

# SPICE-Schaltungs- Simulation mit LTspice IV



**Tutorial** zum erfolgreichen Umgang mit der frei im Internet ([www.linear.com](http://www.linear.com)) erhältlichen SPICE-Vollversion der Firma Linear Technologies

## Tutorial-Version 2.2

Copyright by  
Gunthard Kraus, Oberstudienrat i.R. an der Elektronikschule Tettnang  
Gastdozent an der Dualen Hochschule Baden Württemberg (DHBW) in Friedrichshafen

Email: [krausg@elektronikschule.de](mailto:krausg@elektronikschule.de) und [Gunthard.Kraus@gmx.de](mailto:Gunthard.Kraus@gmx.de)  
Homepage: [www.elektronikschule.de/~krausg](http://www.elektronikschule.de/~krausg)

20. April 2012

<b>Inhaltsverzeichnis</b>	<b>Seite</b>
<b>1. Einführung</b>	<b>5</b>
<b>2. Installation</b>	<b>6</b>
<b>3. Übungen am mitgelieferten Demo-Beispiel „Astable Multivibrator“</b>	<b>6</b>
3.1. Aufruf von Schaltplan und Simulation	6
3.2. Darstellung der Signalverläufe	8
3.3. Löschen von Signalverläufen im Ergebnis-Bildschirm	10
3.4. Andere Farbe für eine Ergebniskurve	10
3.5. Änderung der Simulationszeit	11
3.6. Änderung des dargestellten Spannungs- oder Strombereichs	13
3.7. Cursor-Einsatz	15
3.7.1. Verwendung eines Cursors	15
3.7.1. Verwendung eines zweiten Cursors	15
3.8. Differenzmessungen	16
3.9. Strom-Messungen	17
3.10. Änderung von Bauteilwerten	18
<b>4. RC-Tiefpass als erstes eigenes Projekt</b>	<b>19</b>
4.1. Zeichnen des Stromlaufplans mit dem Editor	19
4.2. Zuweisung neuer Bauteilwerte	20
4.3. Untersuchung von einmaligen Vorgängen	21
4.3.1. Die Sprungantwort	21
4.3.2. Ein- und Ausschaltvorgang	23
4.4. Periodische Signale am Eingang	24
4.4.1. Sinussignal mit $f = 1591$ Hz	24
4.4.2. Rechtecksignal mit $f = 1691$ Hz	25
4.4.3. Dreiecksignal mit $f = 1691$ Hz	26
4.5. AC-Sweep zur Ermittlung des Frequenzganges	27
<b>5. FFT (= Fast Fourier Transformation)</b>	<b>29</b>
<b>6. Zweites Projekt: Gleichrichtung</b>	<b>32</b>
6.1. Einpuls-Gleichrichter ohne Trafo	32
6.2. Eine wichtige Sache: ein Transformator für die Gleichrichterschaltungen	33
6.2.1. Die bequemste Lösung: simpler, idealer Trafo	33
6.2.2. Etwas für die Wirklichkeit: Erstellung eines realistischen SPICE-Modells für einen Transformator mit zwei Wicklungen	33
6.2.3. Erzeugung eines passenden Symbols für diesen Transformator	35
6.3. Einpuls-Gleichrichter mit realistischem Trafo	36
6.4. Verwendung der Diode 1N4007 in der Gleichrichterschaltung	38
6.5. Zweipuls-Gleichrichter mit realistischem Trafo	40
<b>7. Drittes Projekt: Drehstrom</b>	<b>42</b>
7.1. Programmierung eines Drehstromsystems	42
7.2. Drehstrom-Lichtmaschine im Auto	43
<b>8. Viertes Projekt: Darstellung von Bauteil-Kennlinien</b>	<b>45</b>
8.1. Ohm'scher Widerstand	45
8.2. Diode	46
8.3. NPN-Transistor	47
8.4. N-Kanal-Sperrschicht-FET	49

<b>9. Fünftes Projekt: Schaltungen mit Transistoren</b>	<b>50</b>
<b>9.1. Einstufiger Verstärker</b>	<b>50</b>
9.1.1. Ansteuerung mit einem Sinus-Signal	50
9.1.2. Simulation des Frequenzganges („AC-Sweep“)	52
<b>9.2. Zweistufiger gegengekoppelter Breitbandverstärker</b>	<b>53</b>
9.2.1. Pflichtenheft	53
9.2.2. Simulations-Schaltung und Simulations-Vorgaben	54
9.2.3. Simulation in der Time Domain (= im Zeitbereich)	54
9.2.4. DC-Bias (= Gleichstrom-Analyse)	55
9.2.5. AC-Sweep (= Frequenzgang von 1 Hz bis 200 MHz)	57
<b>9.3. Der Parameter-Sweep</b>	<b>58</b>
<b>10. Sechstes Projekt: OPV-Schaltungen</b>	<b>60</b>
<b>10.1. Einstieg: Umkehrender Verstärker</b>	<b>60</b>
10.1.1. Verwendung des einfachsten OPV-Modells „opamp“	60
10.1.2. Einsatz des Universalen OPVs und die Verwendung von Labels	61
10.1.3. Verwendung eines OPVs vom Typ „TL072“ und Erstellung eines passenden Spice-Symbols	63
<b>10.2. Nicht umkehrender Verstärker mit TL072</b>	<b>68</b>
<b>10.3. Analog-Addierer mit TL072</b>	<b>69</b>
<b>10.4. Probleme beim Betrieb mit nur einer Betriebsspannung</b>	<b>73</b>
<b>11. Siebtes Projekt: DC-DC-Konverter</b>	<b>71</b>
11.1. Bereitstellung des Power-MOSFETs „IRFZ44N“	71
11.2. Der Step-Up-Konverter (= Aufwärtswandler)	73
11.3. Der Flyback-Konverter (= Sperrwandler)	75
11.4. Der Step-Down-Konverter (= Abwärtswandler)	77
<b>12. Achtes Projekt: Phasenanschnitt-Steuerung mit Thyristor</b>	<b>79</b>
12.1. Das eingesetzte Thyristor-Modell	79
12.2. Schalten von Ohm'schen Lasten	80
12.3. Schalten von induktiven Lasten	81
12.4. Zündung des Thyristors über einen Gate-Transformator	81
<b>13. Neuntes Projekt: Echos auf Leitungen</b>	<b>83</b>
13.1. Leitungen -- nur zwei Drähte?	83
13.2. Echos auf Leitungen	85
13.3. Simulation des vorigen Rechenbeispiels mit LTSpice	87
13.4. Leerlauf oder Kurzschluss als Last am Kabelende	90
13.5. Verwendung von Kabel mit Verlusten (Beispiel: RG58 / 50Ω)	92
13.5.1. Wie simuliere ich RG58-Kabel?	92
13.5.2. Simulation der Kabeldämpfung bei 100MHz	93
13.5.3. Speisung der RG58-Leitung mit einer Pulsspannung	96
13.5.4. Ein Kurzschluss am Ende der RG58-Leitung	97
<b>14. Zehntes Projekt: S-Parameter</b>	<b>98</b>
14.1. Jetzt nochmals Echos, aber mit System	98
14.2. Praxisbeispiel: 110MHz – Tschebyschef – Tiefpassfilter (LPF)	101
<b>15. Elftes Projekt: Double Balanced Mixer (= Ringmodulator)</b>	<b>105</b>
15.1. Etwas Grundlagen	105
15.2. Standardschaltung des Ringmodulators	106
15.3. Ein Problem: die erforderlichen Übertrager	107
15.4. Simulation des DBM-Verhaltens mit idealen Übertragern	108

<b>16. Zwölftes Projekt: Digitale Schaltungssimulation</b>	<b>110</b>
16.1. Was man vorher wissen sollte	110
16.2. Einfacher Anfang: die Umkehrstufe (= NOT oder Inverter)	111
16.3. Der AND-Baustein	112
16.4. Das D-Flipflop	113
16.5. Dreistufiger Frequenzteiler mit D-Flipflops	114
<b>17. Dreizehntes Projekt: Rausch-Simulation</b>	<b>115</b>
17.1. Etwas Grundlagen	115
17.1.1. „Rauschen“ -- woher kommt das?	115
17.1.2. Weitere Rauschquellen	117
17.1.3. Rauschtemperatur und Noise Figure eines Twoports	118
17.2. Simulation der Spektralen Rauschleistungs-Dichte	118
17.3. Simulation der Noise Figure in dB	121
<b>18. Untersuchung eines Sinus-Oszillators</b>	<b>125</b>
<b>19. Signalformen und Oberwellen</b>	<b>130</b>
19.1. Grundlagen	130
19.2. Simulation des Spektrums eines einmaligen Pulses	133
19.3. Simulation des Frequenzspektrums bei einem periodischen Pulssignal	135
19.4. Der ideale Sinus	136
19.5. Der einseitig begrenzte Sinus	137
19.6. Der symmetrisch begrenzte Sinus	138
<b>20. Die Sache mit der Impulsantwort</b>	<b>139</b>
20.1. Erstes Beispiel: Dirac-Impuls speist einen 160Hz-RC-Tiefpass	144
20.2. Zweites Beispiel: Dirac-Untersuchung des 110MHz-Tiefpasses aus Kapitel 14.2	144
20.2.1. Bestimmung von S21 (= Forward Transmission)	144
20.2.2. Bestimmung von S11 (= Input Reflection) oder S22 (= Output Reflection)	149
<b>21. Modulation</b>	<b>151</b>
21.1. Prinzip der Amplitudenmodulation (AM)	151
21.2. Amplitude Shift Keying (ASK)	153
21.3. Die Frequenzmodulation (FM)	154
21.3.1. Erzeugung eines FM-Signals mit der Spannungsquelle „voltage“	154
21.3.2. FSK = Frequency Shift Keying	156
21.4. Quadratur-Amplitudenmodulation (QAM)	158
21.4.1. Grundlegende Informationen	158
21.4.2. Erzeugung eines QAM-Signals	159
21.4.3. Filterung des QAM-Signals zur Anpassung an die Übertragungs- Kanalbandbreite	166
21.4.4. Demodulation des QAM-Signals	170
21.4.4.1. Prinzip der Synchron-Demodulation	171
21.4.4.2. Die Simulation der vollständigen QAM-Demodulatorschaltung	172

## 1. Einführung

Die Schaltungssimulation ist aus der modernen Elektronik nicht mehr wegzudenken, denn nur mit ihr sind kürzeste und damit intensiv Kosten sparende Entwicklungszeiten beim Entwurf von Schaltungen oder Änderungen möglich. Außerdem kann jede neue Idee „ohne Lötcolben“ bis ins letzte Detail hinein ausgetestet werden, bevor überhaupt eine Leiterplatte gefertigt wird. Deshalb eröffnen sich nicht nur für die professionellen Entwickler, sondern auch für Hobbyelektroniker, Studenten, Schüler und Auszubildende völlig neue kreative Möglichkeiten.

Das sollte man aber gleich wissen: der Industriestandard für Schaltungssimulationsprogramme lautet

### **SPICE.**

Es wurde noch vor 1980 an der Berkeley-Universität entwickelt und um die eigentliche „SPICE-Machine“ herum (...so heißt das spezielle Simulationsprogramm) haben nun viele, viele Leute Bedienungsoberflächen erdacht und auf dem Markt tummeln sich entsprechend viele Anbieter. Dabei werden die Profi-Industrieprogramme meist mit einem Leiterplatten-CAD-Programm kombiniert -- da kann man hinterher gleich seine komplette Schaltung auf der entworfenen Platine austesten, wobei die Mischung von Analog- und Digitalschaltkreisen oder die Ermittlung der EMV-Störstrahlung oder das gegenseitige Übersprechen der Leiterbahnen selbstverständlich komplett simuliert wird. Diese Programme kosten aber kräftig Geld (...der größte und nobelste Lieferant ist hier ORCAD...) und deshalb gibt es auch einfachere Sachen.

Man findet aber auch sehr schöne „pädagogische“ Programme, die während der Ausbildung oder des Studiums eingesetzt werden können und sollen (Stichwort: „Workbench“) und mit tollen Details aufwarten (z.B. völlig realistische Darstellung und Bedienung der Messgeräte, wie Generatoren oder Oszilloskope oder Spectrum-Analyzer usw). Allen diesen Programmen gemeinsam ist aber ein Nachteil: es gibt zwar kostenlose Demo- und Studentenversionen, aber irgendwo steckt da immer ein Pferdefuß: entweder ist die Zahl der maximal verwendbaren Bauteile begrenzt oder die Schaltplangröße und Platinengröße wird fast bis auf Briefmarkenformat beschnitten oder der maximal belegbare Arbeitsspeicher darf einen Maximalwert nicht überschreiten oder ganz moderne Bauteile mit ihren sehr aufwendigen und umfangreichen Modellen werden gleich mal wegen ihrer Größe abgelehnt oder das Übernehmen eines neuen Bauteils in die Bibliothek gleicht vom Aufwand her einem Hausbau oder....oder....oder....

Deshalb ist es sehr erfreulich, dass man bei genauer Suche (...und nach entsprechenden Tipps von Gleichgesinnten!) auf Programme stößt, die

- a) leistungsfähig, gut bedienbar und dennoch kostenlos sind und
- b) trotzdem keine empfindlichen Beschränkungen bei der eigentlichen Schaltungssimulation aufweisen.

Der absolute Einsteigerhit ist hierbei „**5spice**“. Es ist wegen seines absichtlichen Minimalaufbaus mit Abstand am leichtesten erlernbar, aber sobald man mehr Komfort und weitere zusätzliche Funktionen sucht, stößt man auf

### **LTspice IV (vorher: LT SwitcherCAD)**

von Linear Technologies. Diese Software will nur Schaltungen simulieren und weist keinerlei Einschränkungen auf, da sie als freundliches Angebot der Firma an alle Entwicklungsingenieure gedacht ist (...natürlich zuerst zur Simulation von Schaltreglern und Schaltnetzteilen unter Verwendung von Halbleitern dieser Firma, aber das ist doch absolut in Ordnung...). Sie wird in der Zwischenzeit auch für sehr viele andere Zwecke eingesetzt und hat sich einen großen Freundeskreis erworben.

Etwas drollig ist die Bedienung -- sie besteht nämlich aus einer **kräftigen Mischung aus Mausclicks UND Tastenkommandos UND Kommandozeilen-Eingaben**... und manche Dinge muss man erst mal suchen und finden! Aber wenn man den Dreh heraus hat, läuft es prima. Allerdings wie in einer guten, langjährigen Ehe: Viele Eigenschaften liebt man, aber ebenso muss man vieles in Liebe ertragen...

**Zum Schluss noch eine ganz wichtige Vorbemerkung:**

**Diese unterschiedlichen Eingabemöglichkeiten bedeuten,**

**dass viele Funktionen des Programms (Beispiel: „Simuliere jetzt die Schaltung“) auch auf völlig verschiedene Weise zugänglich sind oder gestartet werden können!**

**Im Tutorial werden jedoch NICHT ALLE dieser unterschiedlichen Methoden einzeln besprochen. Hier muss der Anwender selbst prüfen, wozu etwas da ist bzw. welche Methode er in Zukunft verwenden will.**

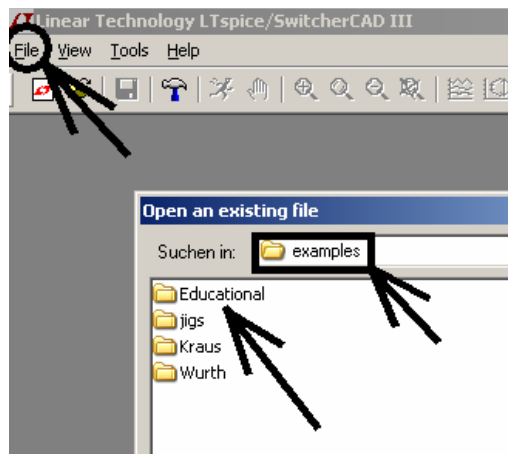
## 2. Installation auf dem PC

Nichts leichter als das: nach dem Download aus dem Internet ([www.linear.com](http://www.linear.com)) klickt man die „\*.exe-Datei“ an und wird in gewohnter Weise durch die Windows-Installation geführt. Der zugehörige Icon wird hierbei auf dem Windows-Startbildschirm automatisch angelegt. Fertig!

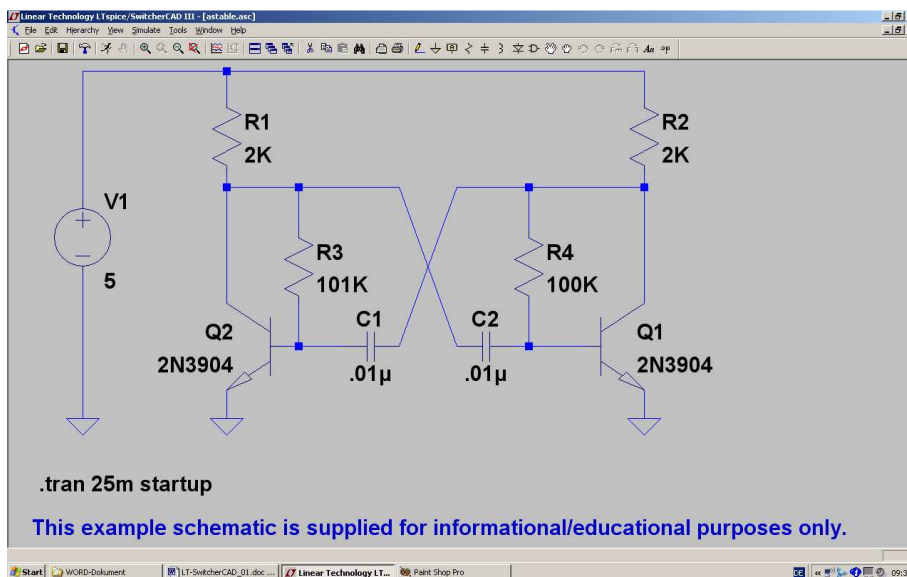
---

## 3. Übungen am mitgelieferten Demo-Beispiel „Astable Multivibrator“

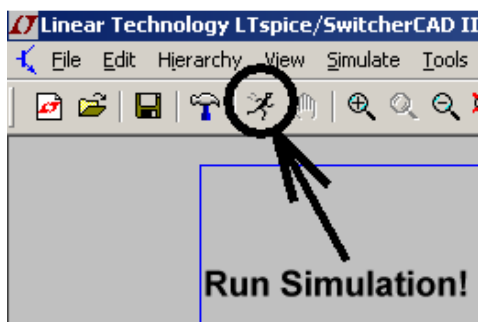
### 3.1. Aufruf von Schaltplan und Simulation



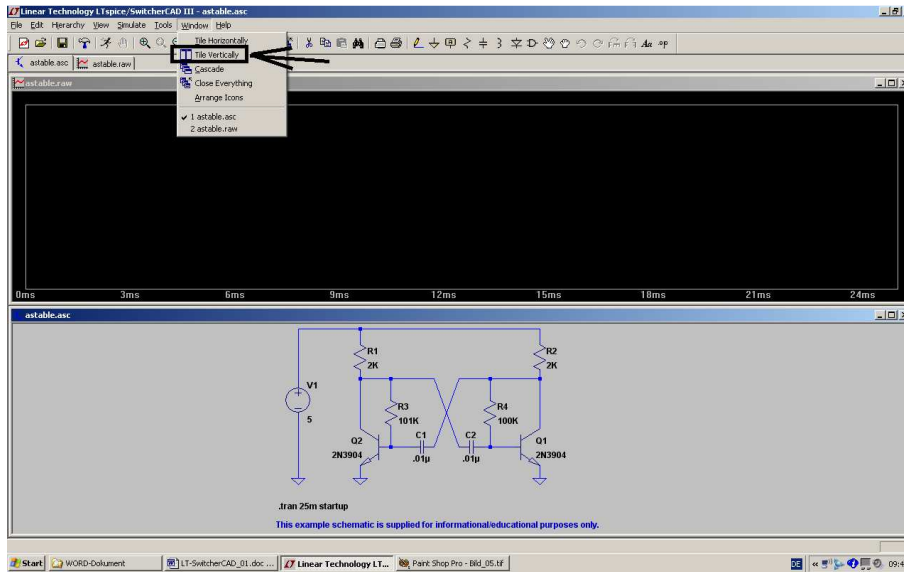
Nach dem Programmstart klicken wir im linken oberen Eck erst auf „File“ und dann „Open“. Im Verzeichnis unseres Programms gibt es dann mehrere Ordner. Wenn wir darin „Examples“ und anschließend „Educational“ öffnen, können wir jetzt einen **astabilen Multivibrator** auswählen (= „astable.asc“).



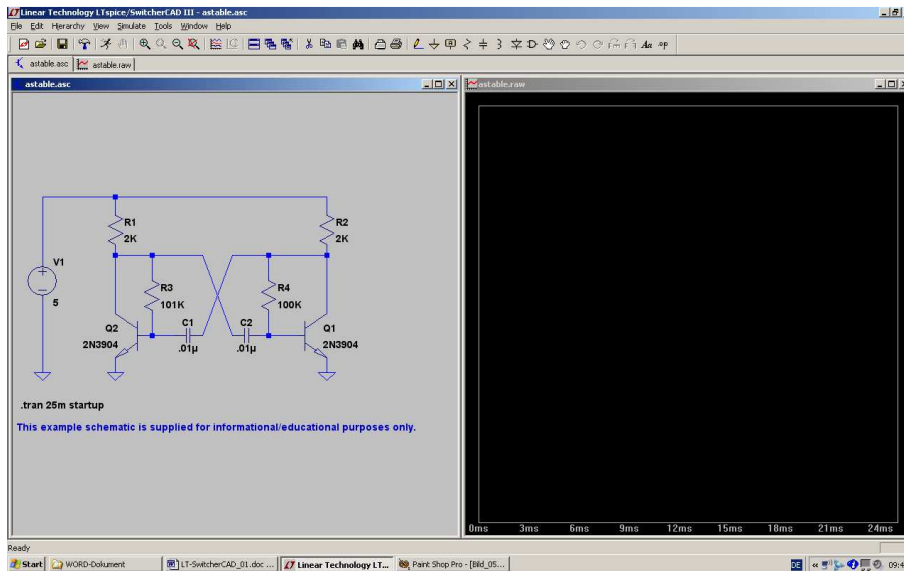
Haben wir alles richtig gemacht, dann empfängt uns der nebenstehende Bildschirm mit der fertigen Schaltung.



In der oberen Menüleiste finden wir (neben einem Hammer!) einen Button mit einem rennenden Männchen drauf. Das ist die **Starttaste für die Simulation** und sie wird nun gedrückt.



Das obere Diagramm ist für die Ergebnisdarstellung gedacht, aber noch leer. Dass die Schaltung zusätzlich in der unteren Bildhälfte präsentiert wird, hat seinen Grund...

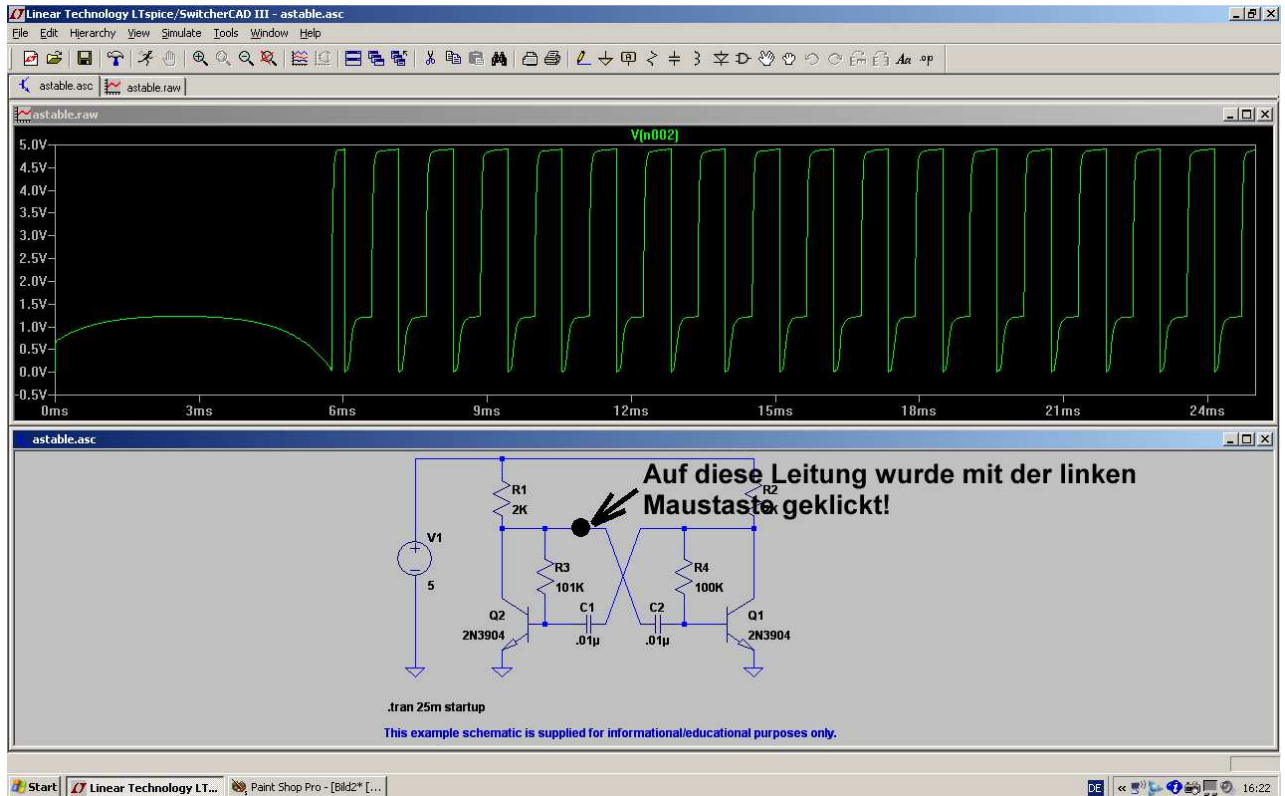


Übrigens: Wem diese Darstellung nicht behagt, weil er lieber links die Schaltung und rechts die Signalverläufe sehen möchte, der öffne das Menü „**Window**“ und rufe „**Tile vertically**“ auf. Dann sieht das Alles so aus:

**Und wer nach der Simulation die Ergebniskurven gern etwas dicker hat, der drücke jetzt schon auf den „Buttton mit dem Hammer“. Da gibt es eine Karteikarte „Waveforms“ und darauf „Plot data with thick lines“.**

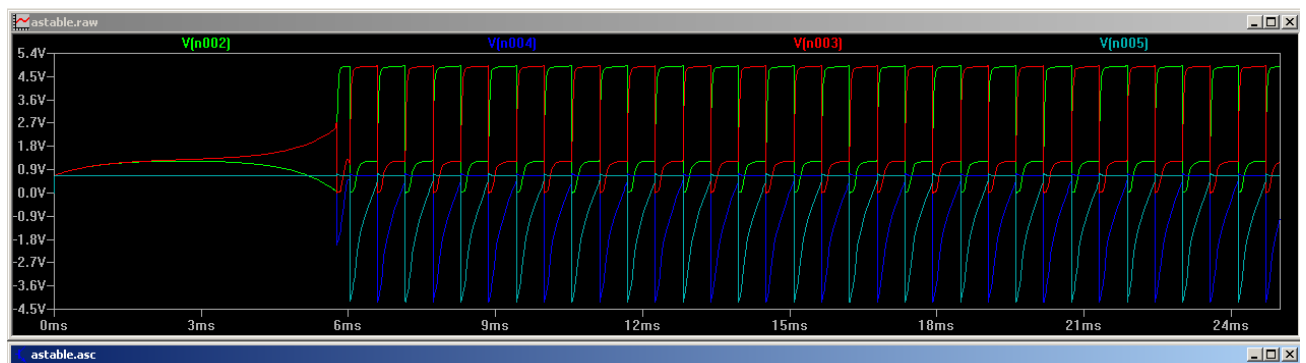
### 3.2. Darstellung der Signalverläufe

Will man den Verlauf der Spannung (gegen Masse) an einem bestimmten Punkt der Schaltung wissen, dann fährt man mit dem Mauscursor darauf. Sobald man genügend in die Nähe kommt, ändert der Cursor seine Form von einem einfachen Kreuz in eine Prüfspitze. Dann reicht ein simpler Mausklick, um die Darstellung zu aktivieren:



So kann man nacheinander alle gewünschten Spannungen sichtbar machen -- *in einem einzigen Diagramm. Aber leider ohne jede Spur innerhalb der Schaltung, wo gerade gemessen wurde! Lediglich die Kennzeichnung anhand der Schaltungsknoten und die unterschiedlichen Kurvenfarben helfen da bei der Identifizierung weiter.....*

Hat man beispielsweise alle Kollektor- und Basisspannungen in der Schaltung gemessen, dann sieht das Ergebnisfenster so aus:



Grundsätzlich kann man sich da auf **drei Arten** helfen:

a) Man informiert sich über die **Knotenbezeichnung innerhalb der Schaltung** (denn darnach werden die simulierten Spannungs- oder Stromverläufe durchnummeriert). Das Schlüsselwort dafür heißt „**Netzliste**“ (= **netlist**) und man findet diese Liste im Menü „**View**“ unter „**SPICE Netlist**“.



```

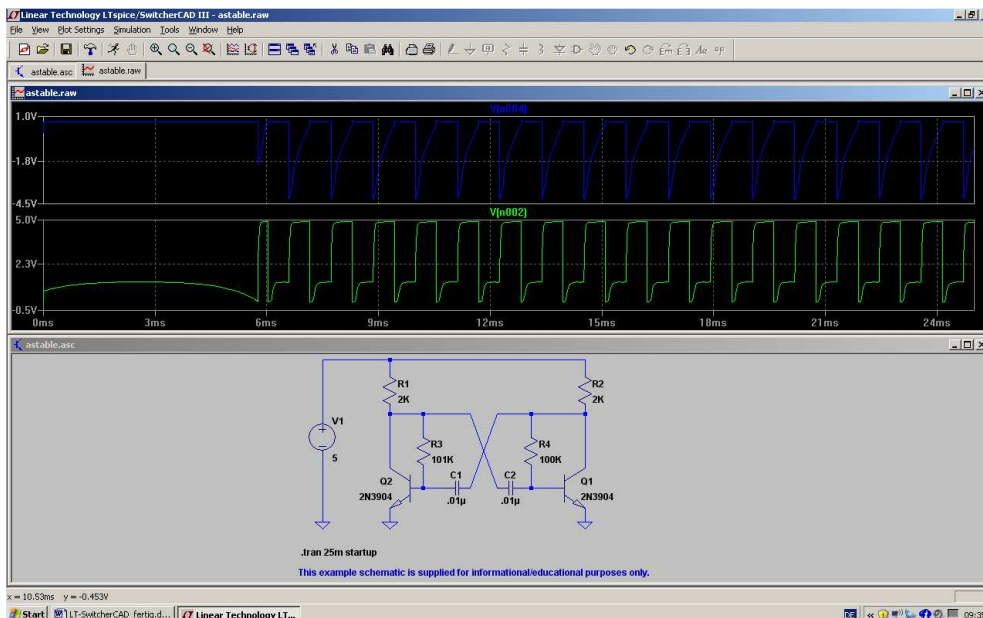
SPICE Netlist: C:\Programme\LTC\SwCADIII\examples\Educational\astable.net
* C:\Programme\LTC\SwCADIII\examples\Educational\astable.asc
R1 N001 N002 2K
R2 N001 N003 2K
R3 N002 N004 101K
R4 N003 N005 100K
C1 N003 N004 .01µ
C2 N005 N002 .01µ
V1 N001 0 5
Q1 N003 N005 0 0 2N3904
Q2 N002 N004 0 0 2N3904
.model 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

```

So sieht die Netzliste für unsere Schaltung aus und sie ist nicht schwer zu verstehen. Jede Zeile steht für ein Bauteil und beginnt mit dem Bauteilnamen. Dann folgen die Knoten, zwischen denen das Bauteil eingefügt ist und den Abschluss bildet der Bauteilwert.



b) Noch einfacher geht es, wenn man mit dem Cursor nochmals an die interessierende Leitung heranfährt. Sobald sich dann nämlich der Cursor in die Prüfspitze verwandelt, wird ganz links unten im Eck eine Meldung mit der Spannungsbezeichnung samt Knotennummer eingeblendet.....



c) Wenn zu viele Verläufe in einem Diagramm angezeigt werden, verliert man leicht die Übersicht. Da ist es besser, gleich mehrere getrennte Fenster zu benutzen.

Dazu klickt man mit der rechten Maustaste auf das Ergebnisfenster und wählt „Add Plot Pane“. Nun haben wir ein weiteres Ergebnisfenster zur Verfügung, in das man kurz hineinklickt

(dadurch wird es aktiviert). Nun reicht ein Klick im Schaltbild auf die gewünschte Leitung (z. B. den Basis-Anschluss des Transistors) zur getrennten Darstellung der Signalverläufe. Oder man klickt mit der rechten Maustaste auf das Diagramm, wählt „Add Trace“ und sucht sich die gewünschte Kurve aus. Diese Prozedur kann man natürlich wiederholen, aber ab 4 Diagrammen wird es schon wieder etwas unübersichtlicher....

**Wie überflüssige Kurven in einem Fenster wieder gelöscht werden können, steht im folgenden Kapitel!**

### 3.3. Löschen von Signalverläufen im Ergebnis-Bildschirm

Wenn z. B. nur ein einziges Fenster für die Ausgabe der Simulationsergebnisse gewählt wurde, sollte man **einzelne oder alle Kurven schnell löschen** können, wenn plötzlich noch ein anderer Punkt der Schaltung interessiert. Das geht so:

- 1) Wir klicken mit der linken Maustaste irgendwo einmal in das Ergebnisfenster, um auf diesen Bildschirmbereich umzuschalten. Der Erfolg kann leicht kontrolliert werden, denn der Rahmen des Fenster färbt sich dann dunkelblau.
- 2) Dann drücken wir die Taste **F5**. Sofort ändert sich der Cursor und wird zu einer **Schere**!
- 3) Diese Schere wird nun auf die Bezeichnung des zu löschenden Kurvenverlaufs am oberen Diagrammrand -- z. B. auf „V(n001)“ -- gestellt und anschließend einmal links mit der Maus geklickt. Damit ist die betreffende Kurve gelöscht und aus dem Diagramm verschwunden.

#### **Wichtig:**

Um jetzt wieder neu eine Spannung messen zu können, muss man zuerst durch **Druck auf die „Escape“-Taste die LösCHFunktion ausschalten** und dann zum **Fenster mit der Schaltung zurückkehren!**

Das geht natürlich wieder mit einem linken Mausklick irgendwo auf die Schaltung. Bitte beobachten, ob sich nun der Rand dieses Fensters dabei wirklich dunkelblau färbt! Wenn ja, dann kann in bekannter Weise erneut mit der Maus an den gewünschten Punkt herangefahren werden, bis der Cursor zur Tastspitze wird. Ein linker Mausklick genügt dann.

### 3.4. Andere Farbe für eine Ergebniskurve

Oft ist die Farbe einer Kurve vor dem dunklen Hintergrund schlecht sichtbar, aber die Umstellung ist eine Kleinigkeit:

**Erst auf die Signalbezeichnung einen rechten Mausklick....**

**....und dann die gewünschte Farbe auswählen!**

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

Default color: [Blue] Watched Cursor:

Expression algebra: [default] Plot:

(n005)

Delete this Trace

6ms 9ms 12ms

R1 2K R2 2K

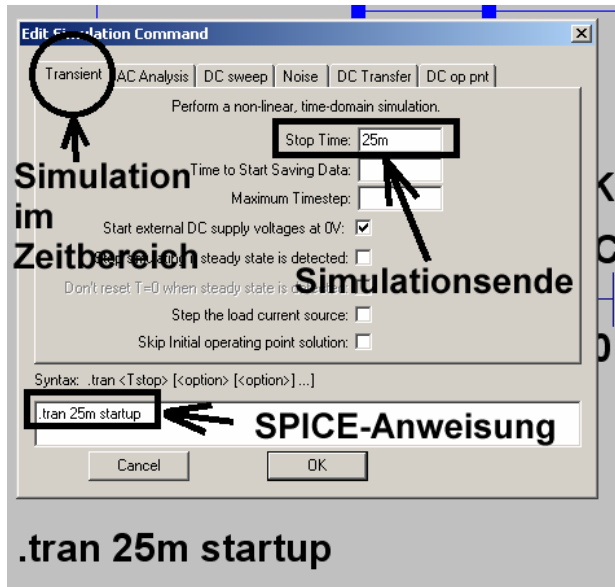
### 3.5. Änderung der Simulationszeit



In dieser Anweisungszeile versteckt sich der SPICE-Simulationsauftrag! Er sagt:

**Simuliere die Verläufe aller Signale in der Zeit von Null bis 25 Millisekunden.**

(Der Simulationsstart ist **IMMER** bei Null)

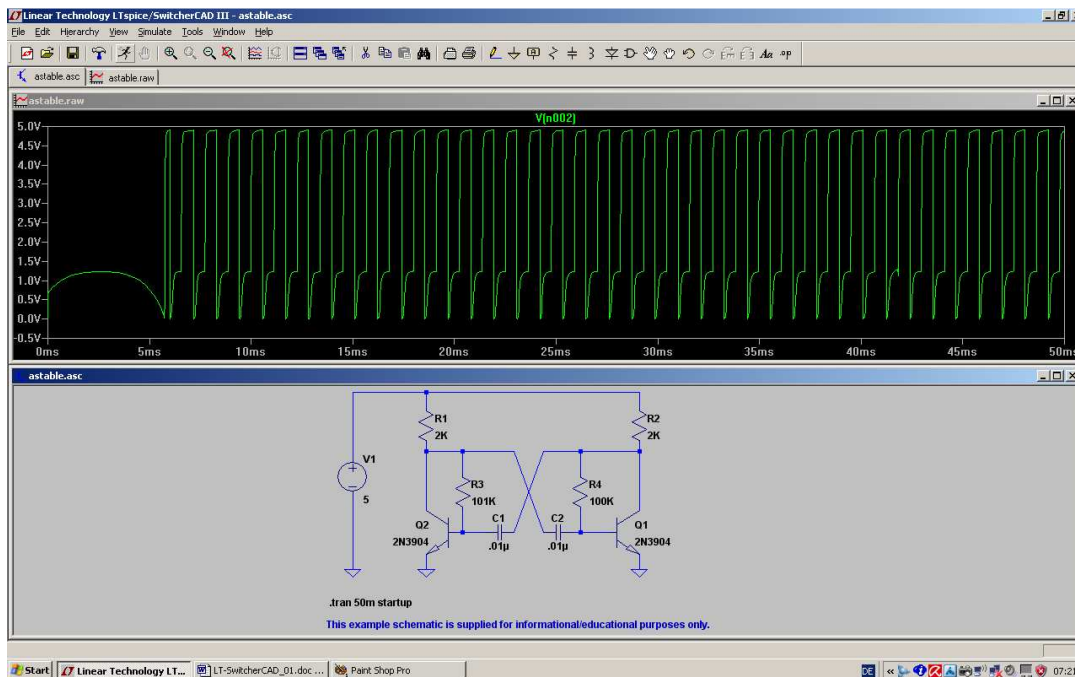


Stellt man nun den Mauszeiger mitten in den Text dieser Zeile und drückt **die rechte Maustaste**, dann erscheint dieses zugehörige Menü.

Wollen wir also kürzer oder länger simulieren lassen, dann tragen wir im Kästchen „Simulationssende“ die neue Zeit ein (...sie wird automatisch in die Anweisung übernommen..), drücken auf OK und anschließend auf den „**RUN**“-Button mit dem **laufenden Männchen**.

#### Aufgabe:

Bitte mal mit einer Stoppzeit von 50 Millisekunden simulieren und nur die Spannung am linken Kollektor (= V(n002) bei Transistor Q2) darstellen!



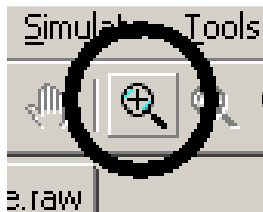
So sollte das dann aussehen!

Wie man sieht, wird bei diesem Programm sehr viel Wert auf die Analyse des Verhaltens der Schaltung nach dem Start gelegt. Das ist kein Wunder, denn die Firma hat es ja zur Analyse von Schaltnetzteilen entwickelt und da ist diese Information außerordentlich wichtig.

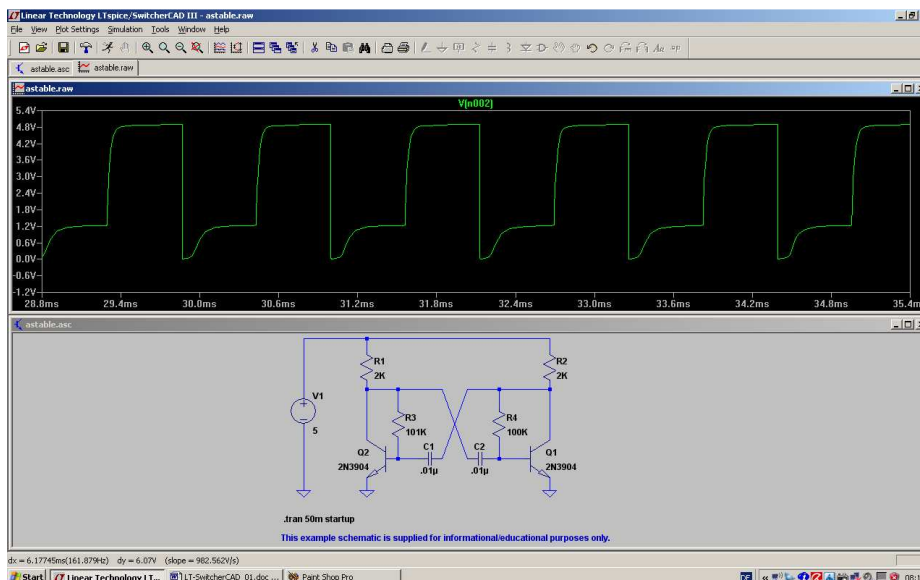
Will man jedoch nur die Verläufe im „eingeschwungenen Zustand“ sehen, dann greift man z. B. zur Zoom-Funktion:

**Aufgabe:**  
**Stellen Sie nur den Spannungsverlauf von ca. 30 bis 35 Millisekunden dar.**

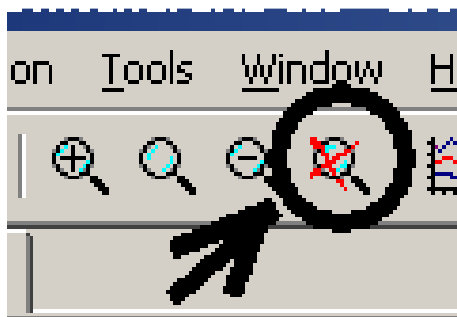
Lösung:



a) Wir klicken auf den „Vergrößerungs-Button und ziehen anschließend bei gedrückter linker Maustaste ein Rechteck um den Spannungsverlauf von 30.....35 Millisekunden.



b) Nach dem Loslassen der linken Maustaste muss der gewünschte Ausschnitt erscheinen.



c) Will man wieder zur ursprünglichen Voll-Darstellung zurückkehren, dann klickt man einfach auf die „Gesamt-Taste“

### 3.6. Änderung des dargestellten Spannungs- oder Strombereichs

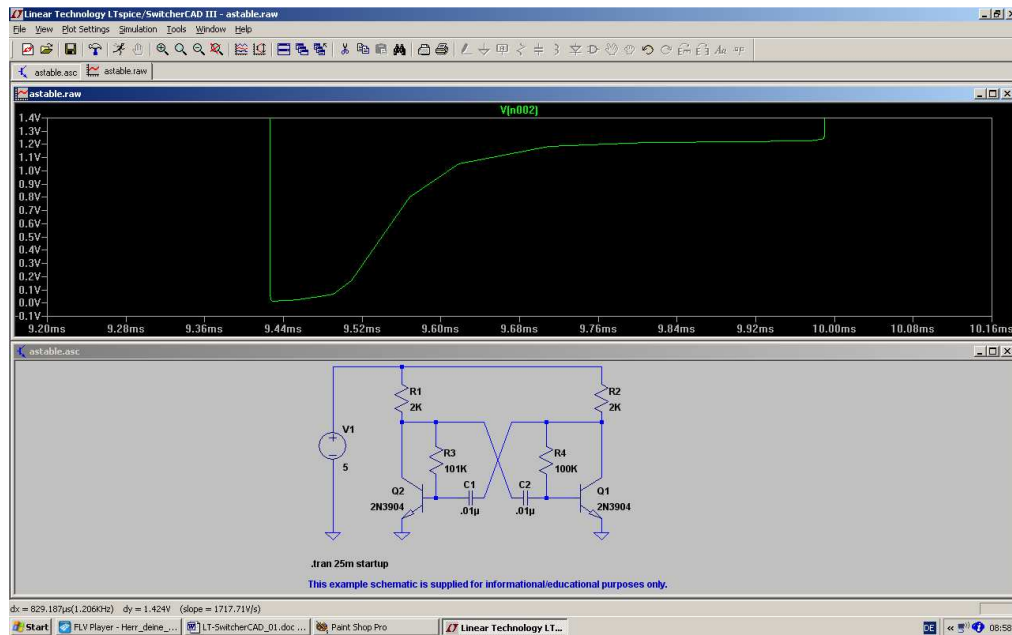
Oft interessieren bei einer Schaltungsentwicklung nur bestimmte Details bei den Signalverläufen und man möchte deshalb den angezeigten Wertebereich eingrenzen. Da gibt es zwei Möglichkeiten:

- a) Man arbeitet mit der eben besprochenen **Zoom-Funktion** oder
- b) Man schaltet die „**automatische Bereichswahl**“ (= **Autoranging**) **aus** und **gibt den angezeigten Wertebereich selbst vor**.

