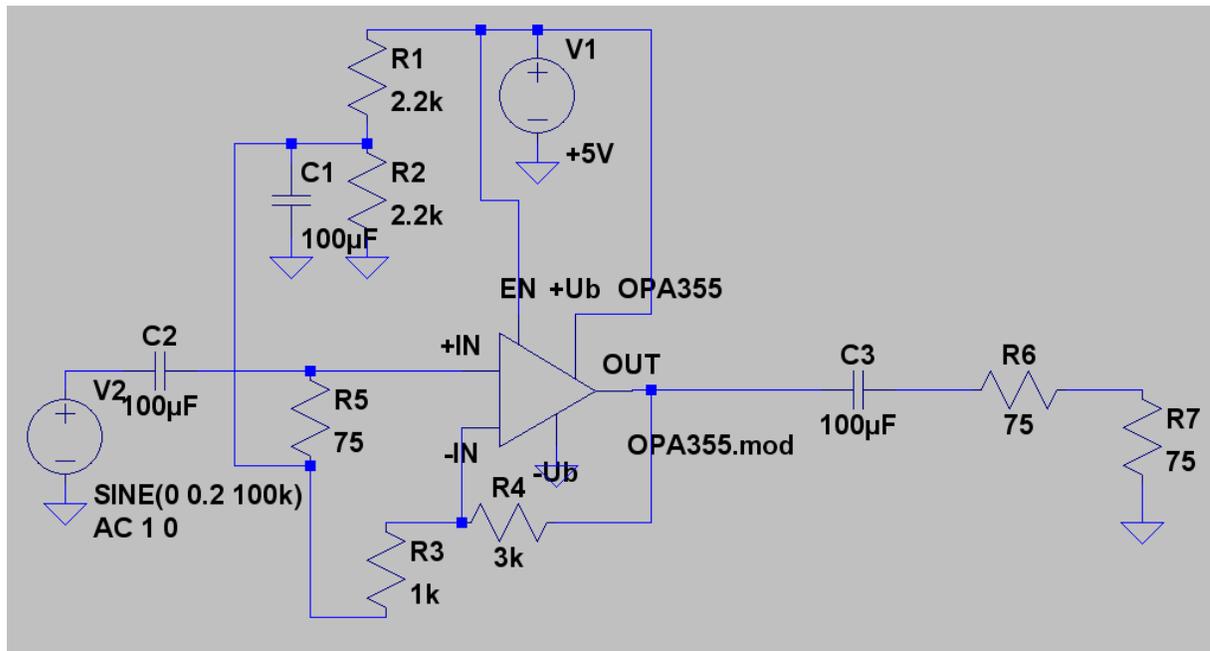
This should be the result...



...and this is the correct attribute list. Only with these entries will everything run well.

Warning:
The value for the attribute “Value2” MUST NEVER BE EMPTY!!!
(...the best is to use the SPICE model name...)
If this entry is missing, then you can get cryptic error messages and the simulation aborts...

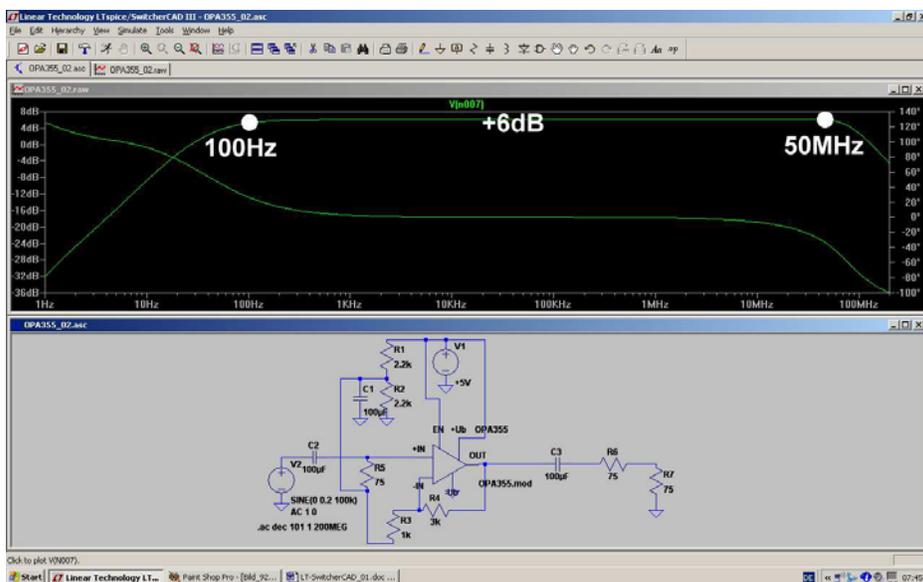
Let us now draw the complete circuit:



Explanations:

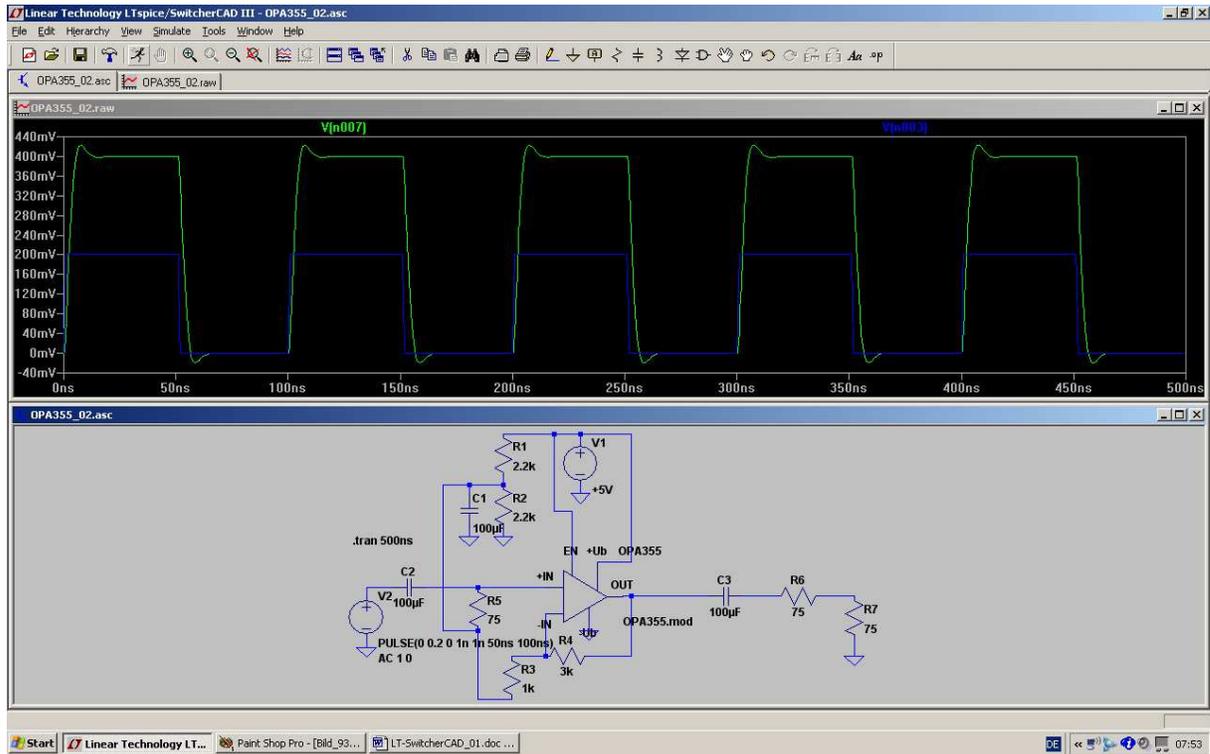
- The +5V supply must be connected to the pins „+Ub“ and „Enable“. Also the voltage divider (consisting of R1 and R2) must be connected to +5V to get the necessary +2.5V at the input pins.
- R5 gives the input resistance of 75Ω, R6 the output resistance of 75Ω.
- R4 and R3 form a feedback network. So the „non-inverting function“, and an unloaded gain =4 and a loaded gain = 2 (+6dB) are realised

This is the result of the simulation for the output voltage at R7:



Fine!

And something interesting is to feed the circuit with a pulse voltage (peak value = 0.5V / frequency = 10MHz):



Very nice, indeed

10.3. Use of Labels

In general OP Amp circuits need a positive and a negative supply voltage. The fully drawn circuit then gets a little bit untidy, so applying some **Labels** simplifies the appearance.

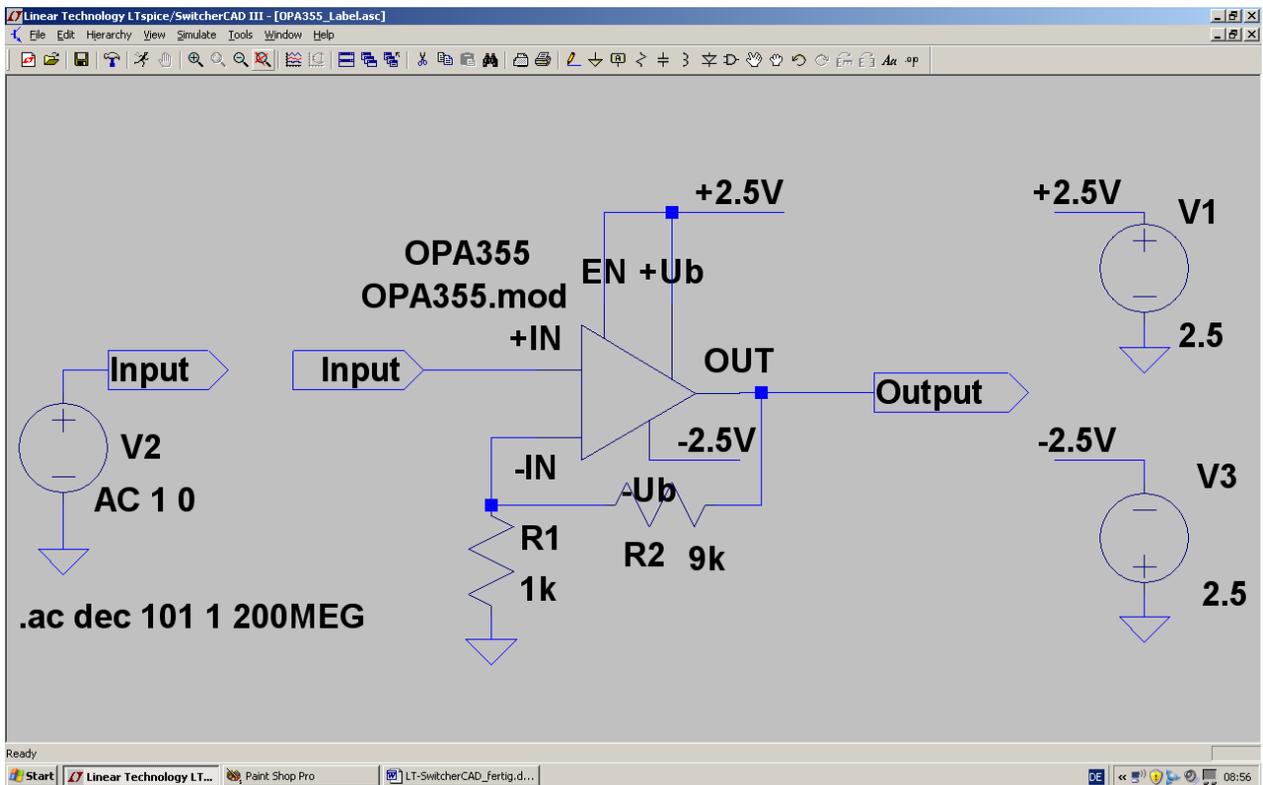
When using a label you replace the wiring by a simple connection with a name for the node at all points where it is needed and you no longer need to draw all the necessary wires.

Let us use the OPA355 from the last chapter and build a new gain block (gain = 10 (+20dB) ; non- inverting circuit; 2 supplies of +2.5V and -2.5V).

Also use a label for the input and for the output nodes.