Zu a):

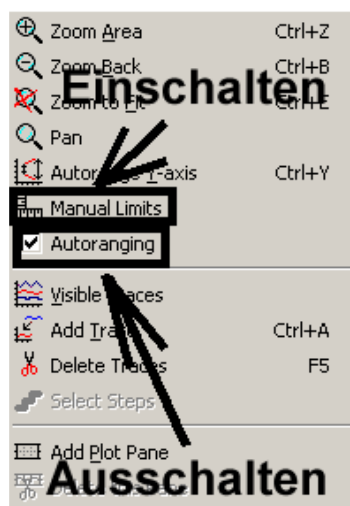
**Aufgabe:** Bei der Kollektorspannung des linken Transistors unseres astabilen Multivibrators soll die kurze negative Spitze der Spannung genauer untersucht werden. Zoomen Sie diesen Bereich heraus.

Lösung:



Zu b):

**Aufgabe:** Sorgen Sie dafür, dass bei der senkrechten Diagrammachse nur der Bereich von 0...+1,5V dargestellt wird.



#### 1. Schritt:

Man klickt mit der **RECHTEN Maustaste** auf die Werteskala der senkrechten Achse. Dadurch öffnet sich das zugehörige Menü.

#### Hinweis:

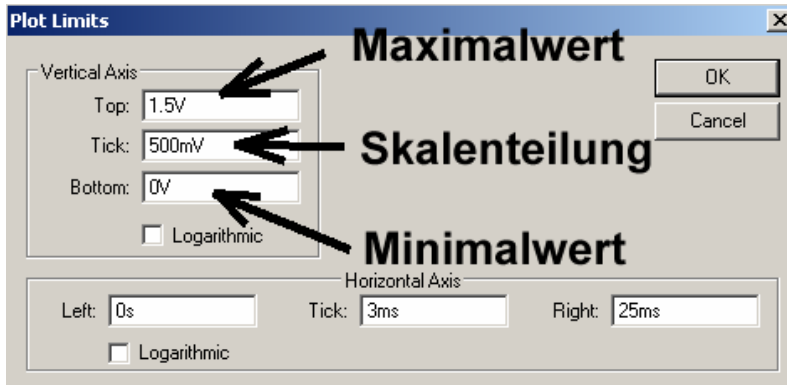
wer einfach nur die Maus auf die linke oder untere Werteskala rollt, bekommt nun **ein Lineal** als Cursor angeboten. Mit einem linken Klick öffnet sich dann

### ein KLEINES Menü

für Änderungen an der Achsendarstellung.

Bitte mal ansehen, aber wir wollen uns doch gleich mit der „Großen Lösung“ beschäftigen und die geht so:

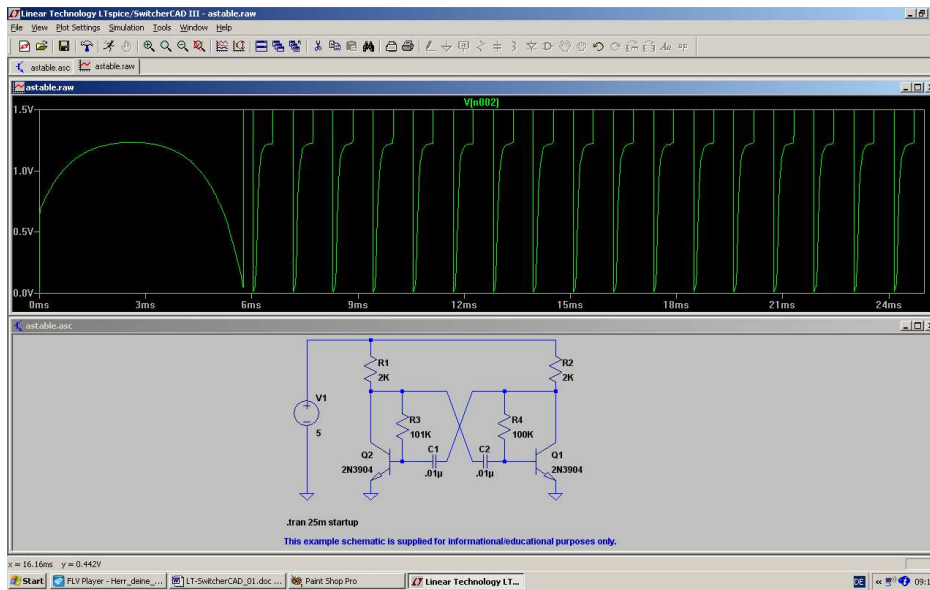
Im auftauchenden Menü (Siehe nebenstehendes Bild) wird „**Autoranging**“ aus- und „**Manual Limits**“ eingeschaltet.



**2. Schritt:**

Darin werden nun die neuen Grenzen für die senkrechte Achse eingetragen (...“Tick“ steht hierbei für „Skalenteilung“).

**Hier können wir auch die Einteilung für die waagrechte Achse ändern und einen anderen Zeitabschnitt einstellen!**

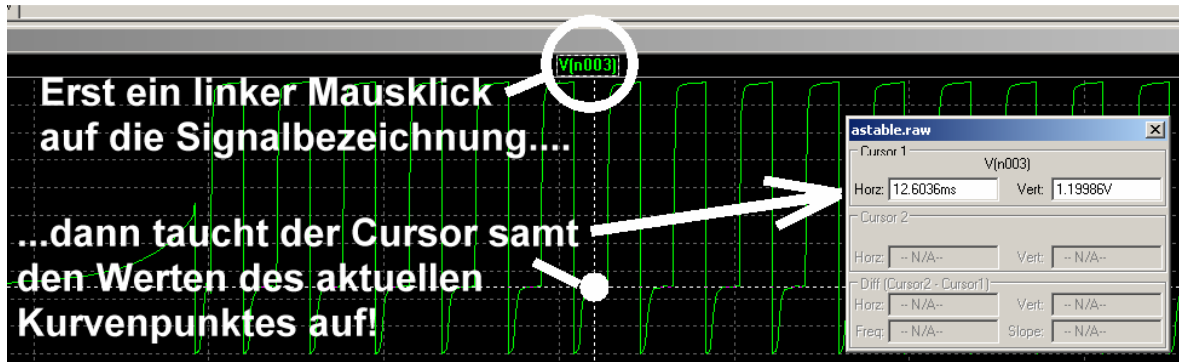


Damit sind wir am Ziel und können wieder zum Fenster mit der Schaltung zurückkehren, um Schaltungsänderungen vorzunehmen.

### 3.7. Cursor-Einsatz

#### 3.7.1. Verwendung eines Cursors

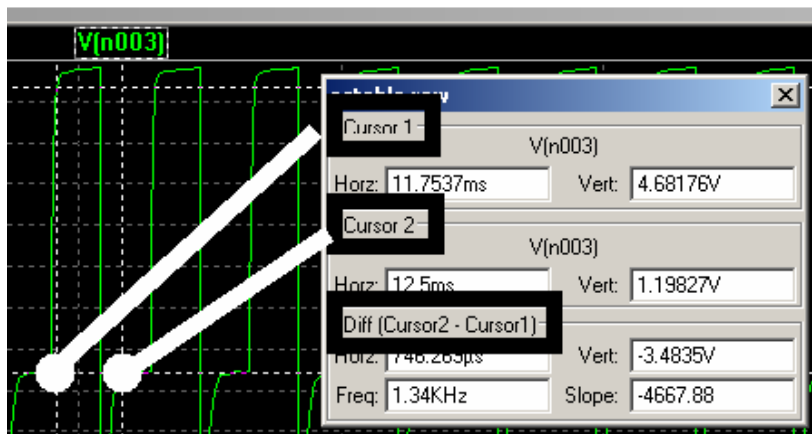
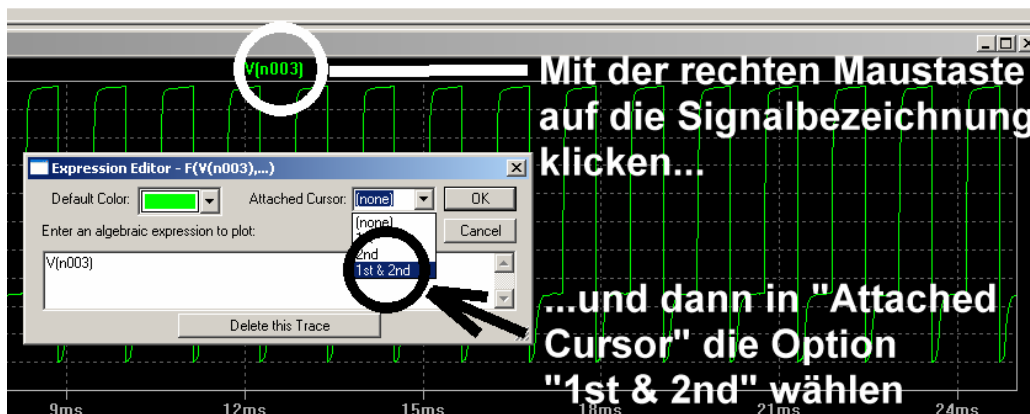
Oft möchte man eine Ergebniskurve „abfahren“, um an bestimmten Zeitpunkten den dort gültigen Spannungswert oder Stromwert zu ermitteln. Das geht so:



Fährt man nun die Maus genau auf das „Fadenkreuz“, dann wird nicht nur die Cursor-Nummer (hier: „1“) eingeblendet. Zusätzlich lässt sich nun das Fadenkreuz durch „Ziehen bei gedrückter Maustaste = DRAG“ entlang der Kurve verschieben. Damit kann an jeder Stelle der Kurve die Zeit sowie der genaue Amplitudenwert aus dem eingeblendeten Fenster entnommen werden.

#### 3.7.2. Verwendung eines zweiten Cursors

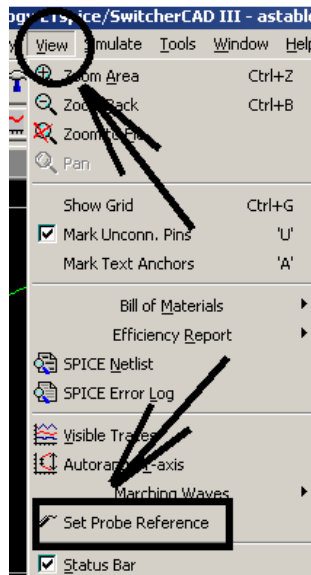
Dazu klickt man mit der **RECHTEN** Maustaste auf die Signalbezeichnung und geht im auftauchenden Fenster in das Menu „Attached Cursor“



Die beiden Cursors lassen sich nun getrennt verschieben. Ihre aktuellen Werte sowie die Differenz werden wieder in einem Fenster eingeblendet.

### 3.8. Differenzmessungen

Grundsätzlich werden alle Spannungen gegen den Referenzpunkt „Masse“ gemessen und dargestellt. Da kämpft man dann mit einem altbekanntem Problem: wo steckt das „Differential-Voltmeter“, mit dem die Spannung direkt zwischen zwei Bauteil-Anschlüssen ermittelt werden kann?

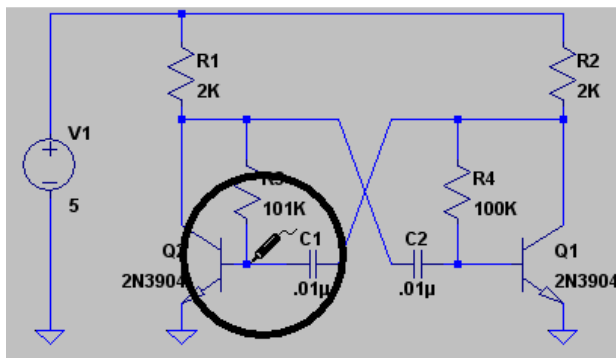


Nun, beim LT-SwitcherCAD geht das etwas anders: man muss zuerst bewusst einen neuen Referenzpunkt in die Schaltung setzen und dann am anderen Bauteilanschluss klicken!

Das geht so:

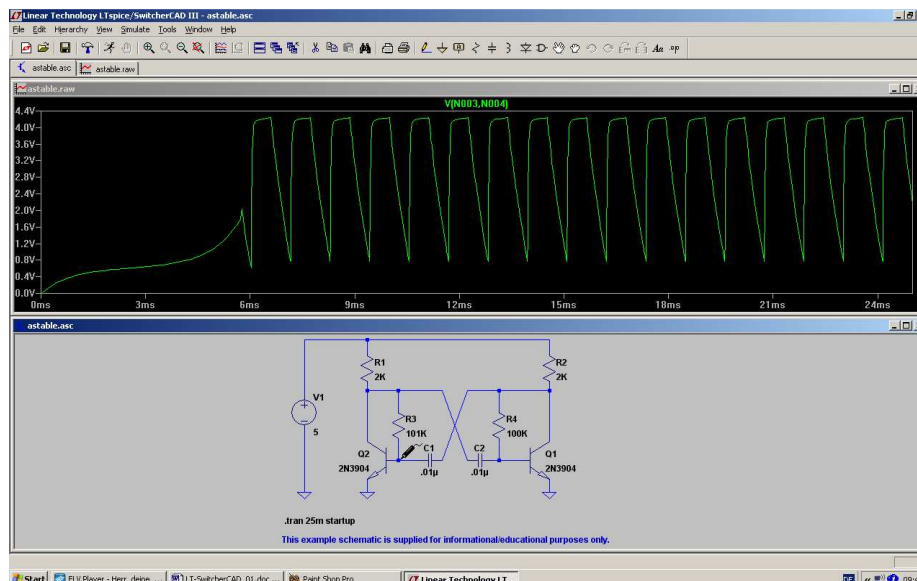
Man öffnet das Menü „View“ und findet darin die Funktion „Set Probe Reference“, die angeklickt wird.

Nehmen wir nun einmal an, dass uns die Spannung am Kondensator C1 (= an der Basis des linken Transistors) interessiert.



Nach „OK“ hängt plötzlich eine Prüfspitze am Cursor, die man direkt am Basisanschluss des linken Transistors durch Anklicken dieses Punktes absetzt.

Fährt man nun den Cursor auf den zweiten Anschluss des Kondensators (= Kollektor des rechten Transistors) und klickt auf die Leitung, dann ist man bereits am Ziel:



Falls gewünscht, kann man sich nun wieder interessierende Details herauszoomen.

Übrigens: Den Reference-Point bringt man durch einen Druck auf die Escape-Taste zum Verschwinden, wenn man wieder „normal“ (= gegen Masse) messen möchte.

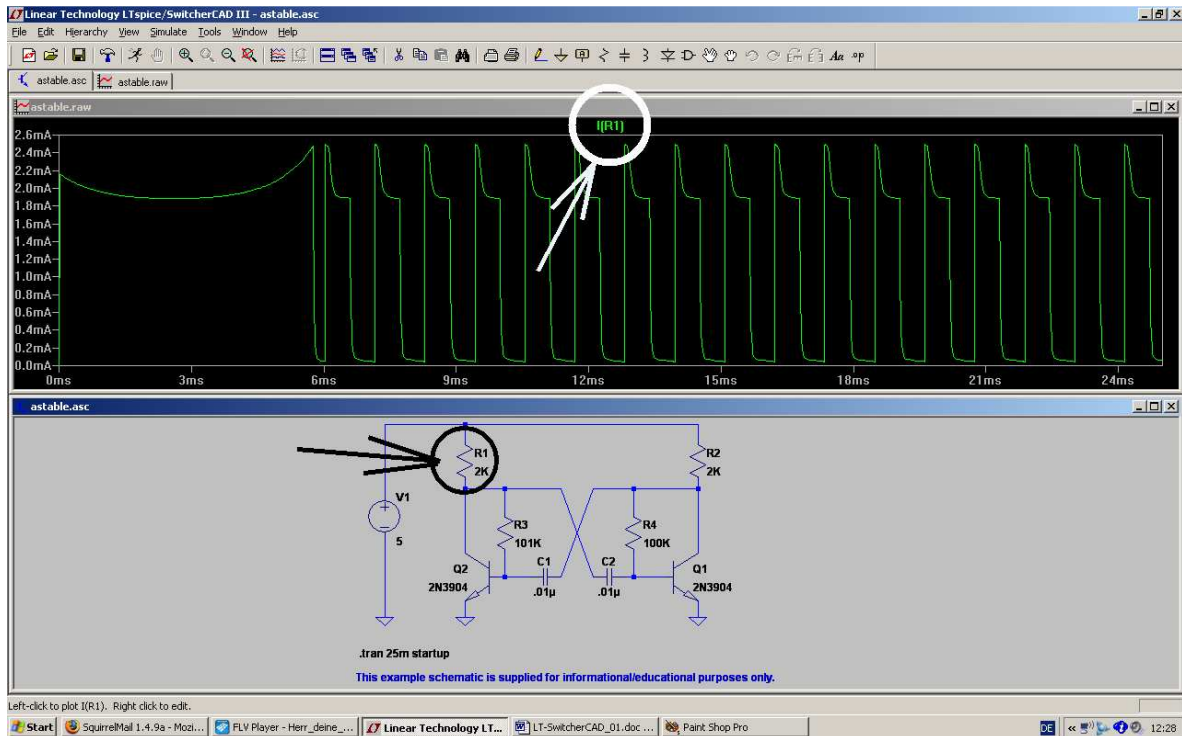


### 3.9. Strom-Messungen

Dazu fährt (bei aktiviertem Schaltplanfenster) den Cursor **exakt auf das Schaltzeichen des Bauteils**, dessen Strom man simuliert haben möchte. Der Cursor nimmt dann plötzlich **das Aussehen einer Stromzange** an und ein linker Mausklick reicht, um die Anzeige des Stromverlaufs auszulösen.

#### Beispiel:

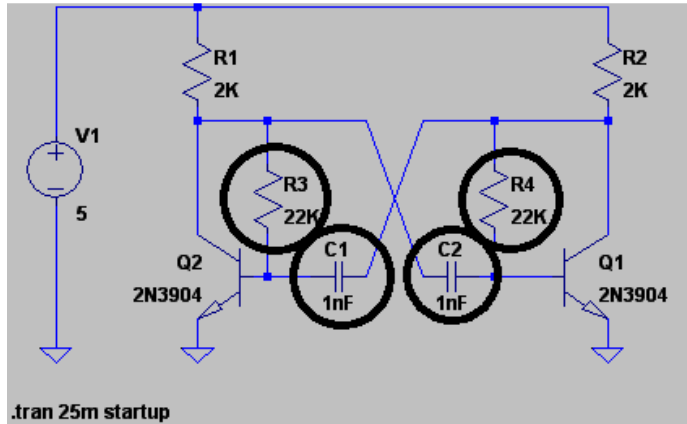
Stromverlauf im Widerstand R1 (= Kollektorwiderstand des linken Transistors):



### 3.10. Änderung von Bauteilwerten

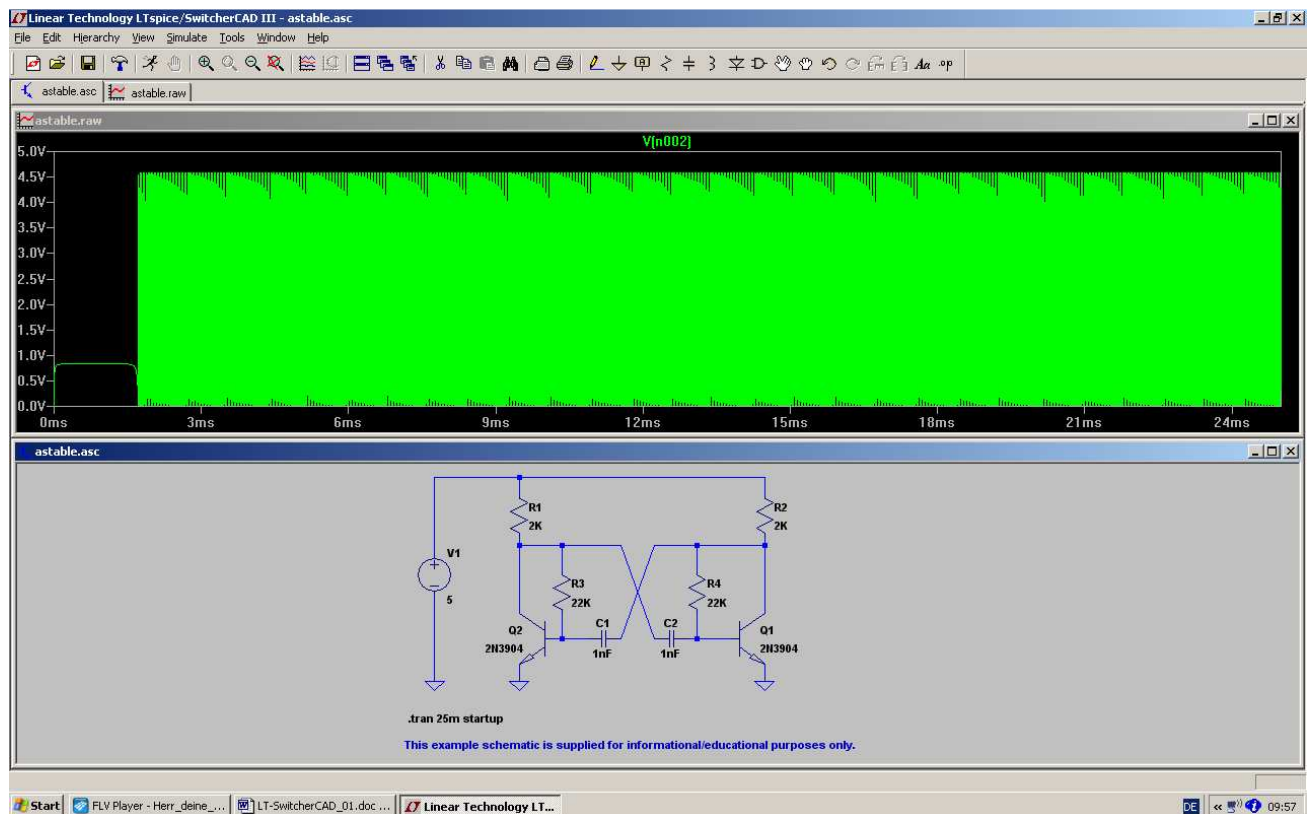
Nun wollen wir die Simulation mit anderen Bauteilwerten wiederholen und ändern beide Kondensatoren auf je 1000pF sowie die beiden Basiswiderstände auf je 22k $\Omega$  ab.

Dazu stellt man bei jedem Bauteil **den Cursor exakt auf die „WERTANGABE“** und drückt dann **die rechte Maustaste**. Im auftauchenden „Property Menu“ trägt man dann den korrigierten Wert ein und bestätigt mit OK.



So muss die Schaltung schließlich aussehen:

Jetzt wird die Simulation wiederholt (= Klick auf das rennende Männchen...) und schon bekommen wir folgendes Bild für die Kollektorspannung des linken Transistors:

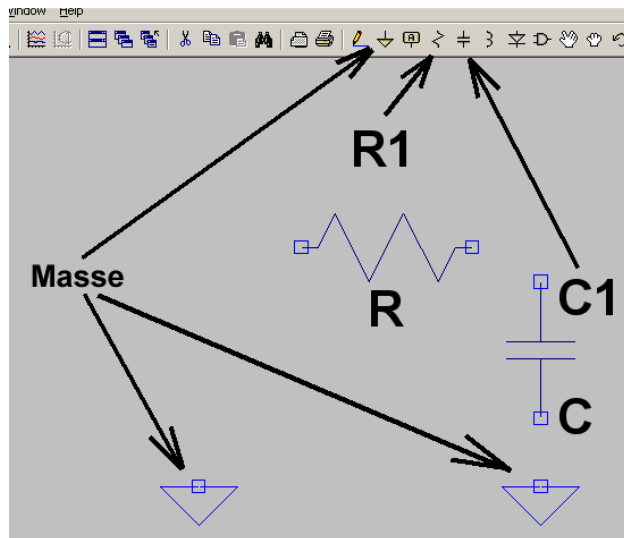


Da muss man nun natürlich mit der Zoom-Funktion dringehen oder gleich die Simulationszeit ändern, wenn man Details sehen oder die neue Frequenz messen möchte.

## 4. RC-Tiefpass als erstes eigenes Projekt

### 4.1. Zeichnen des Stromlaufplans mit dem Editor

Wir holen uns zuerst unter „File“ ein neues Schematic-Blatt und speichern es in einem **eigenen neuen Ordner** (z. B. „Training“) unter einem **eigenen Namen** (z. B. „RC-LPF\_01“).



Dann drücken wir den „**Button mit dem Widerstand**“, holen uns dort unser Widerständchen ab (...es hängt jetzt am Cursor...) **drehen** es mit dem Tastenkommando

**<CTRL> + <R>**

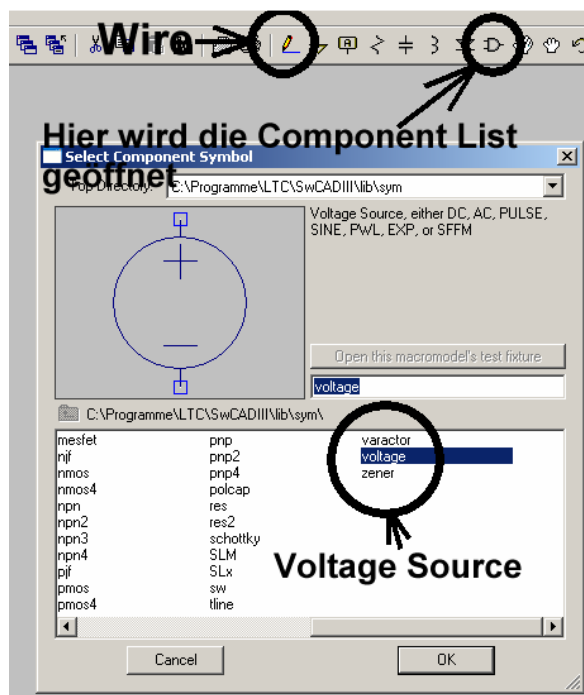
um 90 Grad und setzen es in der Schaltung ab. Anschließend klicken wir mit der **rechten Maustaste** kurz auf den Bildschirm (...oder drücken „ESCAPE“...), denn erst dadurch verschwindet das Widerstandssymbol am Cursor wieder.

Jetzt wiederholen wir die Prozedur mit dem **Kondensator** sowie **zwei Massezeichen** und haben dann den nebenstehenden Stand:

Dazu zwei Tipps:

- Möchte man ein Bauteil (oder später eine Verbindungsleitung) wieder **löschen**, dann drückt man erst auf die **Taste F5**. Sofort erscheint die **schon bekannte Schere als Cursor!** Führt man damit direkt auf das Symbol und klickt links, dann verschwindet das betreffende Teil wieder vom Schirm.
- Soll ein Bauteil dagegen auf dem Bildschirm **verschoben** werden (= MOVE), dann ist ein Druck auf die **Taste F7** erforderlich. Der Cursor wird zu einer **Hand**, die man auf das zu verschiebende Bauteil stellt. Klickt man dann mit der linken Maustaste, so ändert das Teil seine Farbe und hängt plötzlich am Cursor! So kann es nun beliebig mit der Maus verschoben werden und ein neuer linker Klick setzt es an der neuen Position ab.

**Bitte dran denken: aus solchen Sonderfunktionen kommt man immer mit der „ESCAPE“-Taste wieder zurück zum Hauptmenü!**



Jetzt fehlt uns noch die Spannungsquelle und die ist leider in der Menüleiste nicht aufgeführt. Da hilft nur die „Bauteiltaste“ in der Menüleiste.

In der auftauchenden Component-List (...darin ist alles aufgeführt, womit man arbeiten kann...) gehen wir zu „**voltage**“ und bestätigen mit OK. Dann hängt die Quelle am Cursor und kann in die Schaltung eingesetzt werden.

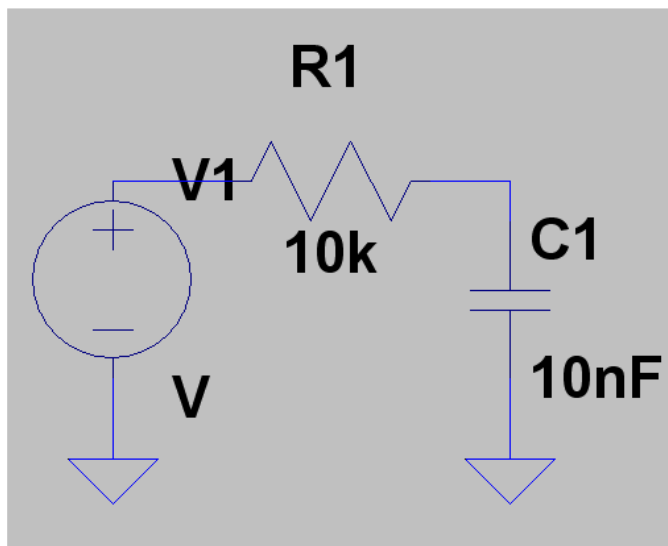
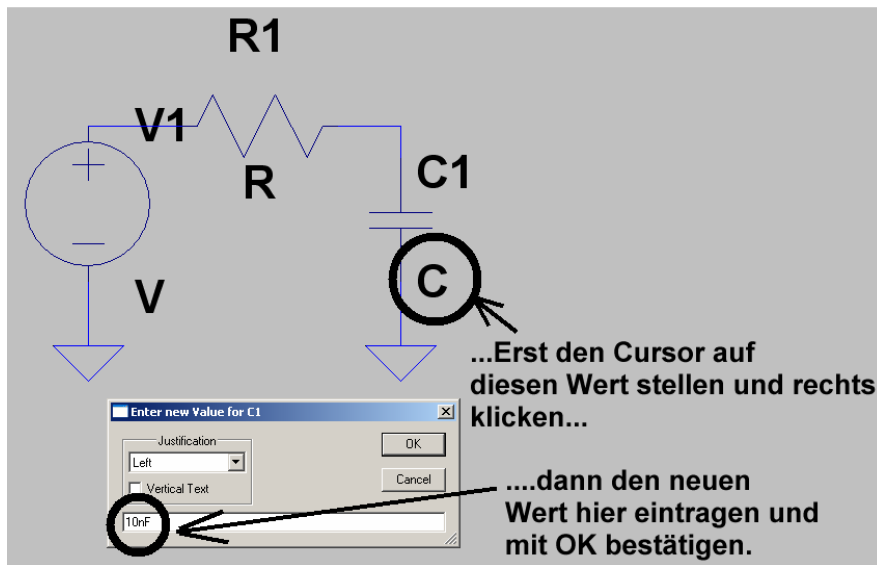
Jetzt wollen wir verdrahten und benötigen dafür die **WIRE-Funktion**. Bitte auf die Taste mit dem Bleistift klicken und genau hinschauen: auf dem Bildschirm muss nun ein Fadenkreuz zu sehen sein!

Klickt man nun auf einen Bauteilanschluss und rollt dann die Maus, dann spult sich eine Verbindungsdrabt ab. Man zieht ihn bis zum nächsten Bauteilanschluss und klickt erneut. Dann sollte die Verbindung hergestellt sein.

**Wichtig:**

Für eine „**abgeknickte Verbindung**“ muss man erst an der **Knickstelle einmal klicken**. Nur dann lässt sich der Rest der Leitung in einer anderen Richtung verlegen.

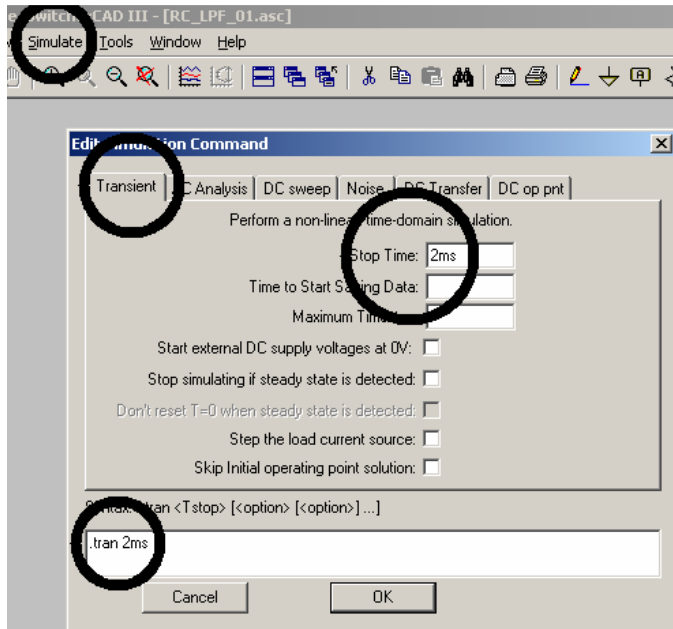
## 4.2. Zuweisung neuer Bauteilwerte



### 4.3. Untersuchung von einmaligen Vorgängen

#### 4.3.1. Die Sprungantwort

Dazu programmieren wir zuerst eine „Simulation im Zeitbereich“, also in der **Time Domain (= Transient Simulation)**. Außerdem lassen wir die Eingangsspannung in sehr kurzer Zeit von 0V auf +1V springen.

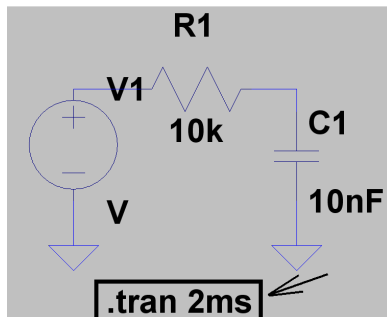


Hinter dem **Simulate**-Menü versteckt sich die Zeile „**Edit Simulation Command**“.

Wenn wir darauf klicken, öffnet sich das nebenstehende Transient-Menü. Darin tippen wir eine Stop Time von 2 Millisekunden ein und prüfen, ob das korrekt in das Anweisungsfenster als

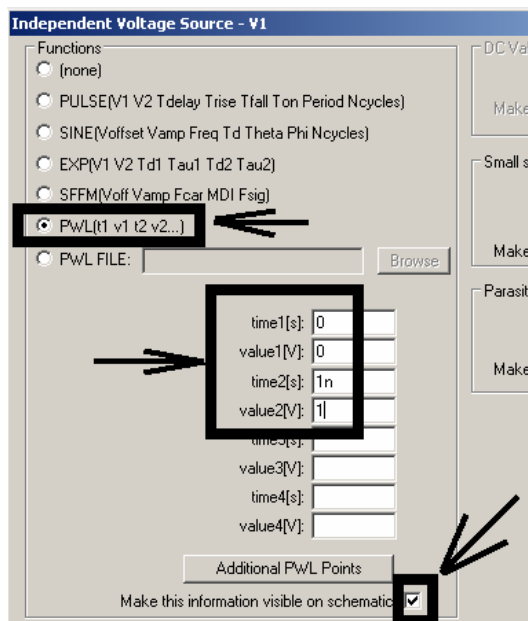
**.tran 2ms**

übernommen wurde. Bitte schließlich mit OK bestätigen.



Dieser Eintrag „**.tran 2ms**“ hängt am Cursor und muss sich nun im Schaltplan absetzen lassen..

Jetzt fehlt nur noch die passende Eingangsspannung, nämlich ein idealer Sprung von Null Volt auf +1 Volt. **Dazu klicken wir mit der RECHTEN MAUSTASTE auf das Symbol der Spannungsquelle und öffnen das „ADVANCED“ – Menü.**



Wir setzen den Spannungsverlauf aus lauter geraden Stücken zwischen mehreren vorgegebenen Punkten zusammen. Der zugehörige Facha Ausdruck lautet:

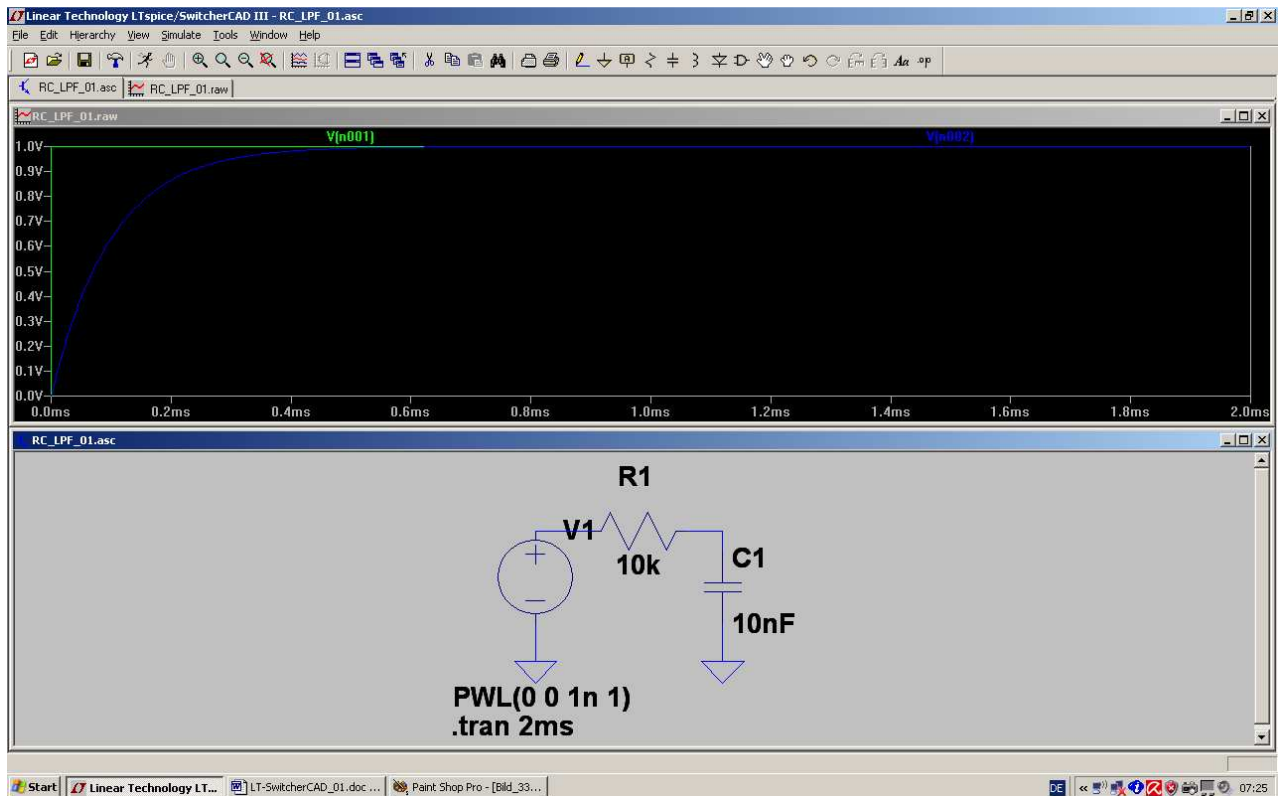
**PWL = piecewise linear**

und er wird im linken Menü aktiviert.

Beim Nullpunkt wählen wir Null Volt und nach einer Nanosekunde soll die Spannung auf +1V angestiegen sein (...und auf diesem Wert bleibt sie anschließend). Diese Wertepaare trägt man in die auftauchende Tabelle ein und macht noch zum Schluss die Anzeige dieser Werte sichtbar. Dann folgt „OK“.

Wer möchte, kann sich anschließend noch diese Zeile auf dem Bildschirm passend hinschieben (Wissen Sie es noch? Erst F7 drücken, dann mit der linken Maustaste auf die Zeile klicken. Sie hängt jetzt am Cursor und kann neu platziert werden. Schließlich mit einem neuen linken Klick absetzen und dann mit Escape die MOVE-Funktion verlassen).

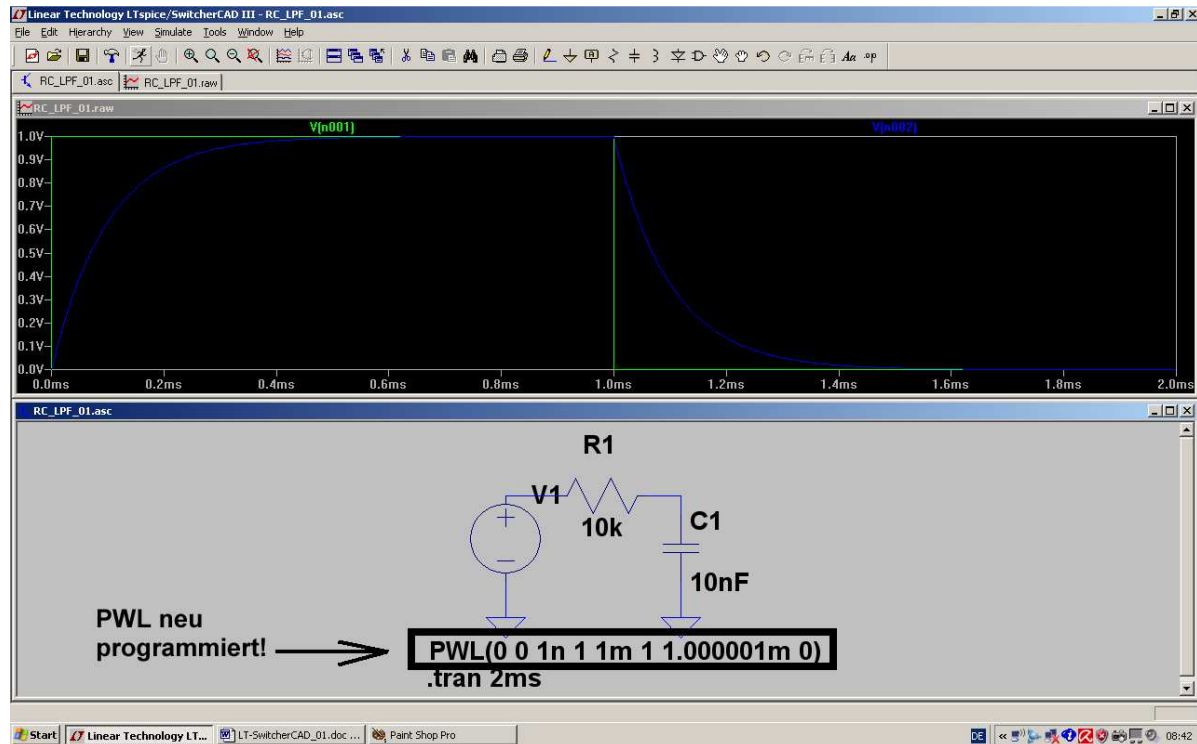
Nun kommt der Druck auf die Run-Taste mit den laufenden Männchen, das Anklicken von Eingangs- und Ausgangspunkt in der Schaltung und das Bewundern des Ergebnisses:



### 4.3.2. Ein- und Ausschaltvorgang

Kein Problem, denn dazu erweitern wir einfach unsere PWL-Liste um 2 Wertepaare. Sie muss dann so aussehen:

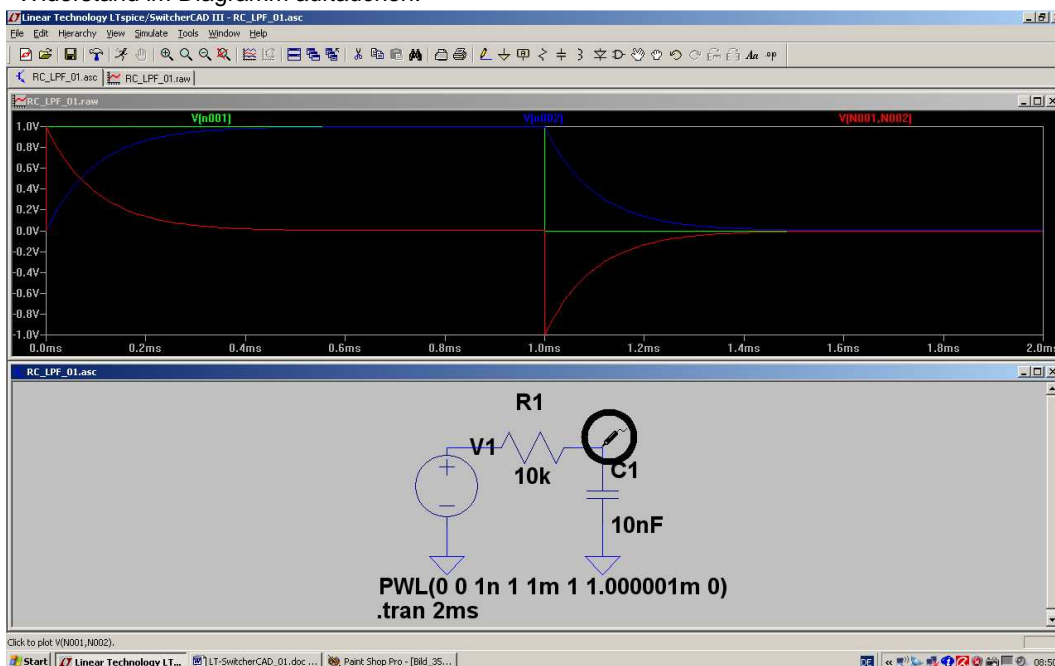
<b>0</b>	<b>0</b>	(Null Volt beim Startpunkt)
<b>1n</b>	<b>1</b>	(1 Volt nach einer Nanosekunde)
<b>1m</b>	<b>1</b>	(immer noch 1 Volt nach einer Millisekunde)
<b>1.000001m</b>	<b>0</b>	(eine Nanosekunde später wieder auf Null Volt zurück)



#### Zusatzaufgabe:

Stellen Sie zusätzlich im Diagramm die Spannung am Widerstand dar.

**Lösung:** wir setzen einen „Reference Point“ am Ausgang (= Verbindung zwischen Widerstand und Kondensator). Tippt man dann an die Verbindung von Spannungsquelle und Widerstand, so sollte die gesuchte Spannung am Widerstand im Diagramm auftauchen.



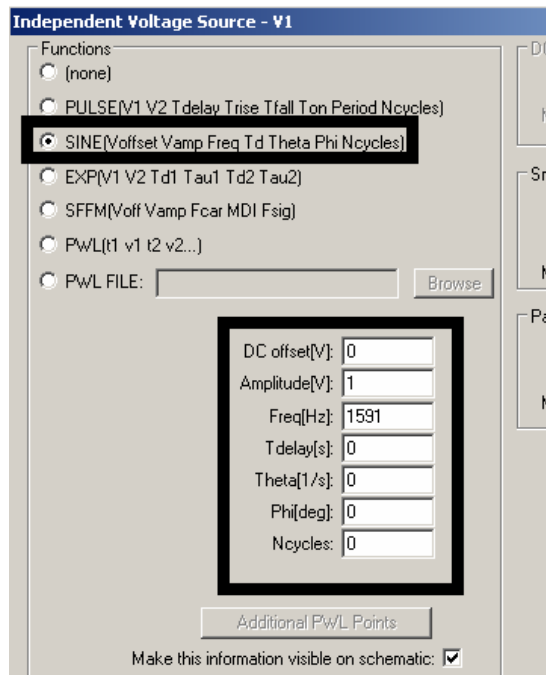
## 4.4. Periodische Signale am Eingang

### 4.4.1. Sinussignal mit $f = 1591 \text{ Hz}$

Wir steuern den Tiefpass etwa bei seiner Grenzfrequenz an und speisen ihn dort mit einer Amplitude von 1V Spitzenwert.

Übrigens wird  $f_{\text{Grenz}}$  so berechnet:

$$f_{\text{Grenz}} = \frac{1}{2 \cdot \pi \cdot R \cdot C} = \frac{1}{2 \cdot \pi \cdot 0,1\text{ms}} = 1591\text{Hz}$$

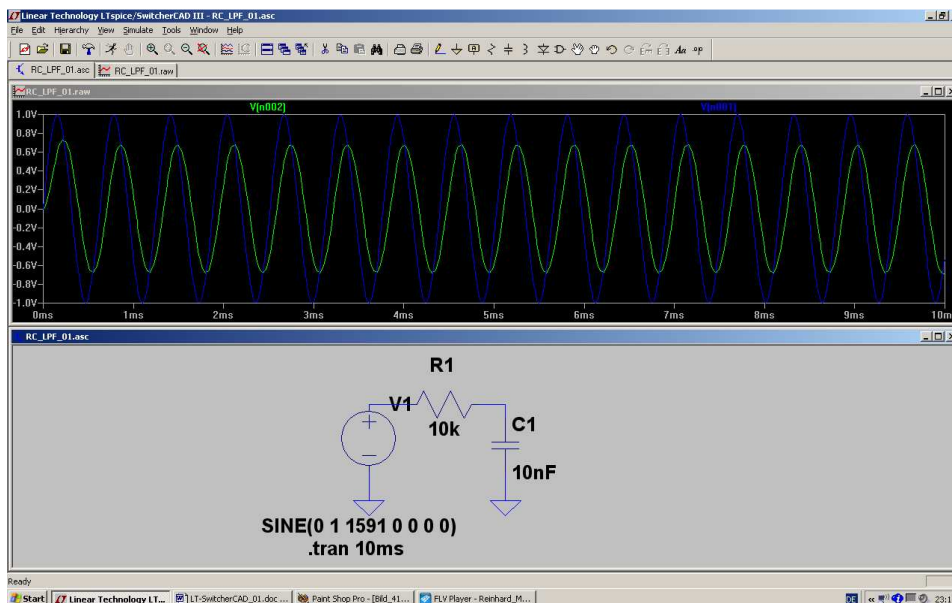


Dazu muss wieder das Symbol der Spannungsquelle rechts angeklickt und auf „Sinusbetrieb“ umgestellt werden.

Außerdem sind folgende Einträge nötig:

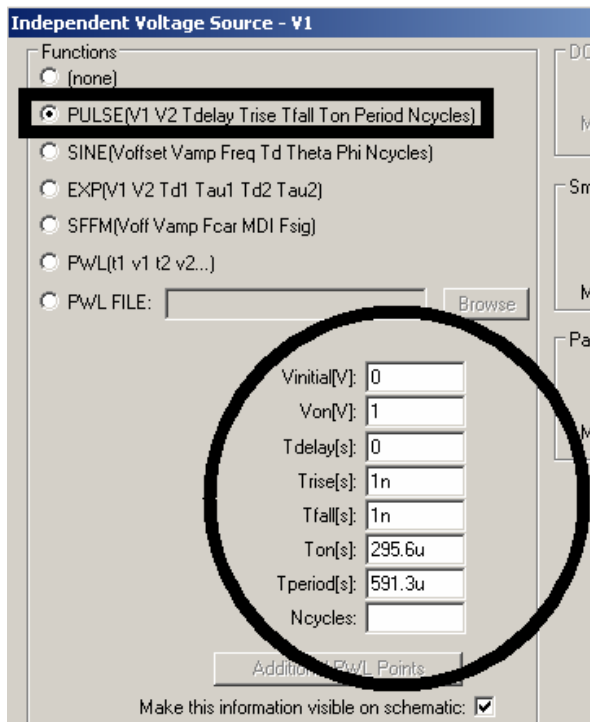
- DC Offset** = Null
- Amplitude** = 1 Volt (= Spitzenwert)
- Frequenz** = 1691 Hz
- Tdelay** = Null (= Startverzögerung)
- Theta** = Null (= Dämpfungsfaktor, Null ergibt Dauerschwingung)
- Phi** = Null (= Startphase der Schwingung)
- Ncycles** = Null (ist dann wichtig, wenn ein „Burst“ aus wenigen Schwingungen programmiert werden soll)

Als Nächstes geht man in das „Simulate“-Menü, öffnet die Zeile „Edit Simulation Command“ und gibt einen Stopp bei 10 Millisekunden vor. Nach der Simulation sieht man sehr schön die (zur Grenzfrequenz gehörende) Abnahme der Amplitude auf 70% und 45 Grad Phasen-Nacheilung bei der Ausgangsspannung.





## 4.4.2. Rechtecksignal mit $f = 1691 \text{ Hz}$



Auch hier ist der erste Schritt wieder das Programmieren der Spannungsquelle.

Wir brauchen „PULSE“ und folgende Einträge in den Feldern:

**Vinitial = Null (= Minimalwert der Spannung)**

**Von = 1 Volt (= Maximalwert der Spannung)**

**Tdelay = Null (= Startverzögerung)**

**Trise = 1 Nanosekunde (= Anstiegszeit)**

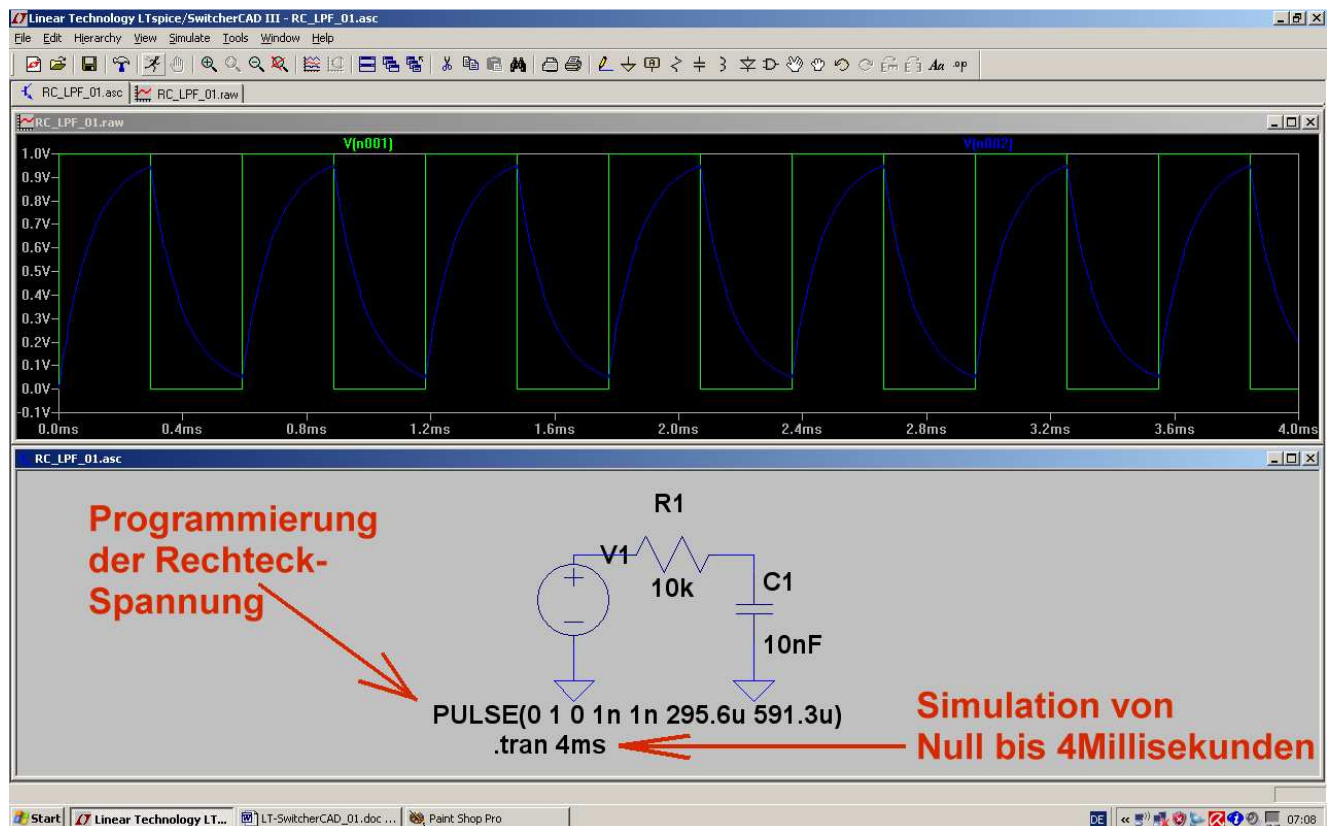
**Tfall = 1 Nanosekunde (= Abfallzeit)**

**Ton = 295,6 Mikosekunden (= Pulsdauer)**

**Tperiod = 591,3 (= Periodendauer)**

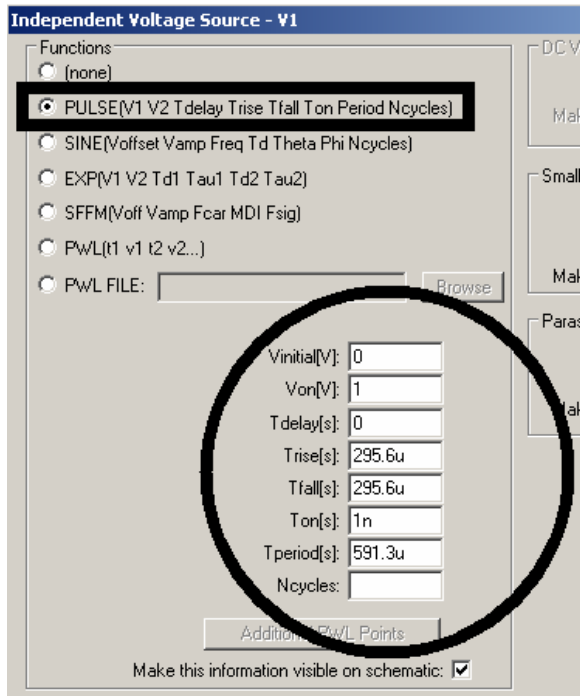
**Ncycles bleibt ohne Eintrag (= Zahl der Zyklen, wenn ein Burstsinal erzeugt werden soll)**

Wenn wir noch (über „Simulate“ und „Edit Simulation Command“ eine Simulationszeit von 0 bis 4 Millisekunden vorgeben, erhalten wir folgendes Bild:



### 4.4.3. Dreiecksignal mit $f = 1691 \text{ Hz}$

Da wird es etwas schwieriger, denn dafür müssen wir die **Pulsquelle etwas zweckentfremden**. Wählt man nämlich die Anstiegs- und Abfallzeit sehr groß und die Pulsdauer extrem kurz, dann muss logischerweise ein Dreiecksignal entstehen.



Also programmiert man die Spannungsquelle folgendermaßen:

**Vinitial = Null (= Minimalwert der Spannung)**

**Von = 1 Volt (= Maximalwert der Spannung)**

**Tdelay = Null (= Startverzögerung)**

**Trise = 295,6 Mikrosekunden (= Anstiegszeit)**

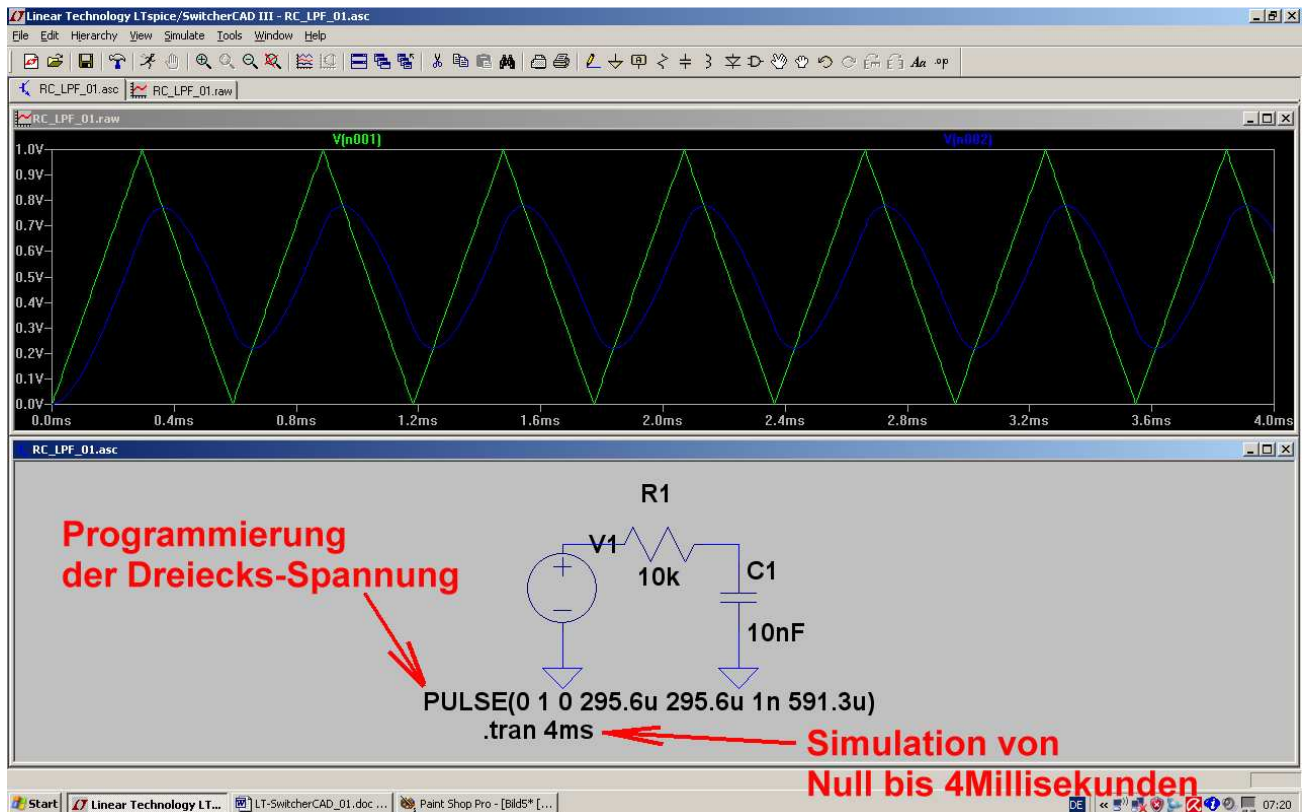
**Tfall = 295,6 Mikrosekunden (= Abfallzeit)**

**Ton = 1 Nanosekunde (= Pulsdauer)**

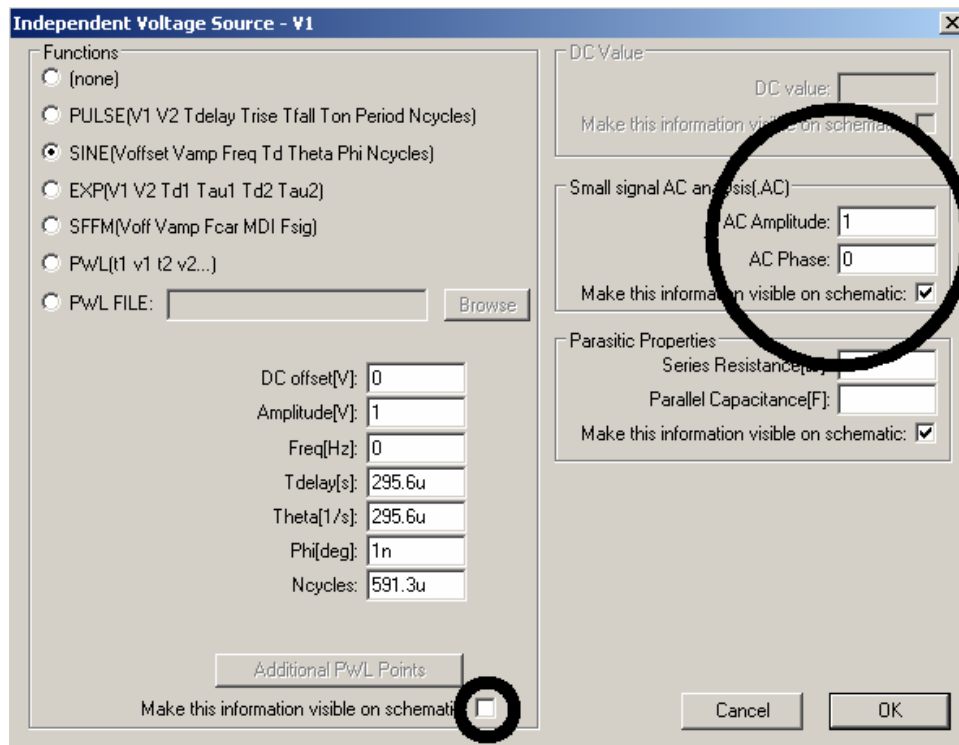
**Tperiod = 591,3 (= Periodendauer)**

**Ncycles bleibt ohne Eintrag (= Zahl der Zyklen, wenn ein Burstsinal erzeugt werden soll)**

Und so sieht das anschließend aus, wenn wieder die Zeit von 0...4 Millisekunden dargestellt wird:



## 4.5. AC-Sweep zur Ermittlung des Frequenzganges

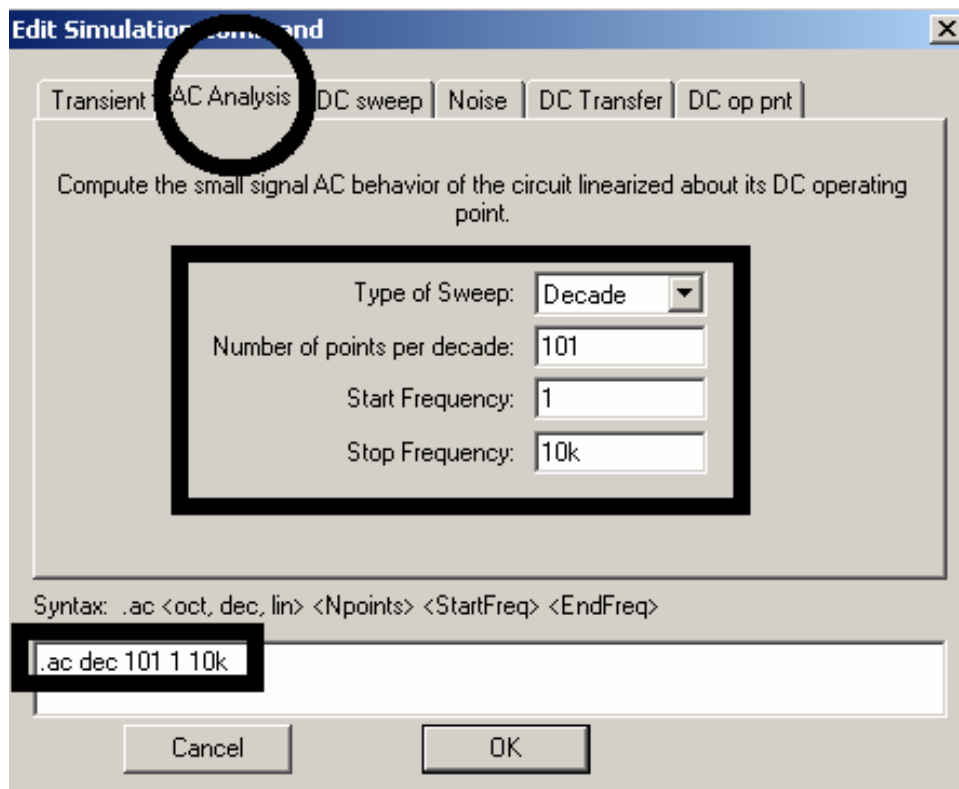


**1. Schritt:**  
Wir stellen die Spannungsquelle auf „AC analysis“ um und geben vor:

**AC Amplitude = 1 Volt**

**AC Startphase = Null**

Die noch aus dem vorigen Beispiel stammende Programmierung der Dreiecksspannung hat keinen Einfluss. Wir machen sie durch Entfernen des zugehörigen Häkchens auf dem Bildschirm unsichtbar.



**2. Schritt:**  
Wir rufen unter „Simulate“ die Zeile „Edit Simulation Command“ auf.

Im nebenstehenden Menü schalten wir gleich auf „AC Analysis“ um und geben folgende Dinge vor:

**Dekadischer Sweep**

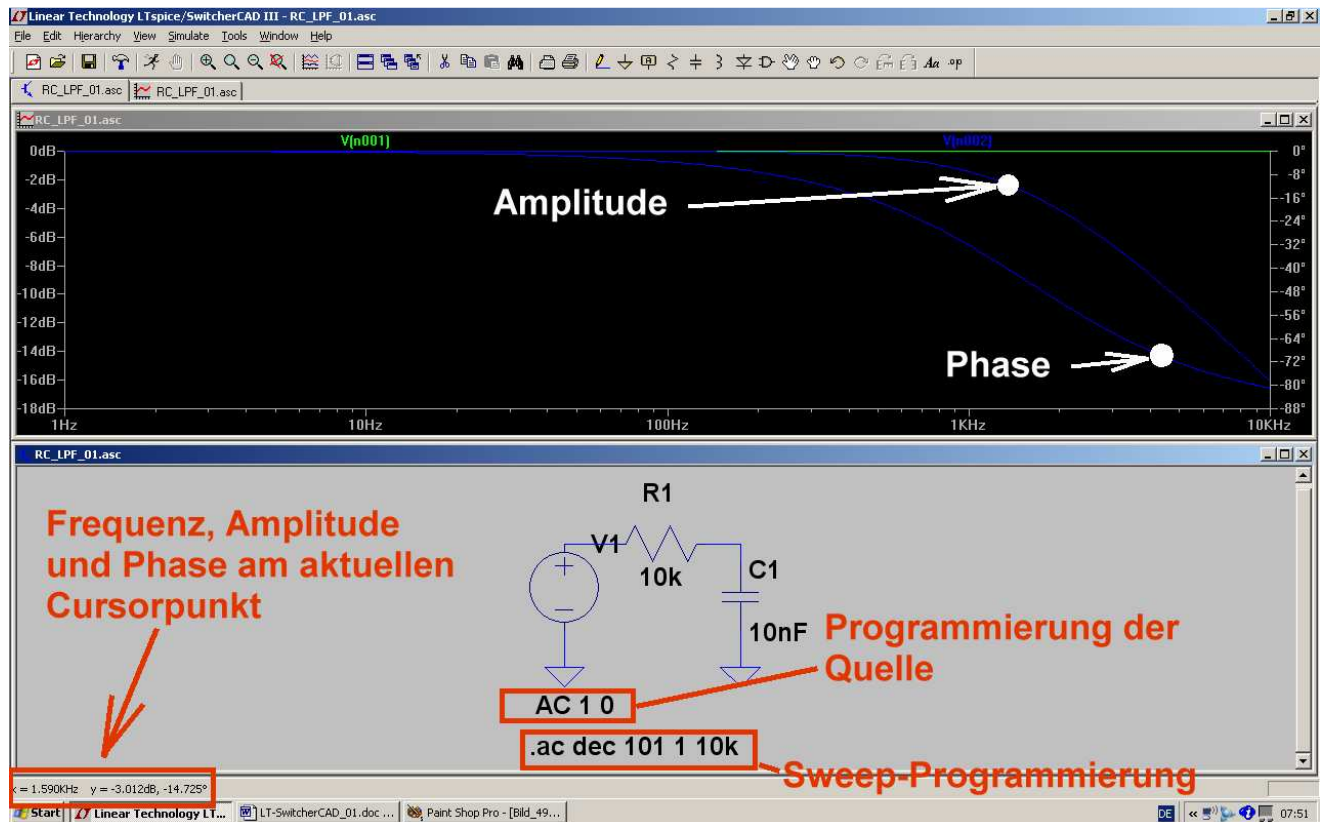
**101 Punkte pro Dekade**

**Startfrequenz = 1Hz**

**Stoppfrequenz = 10kHz**

Bitte die korrekte Übernahme in die Kommandozeile prüfen!

So einen Bildschirm müssen wir dann nach der Simulation vor uns haben.  
 Bitte mal die Sache mit der Anzeige über den Cursor selbst ausprobieren (...er steht gerade auf der Grenzfrequenz...)



Wenn die Phasenanzeige stört:

Bitte auf die Skalenteilung der rechten senkrechten Achse klicken und im auftauchenden Menü „Don't plot phase“ aktivieren.

Damit haben wir das erste eigene Beispiel geschafft!

## 5. FFT (= Fast Fourier Transformation)

Oft interessiert nicht nur der zeitliche Verlauf der einzelnen Spannungen (oder Ströme) in der Schaltung. Sobald nämlich ihre Kurvenform nicht mehr sinusförmig ist, gibt es außer der Grundwelle noch „Oberwellen“, also Signale mit doppelter, dreifacher, vierfacher Frequenz usw. Die **FFT** berechnet ein solches Frequenzspektrum aus dem zeitlichen Verlauf des Signals.

**Allerdings sind eine ganze Reihe an Vorüberlegungen bereits für die Simulation im Zeitbereich nötig, wenn die FFT anschließend befriedigende und korrekte Werte liefern soll:**

=====  
Die **Startfrequenz des simulierten Frequenzspektrums UND die Breite der dargestellten Linien UND der kleinste mögliche Frequenzschritt** im Spektrum werden durch die **Simulationszeit** bestimmt:

$$\text{Maximaler Frequenzschritt} = \text{Startfrequenz} = \text{Linienbreite} = 1 / \text{Simulationszeit}$$

Außerdem muss bei periodischen Signalen die **Simulationszeit immer ein ganzzahliges Vielfaches der Periodendauer** des simulierten Signals sein (...es darf keine Periode angeschnitten werden“). Wird die Forderung verletzt, dann erhält man plötzlich zusätzliche „gelogene“ Linien und / oder ein starkes „Grundrauschen“.

(Zusatzbemerkung für ganz obergenaue Spezialisten: genau genommen muss dieser Wert der Simulationszeit noch um einen Timestep gekürzt werden, denn sonst wird schon wieder eine „neue Periode angeschnitten“. Diesen Fehler merkt man aber nur, wenn sehr wenig Perioden für die FFT benützt werden oder der Timestep sehr grob ist).

=====  
Der „**Maximum Timestep**“ im Zeitbereich erfordert ebenfalls mehrere Überlegungen:

- Er soll so klein sein, dass die simulierte Kurve nicht „eckig“ aussieht.
- Die zugehörige „**minimale Samplefrequenz = 1 / Maximum Timestep**“ muss mindestens doppelt so hoch sein wie die allerhöchste im simulierten Signal vorkommende Oberwellen-Frequenz. Erst dann ist die „Shannon-Bedingung“ erfüllt und man braucht nicht mit „Aliasing-Effekten“ zu rechnen.
- Für die FFT muss eine bestimmte **Mindestmenge „echter“ Samples** zur Verfügung stehen, denn das bestimmt (**neben dem Maximum Timestep und der dadurch festgelegten Samplefrequenz**) die **höchste im Spektrum dargestellte Frequenz UND den Dynamikbereich der Amplitudenanzeige**. Deshalb macht es keinen Sinn, mehr Samples für die FFT vorzugeben als tatsächlich simuliert wurden (...gibt sonst Probleme und gelogene Details beim Spektrum). Diese Anzahl echter Samples kann leicht durch die Beziehung

$$\text{Echte Samplezahl} = \text{Simulationsdauer} / \text{Maximum Timestep}$$

ermittelt und kontrolliert werden.

=====  
LTSpice nimmt bei den errechneten Samples im Zeitbereich meist eine **Datenkompression** vor -- und genau das wollen wir nicht, weil wir ja möglichst viele „echte“ Werte für die FFT brauchen. Diese Datenkompression müssen wir durch eine selbst zu schreibende SPICE-Direktive

**.options plotwinsize=0**

abschalten.

=====  
Die für eine FFT verwendete **Anzahl an Samples muss immer eine Zweierpotenz** sein. Ist kein Problem, denn LTSpice listet uns automatisch die gemäß dieser Bedingung zugelassenen Samplemengen zur Auswahl in einem Menü auf. Unter immer muss man im Hinterkopf behalten: **Mehr Samples für die FFT geben zwar bessere Amplitudenauflösung und höhere Grenzfrequenz, erhöhen aber entsprechend die Rechenzeit und die Größe der Ergebnisdatei.....und setzen genügend „echte“ Samples voraus.**

=====  
**Noch etwas für die Spezialisten:** die Samplerate ist nicht immer konstant, denn wir geben ja nur den „Maximum time step“ vor und sorgen damit für eine bestimmte Mindest-Abtastrate. Bei kniffligen Signalstellen wird SPICE diese Abtastrate automatisch und ohne Vorwarnung erhöhen und wir merken das im Spektrum z. B. an unerklärlichen zusätzlichen Linien mit kleiner Amplitude oder an einem erhöhten „Grundrauschen bei hohen Frequenzen“. Der Effekt entspricht einer nicht genau vorhersehbaren zusätzlichen FM und wird gemindert, je kleiner wir den „Maximum Time Step“ bei der Simulation wählen. Dann steigt bei der Simulation der Samplevorrat entsprechend an, aber die zeitlichen Schwankungen beim Abstand von z. B. zwei aufeinander folgenden Werten sind kleiner. Das hilft umso mehr, je kleiner die Anzahl der für die FFT benützten Samples im Vergleich zu dem bei der Simulation angelegten Wertevorrat gewählt wird.

### Doch jetzt ein praktisches Beispiel:

Wir verwenden den vorigen Tiefpass und speisen ihn mit einem Rechtecksignal (Frequenz = 1kHz, Minimalspannung = Null Volt, Maximalspannung = +1V).

Mit einer Simulationszeit von 20ms und einem „Maximum Timestep“ von 100ns erfüllen wir folgende Forderungen und Bedingungen:

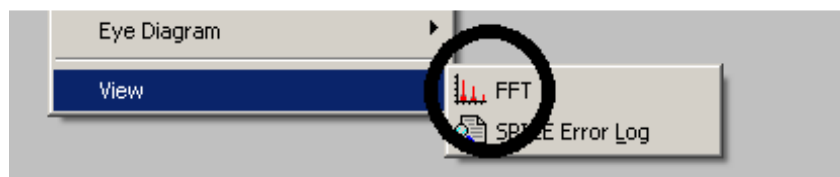
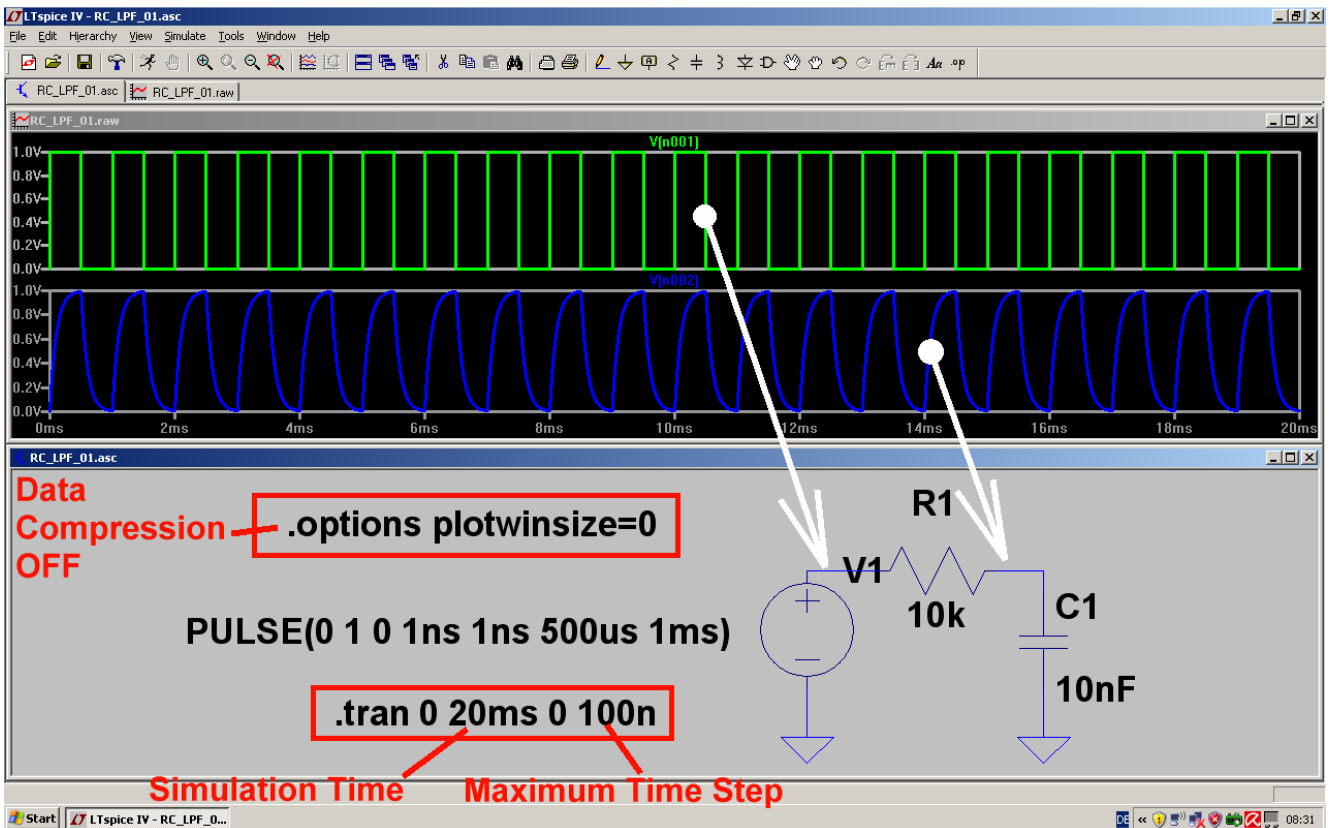
- Die Simulationszeit von 20ms ist ein ganzzahliges Vielfaches der Periodendauer (1ms).
- Der Maximum Timestep von 100ns ergibt eine minimale Samplefrequenz von  $1/100\text{ns} = 10\text{MHz}$ . Die Schmerzgrenze = Shannon-Grenze = höchste zulässige Oberwellenfrequenz würde dann bei 5MHz liegen.
- Die Anzahl der „echten Samples“ ergibt sich nun zu  $20\text{ms} / 100\text{ns} = 200\,000$  Samples. Also können wir im FFT-Menü gefahrlos

**„131 072 sampled data points in time“**

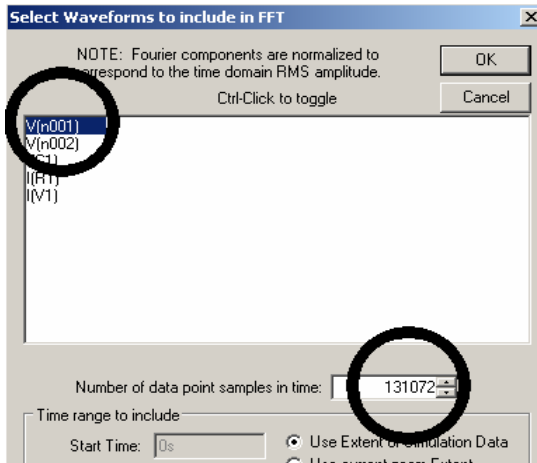
auswählen.

Jetzt zeichnen wir die Schaltung, programmieren die Spannungsquelle, legen Simulationszeit samt Timestep fest und schreiben die Direktive zum Abschalten der Datenkompression (..findet sich im EDIT-Menü).

Die Ergebnisdarstellung mit zwei Plot Panes erfolgt mit „Tile Horizontally“. Außerdem wird gleich auf „dicke Linien“ umgestellt. Das geht über „den Button mit dem Hammer“ auf die Karteikarte „Waveforms“. Dort steht schließlich „Plot data with thick lines“.



Nun klicken wir mit der **RECHTEN MAUSTASTE** auf die Rechteckspannung im Ergebnisdiagramm und wählen (ganz unten) erst „View“, dann „FFT“ im Menü

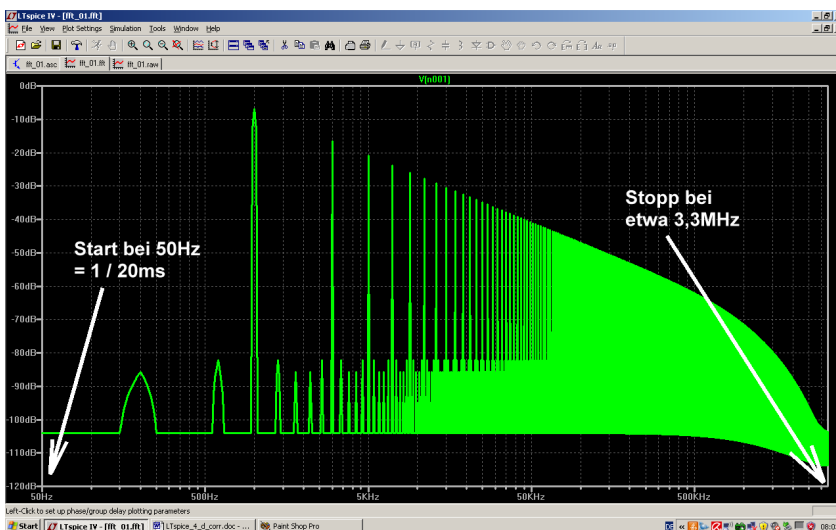


Dann aktivieren wir durch einen Mausklick die **Anzeige der Eingangsspannung V(n001)**.

Für die Berechnung des Spektrums stehen uns mindestens 200 000 echte Samples zur Verfügung. Also geben wir unbesorgt

**131 072 Datenpunkte (= Samples)**

vor und klicken dann OK.



Das ist der Erfolg. Die **Startfrequenz und die Frequenzauflösung** betragen:

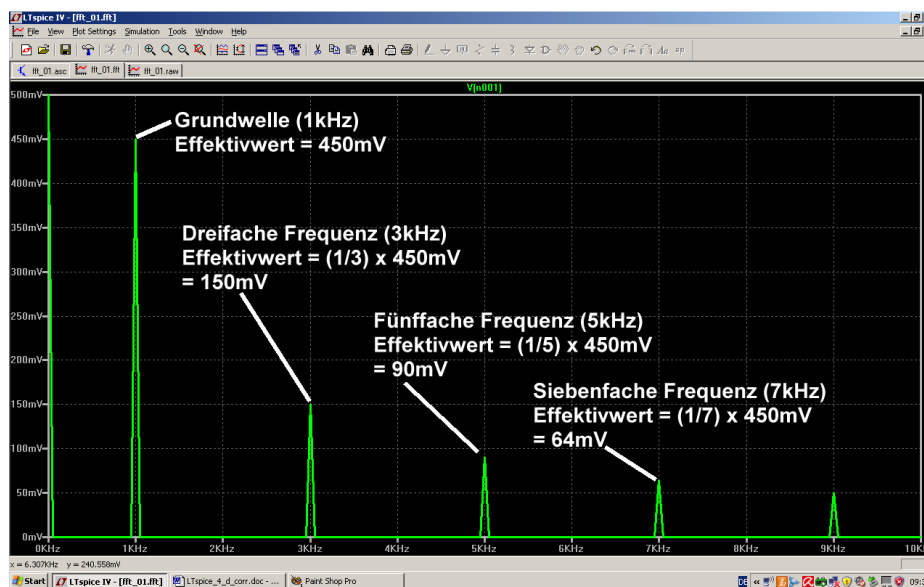
**50Hz = 1/20ms.**

Eine höhere obere Frequenzgrenze beim Spektrum (und größeren Dynamikbereich) gibt es dagegen nur mit mehr Samples.

Doch jetzt wollen wir auf eine lineare Darstellung wechseln und darin kontrollieren, ob die Theorie recht hat. Sie behauptet:

**Im symmetrischen Rechteck**

**gibt es nur Ungeradzahlige Oberwellen (dreifache, fünffache, siebenfache...Grundfrequenz). Ihre Amplituden nehmen exakt mit dem Grad der Oberwelle ab.**



Also fahren wir mit dem Cursor auf die senkrechten Achse, bis das Lineal auftaucht. Ein linker Mausklick genügt und im auftauchenden Menü können wir „linear“ wählen. Ebenso machen wir es bei der waagrechten Frequenzachse. Dort wählen wir eine lineare Darstellung von 0...10kHz mit einem Tick von 1kHz. Und Siehe da.....:

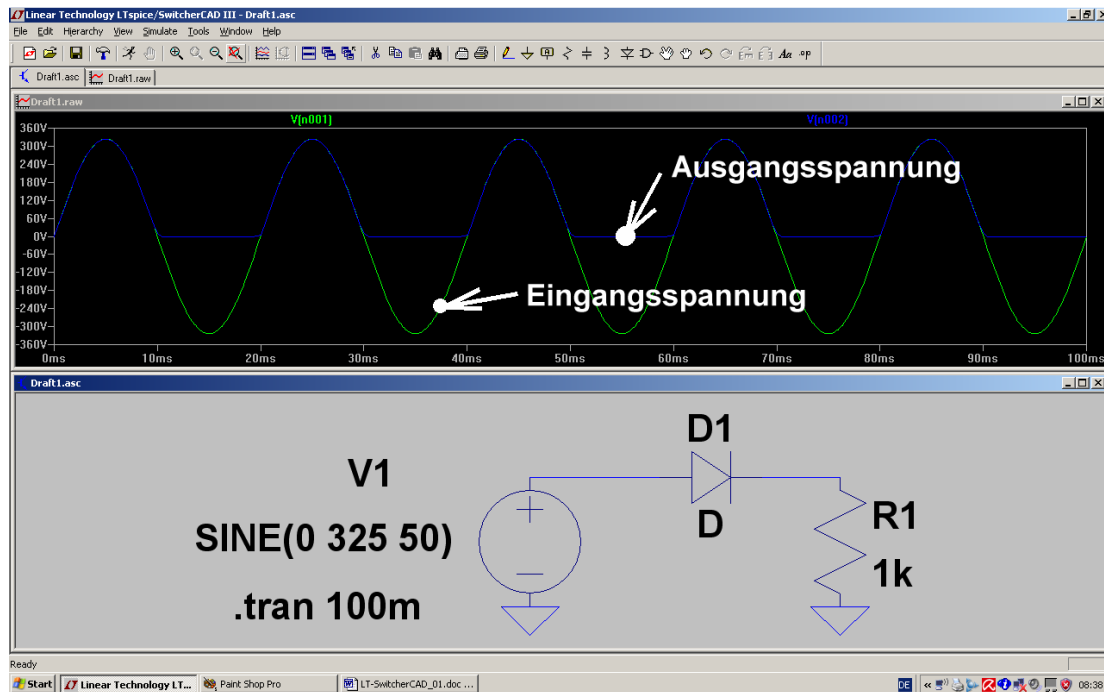
**Übrigens:**  
**Wer sich zusätzlich für die Phase interessiert, der klicke auf die**

**rechte senkrechte Achse und dort auf „Phase“.**

## 6. Zweites Projekt: Gleichrichtung

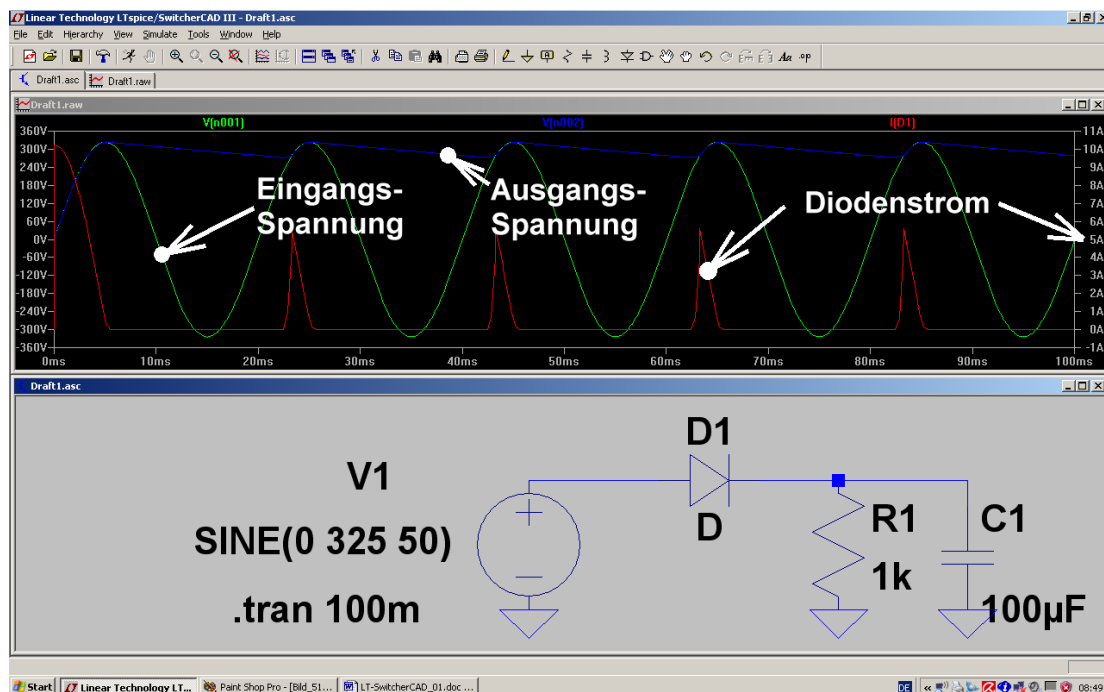
### 6.1. Einpuls-Gleichrichter ohne Trafo

Das darf nun kein Problem mehr sein: erst die Schaltung aus Spannungsquelle, Diode und Widerstand (1k) zusammenstellen, dann eine sinusförmige Eingangsspannung (50 Hz / Spitzenwert = 325 V) programmieren und eine Simulationszeit von 0 bis 100 Millisekunden vorsehen. Nach der Simulation sollte dies zu sehen sein:



**Aufgabe:** Schalten Sie am Ausgang noch einen Ladekondensator von 100 Mikروفarad parallel und simulieren Sie zusätzlich den Strom in der Diode (= fahren Sie dazu mit dem Cursor auf den linken Anschluss der Diode, bis sich der Cursor zur Stromzange verändert).