**This has another advantage:
In the waveform viewer you now get the label as information for the trace
instead of the node number....**

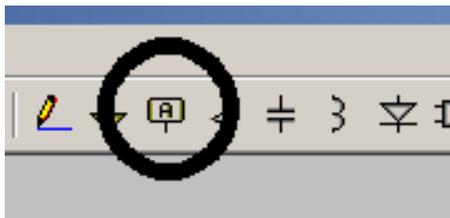
So the schematic looks like this, preset for a sweep from 1Hz to 200MHz:

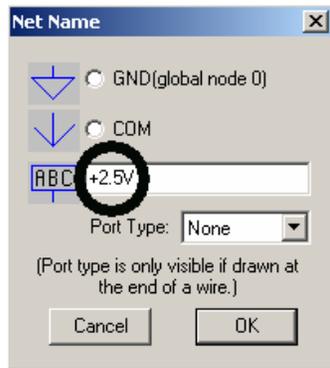


And now let us do the work, starting with the 2 power supplies.

Step 1:

Delete all wiring between the power supply and the circuit and press the "label button":





Step 2:

Now enter **+2.5V** in the field and click on OK. The label “hangs” on the cursor and can be placed on the schematic and wired.

Note:

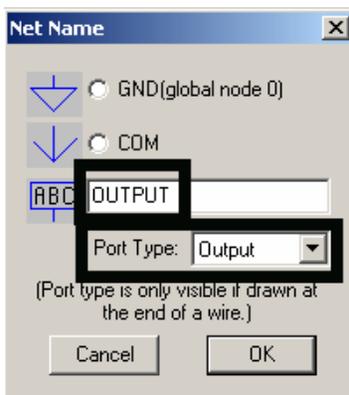
Before placing the label, it looks like a huge rectangle with an additional little rectangle on the centre of the lower edge. This little rectangle is the “catching point” and must be connected to the symbol of the power supply!

Step 3:

Now click and place the label at every point in the schematic which needs the +2,5V supply. When all the instances are placed then connect each one to the corresponding circuit point.

Note: if wires are already present the labels can be placed directly on the wire using the “catching point” and they will automatically be connected.

Repeat steps 2 & 3 for the “-2.5V supply



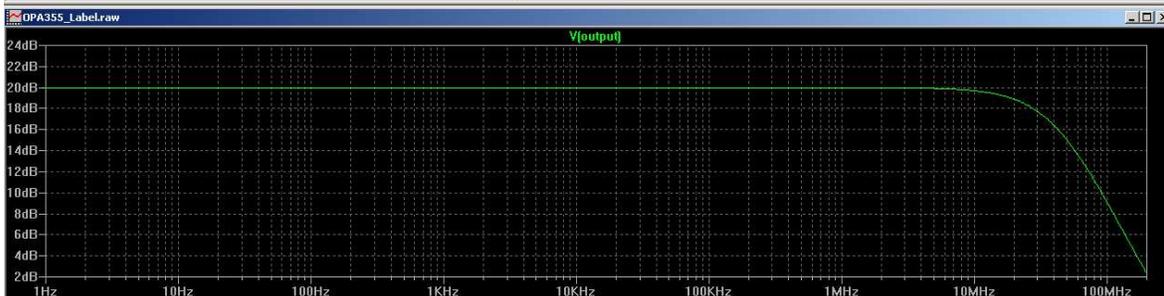
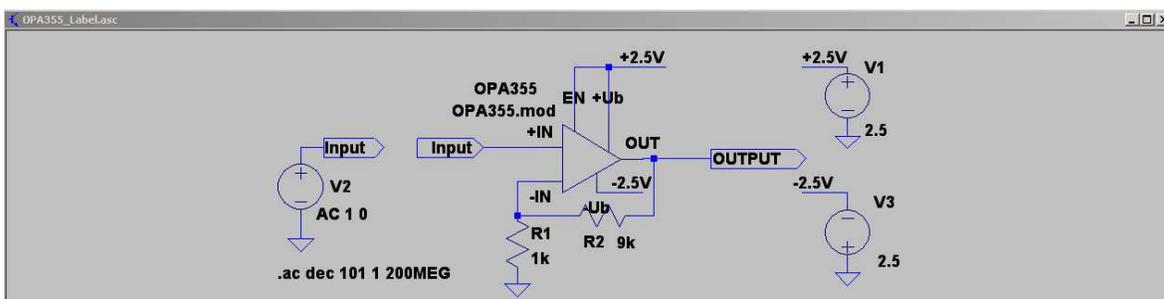
Lastly, repeat this procedure to place the „input“- and „output“ labels.

But please note:

The input of the OP Amp needs the label „INPUT“-- but select the Port Type as „Input“.

The output of the OP Amp needs the label “OUTPUT” -- but select the Port Type as “Output”.

And this is the sweep result (only gain, without phase):



11. Project 7: DC-DC Converters

First a **warning**:

These circuits use complex simulations and realistic inductance properties because an oversimplified inductor model will cause errors or simulation aborts in many applications. Very expensive simulation software versions use special (and secret) tools to avoid these problems, but we have only the normal Berkeley SPICE kernel.....

With the following rules you can avoid most of these problems:

- a) **Never use an ideal inductance -- always add a series resistor (even 0.01Ω can do the job!).**
- b) Every inductor has a winding capacitance, and often the circuit has additional stray capacitance. This always gives **resonance effects** with (sometimes) unexpected results and problems during the simulation. So use damping resistors or long simulation times to get correct simulation results.

11.1. Model for the power-MOSFET „IRFZ44N“

For all the following experiments we need a good and fast electronic switch -- today this is always a power MOSFET. Choose the IRFZ44N (produced by IRF = International Rectifiers) which is a good choice and widely available part.

Sorry, but now you have to create a symbol again

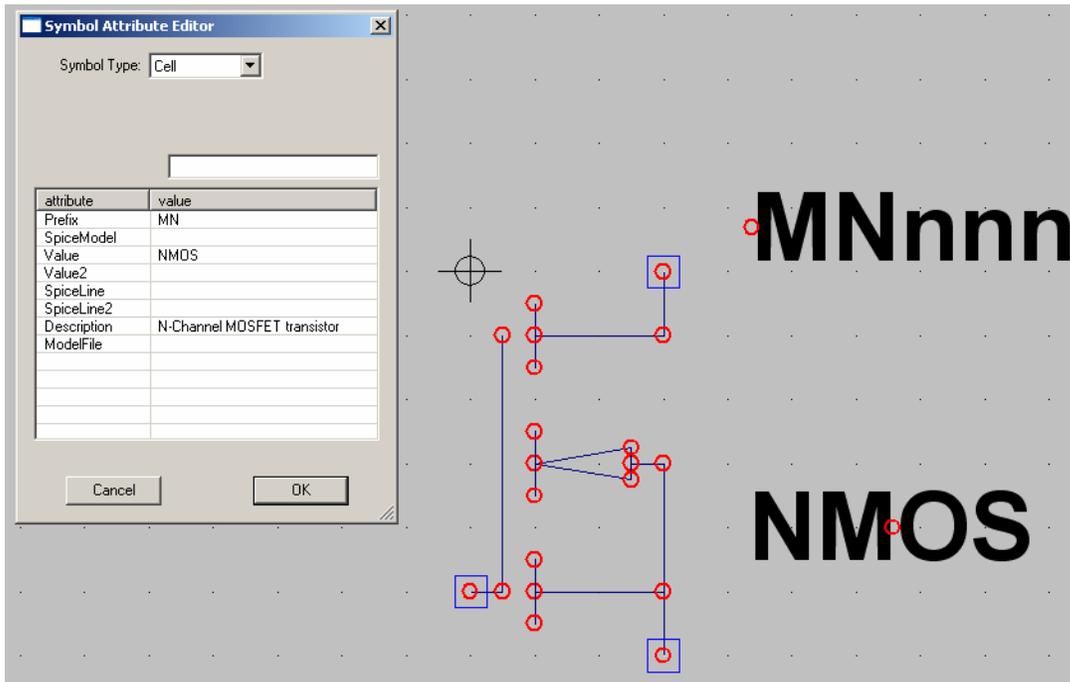
Let's go:

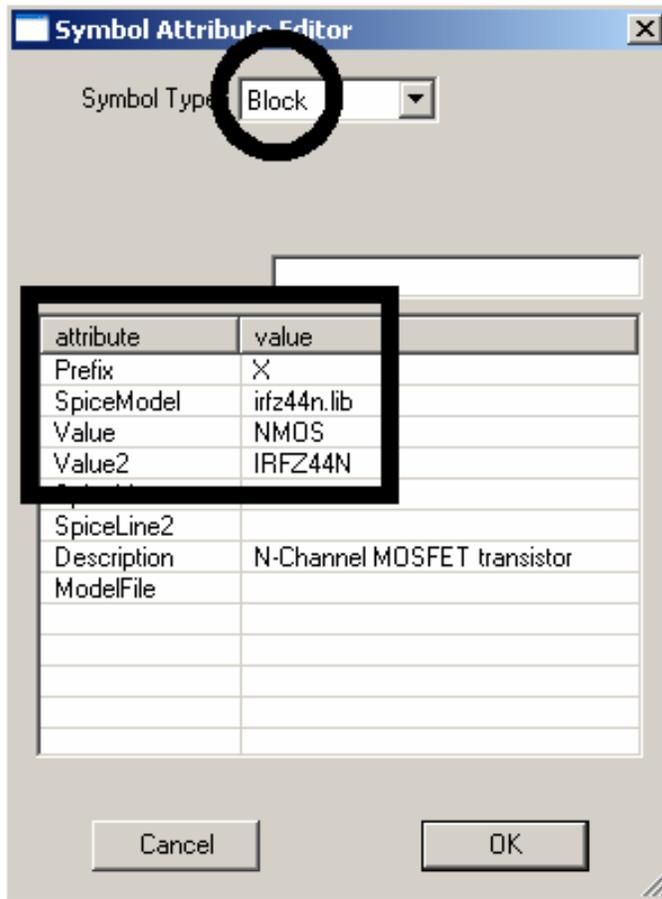
Step 1:

Search with Google for "irfz44n spice model" and find the file "irfz44n.spi". Mark the content, copy into a new editor file and save this file as "irfz44n.lib" in the folder "lib / sub" of LTspice.

Step 2:

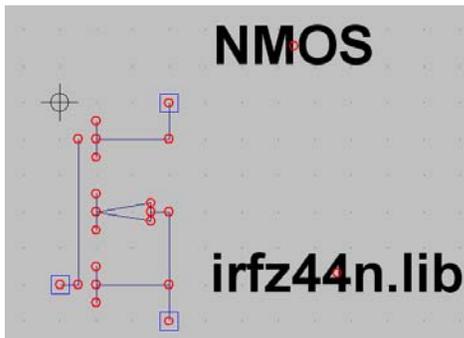
Choose "New Symbol" in the "File" Menu. Open "File" once more and -- with "open" -- follow the path "lib / sym" and find "nmos.asy" (.To see it you must make "All Files" visible). Then open the attribute list of this part by following the path "Edit / Attribute / Edit Attributes".





Step 3:
And now carefully modify the entries as given in the figure on the left.

(Pay special attention to the first line: „Prefix“ must be changed from MN = model to „**X**“ = **subcircuit**)