Lösung:



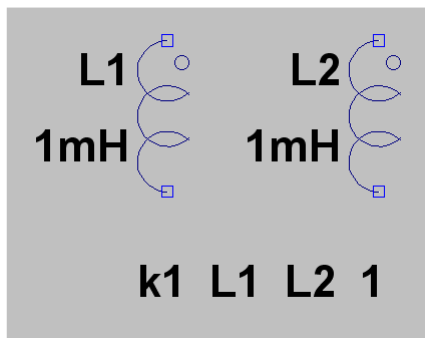


## 6.2. Eine wichtige Sache: ein Transformator für die Gleichrichterschaltungen

### 6.2.1. Die bequemste Lösung: simpler, idealer Trafo

Leider enthält die SwitcherCAD-Bibliothek keine speziellen Transformatormodelle, sondern nur die aus dem originalen SPICE-Manual stammende, einfache (= aber deshalb sehr gut verständliche) Lösung für den idealen Transformator. Sie sieht so aus:

**Man trägt die vorgesehene Anzahl an Wicklungen mit ihren Induktivitätswerten in die Schaltung ein und schreibt zusätzliche Spice-Direktiven für die magnetischen Kopplungsfaktoren „kn“ zwischen den einzelnen Wicklungen.**



Beispiel für einen idealen Übertrager mit 2 Wicklungen und dem Übersetzungsverhältnis 1:1

a) Als Schaltzeichen wird „ind2“ aus der Symbolbibliothek geholt und platziert. Es enthält außer der Induktivität (= Wicklung) noch die bekannte und bei Trafos übliche Kennzeichnung des **Wicklungsanfangs**.

b) Die Zeile „k1 L1 L2 1“ sagt: „die magnetische Kopplung zwischen der Wicklung L1 und der Wicklung L2 beträgt „1“ = „100%“.

c) **Achtung:**

**Eine direkte Eingabe des Übersetzungsverhältnisses ist bei SPICE**

**nicht möglich**, da immer nur mit den Bauteil-Eigenschaften (hier: Induktivitätswerte und magnetische Kopplung) simuliert wird. Also müssen wir das anders lösen und gehen auf die Induktivitäten los.

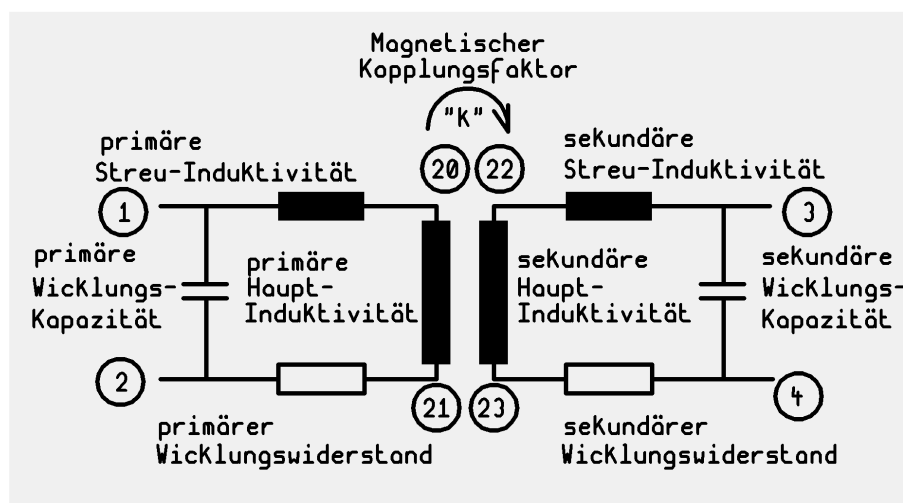
Dabei gilt folgendes:

$$\frac{L_{\text{Primär}}}{L_{\text{Sekundär}}} = (\dot{u})^2$$

und diese Formel stellen wir (wenn wir L1 vorgegeben haben und das Übersetzungsverhältnis kennen) einfach nach Lsek um. Ist also sehr einfach!

Nun soll als Nächstes mal gezeigt werden, wie man sich hilft, wenn man einen echten, gewickelten Übertrager vor sich hat und das ist z. B. bei Schaltnetzteilen unerhört wichtig. Dort wird nämlich sonst der Unterschied zwischen Simulation und wirklicher Schaltung so groß, dass die Simulation keinen Sinn mehr macht. (Die Ausgangsbasis war das SPICE-Modell für den Ausgangsübertrager eines röhrenbestückten HiFi-Verstärkers. Es wurde im Internet entdeckt und entsprechend umfunktioniert).

### 6.2.2. Erstellung des SPICE-Modells für einen realistischen Transformator mit zwei Wicklungen



Grundlage ist das übliche Standard-Ersatzschaltbild, in dem sowohl die Streuung wie auch die „Kopplung“ (= Anteil des primären Magnetfeldes im Sekundärkreis) berücksichtigt werden. Ebenso werden die Wicklungswiderstände und die Wicklungskapazitäten einbezogen.

Die Knoten 1 und 2 stellen die Anschlüsse der Primärwicklung, die Knoten 3 und 4 dagegen die der Sekundärwicklung dar. Für die „inneren Knoten“ dieses „Subcircuits“ wählt man höhere Knotennummern, um bei Erweiterungen irgendwelche Kollisionen zu vermeiden.

die der Sekundärwicklung dar. Für die „inneren Knoten“ dieses „Subcircuits“ wählt man höhere Knotennummern, um bei Erweiterungen irgendwelche Kollisionen zu vermeiden.

Nun zum untenstehenden SPICE-Modell mit dem **Subcircuit „xformer\_01“ samt den Anschlüssen 1 / 2 / 3 / 4:**

**Lleak** bilden die Streu-Induktivitäten. Sie repräsentieren diejenigen Magnetfeldanteile, die nicht im Kern und damit von der Primär- zur Sekundärwicklung verlaufen. Der Rest der Bauteilbezeichnungen ist (bei einem Blick auf das obige Ersatzschaltbild) selbsterklärend.

Ein **Sternchen** („Asterisk“) kennzeichnet eine Kommentarzeile. Und so sieht das Ergebnis aus:

```

*
* [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

```

Jede Bauteilzeile beginnt mit der Bezeichnung des Teils. Dann folgen die Schaltungsknoten, zwischen denen es hängt. Und den Schluss bildet der Bauteilwert.

Die Kopplung erreicht in der Praxis nie ganz den Wert „1“ und so sollte man sie auch eingeben (hier: k = 0,999). Manche SPICE-Versionen protestieren sogar, wenn das nicht so gemacht wird.

Bei der Primär- und Sekundärwicklung wurden gleiche Werte eingetragen, deshalb handelt es sich hier um einen 1:1 – Übertrager.

Die eigentliche Schaltungsbeschreibung beginnt mit **.SUBCKT .....** und endet mit **.ENDS**

=====

**Sehr wichtig:**

a) Dieses fertige Modell wird nun mit der Bezeichnung **xformer\_01.lib** im Ordner

**Programme / LTC / LTSpiceIV / lib / sub**

bei den schon vorhandenen übrigen SPICE-Modellen gespeichert. Bitte sorgfältig auf den Pfad UND die korrekte Endung achten...

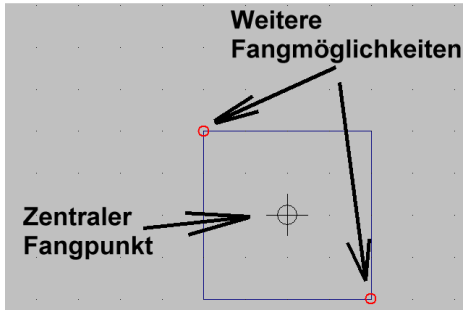
b) Es gibt keine **Möglichkeit, das Übersetzungsverhältnis direkt einzugeben**. Das läuft ausschließlich über eine Veränderung der Sekundären Hauptinduktivität und das machen wir im nächsten Projekt.

=====

### 6.2.3. Erzeugung eines passenden Symbols für diesen Transformator

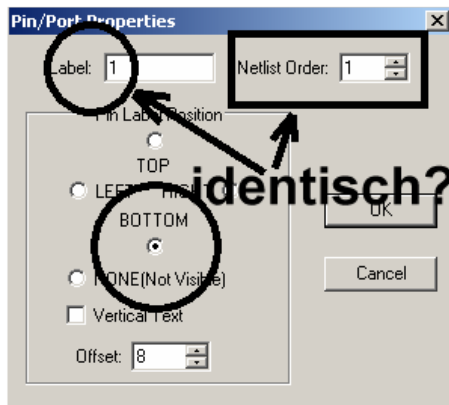
#### 1. Schritt:

Im Menü „File“ findet sich die Option „New Symbol“ -- sie wird angeklickt. Dadurch erscheint ein neuer Bildschirm mit einem „Fadenkreuz“ im Zentrum (..es bildet später den „Fangpunkt“ des Symbols beim Anklicken).



#### 2. Schritt:

Unter „DRAW“ wählen wir das Rechteck („Rect“) und zeichnen es in der Größe 20mm x 20mm um den Fangpunkt herum (...über dem Bildschirm liegt ein feines 5mm – Punktraster). Die roten Kreise an den Ecken des Rechtecks müssen sein, denn sie bilden später weitere Fangmöglichkeiten.

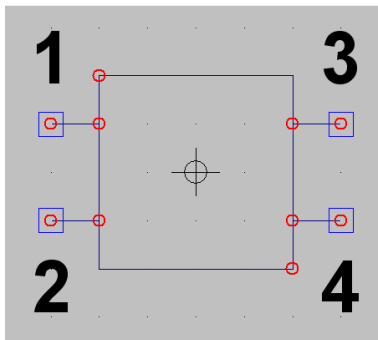


#### 3. Schritt:

Unter „Edit“ findet sich „Add Pin / Port“. Damit können wir die erforderlichen vier Anschlüsse erzeugen. Im zugehörigen Menü müssen wir jedoch jedes Mal die Pin-Nummer UND die Position des Anschlusses gegenüber der Pinbezeichnung („Label“) eintragen. Die nebenstehenden Einträge gelten für Pin 1.

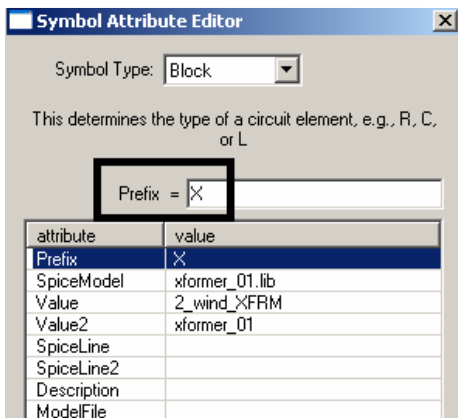
**Bitte darauf achten, dass der Pin-Label und die Netlist-Order übereinstimmen!**

Nach dem Klick auf OK hängt der Pin am Cursor und kann dann ca 5mm vom Symbol abgesetzt werden (Siehe nächstes Bild).



#### 4. Schritt:

Nachdem wir alle 4 Pins erzeugt haben, öffnen wir nochmals „DRAW“ und zeichnen die Verbindungsleitungen von den Pins zum Symbol. So sollte am Ende das fertige Symbol aussehen.



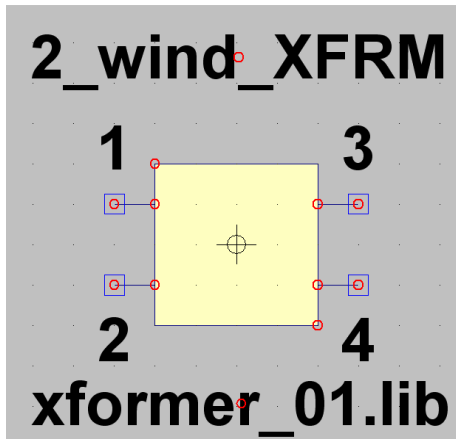
#### 5. Schritt:

Jetzt wird es ernst, denn nun soll aus dem hübschen Bildchen unser echter Transformator werden. Also heißt es, über „EDIT / Attributes / Edit Attributes“ an das nebenstehende Menü heranzukommen.

Dann übernehmen wir nacheinander diese Vorgaben, wobei wir auf die betroffene Zeile in der Liste klicken und im markierten Feld eingeben (**Hier bedeutet z. B. Prefix „X“ einen Subcircuit**).

#### Warnung:

**In der Zeile für Value2 MUSS IMMER ein Eintrag stehen (...am besten dieselbe Bezeichnung wie das Spice-Modell). Wenn der fehlt, kann es eine völlig unerklärliche Fehlermeldung geben....**



### 6. Schritt:

Jetzt gilt es noch, wichtige Informationen beim Aufruf des Symbols für den Anwender sichtbar zu machen. Dazu geht es nochmals in „Edit“ und „Attributes“, jetzt aber zu „Attribut Window“. Wenn wir nun auf „value“ klicken, dann hängt die Bezeichnung „2\_wind\_XFRM“ am Cursor und kann oberhalb des Symbols platziert werden. Dann wiederholt man das Spiel und wählt dieses Mal „SpiceModel“. Ganz korrekt erscheint das Modell-File „xformer\_01.lib“ am Cursor und kann unterhalb des Symbols angeordnet werden. Fertig!

### 7. Schritt:

Den Abschluss bildet das korrekte Abspeichern des Symbols. Wir wählen „File“ und „Save as“, müssen uns aber dann über „LTSpiceIV“ und „lib“ bis zum Ordner „sym“ (= Symbols) durchhangeln. Darin sollte man sich einen neuen Ordner „Xformers“ anlegen und das fertige Bauteil als „xformer\_01.asy“ endgültig speichern.

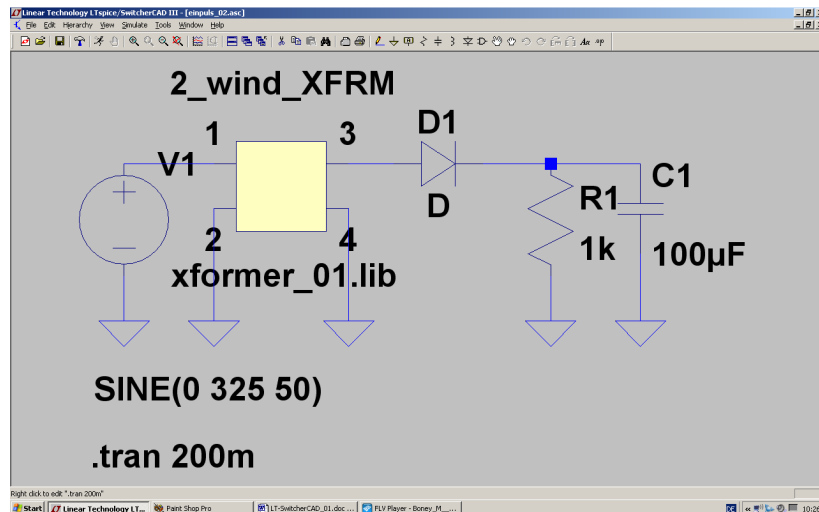
Tatsächlich alles geschafft!

## 6.3. Einpuls-Gleichrichter mit realistischem Trafo

Nun wollen wir die Ansprüche etwas höher schrauben und uns eine Schaltung mit Netztransformator samt Einpuls-Gleichrichter und Ladekondensator vornehmen.

### Aufgabe:

Untersuchen Sie ein Netzteil mit Netztrafo, Gleichrichterdiode, Lastwiderstand  $R_L = 1\text{k}\Omega$  und Ladekondensator  $C = 100\mu\text{F}$ . Der Trafo-Eingang liegt an der Netzspannung mit  $230\text{V} / 50\text{Hz}$ , das Übersetzungsverhältnis sei  $20:1$ . Simulieren Sie die Zeit von Null bis  $200\text{ms}$ .



### Lösung:

Die meisten Bauteile (Spannungsquelle, Kondensator, Widerstand, Diode, Ground-Symbol) kennen wir ja schon vom letzten Beispiel her. Dazu kommt nun der realistische Übertrager aus dem vorigen Kapitel. Bitte gleich alles platzieren sowie die Werte der Bauteile, der Spannungsquelle und die Simulationsanweisung eintragen.

Jetzt müssen wir uns noch um den Transformator kümmern und die Sache mit dem Übersetzungsverhältnis unterbringen.

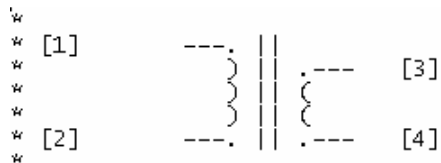
Dabei gilt folgendes:

$$\frac{L_{\text{Primär}}}{L_{\text{Sekundär}}} = (\dot{u})^2$$

Wenn wir (wie in den meisten Fällen für einen fast idealen Trafo zulässig) mal von einer Primär-Induktivität von  $1\text{Henry}$  ausgehen, dann müssen wir in unserem SPICE-File „xformer\_01.lib“ eine Sekundär-Induktivität von

$$L_{\text{Sekundär}} = \frac{L_{\text{Primär}}}{(20)^2} = \frac{1\text{H}}{400} = 2,5\text{mH}$$

benützen.



Für einen fast idealen Trafo kann man dann sowohl die Wicklungs-Widerstände, die Streu-Induktivitäten und die Wicklungs-Kapazitäten stark verkleinern.

**Aber man darf sie nicht ganz auf Null setzen!**

```

.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

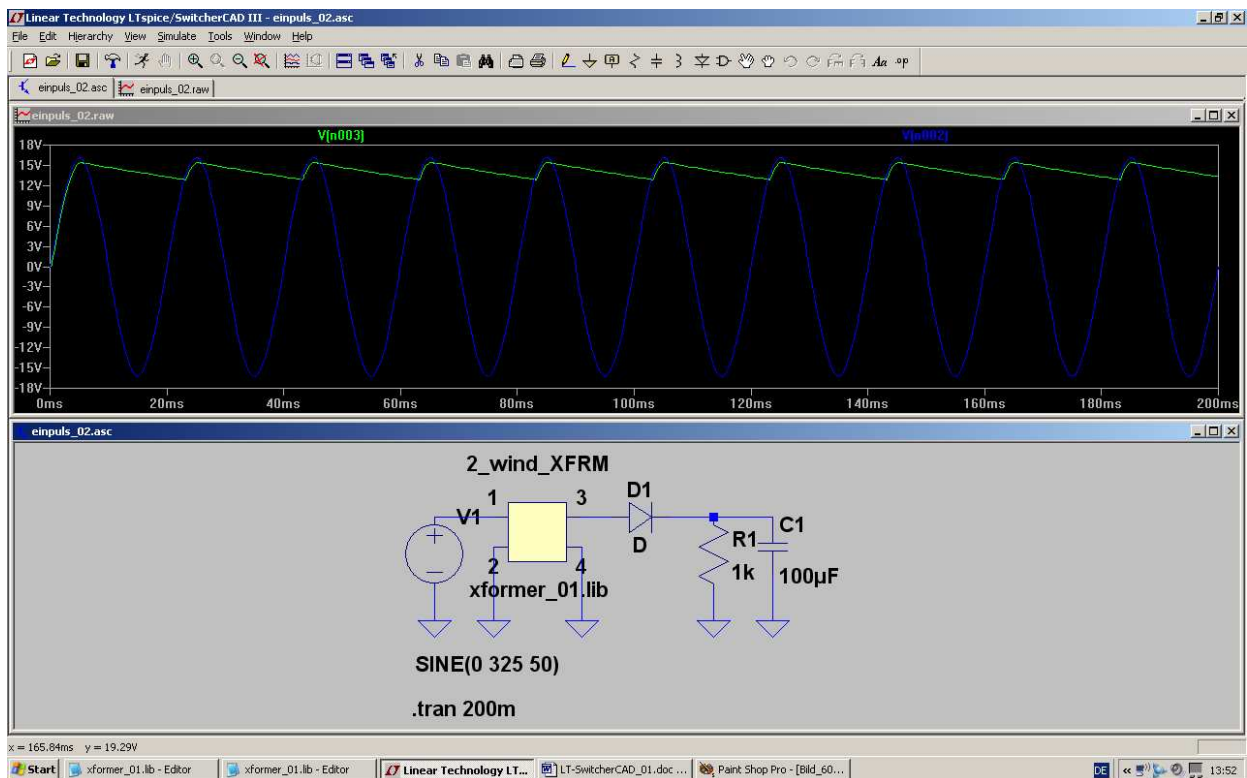
K Lpri1 Lsec1 0.999
.ENDS

```

**Speziell einen Widerstand von Null Ohm in Reihe zu einer Induktivität beantwortet das originale SPICE sofort mit einem Abbruch und einer Fehlermeldung!**

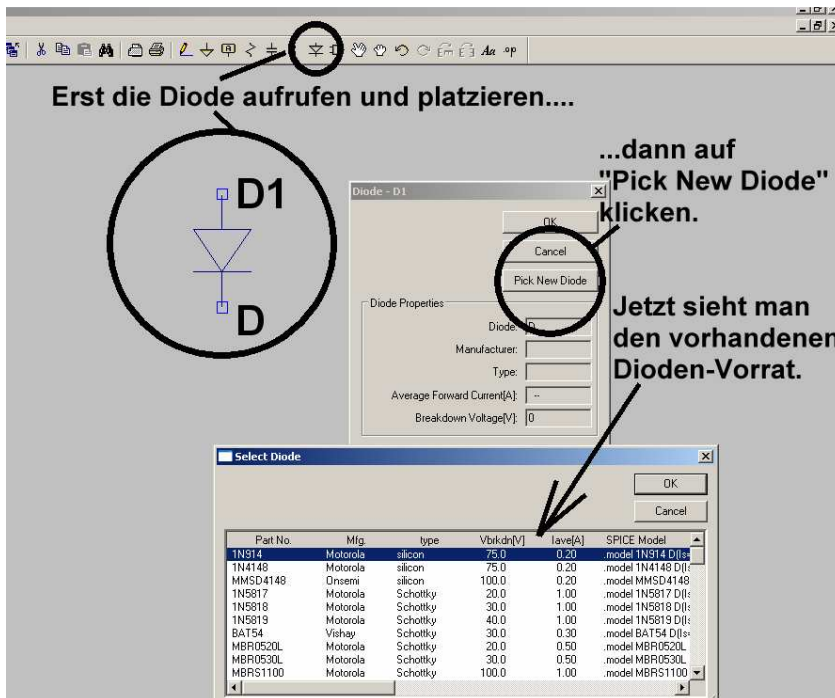
So sieht dann das „xformer\_01.lib“-File aus, das wir anschließend wieder am korrekten Ort (= Ordner „sub“ im „lib“-Verzeichnis von LTSpiceIV) abspeichern müssen:

Der Rest, also die eigentliche Simulation, ist eine Kleinigkeit und bekannt. Deshalb folgt gleich das Ergebnis für die Spannungen vor und nach der Diode:



## 6.4. Verwendung der Diode 1N4007 in der Gleichrichterschaltung

Wir wollen nun anstelle der idealen Diode den bekannten Diodentyp „1N4007“ einsetzen. Leider ist dieses SPICE-Modell nicht in der mitgelieferten Bibliothek enthalten und deshalb üben wir auf diese Weise das Einbinden fremder Modelle.



Ein Tipp: wer wissen möchte, welche Diodentypen bereits beim Programm im Bauteilvorrat mitgeliefert werden, der verfähre nach der nebenstehenden Methode.

Aber jetzt wollen wir die Sache mit der 1N4004 anpacken.

### 1. Schritt:

Im Internet suchen wir (z. B. mit Google) nach der Library „diode.lib“. Sie enthält (als OrCad-Bibliothek) viele Netzgleichrichter.

### Aber Vorsicht:

**Diese Library kommt bei uns als HTML-File an und ist in dieser Form noch nicht zu gebrauchen!**

Da bleibt nur ein Weg: erst den Inhalt mit <STRG> + <a> komplett markieren und mit <STRG> + <c> in die Zwischenablage kopieren. Dann den Text-Editor (z. B. Notepad) öffnen und die Zwischenablage mit <STRG> + <v> in ein neues Blatt einfügen.

Erst jetzt können wir das File als „diode.lib“ im Ordner

### LTC / LTSpiceIV / lib / sub

speichern. (Bitte aber dabei auf die korrekte Endung achten und im Editor vorher beim File-Typ „Alle Dateien“ wählen -- sonst wird nämlich zusätzlich die Endung „.txt“ angehängt und dann findet unser Programm die Datei nicht.....)

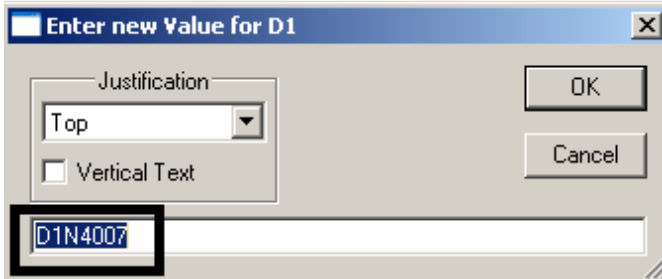
```
*$
.model D1N4004 ako:D1N4001 D(Bv=600)
*$
.model D1N4005 ako:D1N4001 D(Bv=900)
*$
.model D1N4006 ako:D1N4001 D(Bv=1200)
*$
.model D1N4007 ako:D1N4001 D(Bv=1500)
*$
.model D1N4009 D(Is=544.7E-21 N=1 Rs=.1 Ikf=
+ Vj=.75 Fc=.5 Isr=30.77n Nr=2
*$
```

### 2. Schritt:

Mit dem Text-Editor (z. B. Notepad) öffnen wir nochmals das File „diode.lib“ und suchen nach „1N4007“. Dabei stellen wir fest, dass dieses Modell die Bezeichnung

### D1N4007

aufweist!



### 3. Schritt:

Nun fahren wir in unserem Schaltplan mit dem Cursor auf die untere Angabe „D“ beim Diodensymbol und klicken mit der rechten Maustaste darauf.

Jetzt können wir als neuen Wert „D1N4007“ eintragen.

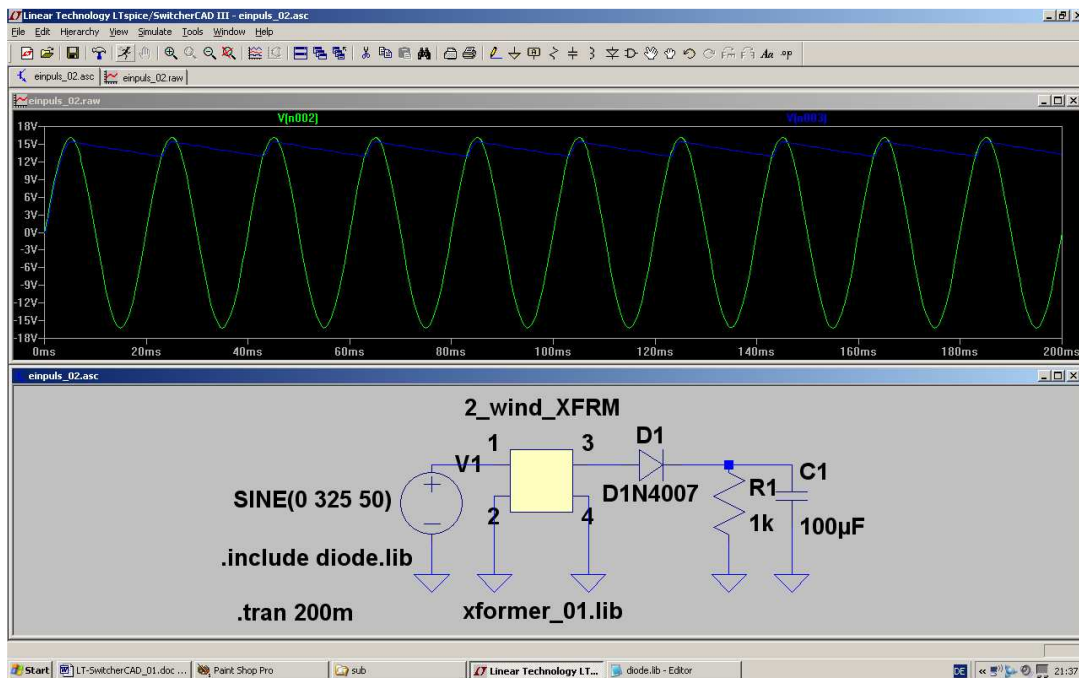
### 4. Schritt:

Aber das ist noch nicht alles. Damit die Einbindung und Zuweisung richtig funktioniert, müssen wir „Edit“ und „SPICE Directive“ aufrufen. Wir tippen ein:

**.include diode.lib**

und setzen diese Zeile irgendwo in unserem Schaltplan ab.

### 5. Schritt:



Jetzt läuft die Simulation korrekt mit dem neuen Diodenmodell.

## 6.5. Zweipuls-Gleichrichter mit realistischem Trafo

Hier ist ein neues Bauteil erforderlich: ein Transformator mit zwei getrennten Sekundärwicklungen. Und wir wollen wieder nicht den einfachen breiten, sondern den steinigern Weg beim Trafomodell wählen...

```
*
*
*
*
* [1] ---|
*           |
* [2] ---| [3]
*           |
*           | [4]
*           | [5]
*           |
*           | [6]
*
*
.SUBCKT xformer_02 1 2 3 4 5 6
*
* Primary
*
Lleak1 1      20      1uH
Lpri1  20     21      1H
Rpri1  21     2        1
Cpri1  1      2        1p

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

* Secondary 2
*
Lleak5 5      24      1uH
Lsec2  24     25      2.5mH
Rsec2  25     6        1
Csec2  5      6        1p

K      Lpri1 Lsec1 Lsec2 0.999
.ENDS
```

### 1. Schritt:

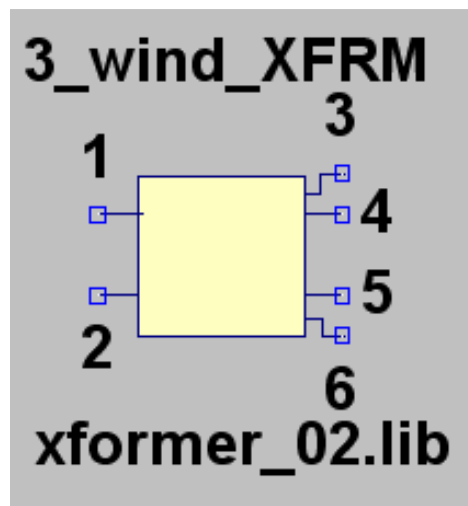
Also gehen wir in den Texteditor, rufen unser File „xformer\_01.lib“ auf und ändern es ab.

Wir sehen gleich wieder ein Übersetzungsverhältnis von  $\dot{U} = 20:1$  vor und tragen die zu einer Primär-Induktivität von 1H gehörenden Sekundär-Induktivitäten von je  $1H / 400 = 2,5mH$  in die entsprechenden Zeilen ein.

Da wir mit einem nur fast idealen Übertrager arbeiten wollen, machen wir die Streu-Induktivitäten, die Wicklungswiderstände und die Wicklungskapazitäten zwar sehr klein, aber nicht Null.

Bitte nicht vergessen: Wir haben es nun mit **drei Induktivitäten zu tun, die miteinander verkoppelt sind** und müssen das im File angeben.

Stimmt alles, dann speichern wir dieses neue File als „xformer\_02.lib“ wieder im „sub“-Ordner der Library.



### 2. Schritt:

Auch ein neues Symbol ist nötig. Die Erstellungs-Prozedur läuft exakt so ab wie im vorigen Beispiel (Kapitel 5.2.2.) und am Ende muss der Übertrager so aussehen.

Bitte aber bei der Erstellung der **Attributliste** wieder dran denken:

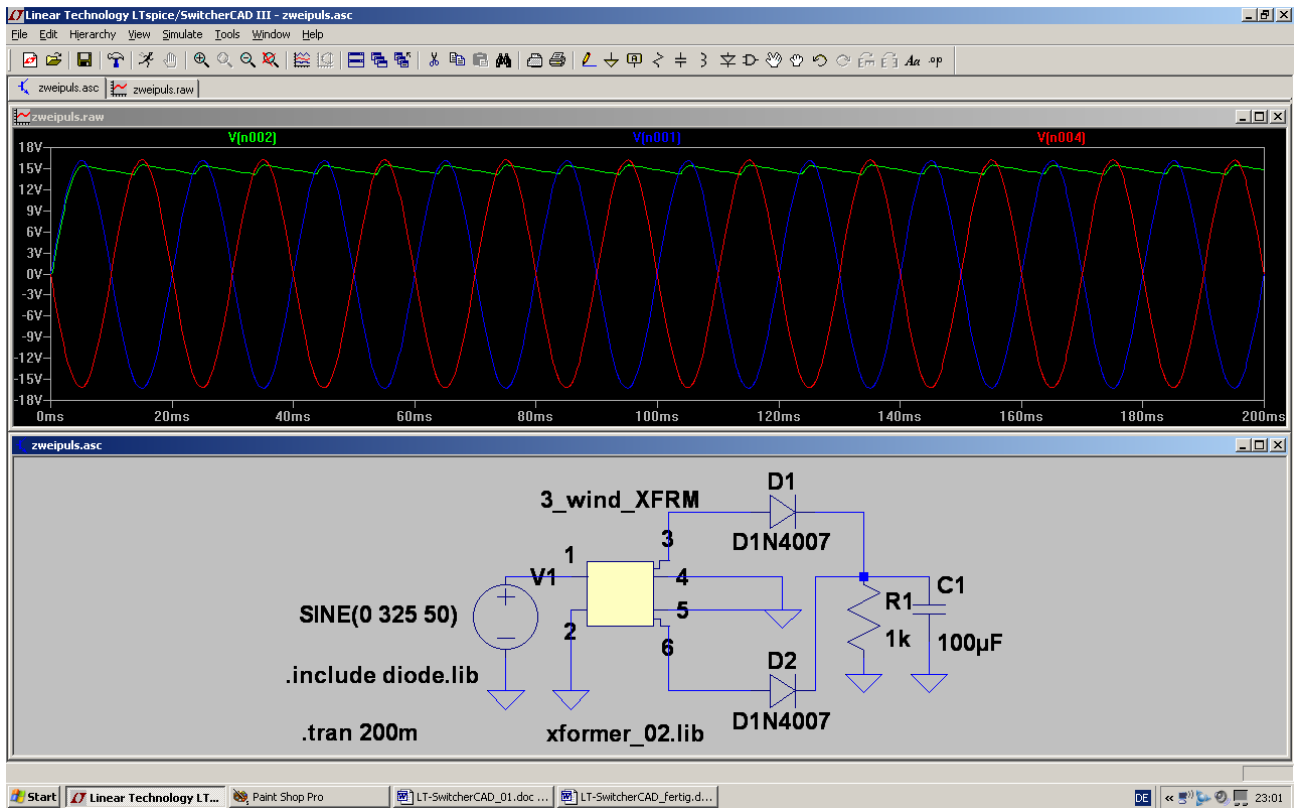
**Warnung:**  
In der Zeile für Value2 MUSS IMMER ein Eintrag stehen (...am besten dieselbe Bezeichnung wie das Spice-Modell).  
Wenn der fehlt, kann es eine völlig unerkklärliche Fehlermeldung geben....

### 3. Schritt:

Nun wird die „Zweipuls-“ (= Mittelpunkts-) Schaltung unter Verwendung dieses Übertragers sowie 2 Dioden vom Typ 1N4007 gezeichnet und alles simuliert.

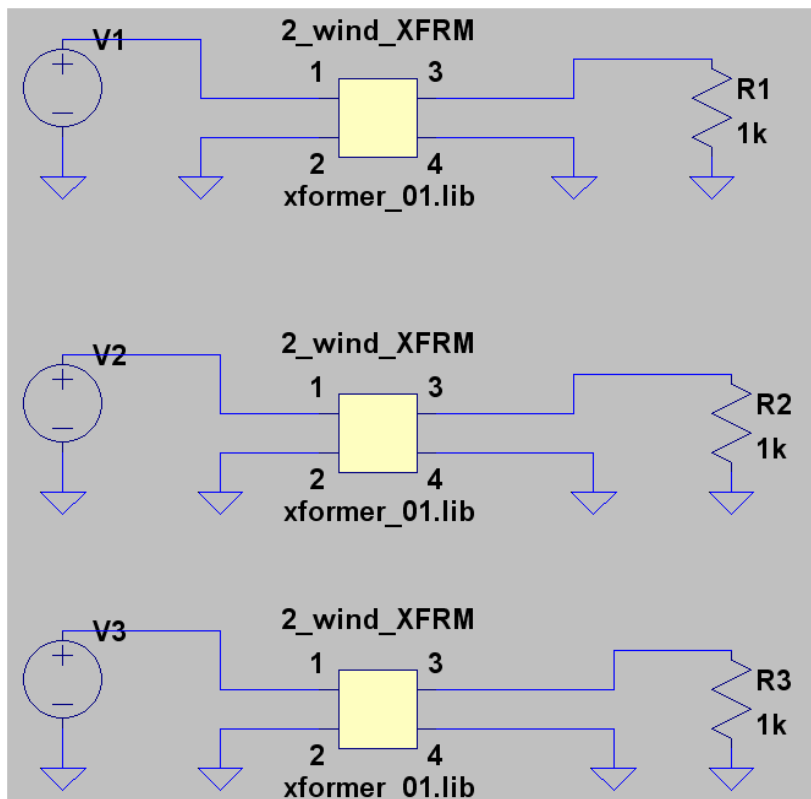
Schaltung und Ergebnis für die beiden Sekundärspannungen des Trafos sowie für die Ausgangsspannung: Siehe nächstes Blatt.





## 7. Drittes Projekt: Drehstrom

### 7.1. Programmierung eines Drehstromsystems



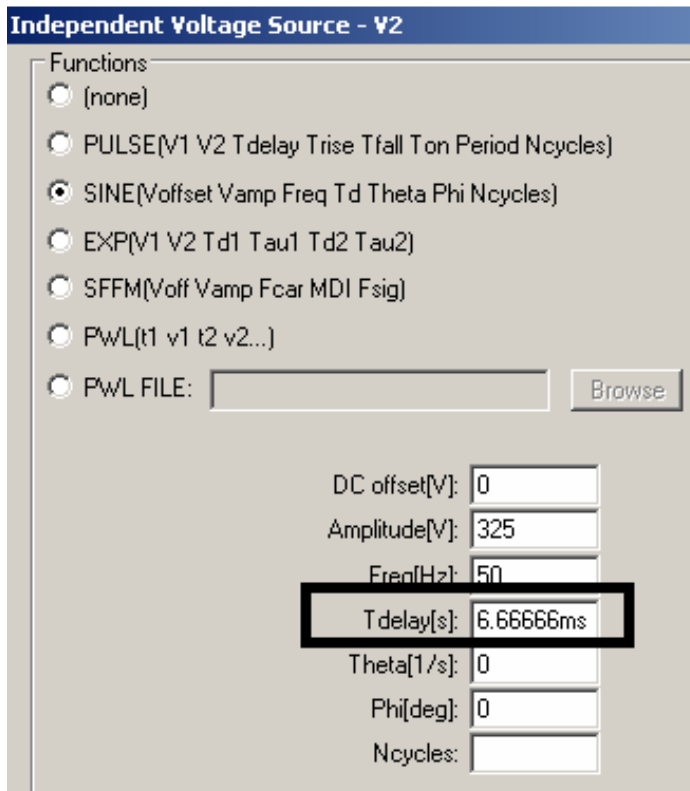
Wir wollen uns mit drei Spannungsquellen, drei realistischen Trenntrafos und drei Lastwiderständen die Sache mit dem Drehstrom etwas näher ansehen.

Beim Transformator verwenden wir die schon bekannte Datei

„xformer\_01.lib“,

aber wir **machen beide Hauptinduktivitäten gleich und erzielen so ein Übersetzungsverhältnis von  $\dot{u} = 1:1$ .**

Diese Bauteile werden geholt und in der nebenstehenden Form verdrahtet.



Anschließend fahren wir bei jeder Spannungsquelle mit dem Cursor auf das Symbol und klicken rechts. Bei allen drei Quellen sind folgende Einträge identisch:

**DC offset = 0 Volt**

**Amplitude = 325 V (= Spitzenwert für einen Effektivwert von 230 V)**

**Frequenz = 50 Hz**

**Phi = 0 degrees (= Phasenlage beim Kurvenstart)**

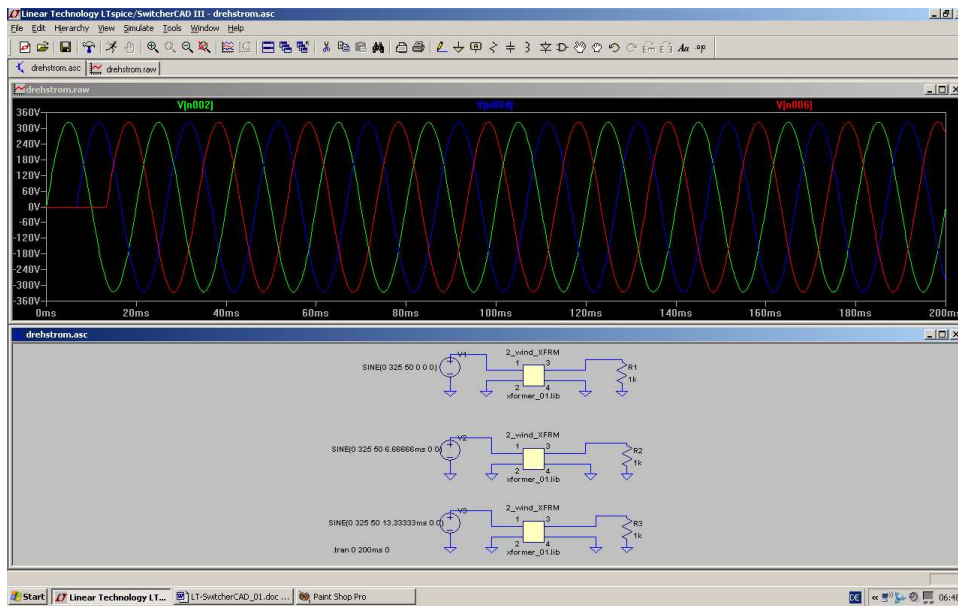
Wichtig:  
Die beim Drehstrom erforderliche Phasenverschiebung von 120 Grad zwischen den einzelnen Kurven erreichen wir durch **drei unterschiedliche Startverzögerungen** (= Tdelay)!