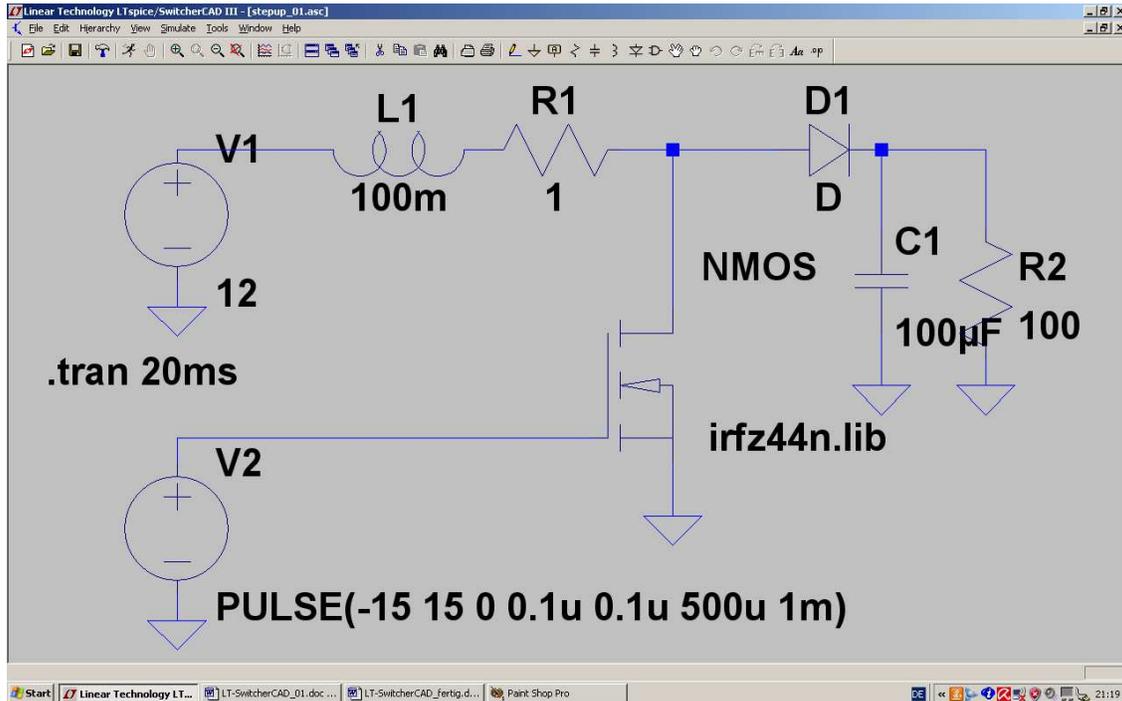


Step 4:
Open „Edit“ and „Attribute Window“. Click on „Value“ and place the description „NMOS“ over the symbol. Then press OK, repeat the procedure but click now on „SPICE File“. The information „irfz44n.lib“ must now be placed beside or below the Symbol.

Now save the complete symbol -- It is best to create a new folder „NMOS-FETs“ in „lib / sym“ and to save the new symbol in it.

That's all -- now let's tackle the first task..

11.2. The Step Up Converter



The schematic shows a +12V power supply , followed by an inductance with $L = 100\text{mH}$ and the obligatory series resistor of 1Ω .

The Power-MOSFET acts as an electronic switch between R1 and D1 to connect this node to ground for a definite time.

When switched ON, current flows from the power supply through the inductor and its series resistor to ground and rises linearly (...this is true, when the voltage across an inductor is constant) .

So more and more magnetic energy is stored in the inductor and this can be calculated from:

$$E = \frac{1}{2} \cdot L \cdot I^2$$

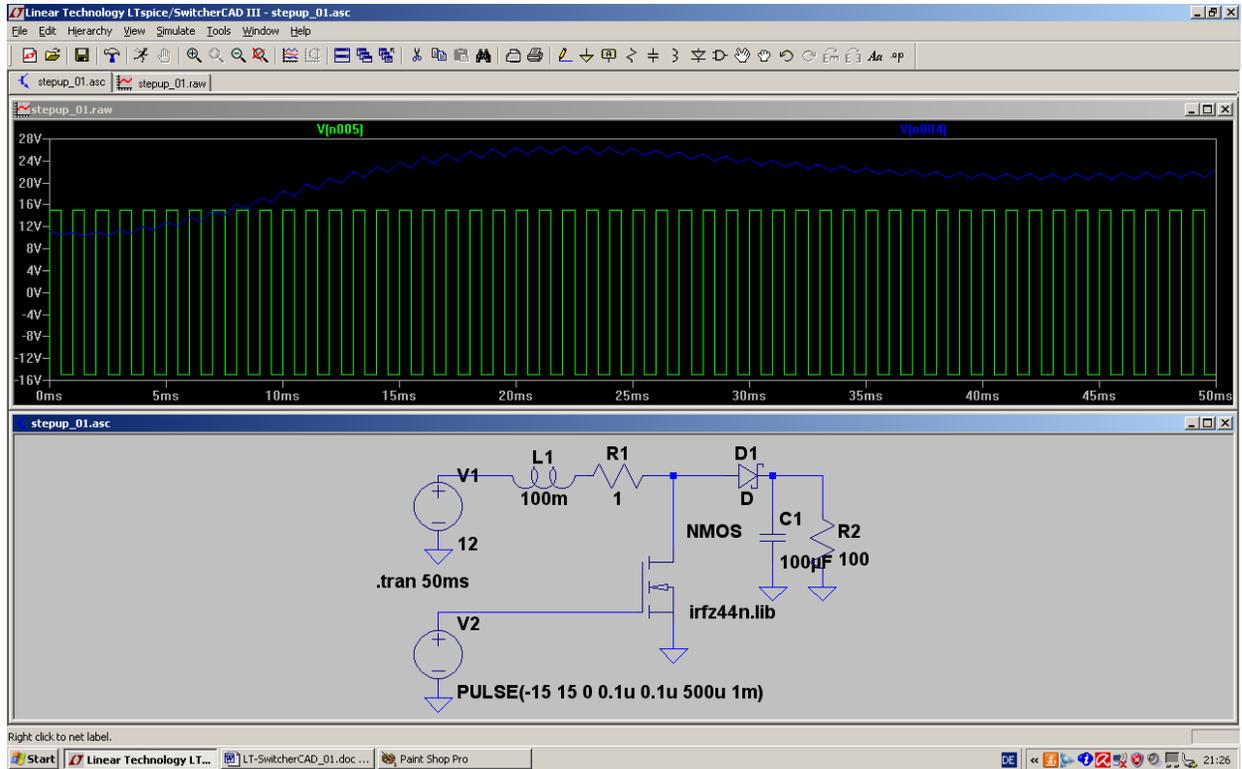
When the FET is switched OFF, this energy cannot dissipate suddenly and the inductor's current cannot switch rapidly (...this would cause a nearly infinite induced voltage...) and so stays at the same value.

That means that the inductor current now flows to the right hand side, switches ON the schottky diode, charges the output capacitor and flows through the output resistor R2. When the gate of the MOSFET is fed with a square wave voltage, the output voltage rises until a "balance" is reached between picked-up-energy (by the Inductor) and the energy delivered to the output resistor R2.

Important: the amplitude of the output voltage is the sum of supply voltage +12V and the inductors voltage!

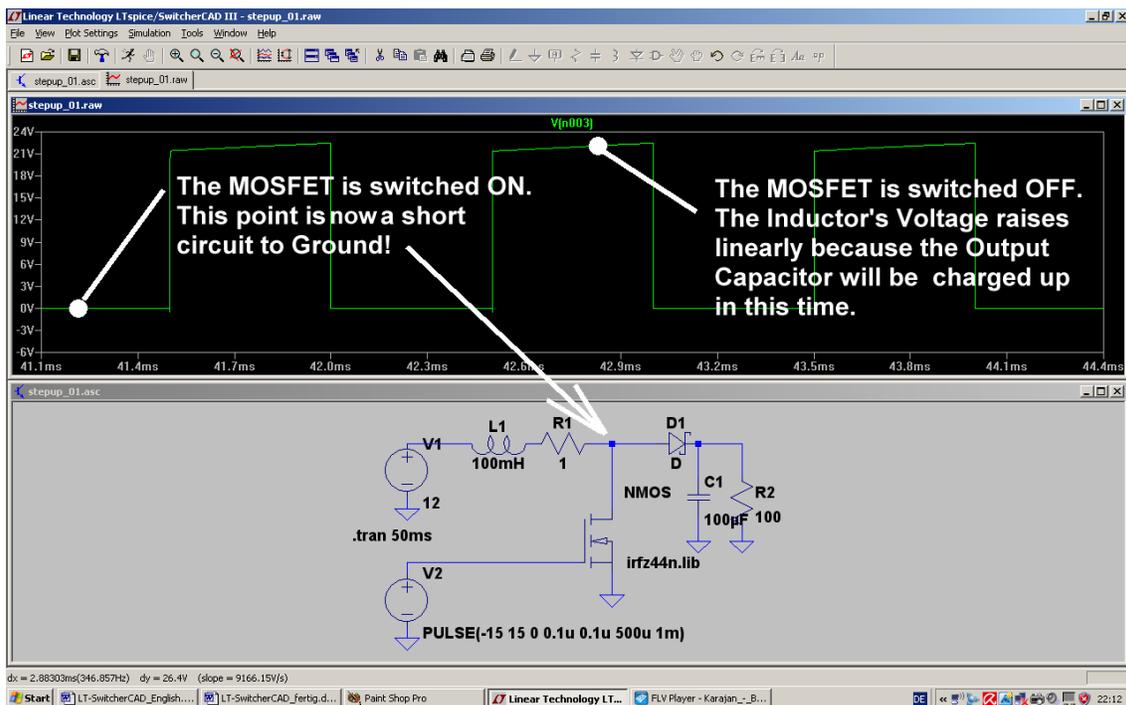
This means that the output voltage of this circuit will always be higher than the supply voltage!

Now draw the schematic and feed the gate with a symmetric square wave voltage (frequency = 1 kHz / $V_{\text{max}} = +15\text{V}$ / $V_{\text{min}} = -15\text{V}$). Choose a simulation time from $0\text{.....}50\text{ms}$. Show the gate voltage and the output voltage.



Clearly, it can now be seen that the output voltage is a mixture of a mean DC Value of +23V and a small „sawtooth-ripple“ (= charging and discharging of the output capacitor).

Lastly, let us have a look at the drain voltage of the MOSFET:

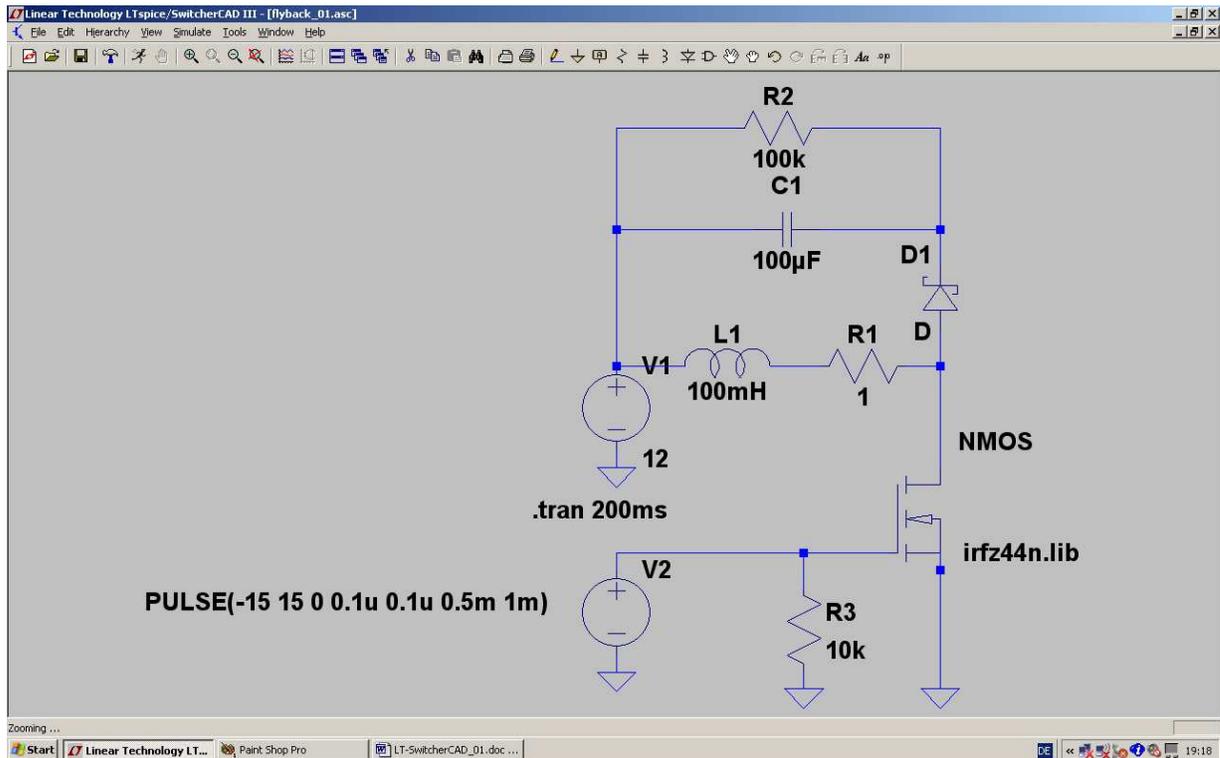


If you want: vary the gate voltage, the inductor's value, the output capacitor's value and simulate....

Note:

In practical circuits the stabilization of the output voltage is done by varying the on to off ratio of the driving squarewave voltage at the gate of the MOSFET.

11.3. The Flyback Converter



Note:

We have nearly the same schematic -- with the exception that the resistor-capacitor load is returned to the supply voltage and not to ground!

When the MOSFET is switched ON, then the inductor current rises again linearly as before. but when the MOSFET is switched OFF, then the current flows through the schottky diode into C1 and R2 to the power supply. This is driven by the stored magnetic energy in L1. This current flow stops when the complete magnetic energy stored in L1 is transferred to C1 and R2.