**Bei V1 beträgt sie Null Millisekunden**

**Bei V2 beträgt sie (siehe Bild) 6,66666 ms**

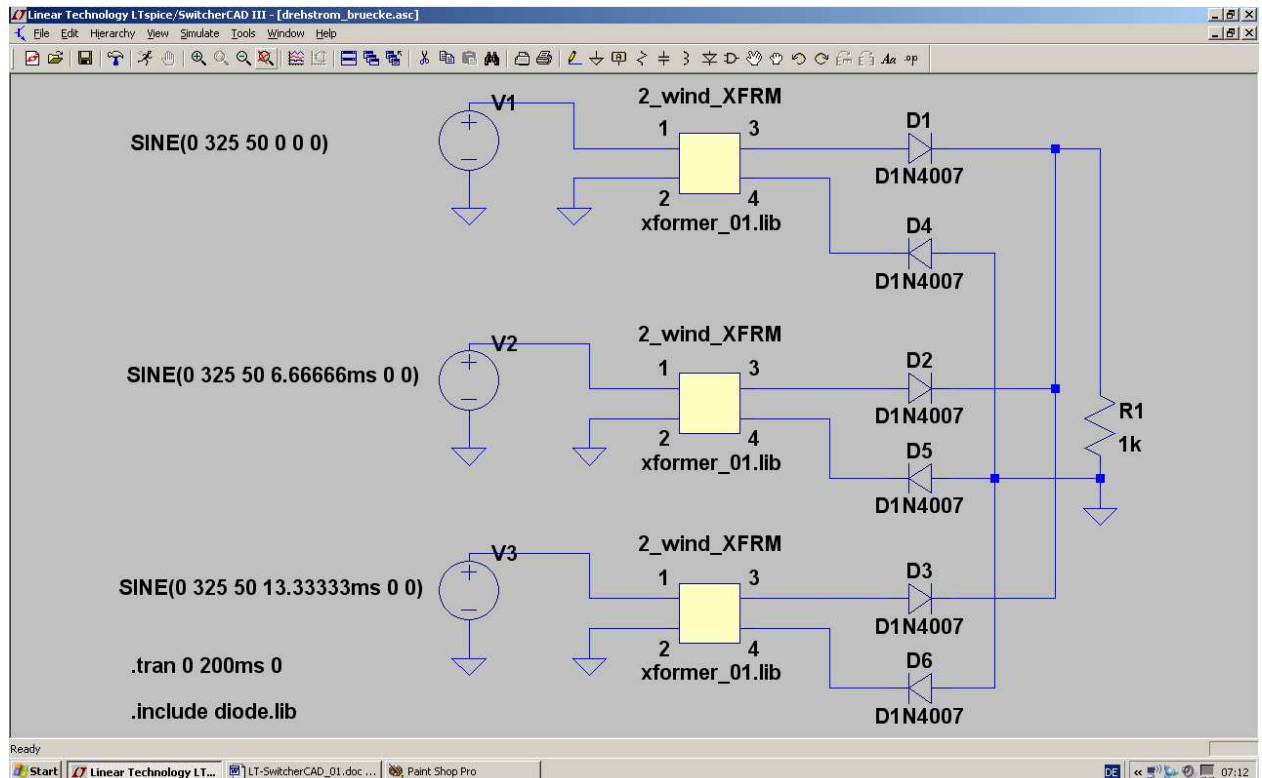
**Bei V3 beträgt sie 13,33333ms**

Programmieren wir nun noch eine Simulationsdauer von 200 ms , dann sieht das Simulationsergebnis so aus:

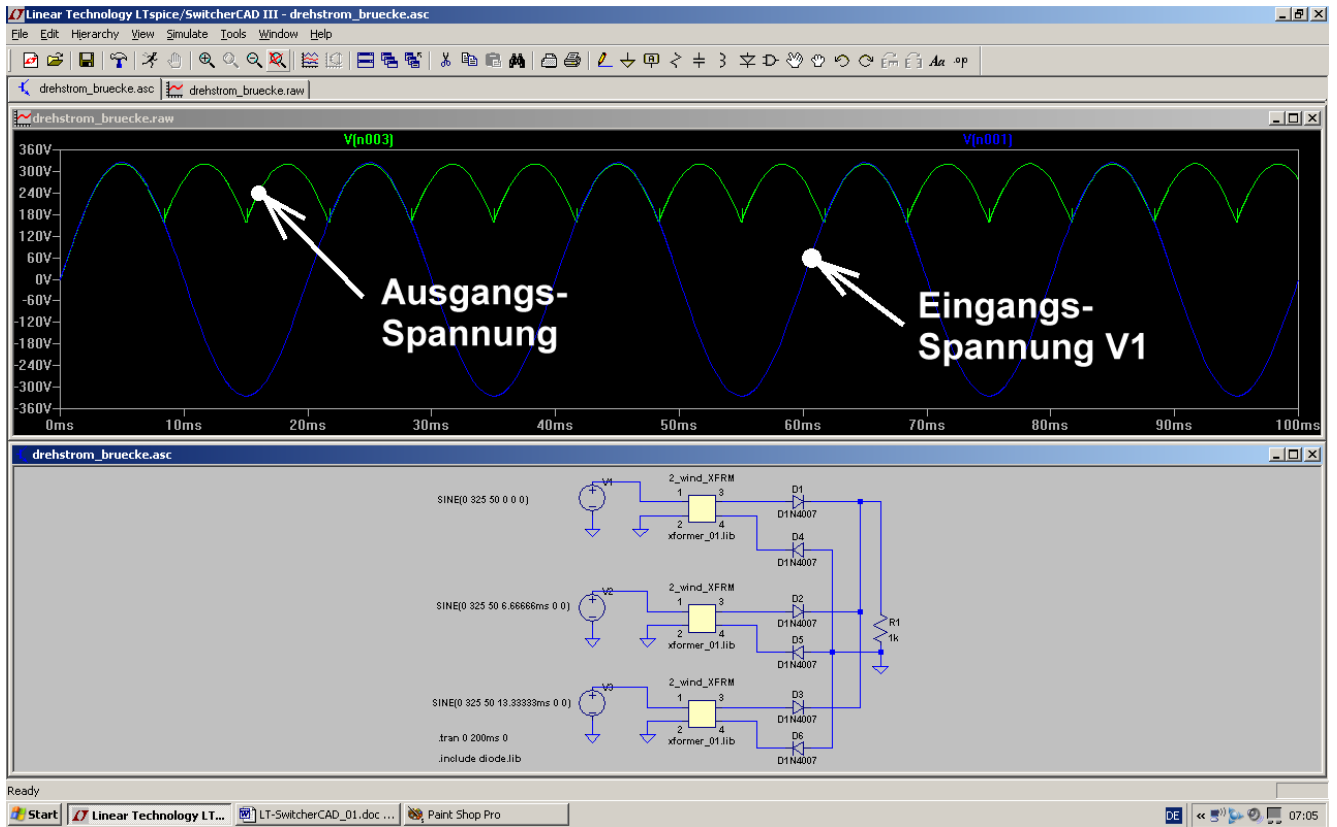


## 7.2. Drehstrom-Lichtmaschine im Auto

Wir speichern die vorige Schaltung als neues Projekt und geben ihr einen anderen Namen. Dann ändern wir die Schaltung ab und brauchen dazu zusätzlich **6 Dioden vom Typ 1N4007**. Zwei der drei Widerstände entfallen und dann können wir die Schaltung neu verdrahten.



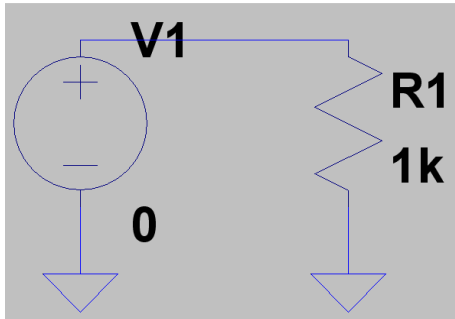
So sieht das Ergebnis der Simulation aus, wenn wir nur die Eingangsspannung V1 sowie die Ausgangsspannung am Lastwiderstand darstellen lassen:



## 8. Viertes Projekt: Darstellung von Bauteil-Kennlinien

### 8.1. Ohm'scher Widerstand

Dieses Bauteil eignet sich besonders gut zur Demonstration des Vorgehens, wenn U-I-Kennlinien sichtbar gemacht werden sollen.

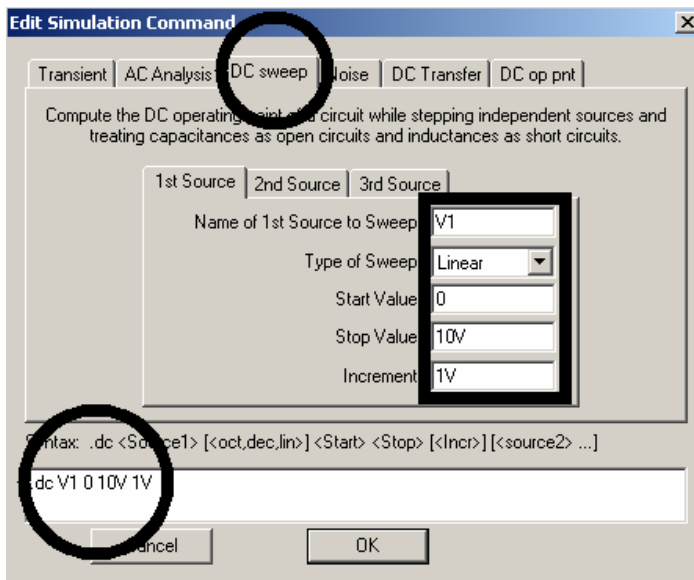


Dazu öffnen wir ein neues Projekt („New Schematic“) und zeichnen uns die nebenstehende Schaltung.

Beim Widerstand „R1“ wählen wir einen Wert von 1kΩ.

Dann müssen wir das Property Menu der Spannungsquelle öffnen (= rechter Mausklick auf das „Plus-Zeichen“ im Symbol....) und dort im rechten oberen Eck beim DC-Value „Null Volt“ hineinkriegen. Wir tippen also in das zugehörige Fenster „0“ ein.

Die erfolgreiche Übernahme prüfen wir an unserem Schaltbild.



Im Menü „Simulate“ wählen wir

### „Edit Simulation Command“

und haben dann den nebenstehenden Bildschirm vor uns.

Auf der Karteikarte „DC Sweep“ programmieren wir nun die Spannungsquelle V1 mit folgenden Vorgaben:

**Startwert = Null Volt**

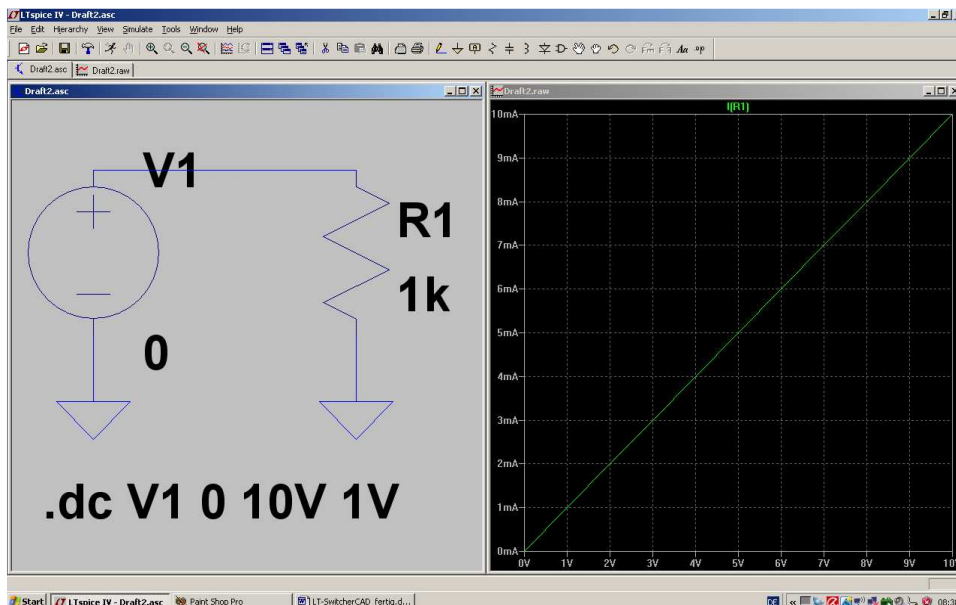
**Endwert = 10 Volt**

**Schrittweite = 1V**

**Linearer Sweep**

Das damit erzeugte Simulation Command nochmals kontrolliert werden.

„.dc V1 0 10V 1V“ findet sich im unteren Feld und sollte



Dann bitte mit OK schließen und die Simulation starten. Fährt man dann mit dem Cursor im Schaltbild auf den oberen Anschluss des Widerstandes, bis sich der Cursor in eine „Stromzange“ verwandelt, dann reicht ein kurzer Klick auf diesen Punkt.

Da haben wir alles...

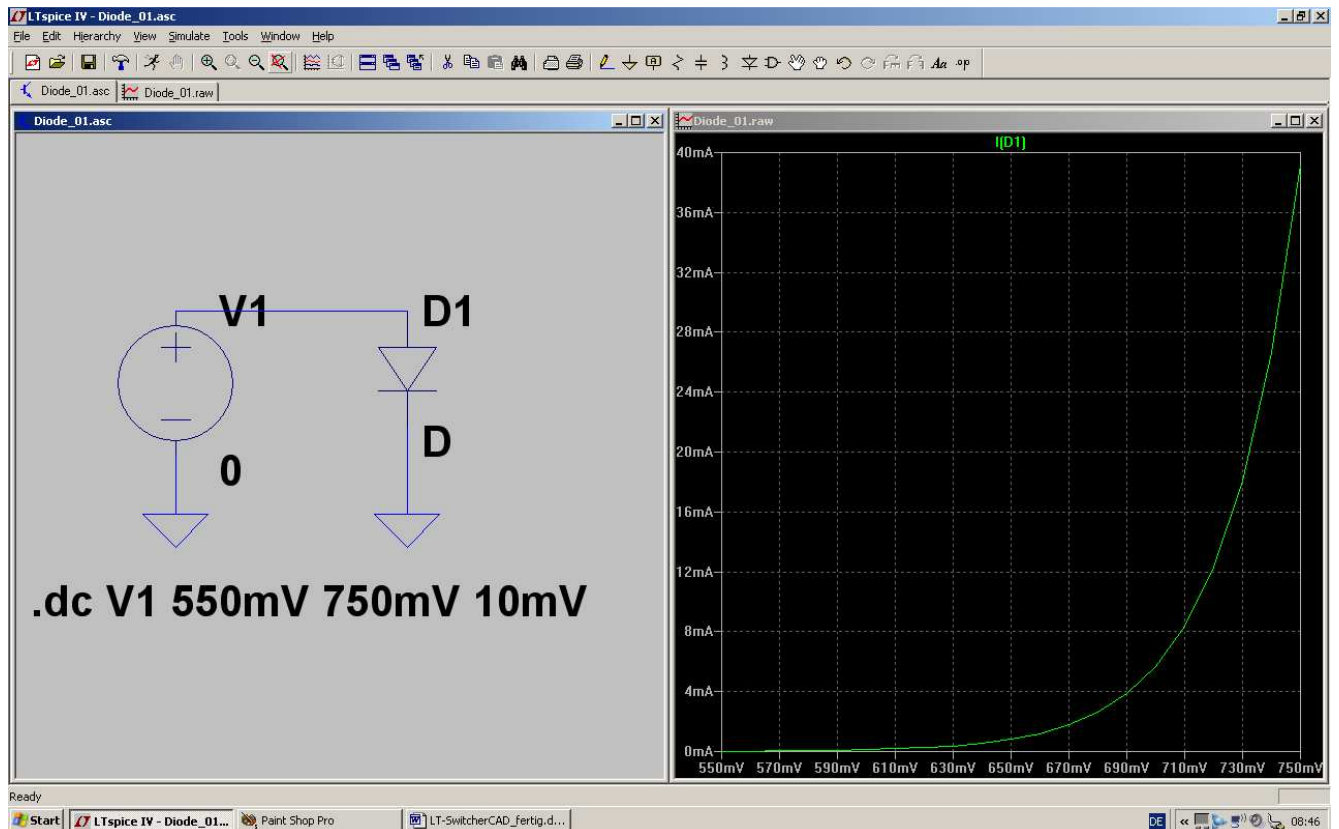
## 8.2. Diode

Das ist jetzt eine sehr einfache Übung:

Erst das Projekt unter einem neuen Namen abspeichern. Dann den Widerstand (nach Druck auf F5) herauslöschen und dafür die Diode (...sie findet sich als Button in der oberen Menüleiste oder als „diode“ in der Bauteile-Liste...) einfügen. Anschließend noch das Simulation Command umändern in

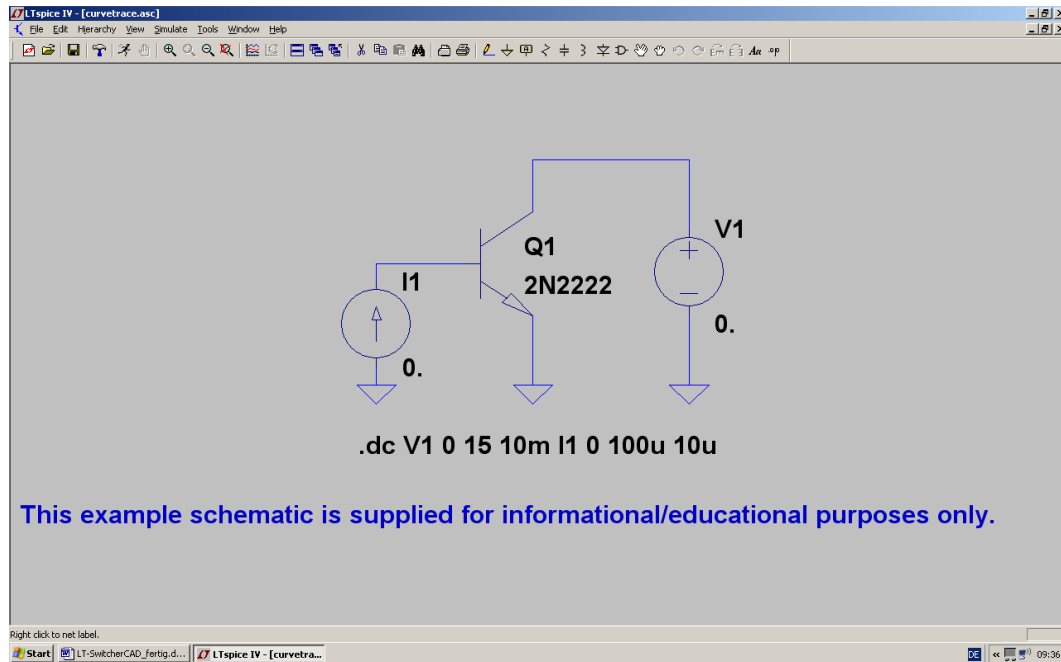
**.dc V1 550mV 750mV 10mV**

Damit wird nur der Spannungsbereich von 550mV bis 750mV dargestellt und in Schritten von 10mV simuliert. Das ist für eine Siliziumdiode genau richtig.



### 8.3. NPN-Transistor

Da gibt es in der „example“-Bibliothek des LTSpiceIV-Programmes ein sehr hübsches Beispiel, das wir uns einfach hernehmen:



Man erkennt:

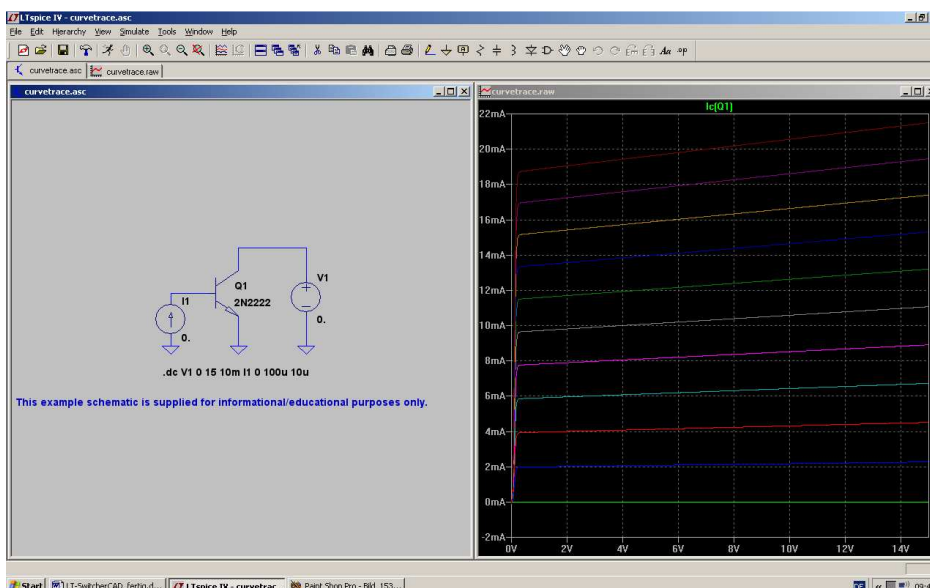
- Am Kollektor des Transistors wird die Gleichspannungsquelle V1 angelegt. Sie liefert zunächst Null Volt.
- Eine Konstantstromquelle I1 speist die Basis des Transistors. Auch ihr Startwert ist „Null“

Das Simulation Command

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

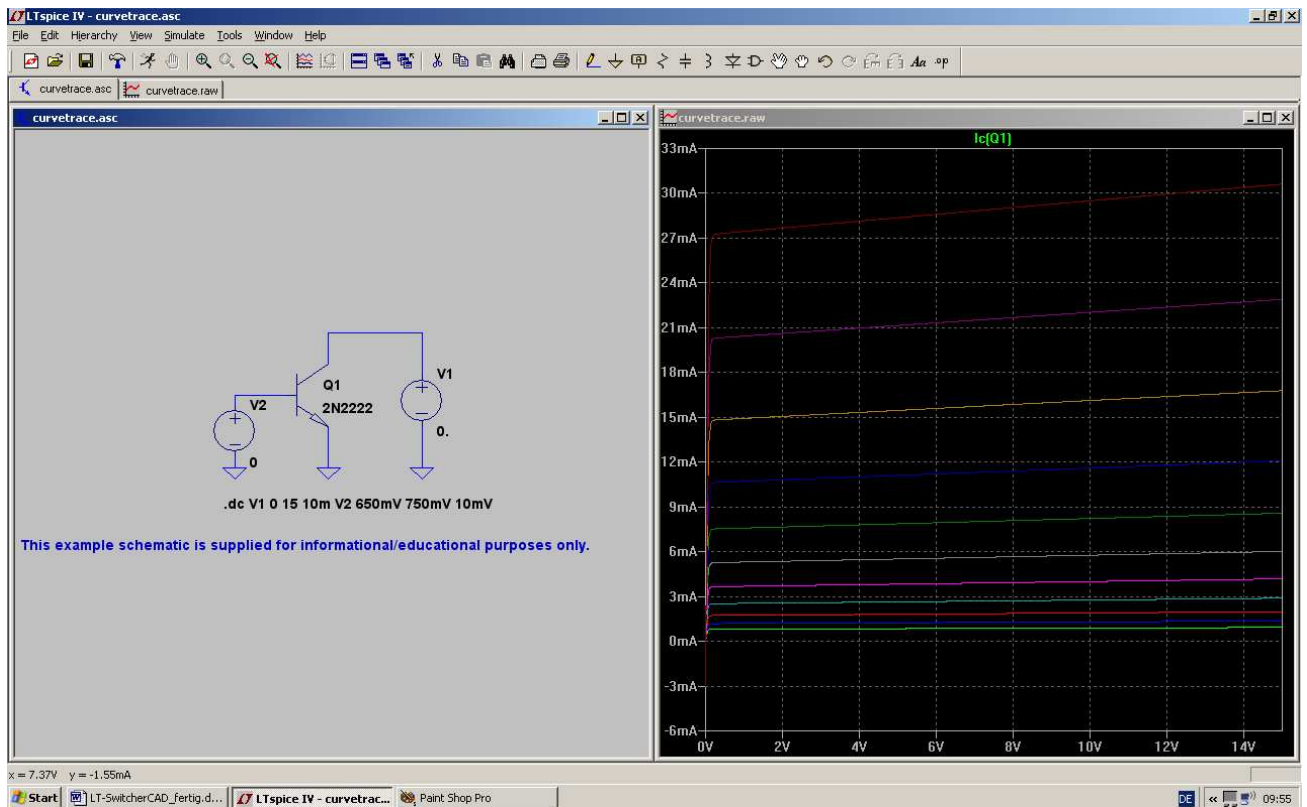
besagt:

Ändere die Spannung V1 von Null Volt bis +15V in Schritten von 10mV. Benütze diese Spannung für die waagrechte Achse des Ergebnisdiagramms (= Abszisse).  
Ändere nun den Strom I1 von Null bis 100 Mikroampere in Schritten von 10 Mikroampere und schreibe alle Kurven in das Ergebnisdiagramm.



Fährt man nun im Schaltplan mit der Maus auf den Kollektor des Symbols, dann wird sich plötzlich der Cursor in eine „Stromzange“ verwandeln. Nun reicht ein linker Mausklick, um das gewünschte Ergebnis zu erhalten.

Wer sich dagegen den Zusammenhang zwischen Basisspannung und Kollektorstrom interessiert, braucht diese Schaltung:



- Die Stromquelle an der Basis wird durch eine Spannungsquelle ersetzt. Ihr Startwert ist „Null“
- Das Simulation Command wird neu verfasst:

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

Die Angaben für die Kollektorspannung V1 bleiben darin unverändert. Die Basis-Spannungsquelle V2 wird in Schritten von 10mV von 650mV auf 750mV geändert.

Ergebnis: Siehe oben...



## 8.4. N-Kanal-Sperrschicht-FET

Gegenüber der Transistor-Simulation des letzten Kapitels ändert sich nur eine Sache: wir müssen jetzt einen solchen FET (Bauteil „njf“ aus der Bibliothek...) verwenden. Da sollte man sich aber die Mühe machen und einen bekannten Typ wählen. Wir einigen uns auf den

### BF245B

Nach ihm sucht man allerdings im mitgelieferten Vorrat vergeblich. (Wissen Sie noch, wie das ging? Erst den FET als „njf“ holen und absetzen. Dann ein rechter Mausklick auf das Schaltzeichen und unter „Pick new JFET“ nachsehen)

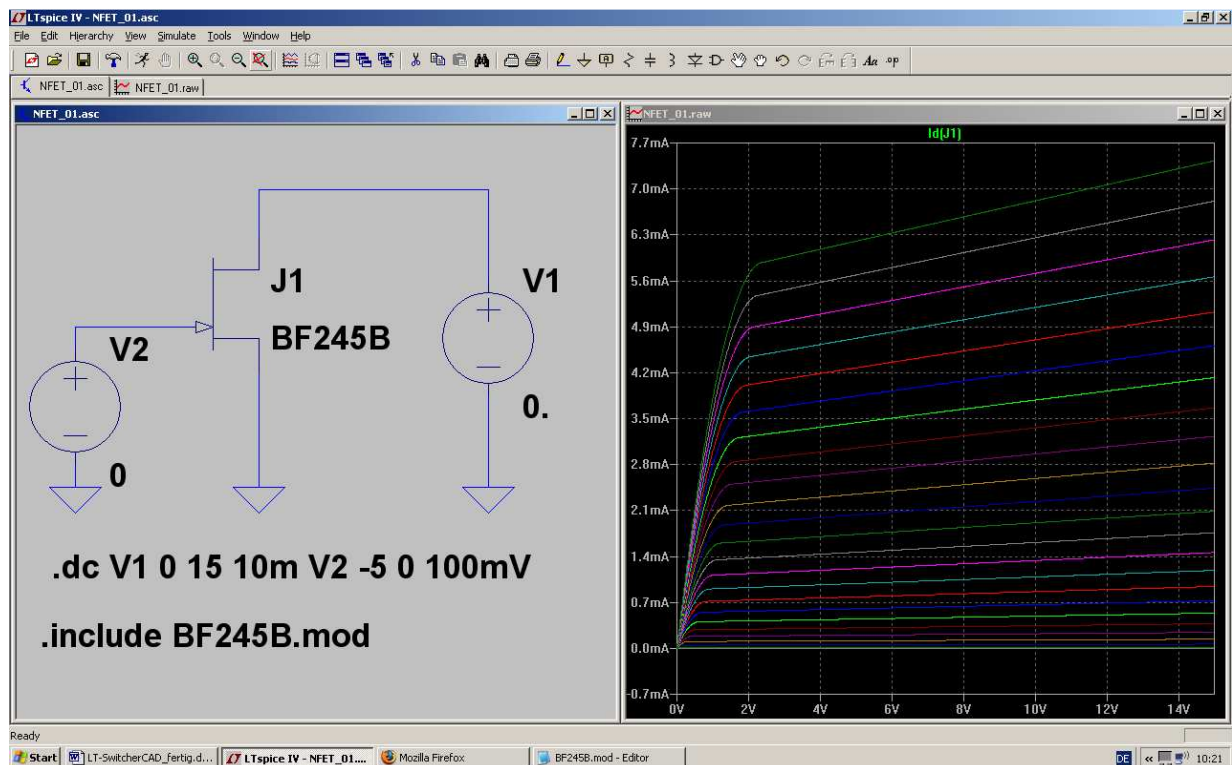
Also geht es ins Internet mit der Such-Eingabe

### bf245b spice model.

Das Ergebnis wird als Text in die Zwischenablage und dann in ein neues Blatt des Texteditors kopiert. Von dort aus kann es endgültig als

### BF245B.mod

in unserer **Bibliothek im Ordner „sub“** abgespeichert werden (...Bitte aber auf die korrekte Endung achten....) Nun sehen wir uns mal an, was noch zu erledigen ist:



Erst wird der FET (= „njf“) in die Schaltung eingebaut und dann mit der rechten Maustaste auf die Modellangabe geklickt (= Zeile „NJF“ rechts unten beim Symbol). Die Modellbezeichnung wird auf **BF245B** abgeändert und dann (über „Edit“) eine **zusätzliche SPICE-Direktive** geschrieben. Sie lautet:

### **.include BF245B.mod**

und damit wird das aus dem Internet stammende Spice-Modell des BF245B dem Simulationsteil zur Verfügung gestellt.

Wenn man nun noch beim Simulation Command den Spannungsbereich von V2 -- wie es sich für einen Sperrschicht-FET bei der Gate-Spannung gehört! -- auf

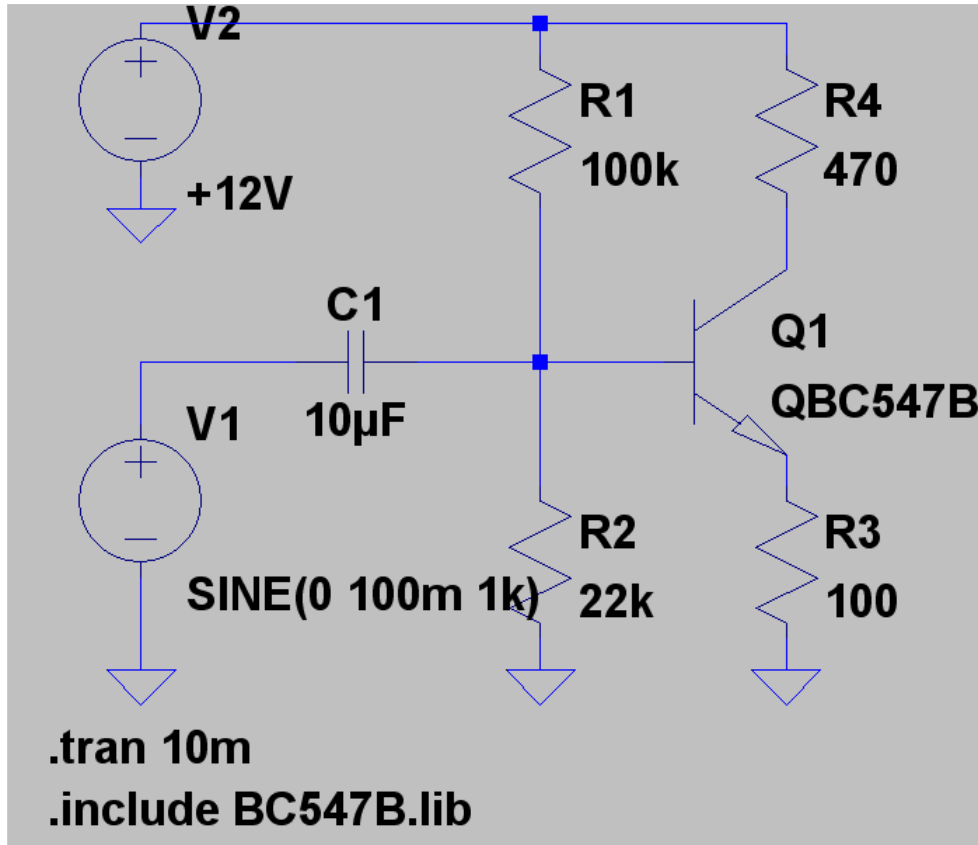
### **-5V bis 0V in Schritten von 100mV**

einstellt, dann sollte alles klappen.

## 9. Fünftes Projekt: Schaltungen mit Transistoren

### 9.1. Einstufiger Verstärker

#### 9.1.1. Ansteuerung mit einem Sinus-Signal



Dazu brauchen wir eine Speisespannung, eine Signalquelle, einen Kondensator, vier Widerstände und einen npn-Transistor.

Werfen wir erst mal einen Blick auf die fertige Schaltung und dann schauen wir, wie wir das hinkriegen.

#### Anleitung:

##### 1. Schritt:

4 Widerstände, 1 Kondensator, 2 Spannungsquellen („voltage“) und 1 npn-Transistor werden platziert und verdrahtet.

##### 2. Schritt:

Den Widerständen und dem Kondensator werden die korrekten Werte zugewiesen.

Die obere Spannungsquelle V2 ist vom Typ „DC“ und liefert +12V.

Die untere Spannungsquelle V1 erzeugt einen Sinus mit 100mV Spitzenwert und der Frequenz  $f = 1\text{kHz}$ .

##### 3. Schritt:

Es wird ein Simulationszeitraum von 0...10 Millisekunden vorgesehen.

##### 4. Schritt:

Im Internet suchen wir nach „**BC547B.lib**“. Der Inhalt wird markiert, in die Zwischenablage kopiert, in ein neues Blatt eines Texteditors eingefügt und schließlich als „**BC547B.lib**“ im Ordner

**LTC / LTSpiceIV / lib / sub**

gespeichert

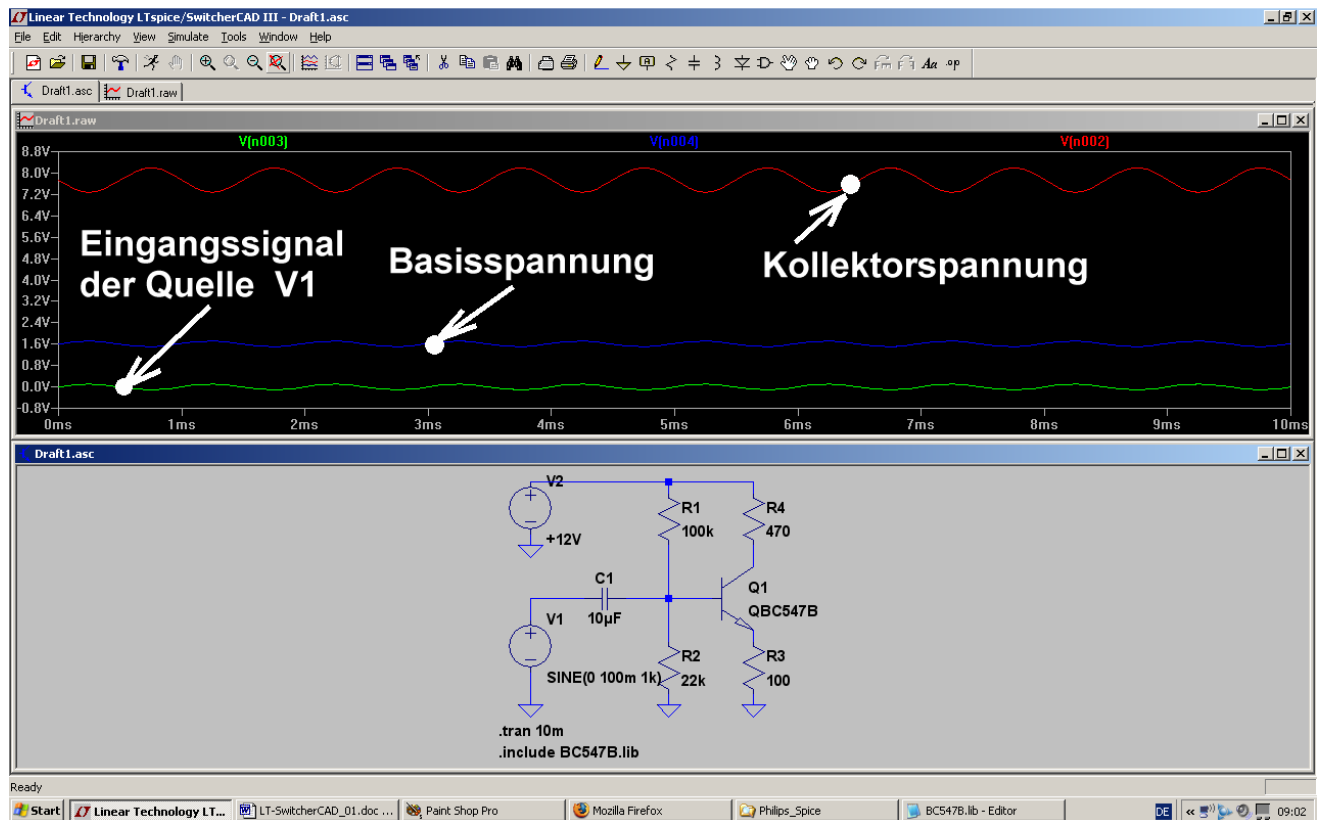
##### 5. Schritt:

Wir stellen den Cursor auf die untere Angabe „**NPN**“ beim Transistorsymbol, klicken rechts, tragen die nun gültige Bezeichnung „**QBC547B**“ ein und schreiben noch das Spice-Command **.include BC547B.lib**. Fertig!

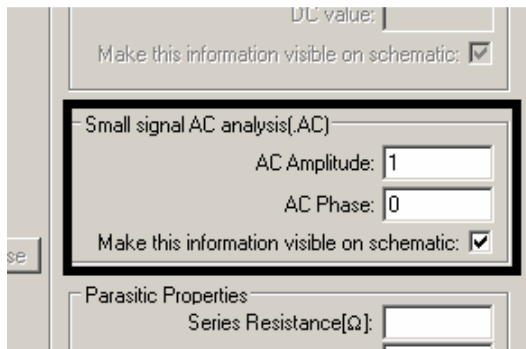
Beim Simulationsergebnis schauen wir uns die Eingangsspannung, die Basisspannung des Transistors und seine Kollektorspannung an. Besonders die Phasenumkehr im Transistor können wir nun sehr schön sehen. Ausserdem lässt sich nun die Behauptung der Theorie, das die Verstärkung der Stufe etwa den Wert

$$VU = RC / RE = 470 / 100 = 4,7 \text{ fach}$$

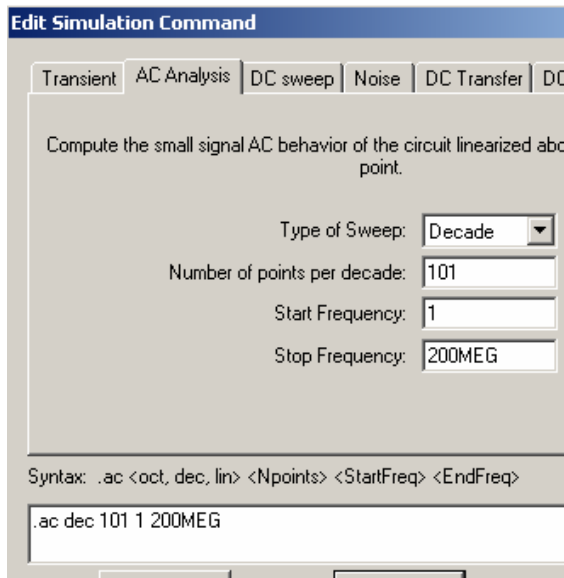
haben, leicht überprüfen. (Dazu braucht man nur die Ausgangsspannung alleine darstellen und ihren Spitzenwert bestimmen. Angesteuert wird die Stufe ja mit einem Spitzenwert von 100mV)



## 9.1.2. Simulation des Frequenzganges („AC-Sweep“)

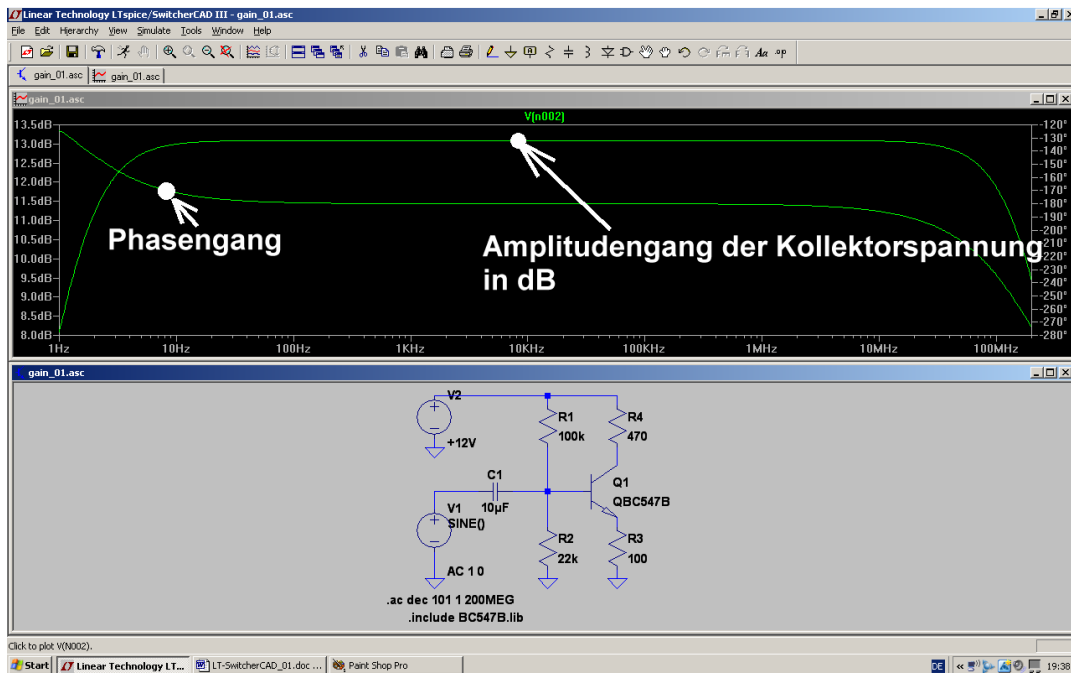


**1. Schritt:**  
Wir stellen die Spannungsquelle V1 auf „Sweep-Betrieb“ um.



**2. Schritt:**  
Über „Simulate“ und „Edit Simulation Command“ kommen wir an das nebenstehende Menü heran. Auf der Karteikarte „AC Analysis“ programmieren wir einen dekadischen Sweep von 1Hz bis 200MHz und wählen 101 Punkte pro Dekade.

So sollte das Simulationsergebnis schließlich aussehen.



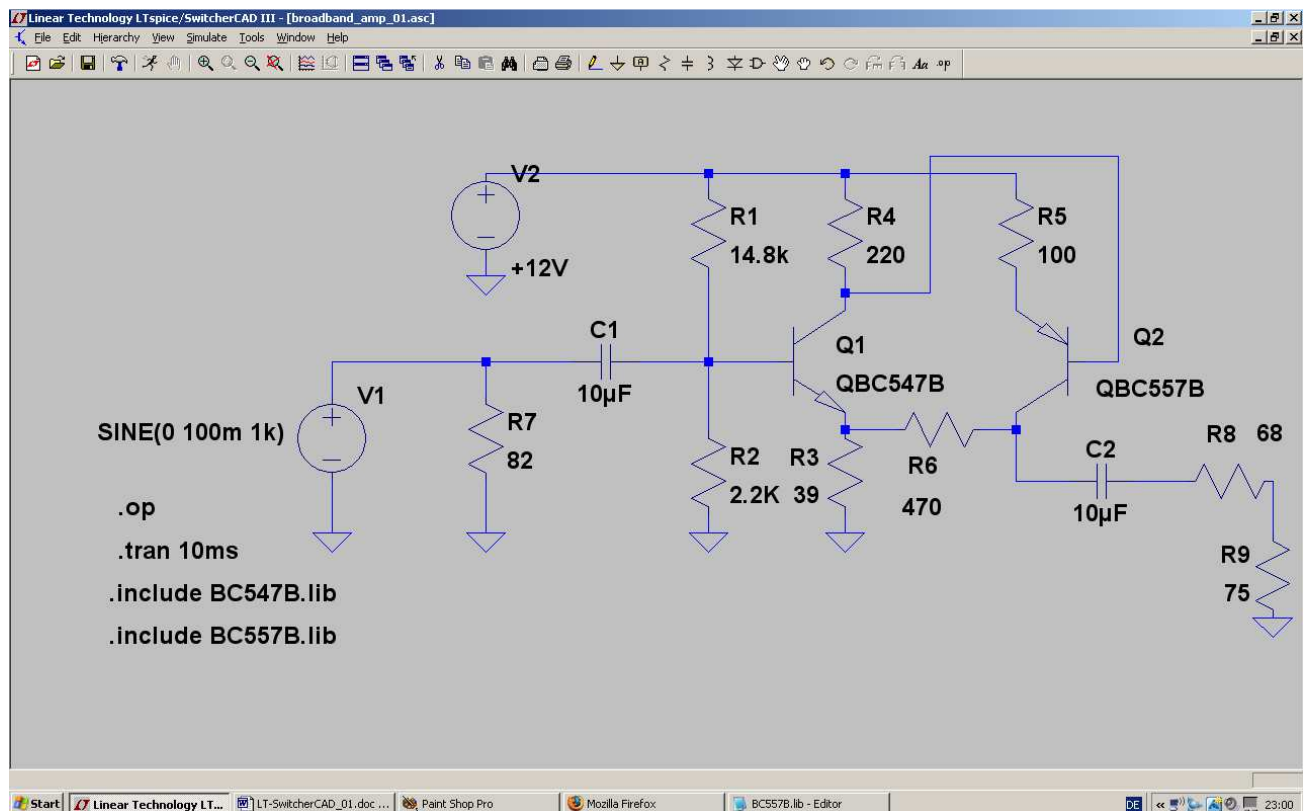
## 9.2. Zweistufiger gegengekoppelter Breitbandverstärker

### 9.2.1. Pflichtenheft

Es soll ein „Gainblock“ für ein 75 Ohm – Kommunikationssystem entwickelt werden. Das bedeutet:

- Eingangs- und Innenwiderstand sollen je 75 Ohm betragen, damit beim Zusammenschalten mehrerer Bausteine immer Leistungsanpassung herrscht und dazwischen 75 Ohm – Koaxialkabel (Kabeltyp: RG59) verwendet werden können.
- Wird der Ausgang mit 75 Ohm belastet, dann soll die sich dort einstellende Ausgangsspannung um den Faktor 2 (das entspricht 6 dB) größer sein als die Eingangsspannung.
- Als Versorgungsspannung ist +12V vorzusehen.
- Die Verstärkung soll im Frequenzbereich von 1 kHz bis 10 MHz konstant sein.
- Es wird eine zweistufige Verstärkerschaltung mit npn- und pnp-Transistor (= BC547B und BC557B) sowie Gegenkopplung eingesetzt.
- Zum Ausgleich von Exemplarstreuungen und zur Einstellung des richtigen Arbeitspunktes soll ein Einstellpotentiometer im Basiskreis des ersten Transistors vorhanden sein.

### 9.2.2. Simulations-Schaltung und Simulations-Vorgaben



Die obige Schaltung wird zuerst gezeichnet, wobei eine **Sache ganz wichtig ist**:

Bei den Widerständen R1 = 14,8k und R2 = 2,2k dürfen wir nur folgende Eingabeform **OHNE KOMMA IN DER WERTANGABE** benutzen, da sonst mit falschen Werten -- ohne Warnung!! -- simuliert wird!

**14.8k** und **2.2k**

Für den **npn-Transistor BC557B** ist natürlich wieder eine Google-Suche im Internet mit anschließender Speicherprozedur erforderlich ist (= HTML-File markieren, Inhalt in das Clipboard kopieren, Texteditor öffnen und Clipboard hineinkopieren, schließlich das File mit korrekter Endung im korrekten Ordner ablegen.....)

Bei beiden Transistoren ist natürlich jeweils die korrekte Namensänderung samt zugehöriger Spice-Direktive im Schaltplan erforderlich.

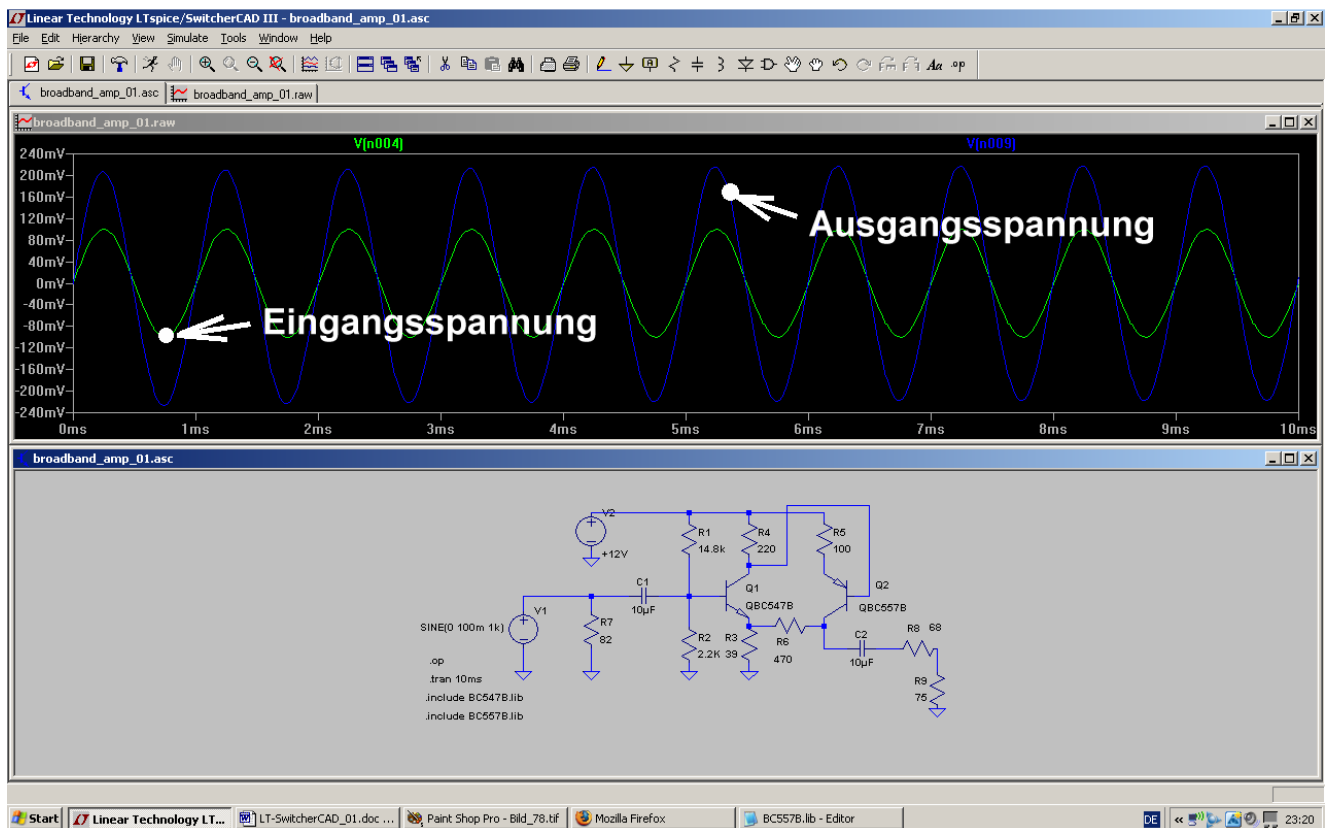
Angesteuert wird die Schaltung im ersten Durchgang mit einer 1kHz-Sinusspannung, die einen Spitzenwert von 100mV aufweist. Simuliert wird von 0...10Millisekunden.

Neu ist die Simulationsanweisung **„.op“**

Sie sorgt dafür, dass uns der DC-Arbeitspunkt („Operation point“) zugänglich gemacht wird -- und zwar in Form einer Liste, die alle vorkommenden Gleichspannungen und Gleichströme des Ruhezustandes enthält.

Gemessen wird anschließend die Eingangsspannung der Quelle V1 und die Ausgangsspannung am Abschlusswiderstand R9 (75Ω)

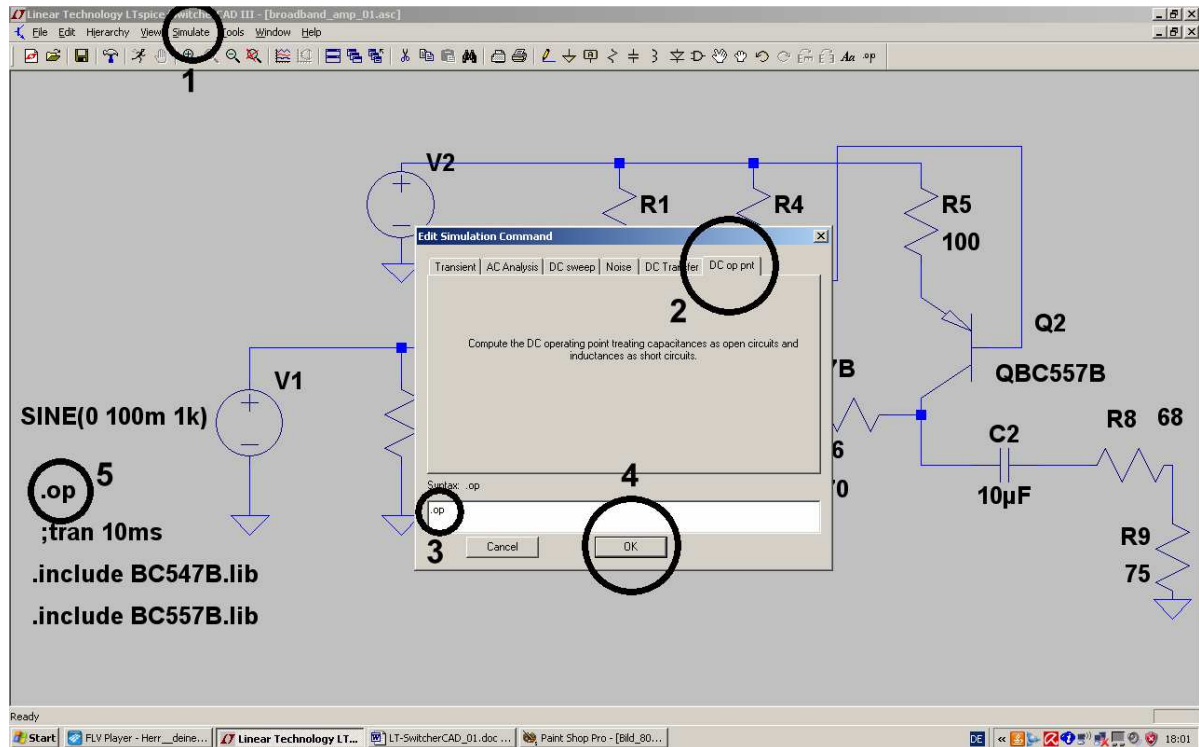
### 9.2.3. Simulation in der Time Domain (= im Zeitbereich)



Man sieht, dass tatsächlich die Ausgangsspannung etwa doppelt so groß ist wie die Eingangsspannung (was einer Spannungsverstärkung von 6dB entspricht). Und zwei Emitterstufen bringen doppelte Phasenumkehr -- also muss das Ausgangssignal wieder gleichphasig mit dem Eingangssignal sein.....

## 9.2.4. DC-Bias (= Gleichstrom-Analyse)

Wenn man sich für die Ruhestrome und –Spannungen im gewählten Arbeitspunkt interessiert, dann geht man so vor:



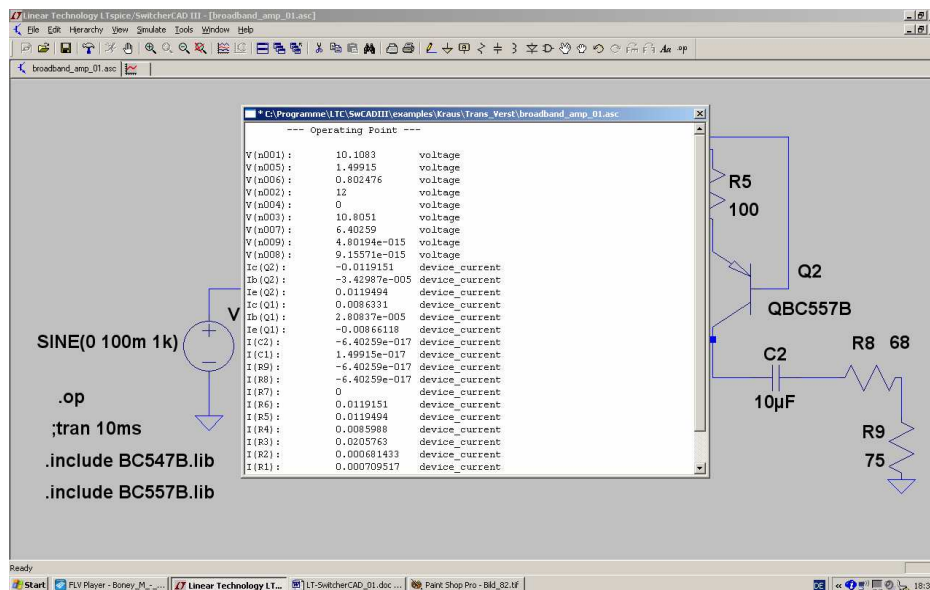
1): Das Menü „Simulation“ wird geöffnet und darin „Edit Simulation Command“ angeklickt.

2): Dann wechseln wir auf die letzte Karteikarte „DC op point“

3). Im Eingabefenster muss nun „.op“ stehen.

4): Das bestätigen wir mit OK und prüfen, ob

5) das Kommando „.op“ korrekt am Cursor hängt und im Simulationsschaltbild abgesetzt werden kann. (**Wichtige Kontrolle: vor dem Kommando „tran 10ms“ muss nun ein Strickpunkt stehen**).



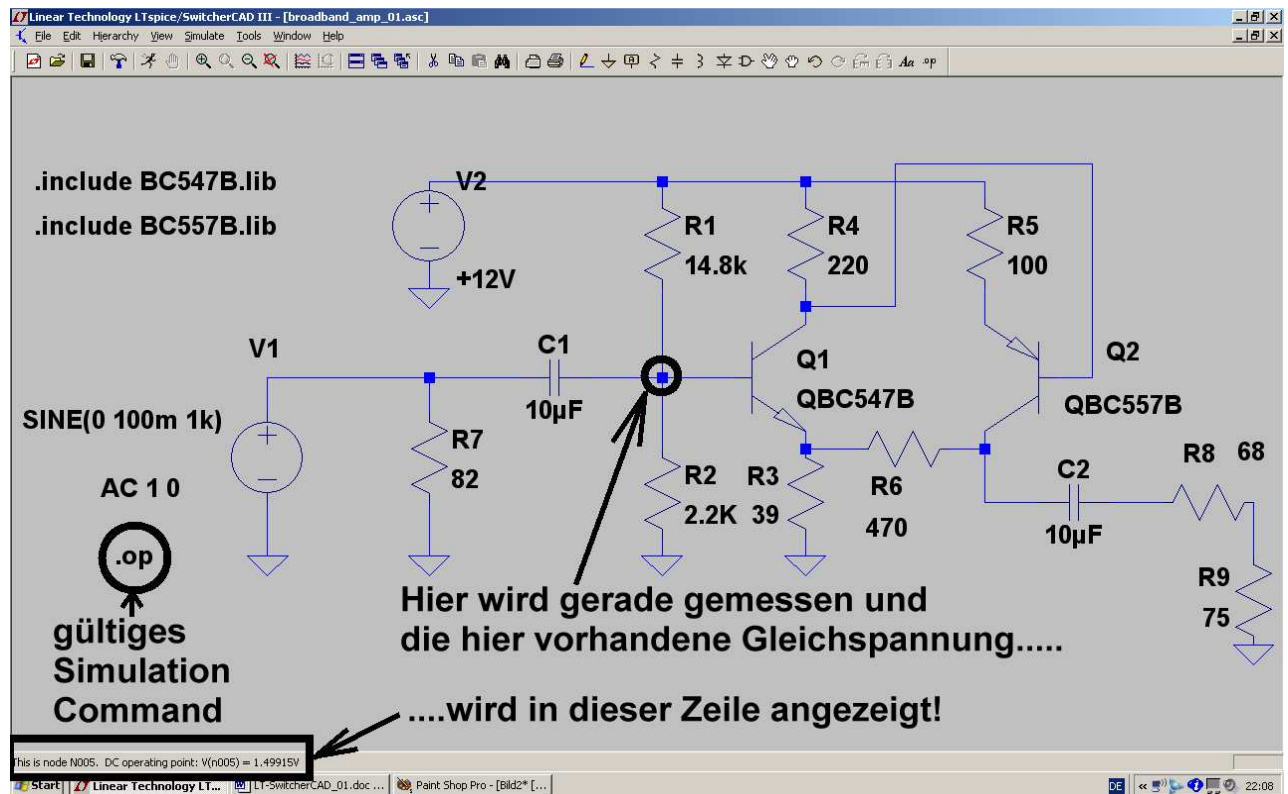
Jetzt dürfen wir wieder mal auf das „rennende Männchen“ (= Simulationsbutton) drücken und auf das Ergebnis warten.

So sieht es aus.

Die Tabelle wird nach Betrachtung wieder geschlossen und dadurch das komplette Schaltbild erneut sichtbar gemacht.

Da gibt es nämlich noch eine weitere hübsche Möglichkeit:

Man kann sich nämlich nun jede einzelne Spannung (oder jeden einzelnen Strom...) direkt anzeigen lassen!



**Wichtig:**

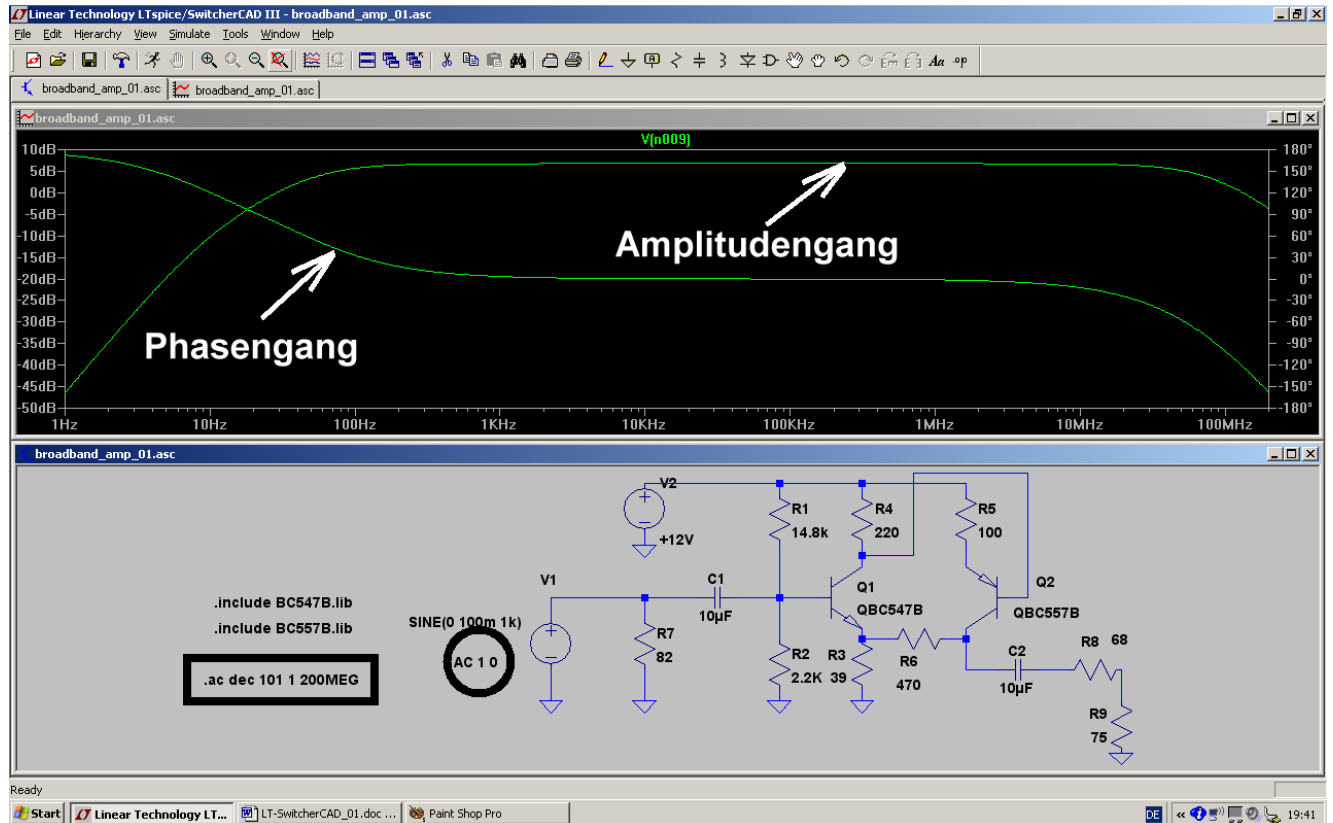
Möchte man nun wieder zur Transientensimulation (= Ansteuerung mit einer Sinusspannung) zurück, dann muss vor „op“ ein Punkt und umgekehrt vor „trans...“ ein Strichpunkt sitzen! Am schnellsten geht das wieder über „Simulation / Edit Simulation Command / Karteikarte „Transient“ und dort einem Klick auf OK.



## 9.2.5. AC-Sweep (= Frequenzgang von 1 Hz bis 200 MHz)

Das sollte nun wirklich schnell gehen:

Bei der Spannungsquelle unter „Small Signal AC Analysis“ die beiden Einträge („1“ für die AC-Amplitude, „0“ für die AC-Phase) vornehmen und ein neues Simulation Command für den AC-Sweep von 1Hz bis 200MHz mit 101 Punkten pro Dekade erstellen.



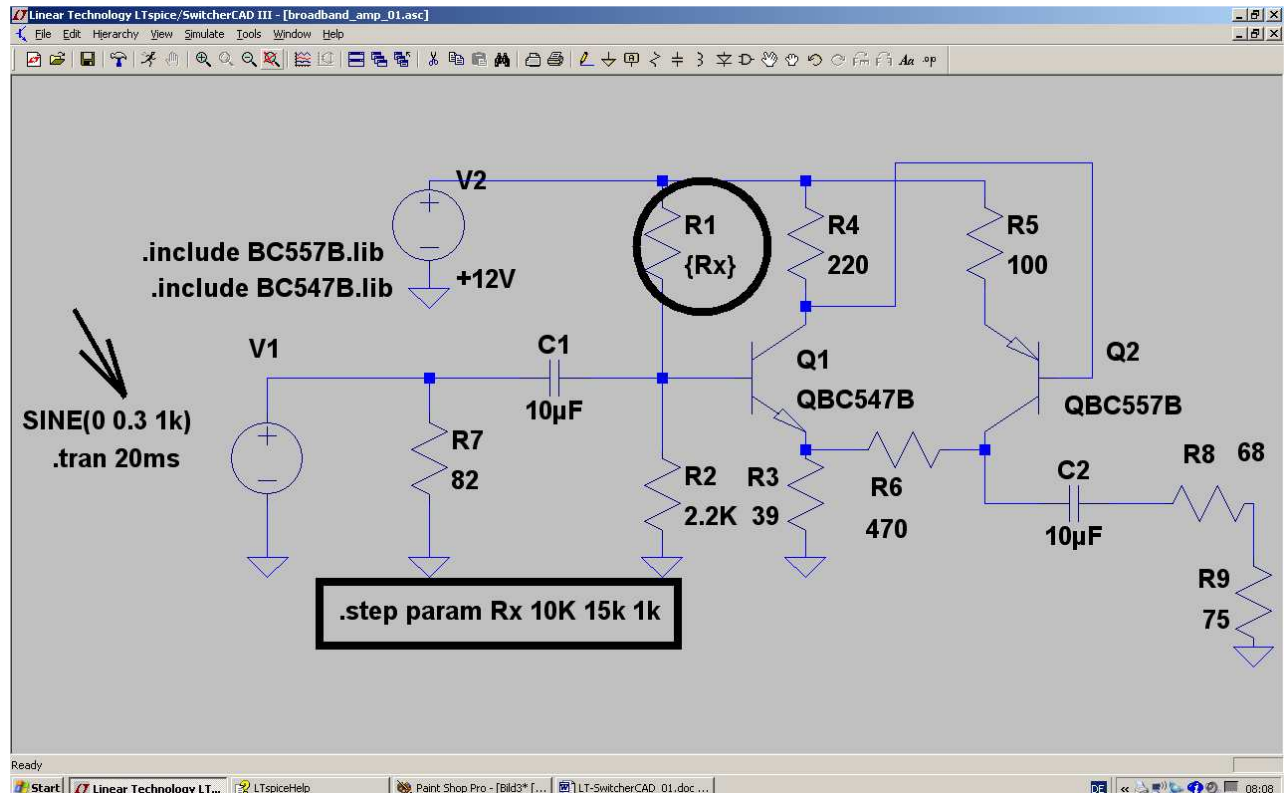
### 9.3. Der Parameter-Sweep

Wer in der Online-Hilfe mal unter „LT SPICE / Dot Commands“ nachschaut, findet eine sehr lange Liste an zusätzlichen Möglichkeiten und Befehlen.

Wir wollen uns daraus das Kommando **.step** etwas näher ansehen.

Das ist eine ganz wichtige Sache, denn damit können wir das „**Ausprobieren von verschiedenen Werten**“ auf **eine wesentlich exaktere und bequemere Grundlage stellen** -- egal ob das die Eingangsspannung, die Speisespannung oder einen Bauteil- oder Modellwert innerhalb der Schaltung betrifft.

Wir nehmen als Beispiel nochmals **Kapitel 7.2.3.** her und **lassen uns den oberen Basis-Spannungsteilerwiderstand R1 in 5 Schritten von 10k bis 15k variieren.** Die Ergebnisse können dann direkt verglichen werden und so hat man sehr schnell die korrekte Potentiometer-Einstellung.

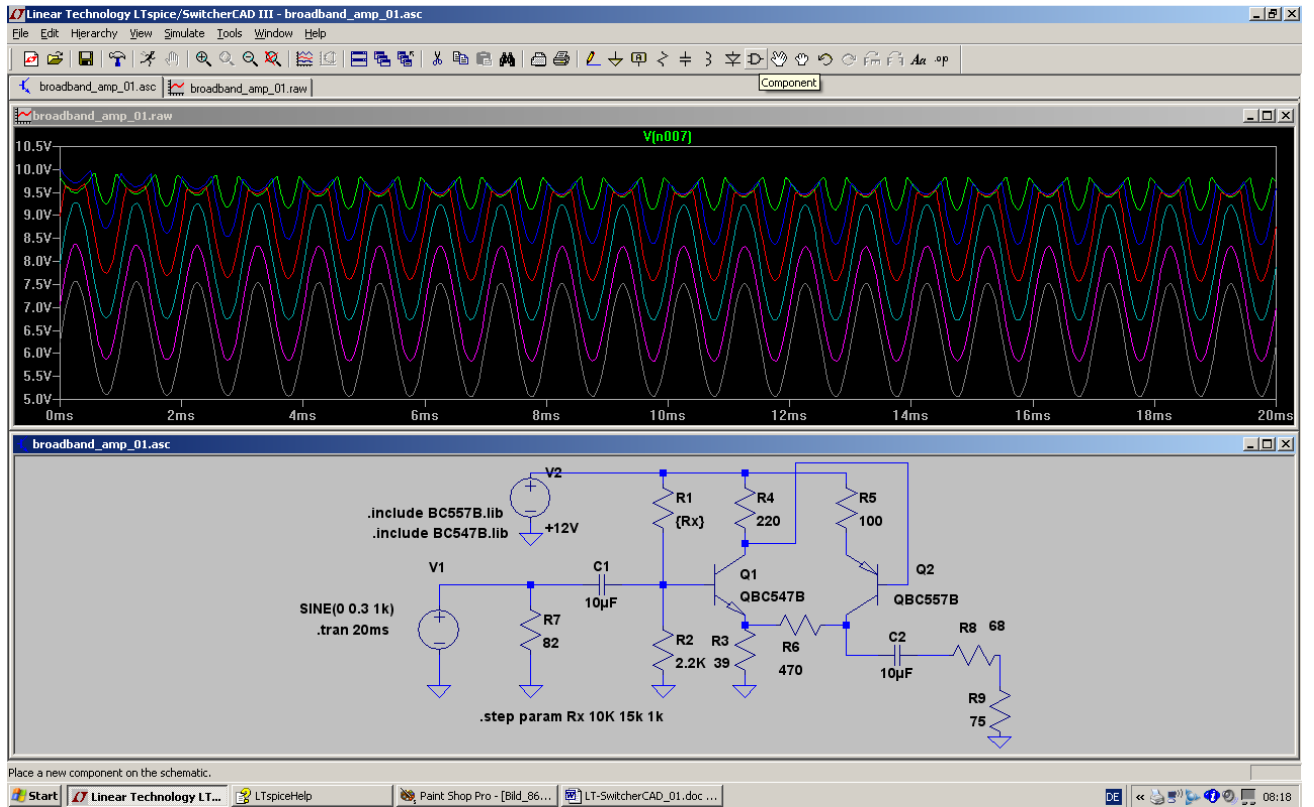


Bitte genau hinschauen:

- Die Amplitude der Eingangsspannung wurde auf 0,3V erhöht
- Beim Widerstand R1 wurde die Wertangabe durch den Parameter **{Rx}** ersetzt
- Schließlich ist noch die Spice-Direktive **.step param Rx 10k 15k 1k** erforderlich.  
Dadurch wird R1 in Schritten von 1 Kilo-Ohm von 10k bis 15k verändert.

Drückt man nun die „Taste mit dem Männchen“, dann wird alles simuliert, aber noch nichts im Ergebnisdiagramm ausgegeben. Dazu müssen wir z. B. erst an den Kollektor von Transistor Q2 tippen, um die dortige Spannung beobachten zu können.

So sieht das Ergebnis dann aus und man kann nun leicht abschätzen, dass ein Wert von etwa 15k bei R1 (= unterste Kurve) erforderlich ist, um eine mittlere Kollektorgleichspannung von ca. 6,5V zu erhalten.

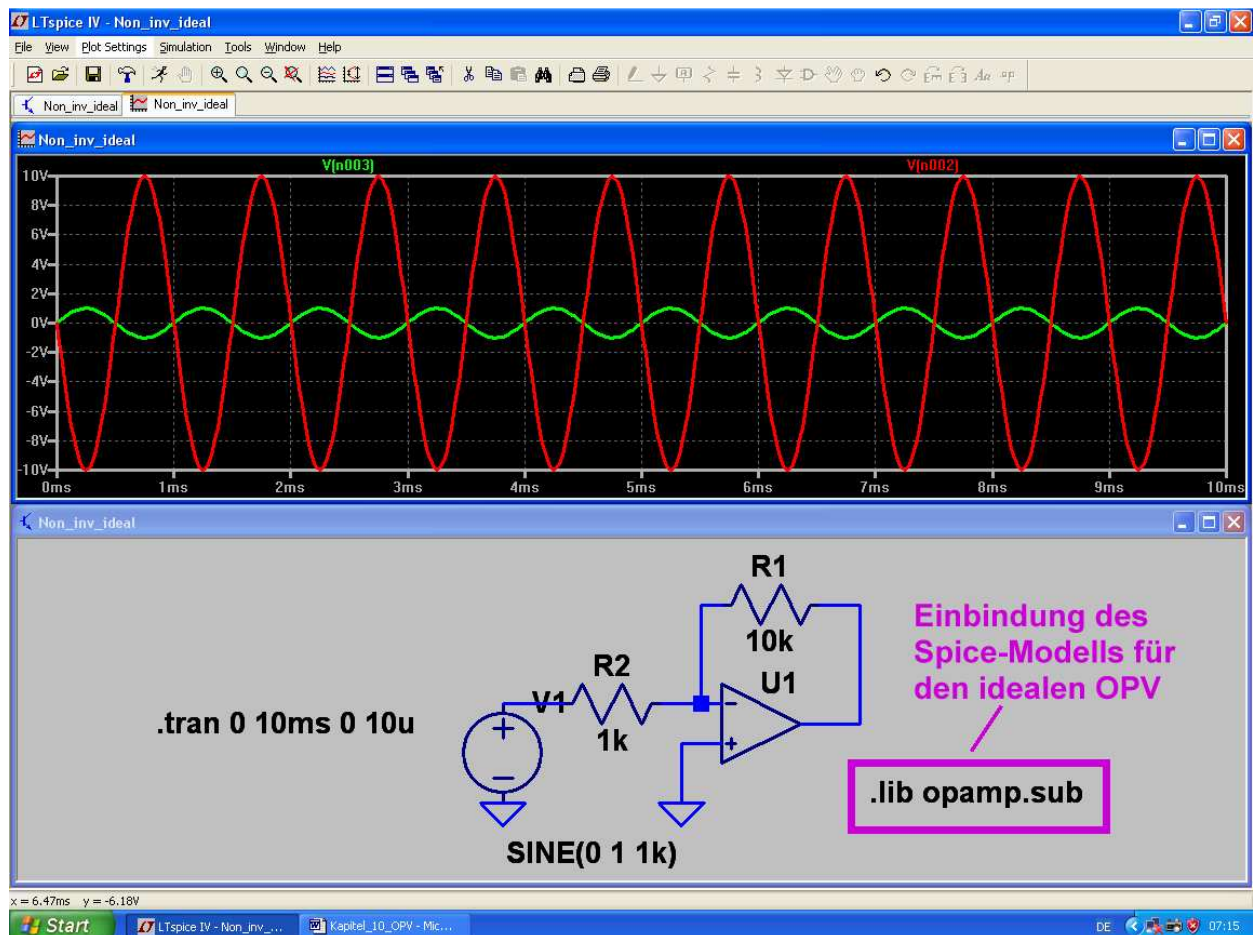


## 10. Sechstes Projekt: OPV-Schaltungen

### 10.1. Einstieg: Umkehrender Verstärker

#### 10.1.1. Verwendung des einfachsten OPV-Modells „opamp“

Will man nur wissen, ob eine eigene Idee grundsätzlich funktioniert, dann reicht die Simulation mit einem idealen OPV-Modell ohne Versorgungsspannungen. Wir finden es in der Bibliothek im Ordner „Opamps“ als „opamp“ und starten damit ein neues Projekt, nämlich einen umkehrenden Verstärker mit 10facher Verstärkung. Gespeist wird der Eingang mit einer Sinusspannung (Spitzenwert = 1 V / Frequenz = 1 kHz). Allerdings darf man die **Spice-Direktive zur Einbindung der Modelldatei** nicht vergessen:

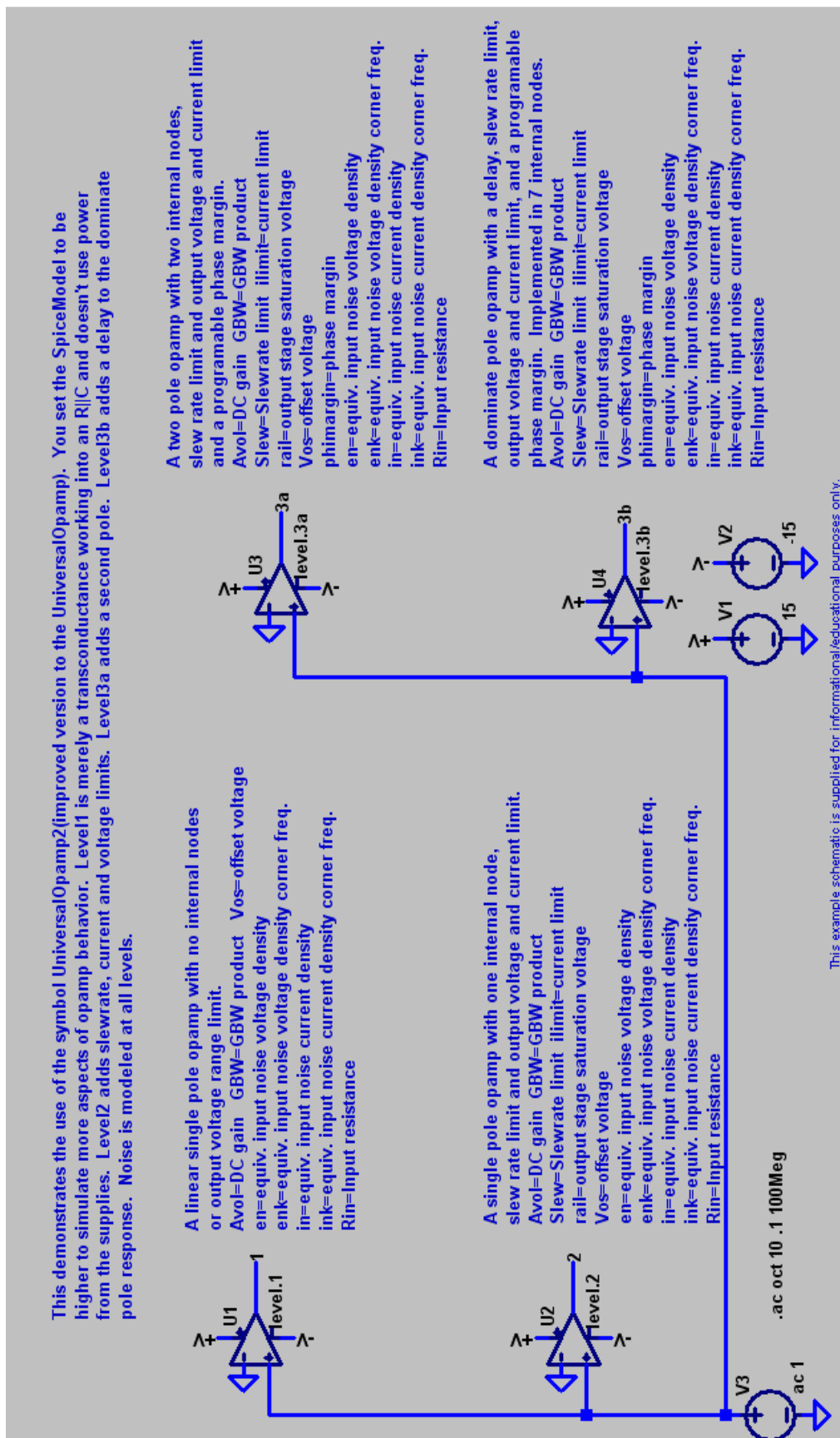


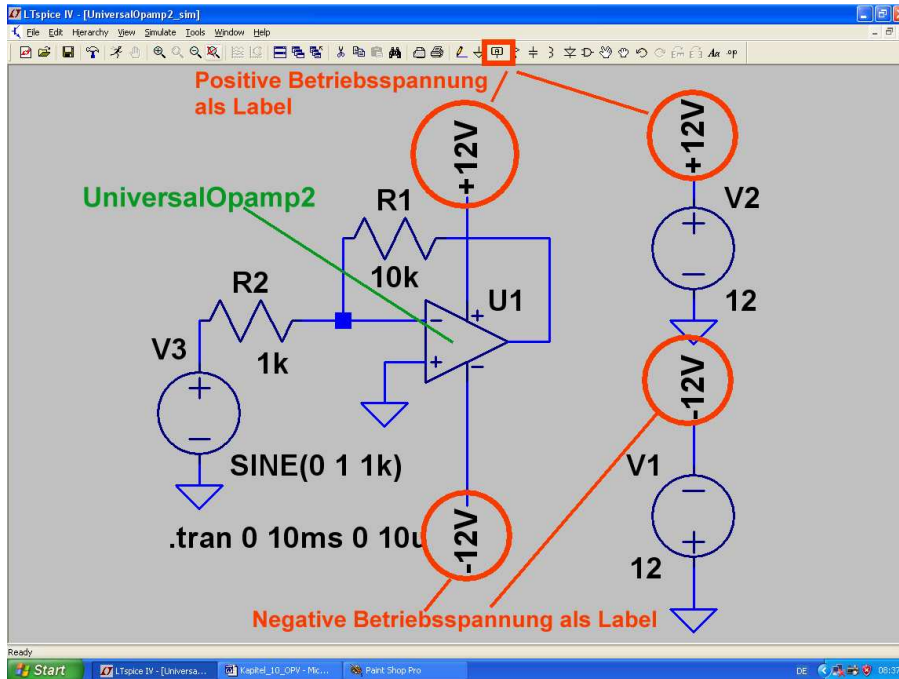
#### Hinweise:

- Die Ergebniskurven im oberen Diagramm wurden mit **dicken Linien** dargestellt. Dazu klickt man auf den „**Button mit dem Hammer**“ in der Menüleiste, wählt die Karteikarte „**Waveforms**“ und aktiviert darin die erste Zeile „**Plot data with thick lines**“ mit einem Häkchen.
- Möchte man die **Farbe** einer Kurve verändern, dann klickt man mit der **linken Maustaste** auf die Bezeichnung dieser Kurve (oberhalb des Diagramms). Im auftauchenden Menü wird eine Palette mit 16 Farben zur Auswahl angeboten.
- Möchte man das **Schaltbild mit etwas kräftigeren Linien** zeichnen, dann klickt man wieder auf den „**Button mit dem Hammer**“. Auf der Karteikarte „**Drafting Options**“ muss man dann die dritte Zeile von unten („**Draft with thick lines**“) aktivieren.

## 10.1.2. Einsatz des Universalen OPVs und die Verwendung von Labels

Sobald die Ansprüche an die Schaltung steigen und noch keine Entscheidung für einen bestimmten OPV-Typ gefallen ist, kann man den „UniversalOpamp2“ einsetzen. Er bietet vier verschiedene Qualitätsstufen und die sind im mitgelieferten Beispiel „**UniversalOpamp2.asc**“ genau beschrieben:

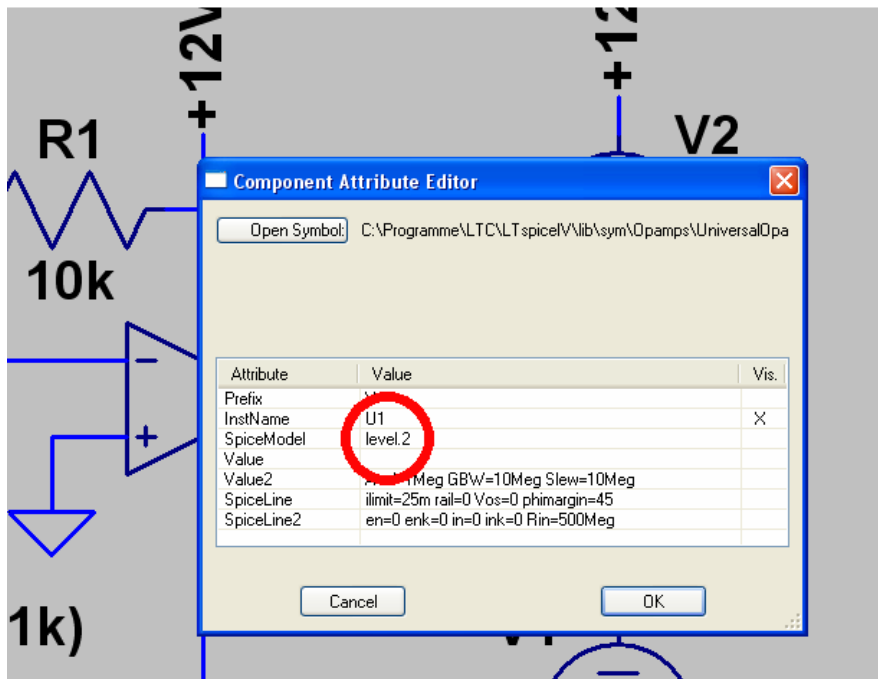




So sieht die Schaltung aus, wenn man in der vorigen Schaltung einen „UniversalOpamp2“ einsetzt. Dabei wurden die beiden Speisespannungsquellen mit „Labels“ versehen, um das Schaltbild übersichtlicher zu gestalten.

#### Hinweis:

Die Bezeichnung „UniversalOpamp2“ sagt aus, dass das Spice-Modell für „Level 2“ aktiv ist. Die Level-Änderung ist sehr einfach:



Man klickt mit der rechten Maustaste auf das OPV-Schaltzeichen. Dadurch öffnet sich das Property-Menü und man kann an der markierten Stelle den gewünschten Level (level.1 / level.2 / level.3a / level.3b) eintragen

#### Anmerkung:

Die Auswirkungen der verschiedenen Levels sieht man natürlich nicht in unserem Beispiel bei der Frequenz  $f = 1\text{kHz}$ .

Sobald man jedoch einen AC-Sweep zur Ermittlung des Frequenzgangs programmiert und sich dabei den Phasenverlauf mit einblenden lässt oder die Schaltung mit einem Pulssignal mit höherer Pulsfrequenz (z. B.  $f = 100\text{kHz}$ ) ansteuert, zeigen sich die Unterschiede.

### 10.1.3. Verwendung eines OPVs vom Typ „TL072“ und Erstellung eines passenden Spice-Symbols

Beim TL072 handelt es sich um einen sogenannten „Wald- und Wiesentyp“: oft verwendet, leicht beschaffbar und nicht teuer. Da jedoch bei LTSpice nur die Modelle der hauseigenen Typen mitgeliefert werden, wollen wir uns mal Schritt für Schritt ansehen, wie man den TL072 für den Einsatz in diesem Programm vorbereitet.

#### 1. Schritt:

Man gibt in die Internet-Suchmaschine den Begriff „**TL072 spice model**“ ein und lädt sich dann die entsprechende Datei z. B. vom Original-Entwickler (Texas Instruments) auf den Rechner.

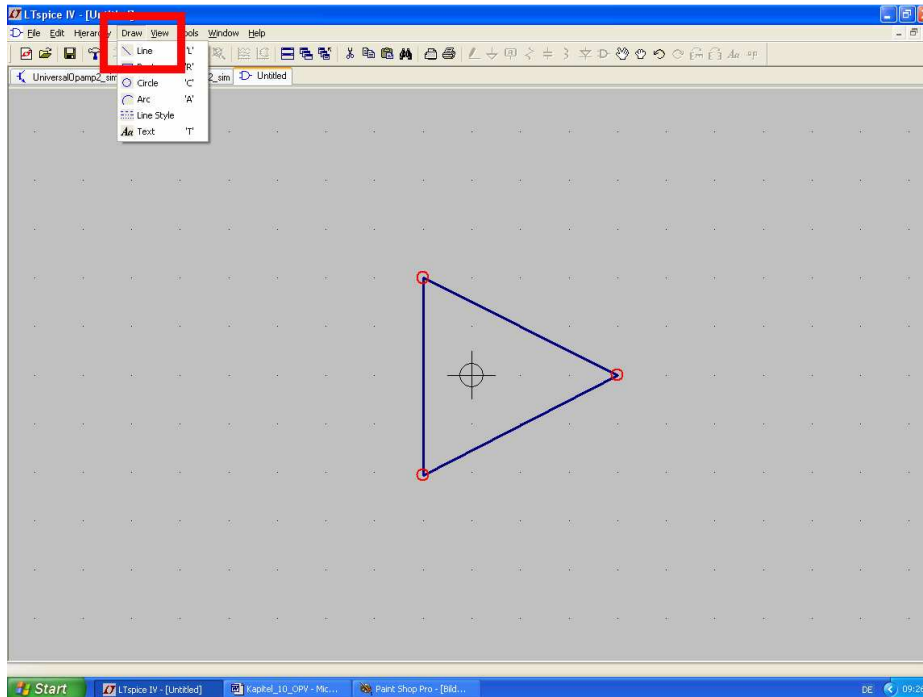
#### 2. Schritt:

Das Dokument wird geöffnet, der Text der Modellbeschreibung markiert und dieser Text in die Zwischenablage kopiert. Nun ruft man einen einfachen Texteditor (z. B. notepad) auf und kopiert die Zwischenablage in ein leeres Blatt des Editors. Das entstehende File wird anschließend (mit dem Dateityp „**Alle Dateien**“) unter der Bezeichnung „**TL072.sub**“ im Ordner „**LTSpice / lib / sub**“ abgelegt. So sieht das am Ende aus:

```
* TL072 OPERATIONAL AMPLIFIER "MACROMODEL" SUBCIRCUIT
* CREATED USING PARTS RELEASE 4.01 ON 06/16/89 AT 13:08
* (REV N/A) SUPPLY VOLTAGE: +/-15V
* CONNECTIONS: NON-INVERTING INPUT
*           | INVERTING INPUT
*           || POSITIVE POWER SUPPLY
*           ||| NEGATIVE POWER SUPPLY
*           |||| OUTPUT
*           |||||
.SUBCKT TL072 1 2 3 4 5
*
C1 11 12 3.498E-12
C2 6 7 15.00E-12
DC 5 53 DX
DE 54 5 DX
DLP 90 91 DX
DLN 92 90 DX
DP 4 3 DX
EGND 99 0 POLY(2) (3,0) (4,0) 0 .5 .5
FB 7 99 POLY(5) VB VC VE VLP VLN 0 4.715E6 -5E6 5E6 5E6 -5E6
GA 6 0 11 12 282.8E-6
GCM 0 6 10 99 8.942E-9
ISS 3 10 DC 195.0E-6
HLIM 90 0 VLIM 1K
J1 11 2 10 JX
J2 12 1 10 JX
R2 6 9 100.0E3
RD1 4 11 3.536E3
RD2 4 12 3.536E3
RO1 8 5 150
RO2 7 99 150
RP 3 4 2.143E3
RSS 10 99 1.026E6
VB 9 0 DC 0
VC 3 53 DC 2.200
VE 54 4 DC 2.200
VLIM 7 8 DC 0
VLP 91 0 DC 25
VLN 0 92 DC 25
.MODEL DX D(IS=800.0E-18)
.MODEL JX PJF(IS=15.00E-12 BETA=270.1E-6 VTO=-1)
.ENDS
```

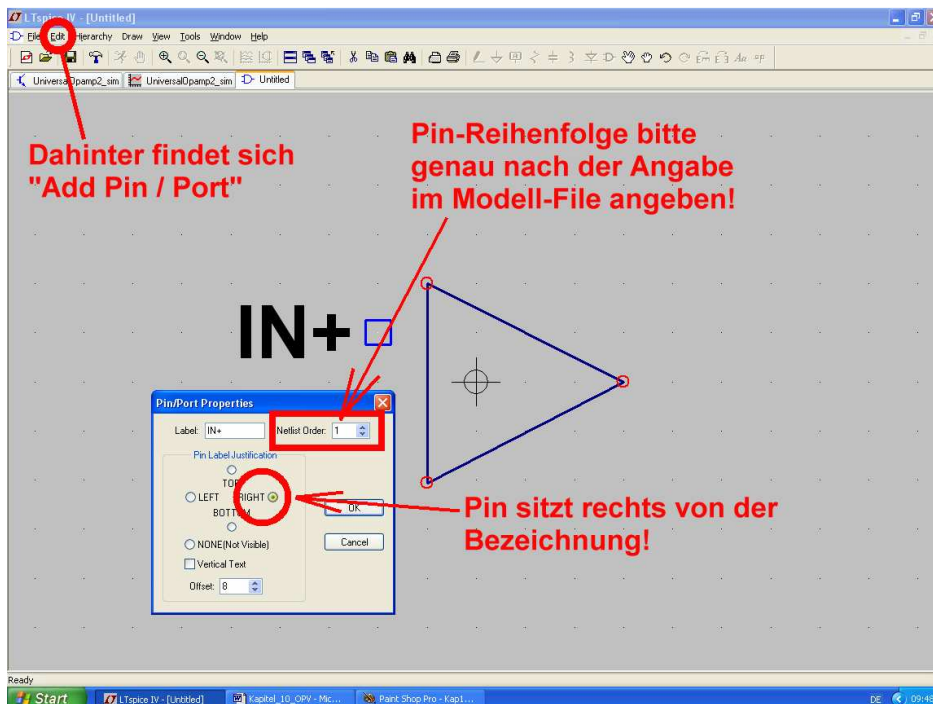
### 3. Schritt:

LTSpice wird gestartet, aber unter „File“ die Option „New Symbol“ angeklickt. Dadurch öffnet sich ein neues Blatt in einem etwas helleren Grau, um es von einem Schaltbild unterscheiden zu können. Über dem Bildschirm liegt ein feines Punktraster und in der Mitte ist das Zentrum für das Symbol durch ein Fadenkreuz markiert.



Unter „Draw“ findet sich mit „Line“ die Möglichkeit zum Zeichnen von Linien. Damit erstellen wir das dreieckförmige OPV-Schaltzeichen.

Die drei Eckpunkte haben dabei einen gegenseitigen Abstand von vier „Kästchen“.



Dahinter findet sich "Add Pin / Port"

Pin-Reihenfolge bitte genau nach der Angabe im Modell-File angeben!

Pin sitzt rechts von der Bezeichnung!

Jetzt werden die Anschlusspins platziert. Der zugehörige Aufruf findet sich hinter „Edit“.

Im nebenstehenden Menü muss man dann auf 2 Dinge achten:

Erstens muss die richtige Pin-Nummer gemäß den Angaben im Modell-File („TL072.sub“) eingetragen werden.

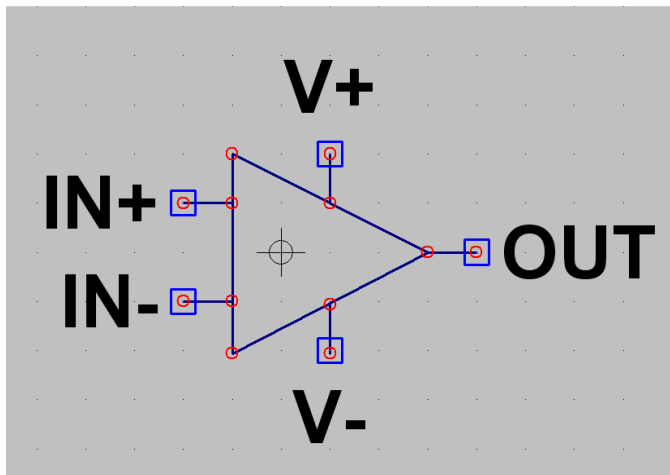
Da gilt für den TL072 folgende Zuordnung:

Pin 1 = Nicht invertierender Eingang =	IN+
Pin2 = Invertierender Eingang =	IN-
Pin 3 = Positive Betriebsspannung =	V+
Pin 4 = Negative Betriebsspannung =	V-
Pin 5 = Ausgang =	OUT

Mit der Taste F7 und darauf folgendem Anklicken des Pinsymbols kann der Pin samt Bezeichnung verschoben werden. Wie immer, wird die Schiebe-Aktion durch „Escape“ beendet.

Zweitens muss man angeben, wo sich der Pin räumlich gegenüber der Bezeichnung befinden soll (hier: rechts).





Sind alle Pins platziert, dann sorgt man im Anschluss wieder mit „**Draw Lines**“ für **Verbindungsleitungen** von jedem Pin zum Schaltzeichen.

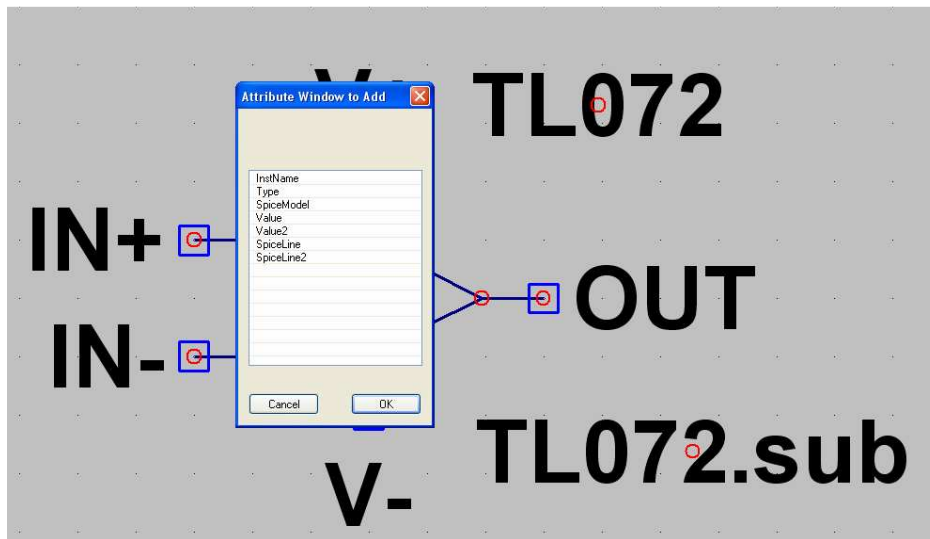
Das ist das Ziel...

Dann werden dem Symbol die einzelnen Eigenschaften zugewiesen. Bitte deshalb die auftauchende Tabelle sehr gewissenhaft ausfüllen:

**Über "Edit" erst "Attributes" und dann "Edit Attributes" aufrufen!**

attribute	value
Prefix	X
SpiceModel	TL072.sub
Value	TL072
Value2	TL072
SpiceLine	
SpiceLine2	
Description	
ModelFile	

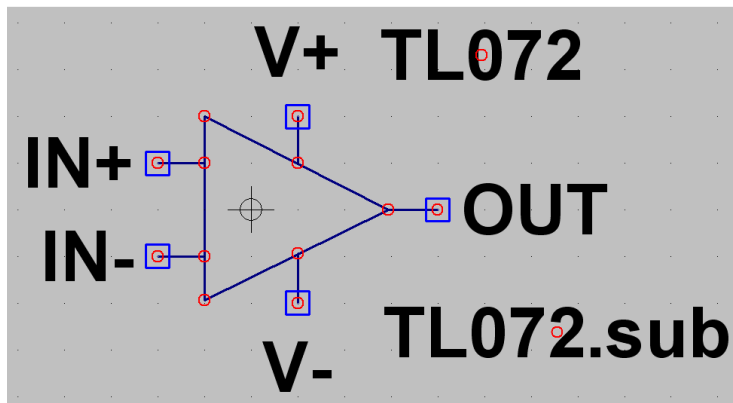
Symbol Type = Cell  
 Prefix = X  
 SpiceModel = 072.sub  
 Value = TL072  
 Value2 = TL072



Nun wiederholt sich das Spiel: über „Edit“ und „Attributes“ kommen wir nun an „Attribute Window“ heran.

In der auftauchenden Liste wird „SpiceModel“ angeklickt und mit OK bestätigt. Die Modellangabe „TL072.sub“ kann nun neben dem Symbol abgesetzt werden.

Dann wiederholt man die Prozedur, wählt aber jetzt „Value“.



Wenn wir diesen Anblick vor uns haben, dann können wir das Symbol speichern.

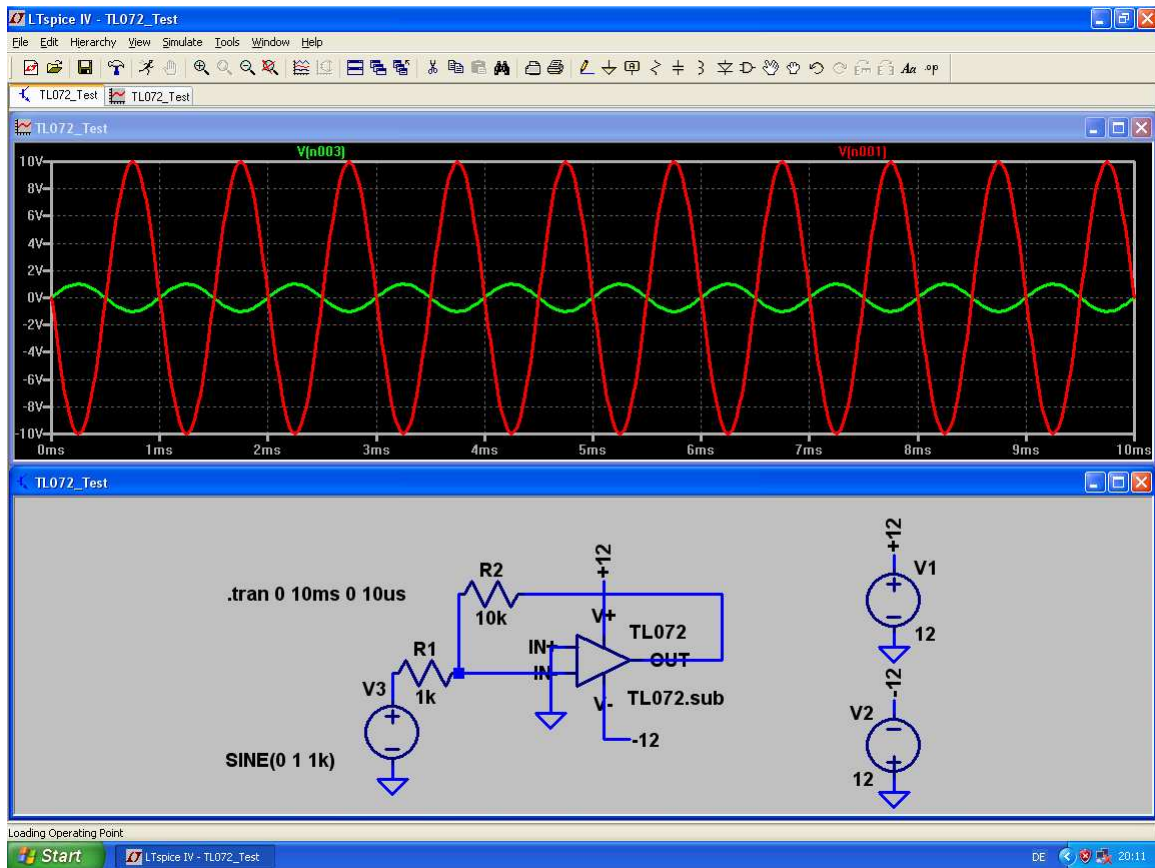
**Achtung:**  
Es muss im Pfad

**LTspice / lib / sym / Opamps“**

unter dem Namen „**TL072**“ abgelegt werden.

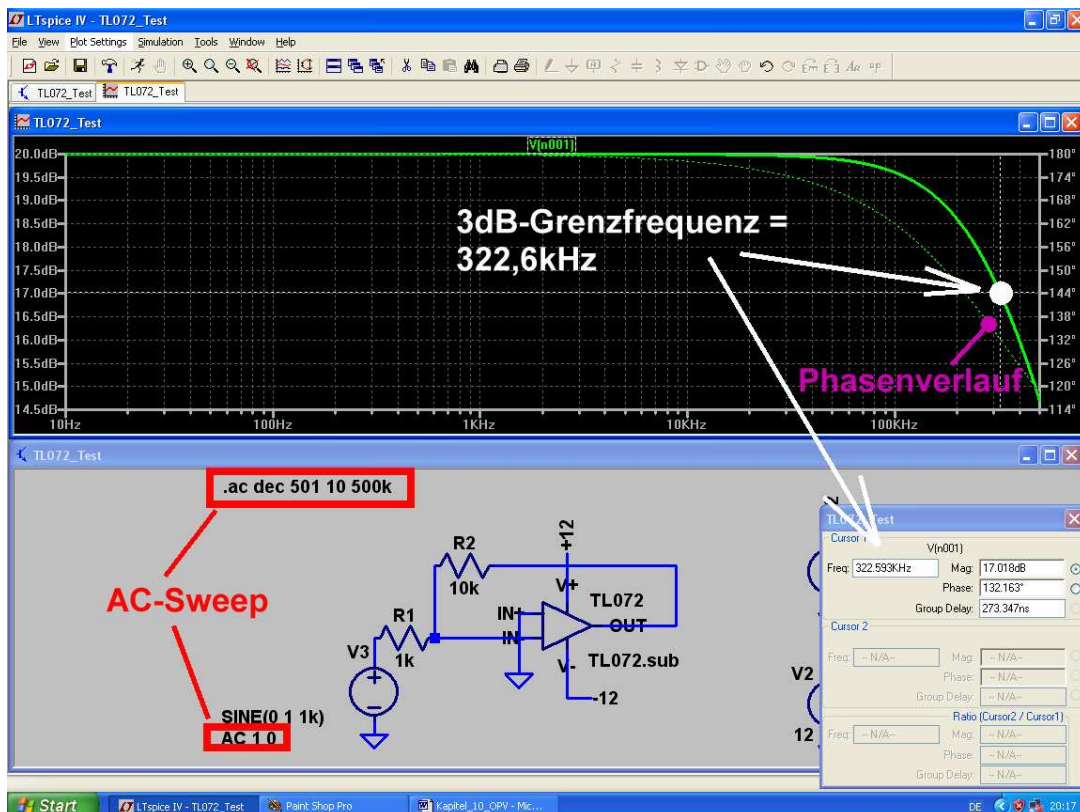
Fertig!

Nun ist natürlich unser invertierender Verstärker mit 10facher Verstärkung auch kein Problem mehr:



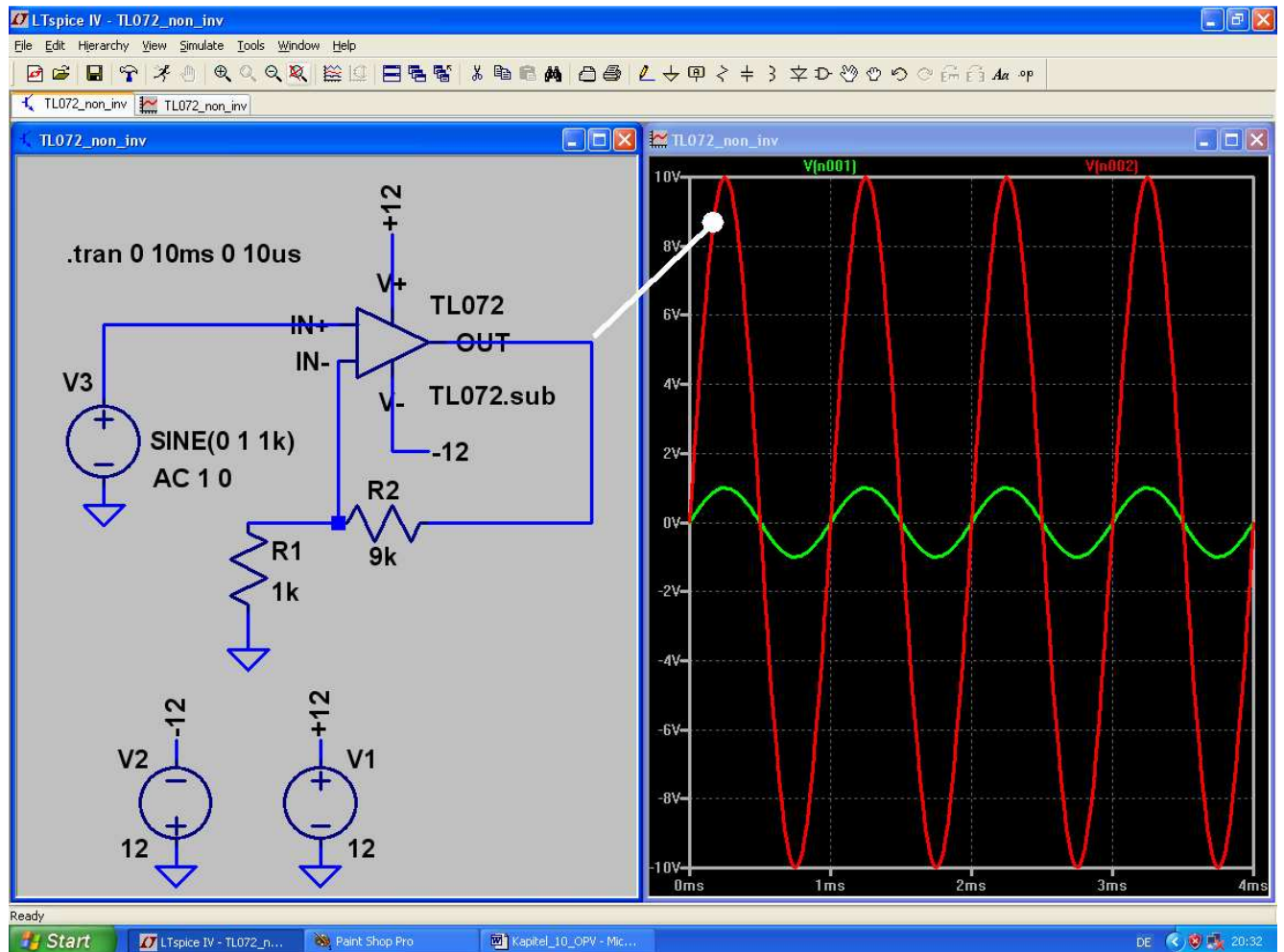
Interessant dürfte dagegen ein AC-Sweep sein, der das Verhalten bei höheren Frequenzen bzw. die obere Grenzfrequenz zeigt. Bitte mal simulieren!

Ergebnis:



## 10.2. Nicht umkehrender Verstärker mit TL072

Gefordert wird wieder 10fache Verstärkung, aber der Einsatz einer nicht umkehrenden Schaltung. Dadurch erreicht man einen sehr hohen Eingangs-widerstand:

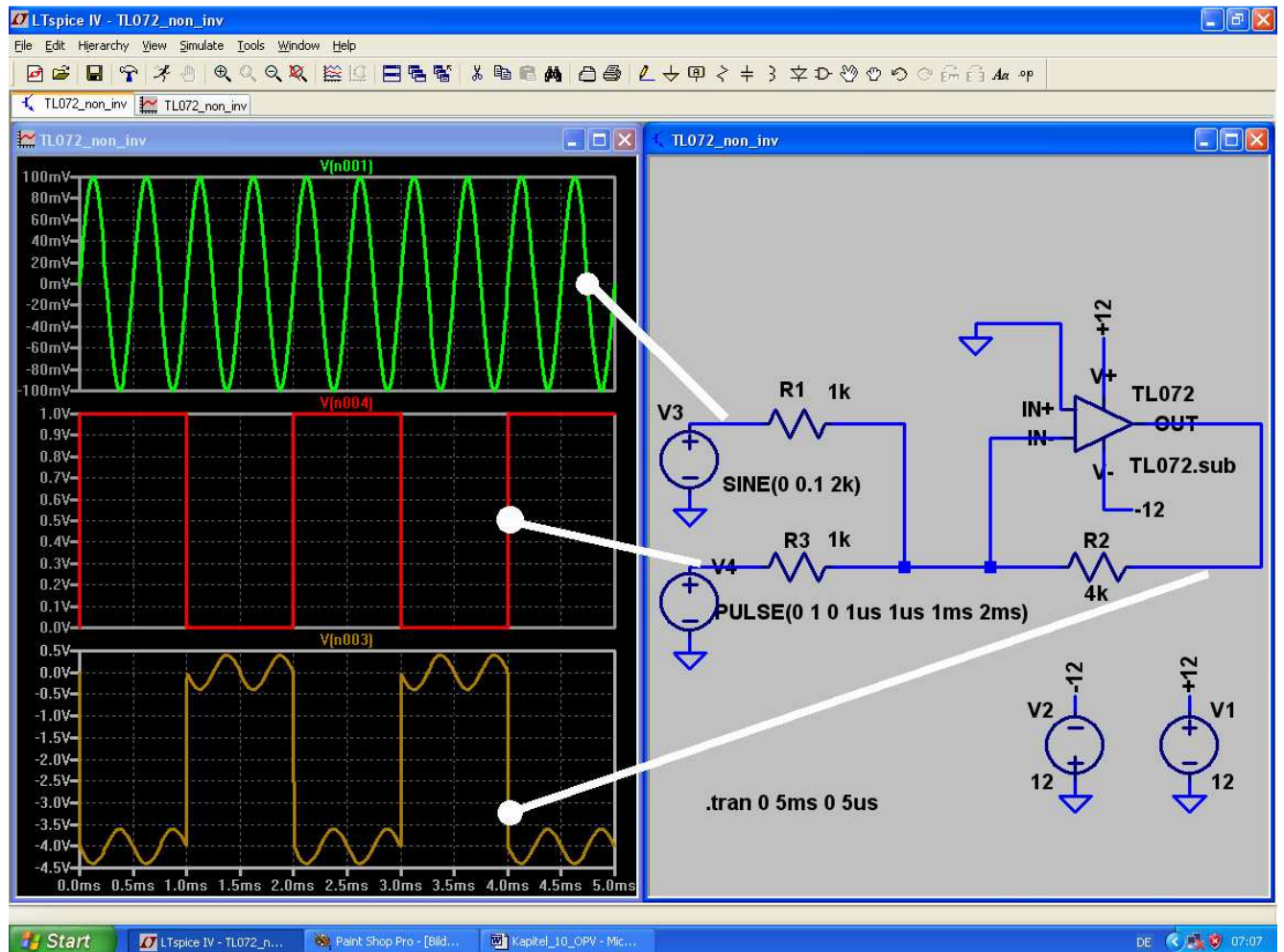


### 10.3. Analog-Addierer mit TL072

Mit OPVs lassen sich bekanntlich analoge Signale sehr leicht zusammenaddieren. Wir wollen das mit einem Sinus ( $f = 2\text{kHz}$ , Spitzenwert =  $0,1\text{V}$ ) und einem Rechtecksignal ( $f = 500\text{Hz}$ ,  $U_{\text{min}} = 0\text{V}$ ,  $U_{\text{max}} = 1\text{V}$ ) ausprobieren und fordern zusätzlich:

- eine 4fache Verstärkung und
- einen Eingangswiderstand von  $1\text{k}\Omega$  bei jedem Eingangskanal.

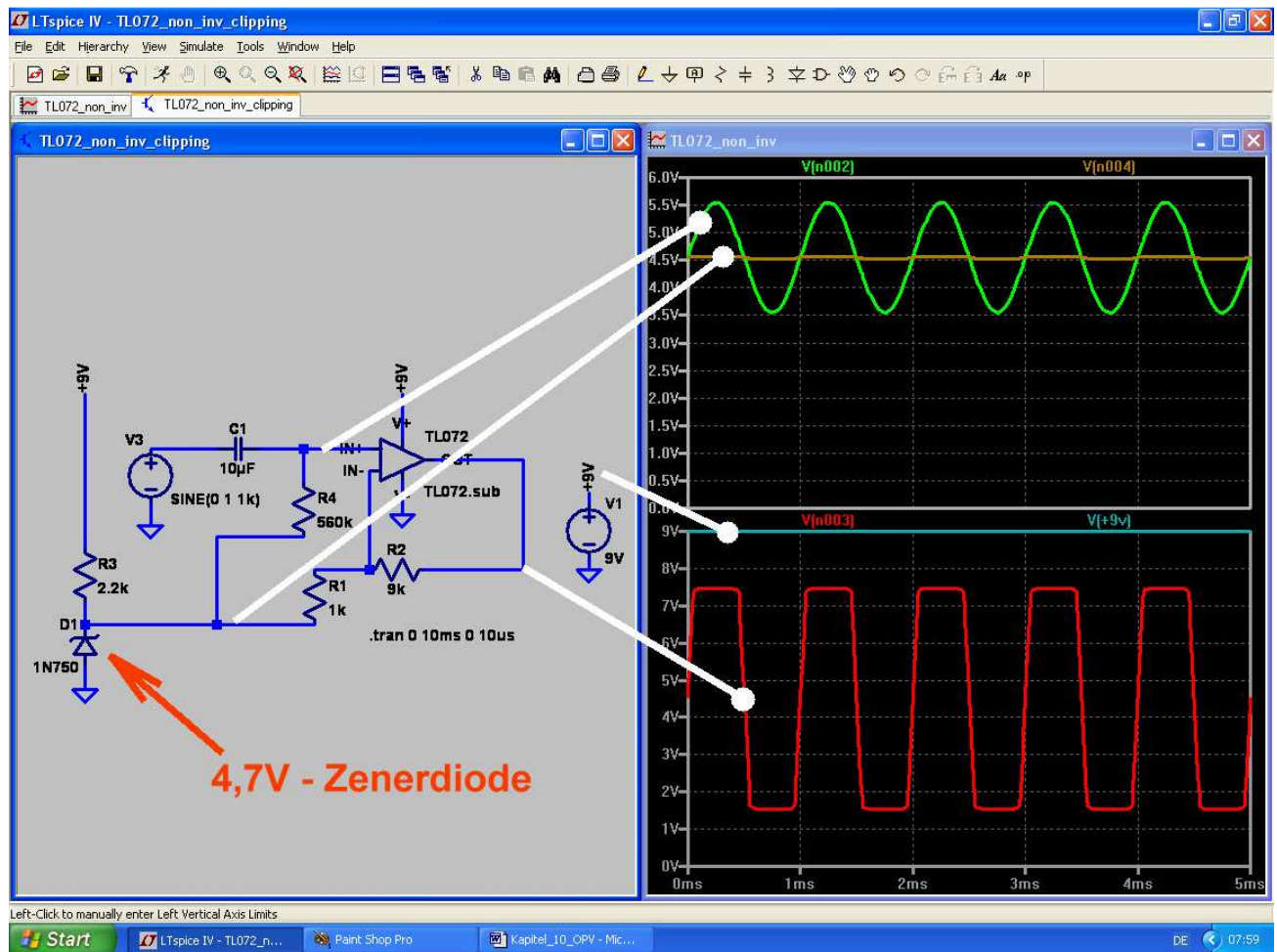
Dazu dient eine umkehrende Schaltung, bei der der Invertierende Eingang des OPVs einen „virtuellen Nullpunkt“ bildet.



## 10.4. Probleme beim Betrieb mit nur einer Betriebsspannung

Oft steht in der Praxis nur eine einzige positive Versorgungsspannung zur Verfügung. Dann muss man Folgendes beachten:

- Der „**V+**“ - Anschluss des OPV wird an diese **Versorgungsspannung** angeschlossen.
- Der „**V-**“ - Anschluss wird mit **Masse** verbunden.
- Das Gleichspannungspotential an den Eingängen des OPVs** muss dann auf **halbe Betriebsspannung** gebracht werden.
- Beim TL072 wird im Ausgang eine „Komplementärendstufe in Kollektorschaltung“ eingesetzt. Die funktioniert gut, hat aber den Nachteil, dass der Spitzenwert der Ausgangsspannung dadurch stets kleiner ist als die halbe Betriebsspannung. Ist eben so! Deshalb wollen wir die maximale unverzerrte Ausgangsspannung durch eine Simulation ermitteln.



### Analyse der Schaltung und Auswertung der Ergebnisse:

Mit der Zenerdiode 1N750 wird eine „Vorspannung“ mit  $U_b / 2 = +4,5V$  aus der Versorgungsspannung von +9V erzeugt. Sie wird über den Widerstand  $R4 = 560k$  dem Nichtinvertierenden Eingang des OPVs zugeführt. Außerdem dient sie (wegen des kleinen dynamischen Widerstandes der Z-Diode) gleichzeitig als „Virtueller Masse-Fußpunkt“ für den Gegenkopplungs-Spannungsteiler aus  $R1$  und  $R2$ .

Im oberen Diagramm ist die Spannung am Nichtinvertierenden Eingang (= Summe aus Vorspannung und zu verstärkender Wechselspannung) zu sehen.

Im unteren Diagramm wird die Ausgangsspannung am Pin „OUT“ des OPVs dargestellt. Deutlich ist zu sehen, dass im oberen und unteren Teil des Spannungsverlaufes nun jeweils 1,5V fehlen -- das ist der Teil, den die Endstufe zum Leben braucht. Zum genauen Vergleich ist auch die Versorgungsspannung mit +9V eingeblendet.

## 11. Siebtes Projekt: DC-DC-Konverter

Dieses Projekt beginnt gleich mit einem Hinweis und einer **Warnung**:

Solche Schaltungen nutzen zum Teil recht verwickelte Vorgänge in Induktivitäten aus. **Das übliche einfache Modell einer Induktivität wird deshalb in der Simulationspraxis dieser Sache nicht gerecht und das Ergebnis ist entweder Nonsens oder die Simulation bricht ab.** Nicht umsonst haben teure Programme hier spezielle zusätzliche Rechenmethoden eingebaut, die trotzdem ein Ergebnis ermöglichen. Aber: nur bei sehr teuren Maschinen vorhanden und dann streng geheimgehalten, wie das funktioniert.

Wir arbeiten jedoch mit dem „Berkely-Standard-SPICE“ und da hilft nur eines:

- a) **Niemals mit idealen Induktivitäten arbeiten, sondern immer einen fühlbaren Reihenwiderstand vorsehen.**
- b) Die Induktivitäten ergeben mit den in der Gesamtschaltung vorhandenen Kapazitäten oft **schwingfähige Gebilde mit recht wilden Simulationsergebnissen.** Da sollte man z.B. die Schaltung so stark am Ausgang belasten, dass diese Effekte stark bedämpft werden und / oder genügend lang simulieren.

### 11.1. Bereitstellung des Power-MOSFETs „IRFZ44N“

Für die folgende Untersuchung der verschiedenen Grundschaltungen benötigen wir einen passenden elektronischen Schalter. Hier haben sich Power-MOSFETs durchgesetzt und die Firma IRF (= International Rectifiers) ist ein bekannter und führender Hersteller. Selbstverständlich stellt sie für alle Produkte SPICE-Modelle bereit, aber die Produktpalette ist ungeheuer groß.

Wir benötigen für unsere Simulationen einen „Standardtyp“ mit dem Namen IRFZ44N und benötigen dafür ein neues Symbol in unserer Software.

**Die folgende Selbstbau-Aktion ist unvermeidbar, weil das Modell nur als „Subcircuit“ erhältlich ist und wir deshalb selbst antreten müssen.....**

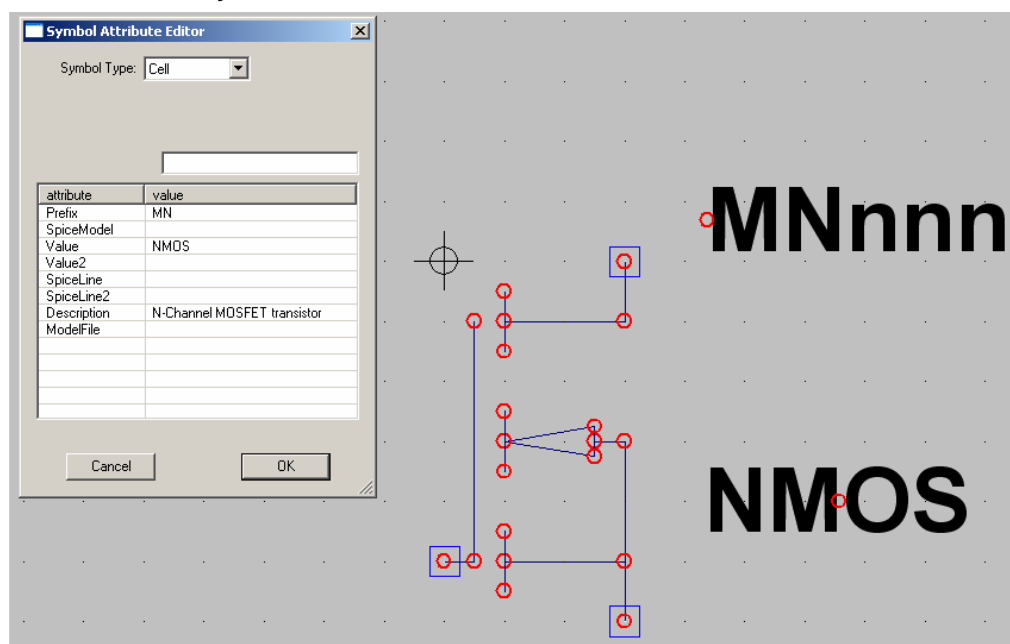
Also wieder mal:

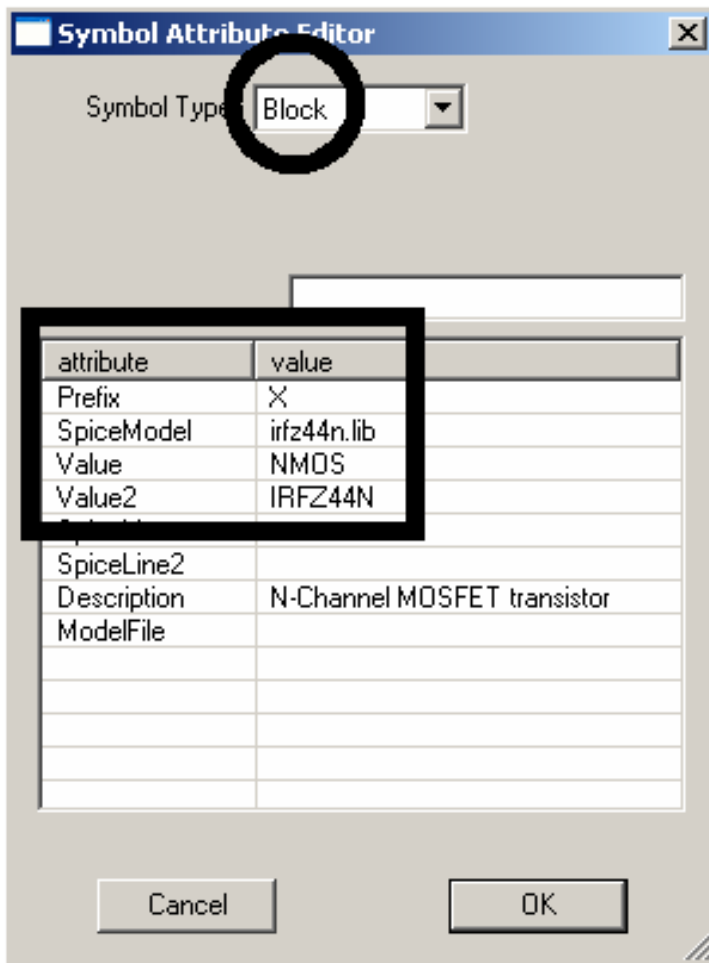
#### 1. Schritt:

Mit der Suchmaschine (Google o. ä.) holen wir unter dem Begriff „**irfz44n spice model**“ direkt die Datei „**irfz44n.spi**“. Deren Inhalt wird markiert, in ein neues Blatt eines Editors kopiert und als „**irfz44n.lib**“ im Ordner „**lib / sub**“ des SwitcherCAD-Programms gespeichert.

#### 2. Schritt:

Unter „**File**“ wählen wir „**New Symbol**“ und gehen dann nochmals in „**File**“. Jetzt öffnen wir im Ordner „**lib / sym**“ die Datei „**nmos.asy**“. Anschließend machen wir über „**Edit**“ und „**Edit Attributes**“ die Attribute-Liste sichtbar:



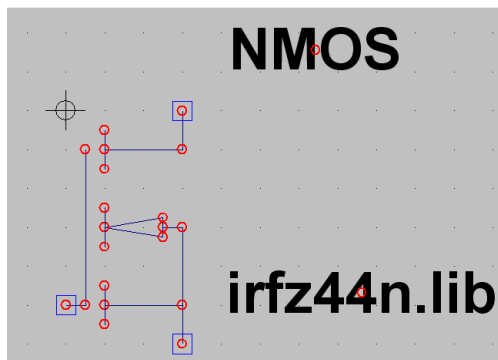


### 3. Schritt:

Und darin muss massiv umgebaut werden.

**Bitte alle Einträge nach nebenstehendem Vorbild sorgfältig vornehmen und prüfen!!**

(Besonders in der ersten Zeile „Prefix“ muss unbedingt von MN = model auf „X“, also **subcircuit**, umgestellt werden!)



### 4. Schritt:

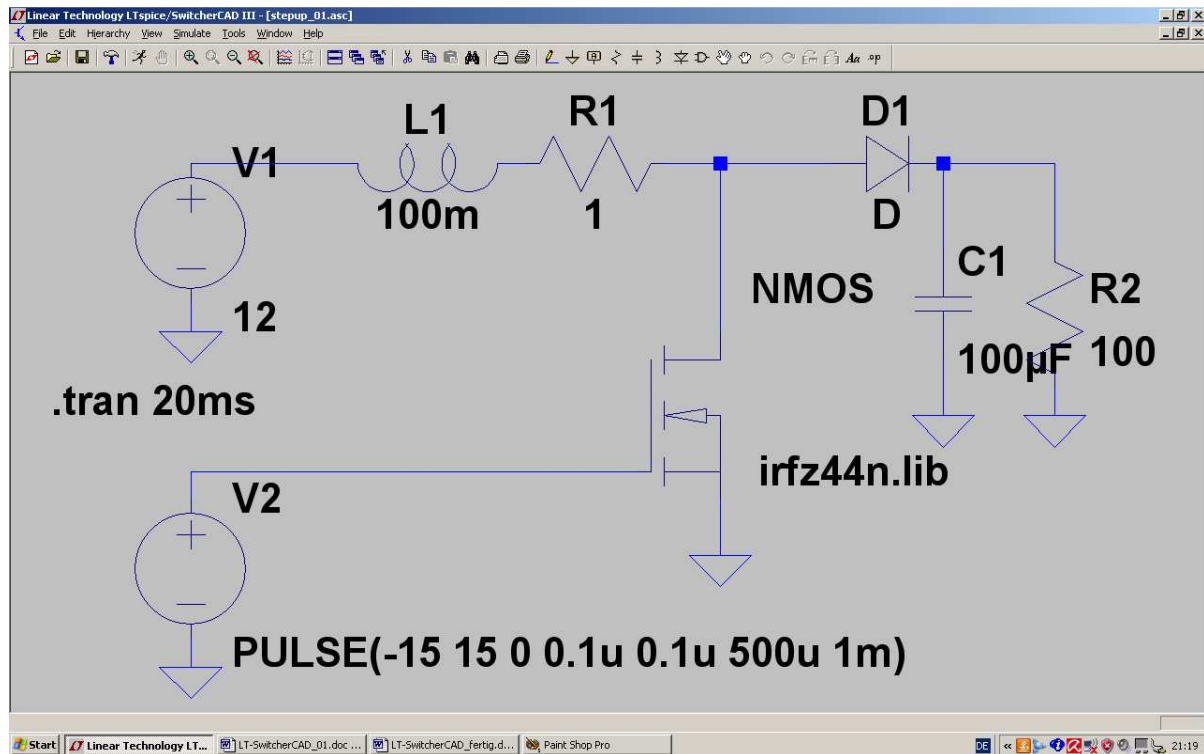
Nun gehen wir nochmals in „Edit“, öffnen diesmal aber „Attribute Window“. Wir klicken „Value“ an und platzieren die auftauchende Bezeichnung „NMOS“ oberhalb unseres Symbols. Nach der Bestätigung mit OK wiederholen wir die Prozedur, aber diesmal klicken wir auf „SPICE File“. Die auftauchende Angabe „irfz44n.lib“ wird unter das Symbol gesetzt.

Erst jetzt dürfen wir das fertige Symbol speichern -- am besten in einem extra angelegten eigenen Ordner „NMOS-FETS“ in „lib / sym“.

Das war's und jetzt geht es an die Schaltungen.



## 11.2. Der Step-Up-Konverter (= Aufwärtswandler)



Die Schaltung besteht zunächst aus der Gleichspannungsquelle V1 mit 12 V, an die unsere Induktivität mit  $L = 100\text{mH}$  (Fachausdruck: „Speicherdrossel“) angeschlossen wird. Es folgt der geforderte Reihenwiderstand, bevor der Power-MOSFET seine Aufgabe als gesteuerter Kurzschluss-Schalter gegen Masse übernehmen kann. Solange er eingeschaltet ist, liegt die Drossel an +12 V und der Strom in ihr steigt etwa linear an. Dadurch wird aber nach der Beziehung

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

immer mehr Energie im Magnetfeld der Spule gespeichert. Sperrt man nun den FET, dann kann diese Energie nicht verschwinden, außerdem kann sich der Strom in einer Spule nicht ruckartig ändern (denn das gäbe unendlich hohe Induktionsspannungen). Also fließt der Spulenstrom nach rechts, schaltet die Schottky-Diode D ein, speist den Lastwiderstand und lädt den Kondensator höher auf. Dieser Vorgang wird (da am Gate des FET eine Rechteckspannung anliegt) dauernd wiederholt, bis ein Gleichgewicht zwischen aufgenommenen und abgegebenen „Energieportionen“ hergestellt ist.

**Wichtig: bei dieser Schaltung addieren sich die Versorgungsgleichspannung und die Induktionsspannung der Spule.**

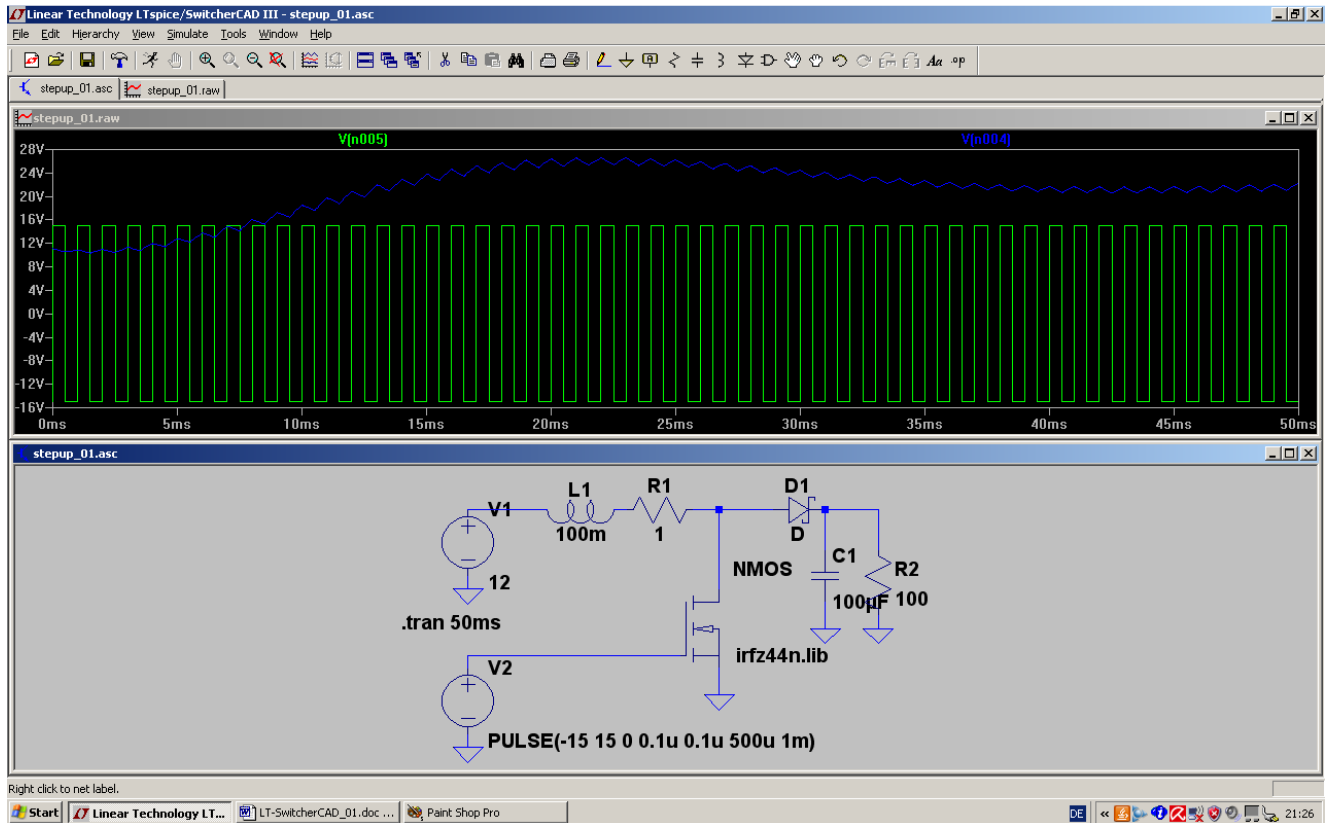
**Folge: die Ausgangs-Gleichspannung wird deutlich höher als die Betriebsspannung!**

Nun wird diese Schaltung gezeichnet und gleich die Puls-Spannungsquelle programmiert.

Wir steuern den Power-MOSFET am Gate mit einem symmetrischen Rechtecksignal an, das eine Frequenz von 1 kHz aufweist.

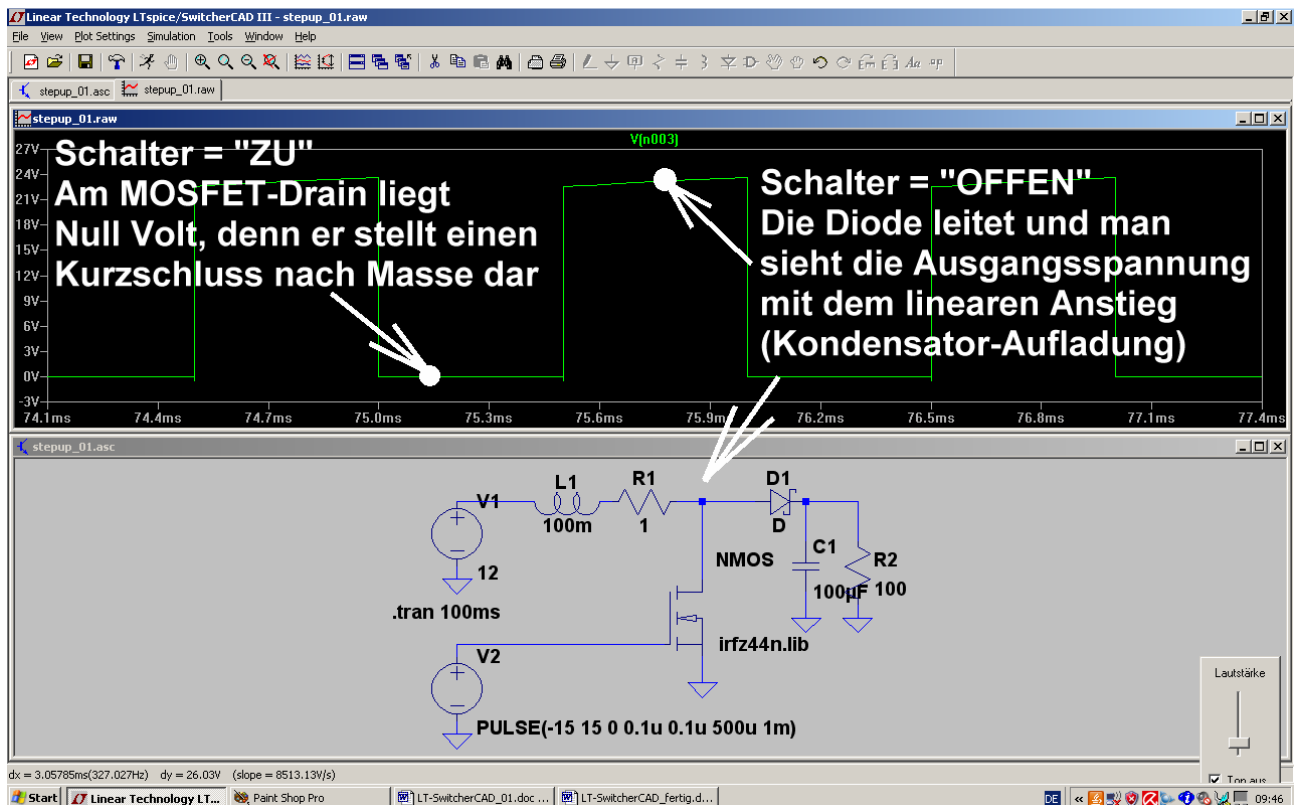
Eingeschaltet wird er mit +15 V, gesperrt dagegen mit -15 V.

Als Simulationszeitraum wählen wir 0.....50ms.



Gut zu erkennen ist der Mittelwert der Ausgangsspannung von etwa +23V und darauf ein kleiner Sägezahn-Anteil (= Auf- und Entladung des Ausgangs-Kondensators).

Werfen wir jetzt noch einen Blick auf die Drainspannung des FETs:

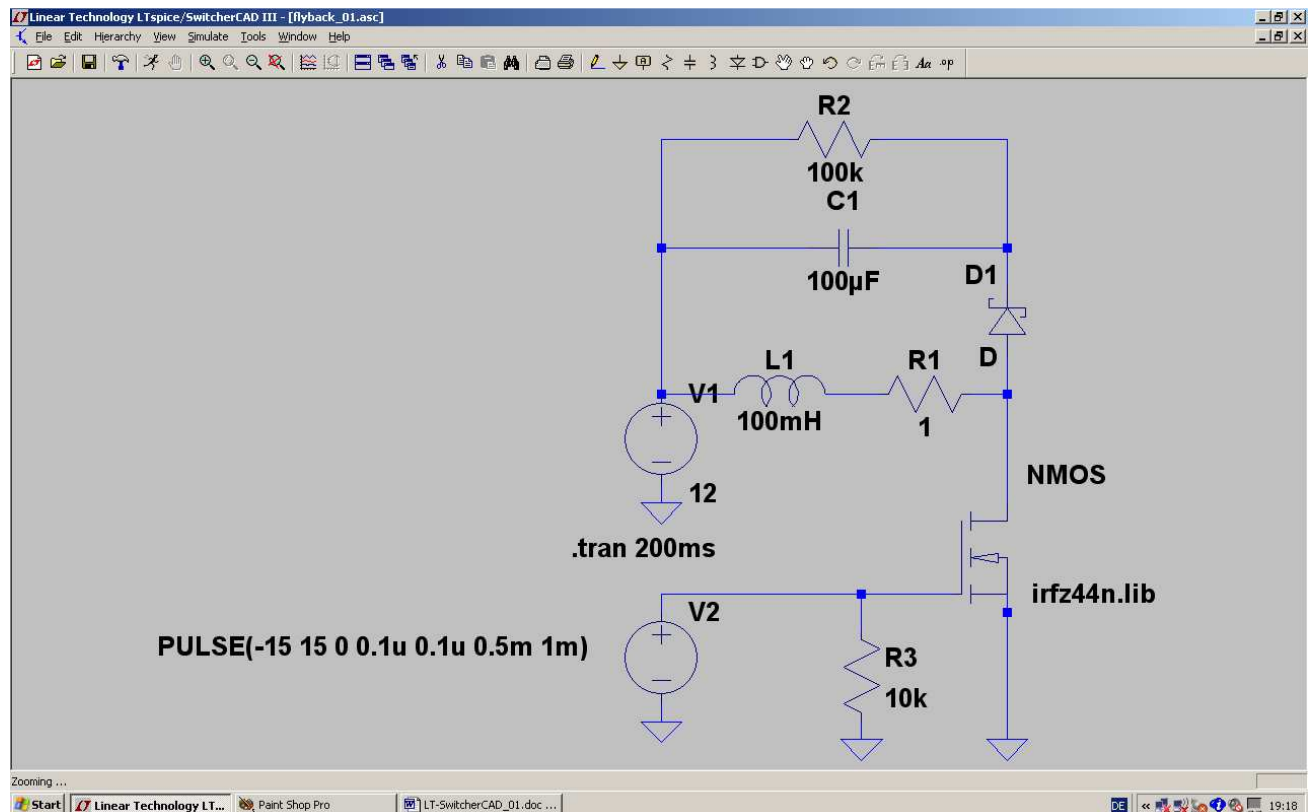


Wer Spaß daran hat, kann sich nun auch die Gate-Spannung auf den Schirm holen und / oder die Simulation für andere Werte der Ansteuerfrequenz, der Induktivität oder des Ladekondensators wiederholen.

**Hinweis:**

**Die Regelung oder Stabilisierung der Ausgangsspannung erfolgt bei dieser Schaltung immer über das Tastverhältnis (= Verhältnis von FET-Einschaltdauer zur Periodendauer)!**

### 11.3. Der Flyback-Konverter (= Sperrwandler)



Genau hinsehen:

hier wird das rechte Ende der Drosselspule bei eingeschaltetem FET direkt an Masse gelegt, während das linke Ende an der (konstanten) Betriebsgleichspannung von +12 V liegt. Dadurch steigt der Spulenstrom linear an und in der Spule wird wieder mal im Magnetfeld Energie gespeichert. Sperrt der FET plötzlich, so muss der Spulenstrom in alter Richtung weiterfließen. Aber da ihm der Weg über den FET versperrt ist, schließt sich der Stromkreis nun über die Schottky-Diode „D“ und den Ladekondensator bzw. Lastwiderstand. Dieser Stromfluss hört dann auf, wenn die in der Spule gespeicherte „Energieportion“ voll an den Ausgangskreis übergeben wurde.

Das führt zu recht interessanten Eigenschaften der Schaltung:

- Die Schaltung darf am Ausgang ohne schlimme Folgen kurzgeschlossen werden, da die Energieübertragung zum Ausgang wie beim Wasserschöpfen abläuft: Eimer füllen (= Spule aufladen) / Eimer ausleeren (= Energie an Ausgang übergeben). Dadurch ist weder der FET noch irgendein Teil der Schaltung beim Kurzschluss gefährdet.
- Allerdings gibt es noch weitere Effekte: entfernt man den Lastwiderstand, dann wird durch die übergebenen Energieportionen der Kondensator geladen und geladen und immer weiter geladen... folglich läuft die Ausgangsspannung immer höher und höher, bis irgendetwas nachgibt! **Zur Demonstration dieses Effektes wurde der Lastwiderstand auf 100kΩ erhöht.**
- Genau hinsehen: **der Pluspol der Kondensatorspannung (= Ausgangsspannung) ist an der Katode** der verwendeten Schottky-Diode, außerdem ist diese Ausgangsspannung im Normalfall immer deutlich höher als die Betriebsspannung!

## Hinweis:

Die Regelung oder Stabilisierung der Ausgangsspannung erfolgt bei dieser Schaltung immer über das Tastverhältnis (= Verhältnis von FET-Einschaltdauer zur Periodendauer)!

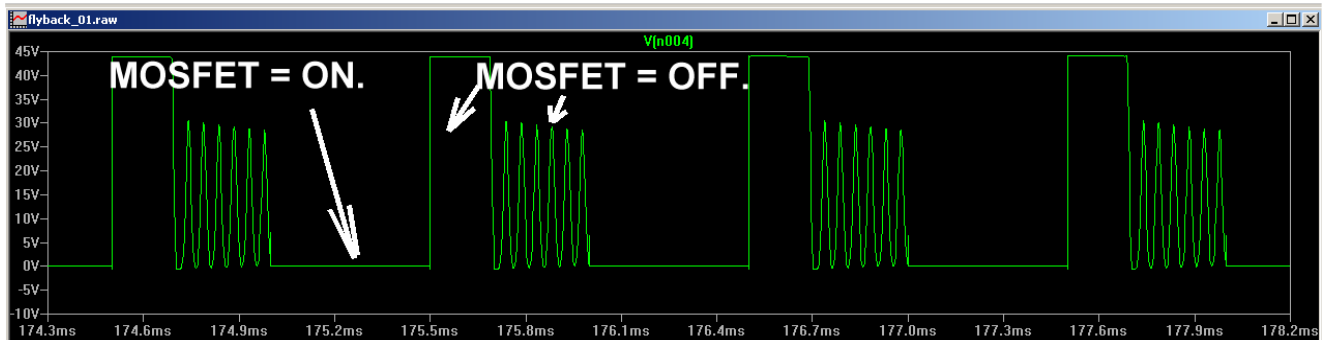
Die obige Schaltung wird nun gezeichnet und simuliert. Als Ansteuersignal dient wieder eine symmetrische Rechteckspannung mit +15 V Maximalwert, -15 V Minimalwert und einer Frequenz von 1 kHz. Da die Schwellspannung der Diode nicht Null Volt beträgt, endet der Aufladevorgang des Ausgangskondensators bereits, bevor die „gespeicherte Magnetfeld-Energie in der Spule aufgebraucht ist.“

Diese Restenergie tobt sich nun als gedämpfte Eigenschwingung (des Resonanzkreises, der aus L und der Drainkapazität des MOSFETs gebildet wird) aus!

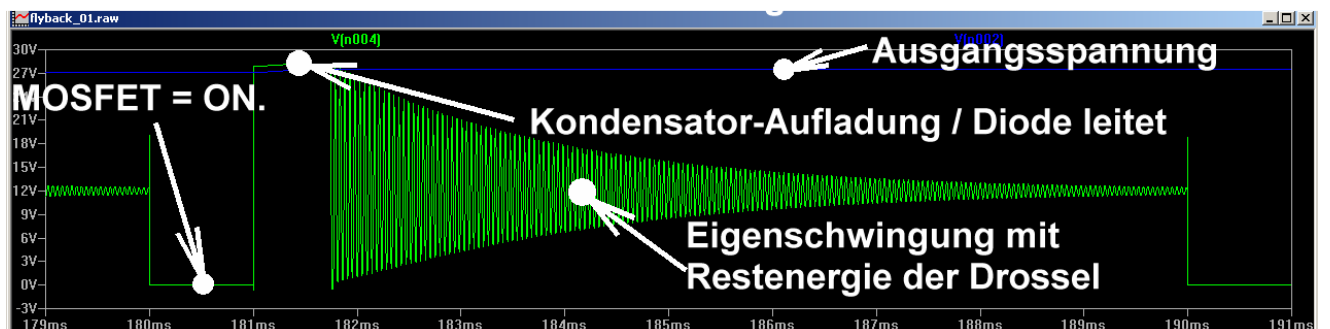
Schauen wir doch mal über die Simulation des Spulenstroms nach, ob das alles stimmt:



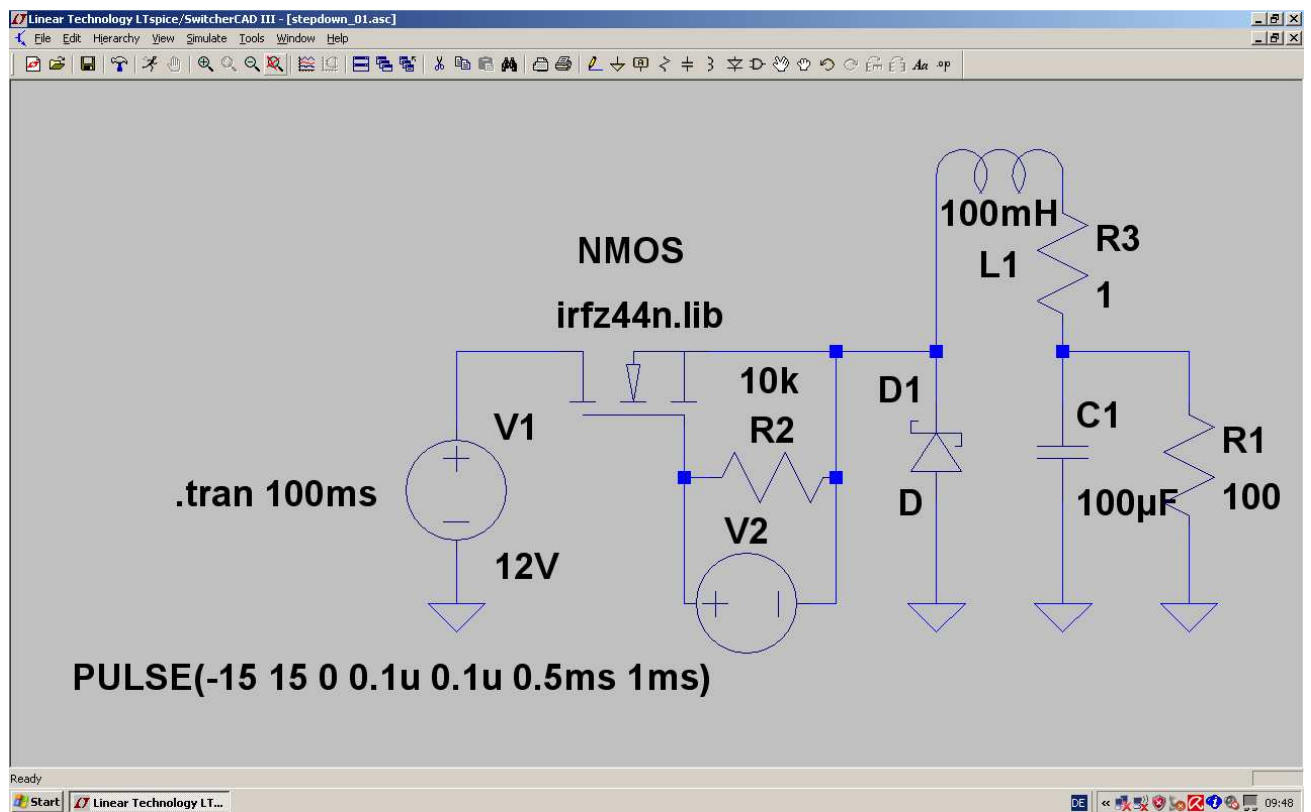
Genau so schön muss man das an der Drainspannung des MOSFETs sehen:



Zur besseren Verdeutlichung hier nochmals eine Simulation mit einer Pulslänge von 1ms und einer Periodendauer von 10ms:



## 11.4. Der Step-Down-Konverter (= Abwärtswandler)



Bei dieser Schaltung ist der MOSFET als Schalter zwischen der Betriebsspannung und der Drossel eingefügt. Die Diode ist in Sperrrichtung zwischen dem Anfang der Drossel und Masse angeordnet (...sie wird hier oft als „Freilaufdiode“ bezeichnet).

Vorgänge in der Schaltung:

- Wird der MOSFET eingeschaltet, dann fließt aus der Gleichspannungsquelle ein linear ansteigender Strom über den FET, die Drossel und den Ausgangskondensator bzw. Lastwiderstand nach Masse. Dadurch wird wieder im Magnetfeld der Spule Energie gespeichert.
- Wird der FET gesperrt, so muss der Strom seinen Weg über die Freilaufdiode nehmen, damit sich der Stromkreis wieder korrekt schließt (...der Weg über die Spannungsquelle ist wegen des gesperrten MOSFETs unterbrochen). Die gespeicherte Magnetfeldenergie versorgt nun den Lastkreis. Bitte beachten: der Strom klingt wieder linear bis auf Null ab **und in dieser Zeit ist die Spannung an TPv2 (...wegen der leitenden Diode!) NEGATIV!!!**

Hinweise:

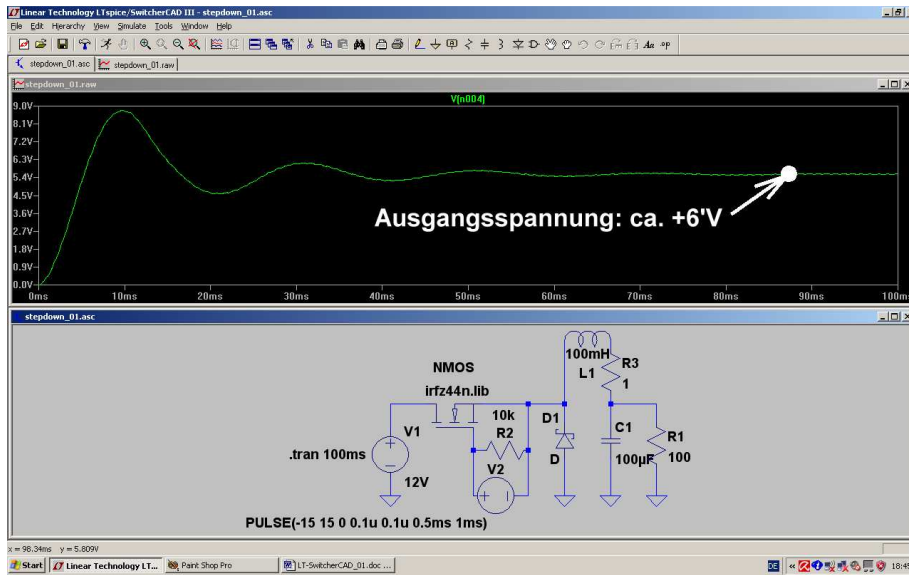
- Die Regelung oder Stabilisierung der Ausgangsspannung erfolgt bei dieser Schaltung immer über das Tastverhältnis (= Verhältnis von FET-Einschaltdauer zur Periodendauer)!
- Die Ausgangsgleichspannung ist bei dieser Schaltung **IMMER** kleiner als die Eingangsspannung, ihren Wert erhält man, wenn man die Eingangsspannung mit dem Tastverhältnis multipliziert.

Bei unserem Beispiel haben wir eine Einschaltzeit von 0,5ms und eine Periodendauer von 1ms vorgesehen. Deshalb sind am Ausgang

$$0,5 \times 12V = 6V$$

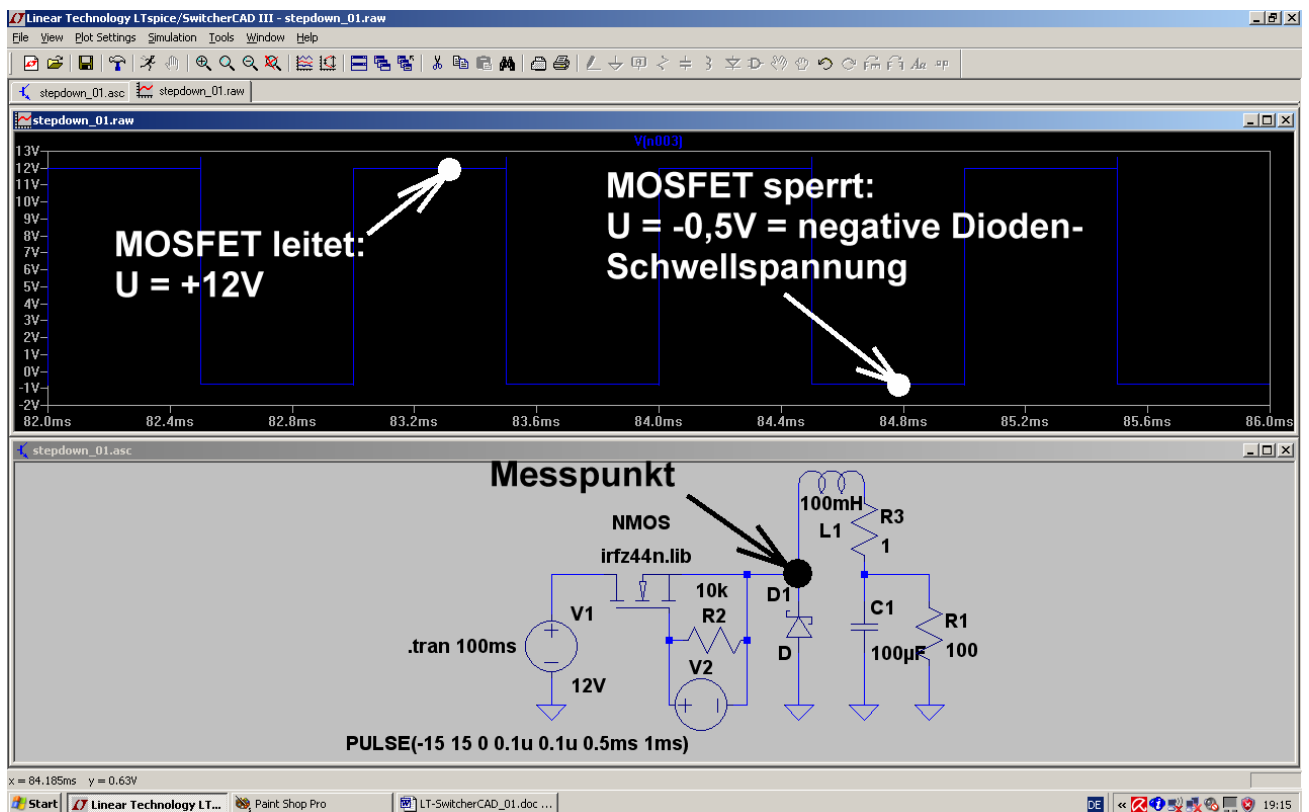
zu erwarten.

Sehen wir uns mal die Simulation der Ausgangsspannung an:



Na, also.....

Und so sieht die Spannung hinter dem MOSFET -- also an der Katode der Diode -- aus:



## 12. Achtes Projekt: Phasenanschnitt-Steuerung mit Thyristor

### 12.1. Das eingesetzte Thyristor-Modell

Recht gut bewährt hat sich für die Simulation der Thyristor-Typ

## 2N5171

Es handelt sich hier um einen 20A / 600V – Thyristor, mit dem in der Praxis doch einiges anzufangen ist. Zur Beschaffung des SPICE-Modells suchen wir im Internet nach der Datei

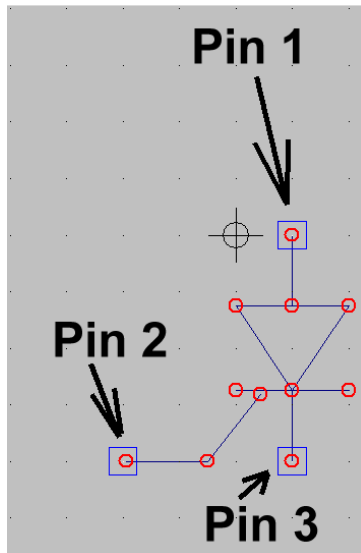
## thyristr.lib

Sie enthält den erforderlichen Subcircuit und wird unter dem Pfad „lib / sub“ im LTSpiceIV-Verzeichnis abgelegt.

### Aber Vorsicht:

Wenn sie dagegen als HTML-File auf unserem Bildschirm auftaucht, wird sie erst geöffnet, dann der Inhalt markiert, in die Zwischenablage kopiert und in ein neues Blatt des Texteditors eingefügt. Das kann dann endgültig als „thyristr.lib“ am angegebenen Ort gespeichert werden.

Und dann geht es mal wieder los:



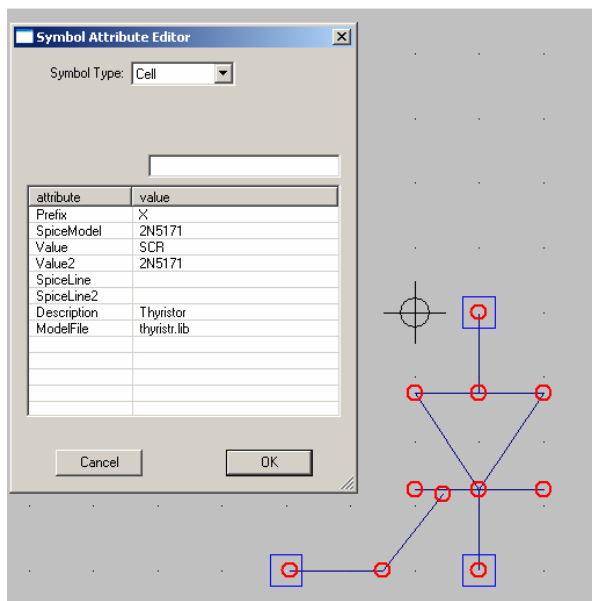
### 1. Schritt:

Unter „File“ wählen wir „New Symbol“ und öffnen anschließend das Symbol für die Diode („diode.asy“ im Ordner „lib / sym“). Wir ergänzen das Schaltzeichen um den Gate-Anschluss und nehmen uns gleich eine wichtige Sache vor:

Entsprechend dem SPICE-Modell müssen wir streng die Zuordnung

**Anode = Pin 1**  
**Gate = Pin 2**  
**Katode = Pin 3**

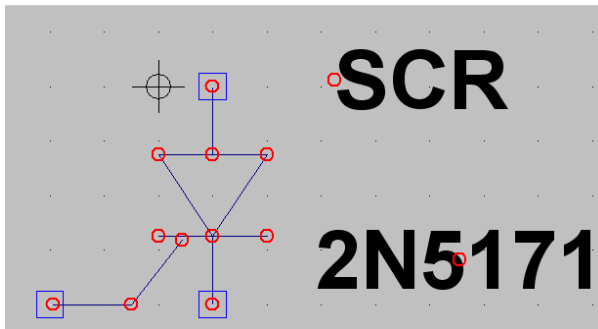
einhalten. Deshalb klicken wir mit der rechten Maustaste nacheinander auf jeden Anschlusspunkt und stellen sicher, dass diese Reihenfolge stimmt.



### 2.Schritt:

Jetzt öffnen wir (unter „Edit / Attributes“) die Option „Edit Attributes“ und erstellen sorgfältig die nebenstehende Attributstabelle.

<b>Präfix</b>	<b>X</b>
<b>SpiceModel</b>	<b>2N5171</b>
<b>Value</b>	<b>SCR</b>
<b>Value 2</b>	<b>2N5171</b>
<b>Description</b>	<b>Thyristor</b>
<b>ModelFile</b>	<b>thyristr.lib</b>

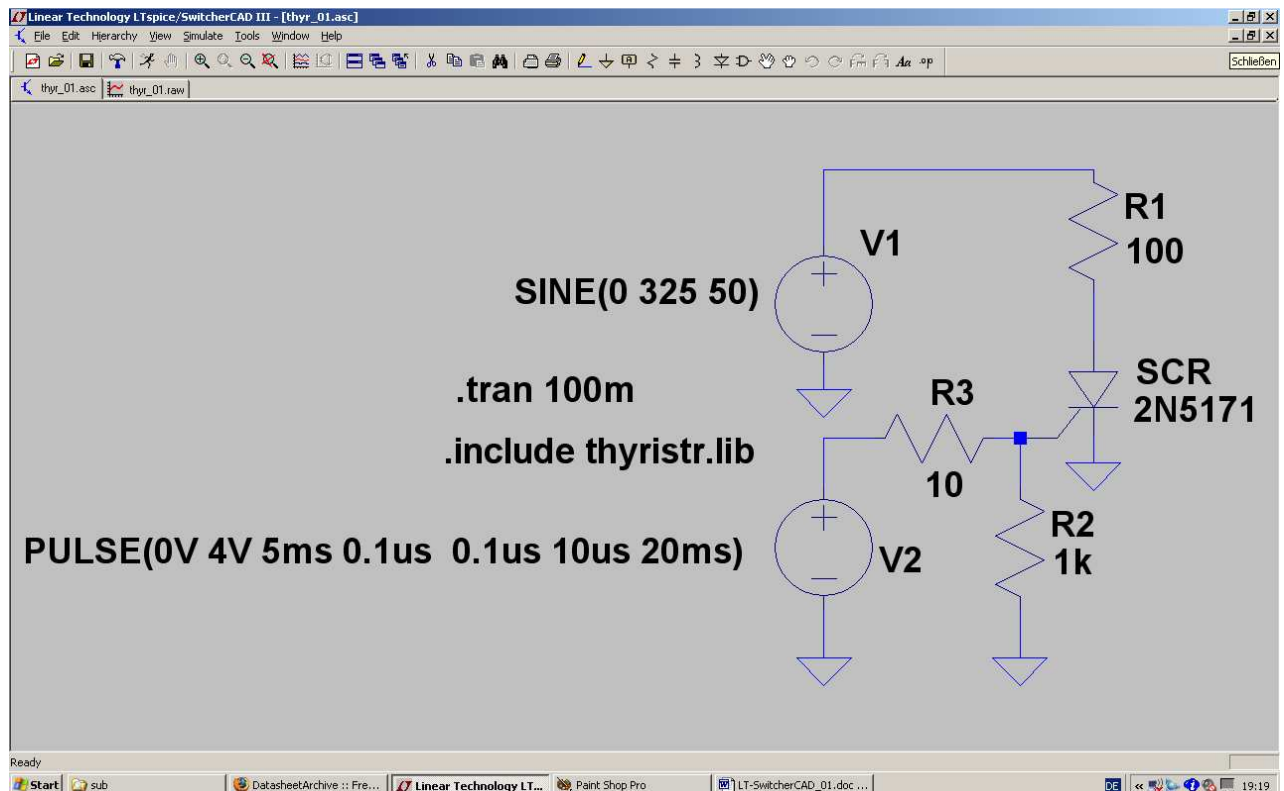


### 3. Schritt:

Wir öffnen nochmals „Edit / Attributes“, wählen aber anschließend „Attribute Window“. In der auftauchenden Liste klicken wir auf „Value“ und platzieren anschließend den Hinweis „SCR“ neben dem Schaltzeichen. Das wird nochmals wiederholt, aber nun „SpiceModel“ gewählt. So sieht der Endzustand aus.

Jetzt muss nur noch das fertige Modell in einem neuen Ordner „Thyristoren“ in „lib / sym“ unter der Bezeichnung „2N5171.asy“ abgelegt werden.

## 12.2. Schalten von Ohm'schen Lasten



### Schaltungsbeschreibung:

Die Sinus-Spannungsquelle V1 (Spitzenwert = 325 V, Frequenz = 50Hz) speist eine Reihenschaltung aus dem Widerstand R1 (100  $\Omega$ ) und dem Thyristor 2N5171. Vom Gate nach Masse ist meist ein Widerstand R2 (hier: 1k $\Omega$ ) geschaltet. Über einen Strombegrenzungswiderstand (R3 = 10 $\Omega$ ) steuert eine Pulsspannungsquelle das Gate des Thyristors an,

### Erläuterungen:

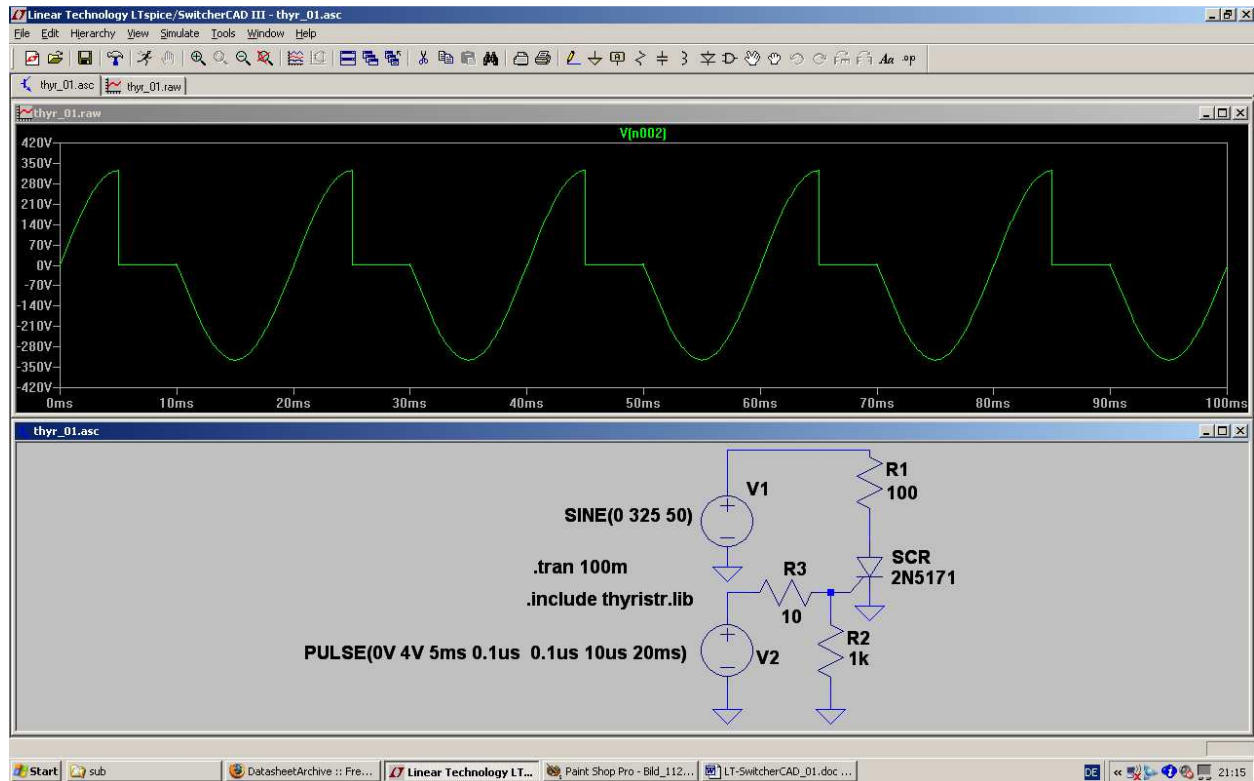
- eine Simulationszeit von 0...100ms wird durch die SPICE-Direktive „.tran 100m“ festgelegt.
- Mit „.include thyristr.lib“ wird die Datei mit den Thyristor-SPICE-Modellen bereitgestellt.
- Die Pulsspannung wird durch folgendes Attribut der Spannung V2 erzeugt:

**PULSE (0V 4V 5ms 0.1us 0.1us 10us 20ms)**

Das ergibt eine Minimalspannung von Null Volt, eine Maximalamplitude von +4V, eine Startverzögerung von 5ms, eine Anstiegs- und Abfallzeit von je 100ns, eine Pulslänge von 10 Mikrosekunden und eine Periodendauer von 20ms

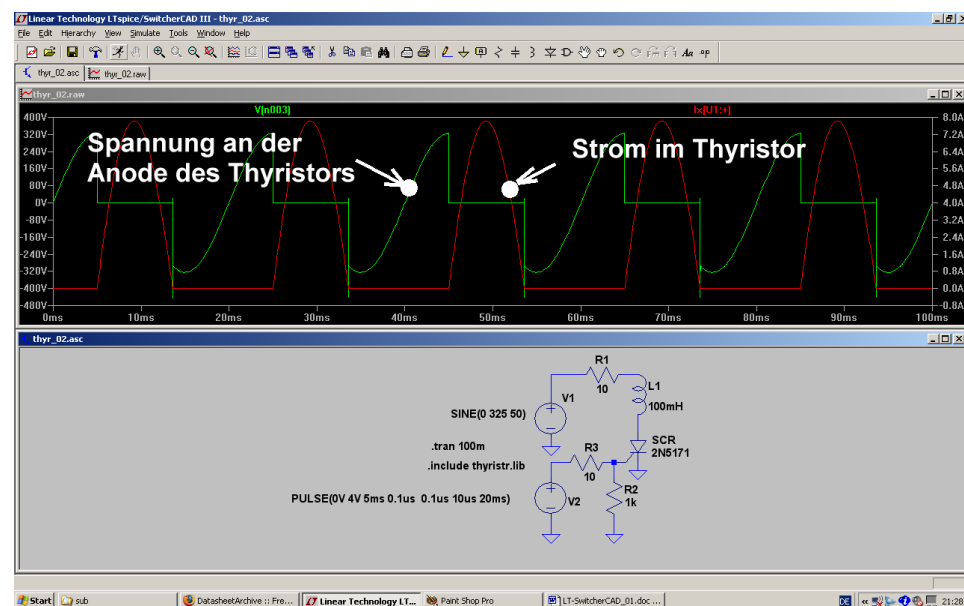


Und so sieht die Spannung an der Anode des Thyristors aus:



### 12.3. Schalten von induktiven Lasten

Hierbei handelt es sich um einen besonders unangenehmen Betriebsfall (...denken wir nur an die vielen Elektromotoren mit Drehzahlregelungen, z. B. in Handbohrmaschinen....), denn sobald nach der Zündung des Thyristors der Strom im Stromkreis ansteigt, speichert die Induktivität Energie in ihrem Magnetfeld. Beim Nulldurchgang der Netzspannung möchte natürlich der Thyristor ausschalten. Aber die Spule bemüht sich (wegen der gespeicherten Energie, die sie wieder los werden möchte!) dagegen aufzukommen und den Strom in der alten Richtung weiterzutreiben. Wie immer im Leben gibt so etwas Kollisionen und Ärger, der sich sehr schön durch die Simulation zeigen lässt.

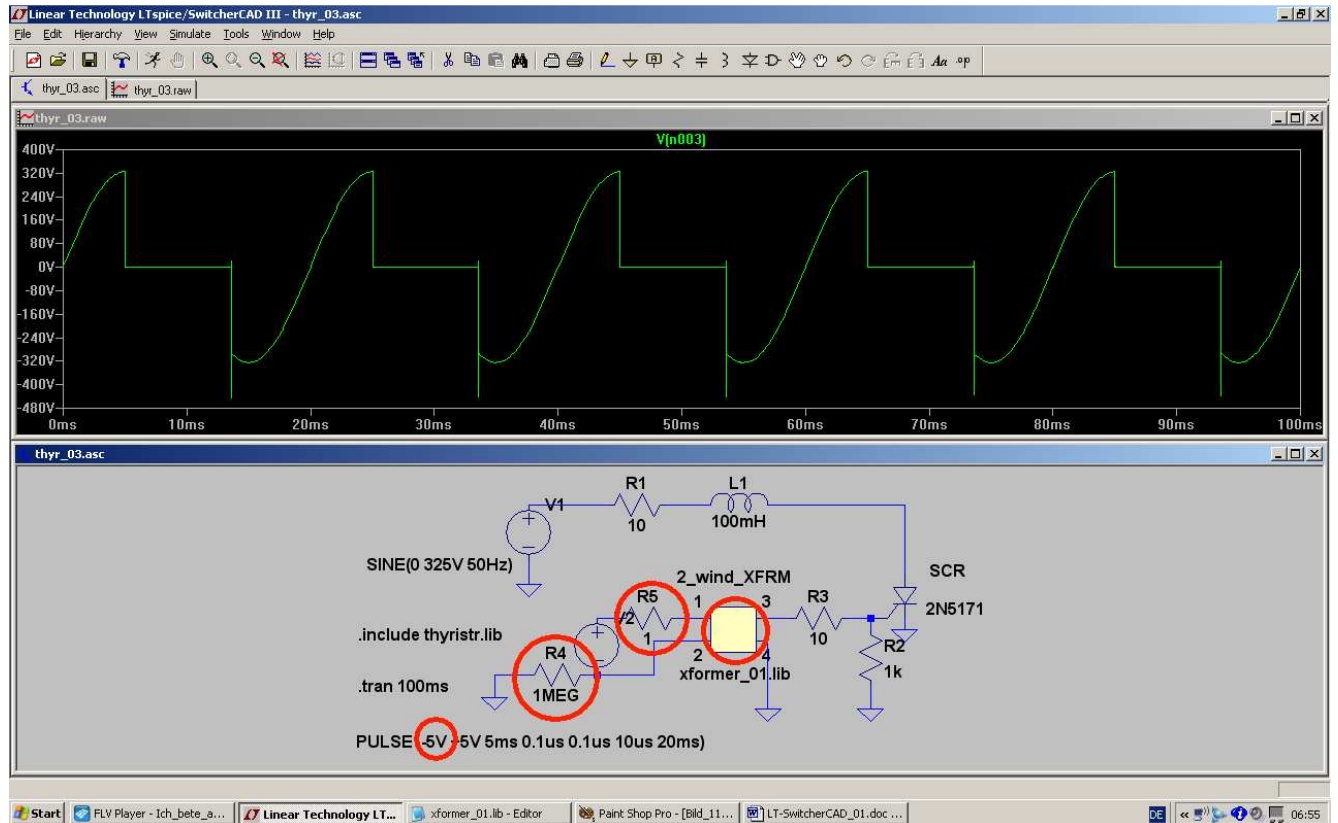


Also schalten wir mal  $L = 100 \text{ mH}$  in Reihe zu einem Lastwiderstand von  $10\Omega$  und drücken dann den Simulationsbutton. Sehr gut ist zu sehen, wie der Thyristor jetzt nicht mehr wie vorher beim Nulldurchgang abschalten kann. Die Induktivität liefert nämlich noch solange Strom, bis die in ihr gespeicherte magnetische Energie verbraucht ist.

Ein kleiner Energierest erzeugt am Ende dann noch eine kurze gedämpfte Schwingung.

## 12.4. Zündung des Thyristors über einen Gate-Transformator

In der Praxis fordert man sehr oft eine galvanische Trennung zwischen Lastkreis und Zündkreis. Deshalb erfolgt die Ansteuerung des Gates mit dem Zündimpuls über einen (meist fertig käuflichen) kleinen Transformator („Zünd-Trafo“). Deshalb nehmen wir uns die vorige Schaltung her, bauen einen solchen Zündtrafo ein und sehen nach, ob alles noch wie gewünscht funktioniert.



Einige Details:

- Die Zündimpuls-Spannungsquelle auf der Primärseite des Trafos ist nun galvanisch getrennt vom Lastkreis, aber SPICE verträgt keine „erdfreien Stromkreise“. Deshalb **muss** (als Gleichstromweg nach Masse bei der Startanalyse!) der **Widerstand R4 = 1 Mega-Ohm eingebaut werden**.
- Was geändert werden **muss**, ist der **Verlauf des Zündimpulses, da der Trafo einen Gleichspannungsanteil im Zündsignal nicht übertragen kann**. Deshalb arbeiten wir nun mit einer symmetrischen Pulsspannung, die die Amplituden „+5V“ und „-5V“ aufweist.
- Eine Spannungsquelle darf bekanntlich nie direkt an eine Induktivität gelegt werden -- das hatten wir schon. Deshalb wird der **Widerstand R5 mit 1Ω zwischen Quelle und Primärseite des Trafos eingefügt**.
- Als Transformator holen wir das vor längerer Zeit selbst erstellte Teil „**xformer\_01**“ aus der Bibliothek und bauen es in die Schaltung ein.