This circuit has some very interesting properties:

- You can short circuit this converter at the output terminal without damage. The energy transfer is like a bucket full of water: when filled (= energy stored in the inductor) it cannot transfer more but this portion of water (= energy) to the output. No damage will occur, neither to the MOSFET nor to the other parts of the converter.
- But be careful: if you run this circuit without the load resistor R2, the transferred energy will add and add and add to the capacitor C1 until a big bang happens! So don't forget this resistor and be sure that its value is always less than **100kΩ**.
- Note that the output voltage's polarization is now inverted to that of the step-up-converter.....and the output voltage's amplitude is normally much higher than the supply voltage!

Note:

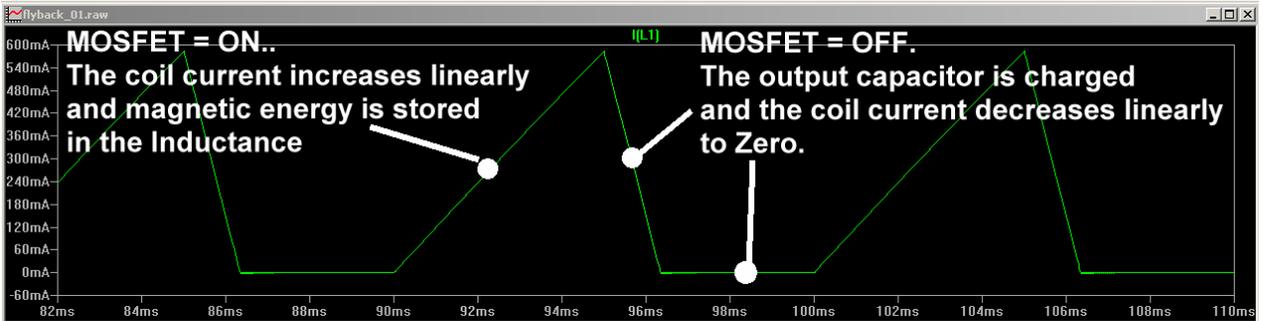
In practical circuits the stabilization of the output voltage is done by varying the on to off ratio of the driving squarewave voltage at the gate of the MOSFET.

Now draw this schematic and simulate. As gate input voltage again use the symmetric square wave signal with $f = 1\text{kHz}$ and $+15\text{V} / -15\text{V}$ as amplitude.

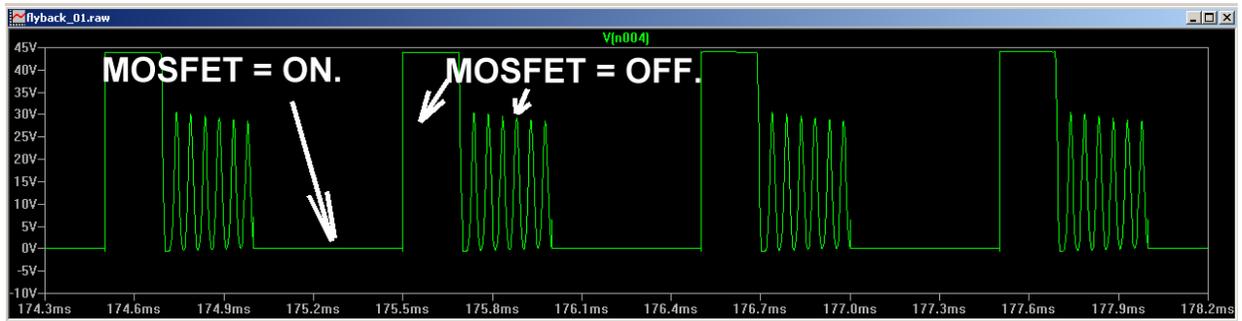
Please have a look at this special effect:

Because the schottky diode's ON-voltage is 0.4V , some energy remains in the inductor $L1$ when the diode switches OFF. This gives a resonant oscillation caused by the self capacitance of $L1$ and the drain capacitance of the MOSFET together with the inductance of $L1$.

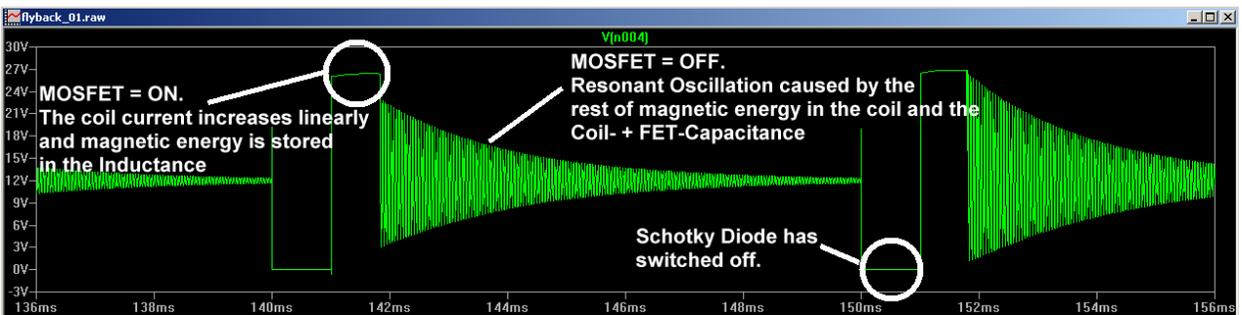
Let's at first look at the inductor current:



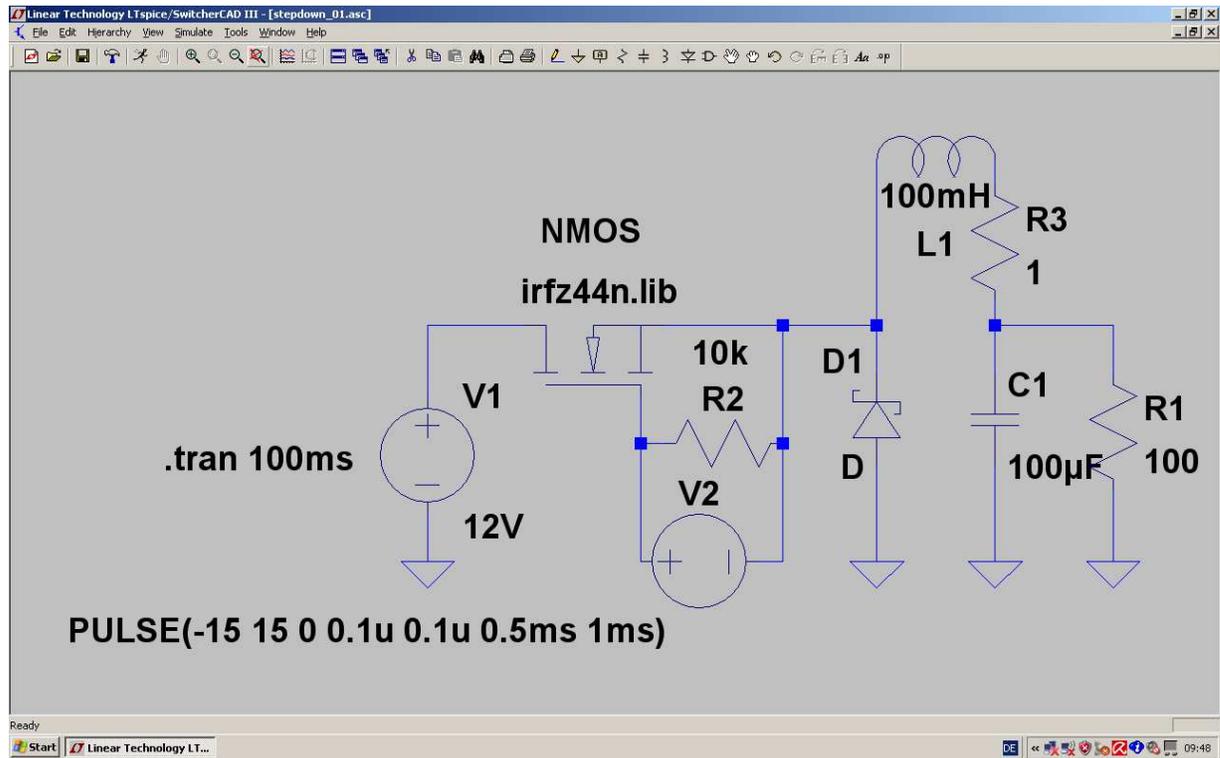
And this is the drain voltage of the MOSFET:



Additional information: this is the simulation with a short ON-Time (1ms) and a long OFF-Time (9ms) of the MOSFET:



11.4. The Step Down Converter



In this circuit the MOSFET serves as a switch between the power supply and the inductor and the schottky diode is connected between the node “MOSFET’s source / inductor” and ground (initially reverse biased).

Description:

- When the MOSFET is switched ON then the linear rising current flows from the power supply through the FET and the inductor to the output capacitor and the load resistor to ground. So the capacitor is charged and energy is stored in the magnetic field of the inductor.
- When the MOSFET is switched OFF the inductor current continues to flow in the same direction, but now the MOSFET is OFF and the schottky diode switches on to maintain current flow. The stored magnetic energy is transferred to the output circuit (= C1 and R1) until the inductor current has decreased to zero. During the time this current flows the voltage across the diode is **NEGATIVE with respect to ground!!!**

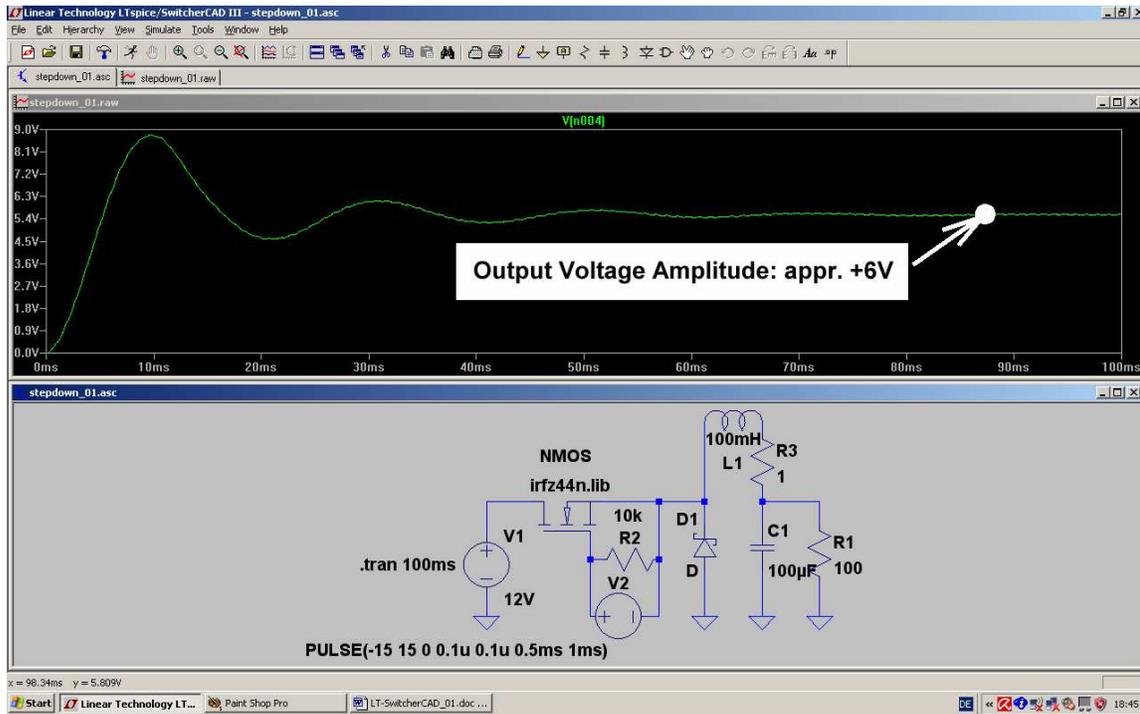
Note:

- In practical circuits the stabilization of the output voltage is always done by varying the on to off ratio of the driving squarewave voltage at the gate of the MOSFET.
- The amplitude of the output voltage is always less than the power supply voltage!

If you want to know the exact output amplitude value, multiply the supply voltage with the ratio „ON-time / period time“. So for our example we get with an ON-time of 0,5ms and a period time of 1ms

$$V_{out} = 0,5 \times 12V = 6V$$

Let's first simulate the output voltage:



And this is the diode voltage:

