

# LABOR ELEKTRO – CAE

---



Fakultät Technik  
Studiengang Mechatronik

## Labor CAE

**Einführung in die Simulation elektrischer  
Schaltungen mit LTSpice**

**6 UE**

Franz Smagacz-Allramseder

Matthias Holetzko

Andreas Schäfer

23.04.2015

# **Übersicht:**

## **A. Einführung**

Historisches Schaltungssimulation mit LTSpice	Seite 3
Nützliche Voreinstellungen:	Seite 4

## **B. Übung: 1.0 Darstellung von Bauteilekennlinien**

Widerstand	Seite 5
Diode	Seite 6
Transistor	Seite 7
OPV-Schaltungen	Seite 8

## **C. Übung: 2.0 Simulation Temperatur Messbrücke**

Entwurf der Brückenschaltung	Seite 11
Grundlegende Funktionen	Seite 12
Starten der Simulation (Gleichspannungsanalyse)	Seite 12
Simmulationsergebnisse	Seite 13
Anzeigen der Knotennamen	Seite 14
Festlegen/Umbenennen von Knotennamen	Seite 15
Spannungen/Ströme/Leitungen darstellen	Seite 16
Modellierung des KTY81 in LTSpice	Seite 20
Anpassen des Schalzeichens mit dem Symboleditor	Seite 21
Linearisierung/Abgleich der Messbrücke	Seite 23
Dimensionierung des Differenzverstärkers	Seite 24
Einbinden des TS912 in LTSpice	Seite 25
Simulation starten mit neuem Bauelement	Seite 28
Simulation der Gesamtschaltung	Seite 29
Referenzspannungsquelle	Seite 30

## **D. Additional:**

### **Übung: 3.0: Digitale Schaltungssimulation**

Die Umkehrstufe (= NOT oder Inverter)	Seite 31
D-Flipflop	Seite 32

<b>Wichtiger Hinweis: Montecarlo-Analyse</b>	Seite 33
--	----------

#### **Anhang:**

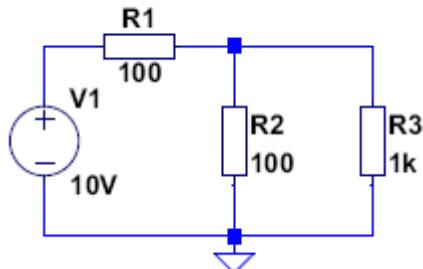
<b>Beispiel: Verwendung der Diode 1N4007</b>	Seite 34
<b>Beispiel: Verwendung der OPVs vom Typ „TL072 und Erstellung eines passenden Spice-Symbols</b>	

# Historisches

## Schaltungssimulation mit LTSpice

Viele Programme zur Simulation elektrischer/elektronischer Schaltungen beruhen auf dem Berechnungsalgorithmus **SPICE (Simulation Program with Integrated Circuit Emphasis)**, der 1973 an der University of California at Berkeley entwickelt wurde. SPICE ist eine “open source” Software (kostenlos), bei der die Eingabe der zu berechnenden Schaltung allerdings sehr mühsam ist und anhand einer speziellen SPICE-Syntax in einer Netzliste erfolgt:

Schaltung:



SPICE-Netzliste

```
* Spannungsteiler1
V1 N001 0 10V
R1 N001 N002 100
R2 N002 0 100
R3 N002 0 1k
.end
```

Viele aktuelle Schaltungssimulationsprogramme nutzen die **SPICE-Syntax** zur Schaltungsbeschreibung und den (inzwischen weiterentwickelten) Berechnungsalgorithmen, bieten aber zusätzlich eine komfortable graphische Benutzeroberfläche zur Schaltplaneingabe. Für unsere Laborübung benützen wir LTSpice.

## LTSpice



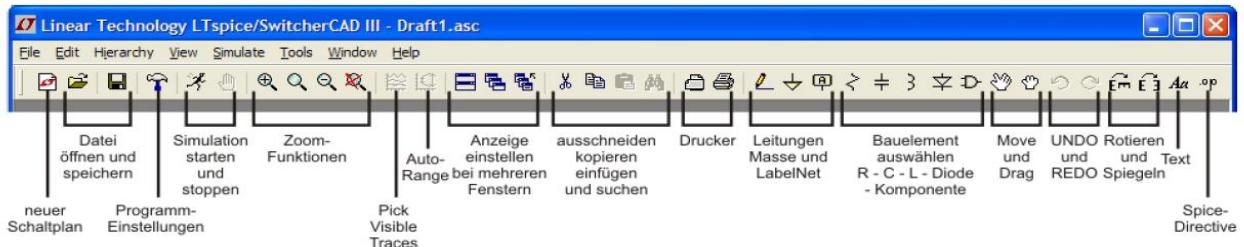
### Eigenschaften

- kostenlos verfügbares Schaltungssimulationsprogramm der Fa. Linear Technology (Halbleiterhersteller)
- keinerlei Einschränkungen im Funktionsumfang
- keine Begrenzung der Schaltungsgröße
- relativ leichte Einarbeitung und intuitive Bedienung
- die meisten Simulationseinstellungen sind direkt auf der Schaltplanseite sichtbar  
(→ keine „versteckten Spezialoptionen“ in Unter-unter-unter-Menüs ...)
- wird sehr aktiv weiterentwickelt (regelmäßige (Online-)Updates)
- umfangreiche Online-Hilfe verfügbar: <http://ltwiki.org>
- sehr aktives Online-Forum: <http://tech.groups.yahoo.com/group/LTspice/>
- viele Beispielsimulationen enthalten

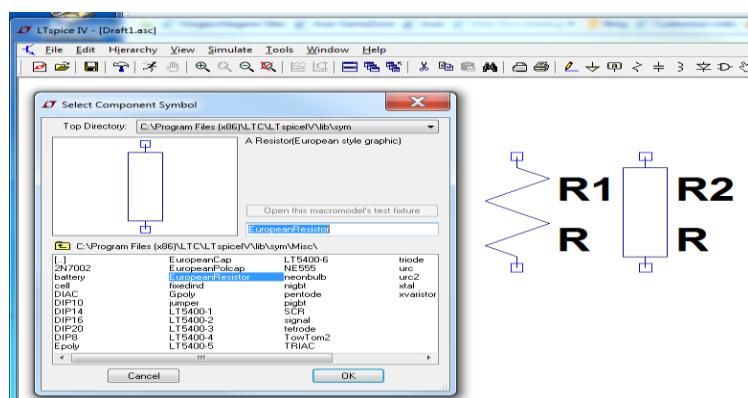
Download der Installationsdatei unter <http://www.linear.com/design-tools/software/>  
(LTspiceIV.exe, 12,8 MB)

**Vorwort:** In dieser Laborübung „Simulation mit „LTSpice“ sollen Sie sich die Grundlagen erarbeiten, mit denen Sie dann spätestens bei Ihrer Studienarbeit Ihre eigenen Schaltungen simulieren und das Ergebnis in Ihrer Arbeit dokumentieren können. Diese Grundlagen werden wir uns mit der Simulation „Kleinprojekt: Temperaturmessbrücke“ erarbeiten. Sie werden lernen wie man eigene Modelle einbindet und dass man mit „SPICE“ auch digitale Schaltungen simulieren kann.

Nützliche Voreinstellungen:



Hauptmenüleiste



Das wie bei uns in Europa übliche,



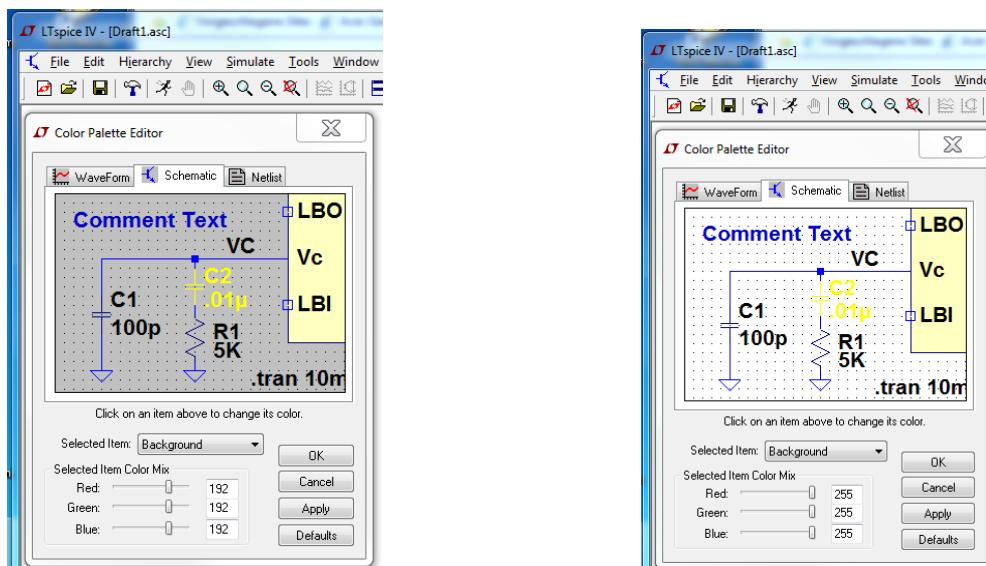
Rechteck für Widerstände ist über "Umwege" auch erreichbar:  
Component / Misc / EuropeanResistor )

Es ist auch möglich diese Einstellung dauerhaft vorzunehmen. **NICHT an den DHBW Rechnern!**

einfachste Methode ...

kopiere aus dem Symbolverzeichnis des Installationspfades (\LTC\LTspiceIV\lib\sym\Misc) den europäischen **Widerstand** (EuropeanResistor.asy) und füge ihn ein Verzeichnis höher (\LTC\LTspiceIV\lib\sym) wieder ein ... benenne den amerikanischen Transistor (res.asy) um in res.asy.old oder wie auch immer und benenne den EuropeanResistor.asy in res.asy um.

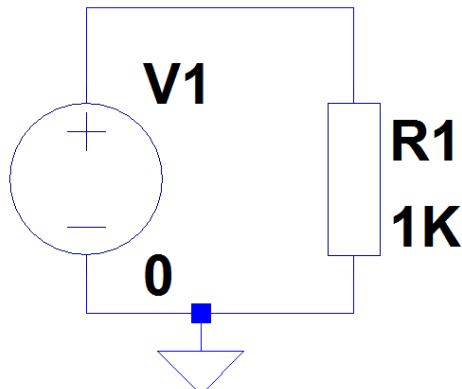
Es muss auch nicht immer diese braune Hintergrundfarbe sein!



# Übung 1: Darstellung von Bauteil-Kennlinien

## Ohm'scher Widerstand

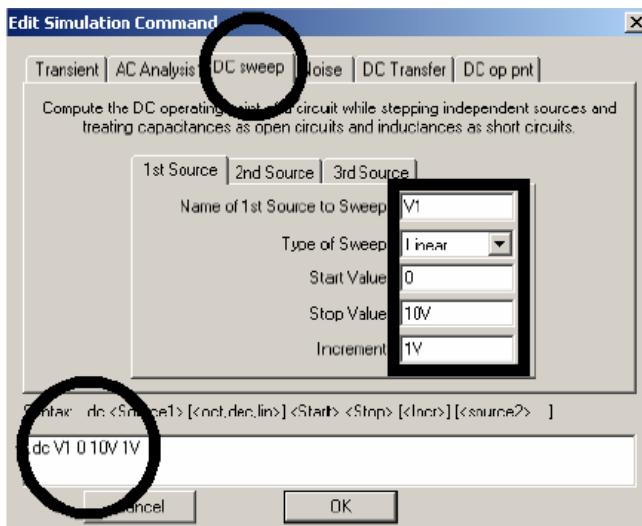
Dieses Bauteil eignet sich besonders gut zur Demonstration des Vorgehens, wenn U-I-Kennlinien sichtbar gemacht werden sollen.



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

Beim Widerstand „R1“ wählen wir einen Wert von 1k Ohm. 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**

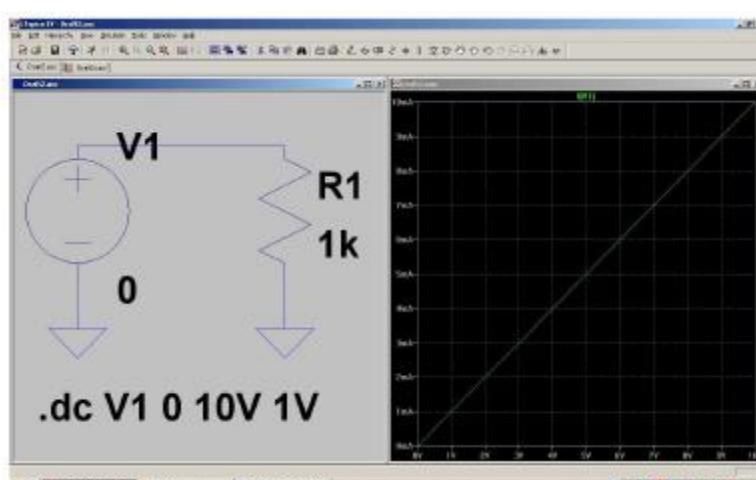
**Endwert = 10 Volt**

**Schrittweite = 1V**

**Linearer Sweep**

Das damit erzeugte Simulation Command „**.dc V1 0 10V 1V**“ findet sich im unteren Feld und sollte nochmals kontrolliert werden.

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.



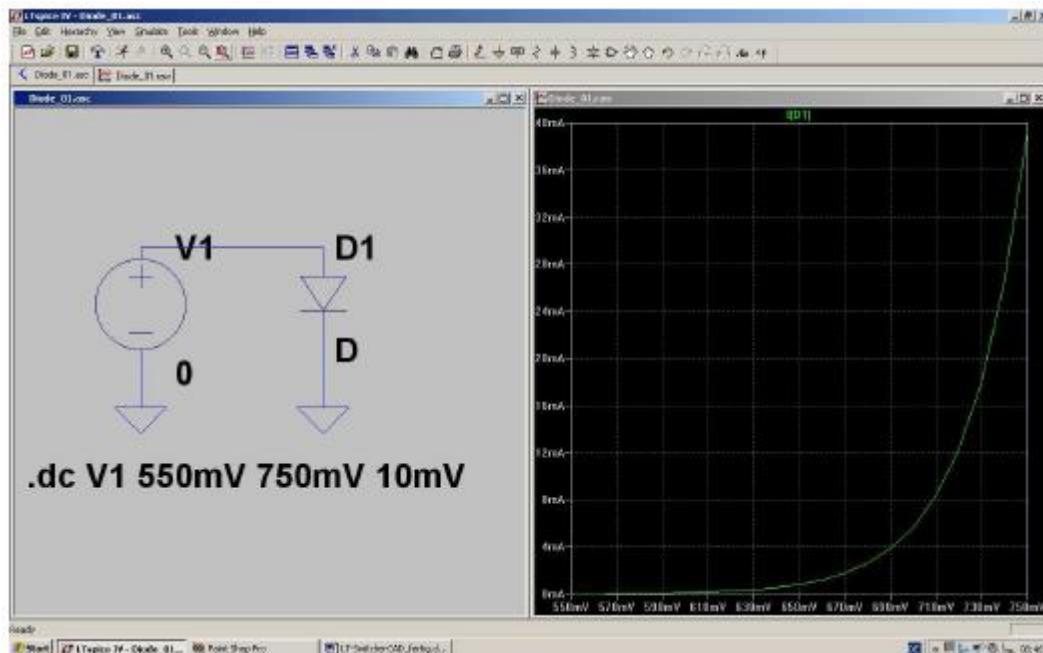
## Diode

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

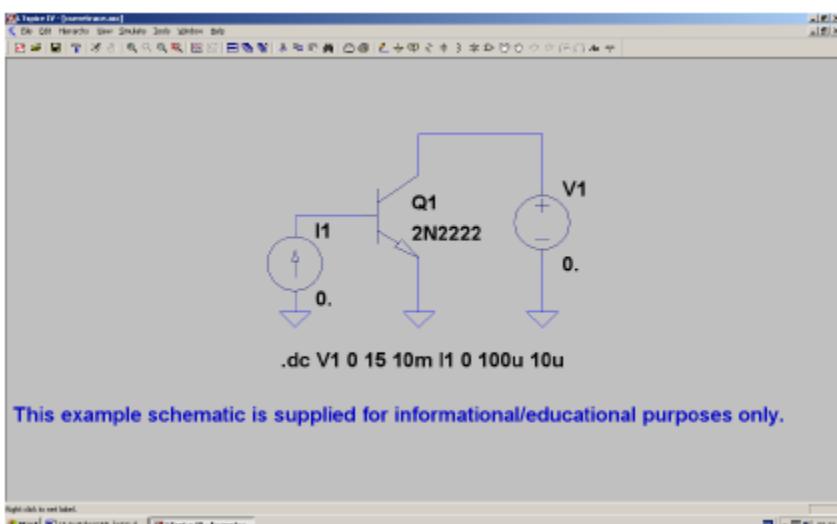
**.dc V1 550mV 750mV 10mV**

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



## NPN-Transistor

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



Man erkennt:

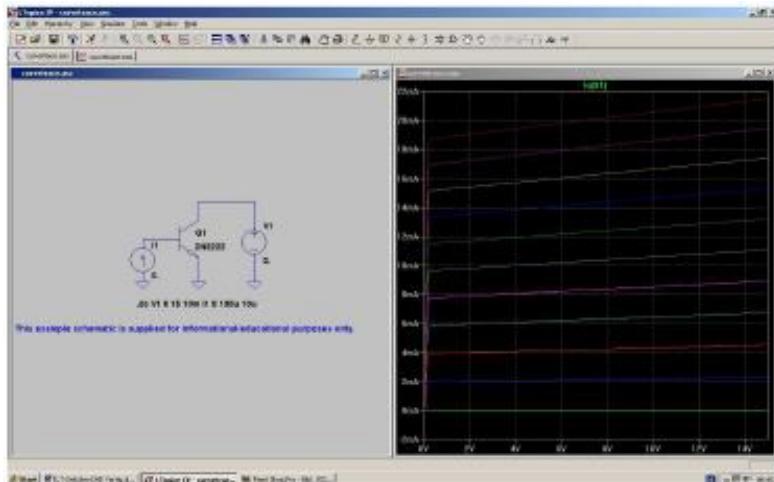
- Am Kollektor des Transistors wird die Gleichspannungsquelle V1 angelegt. Sie liefert zunächst Null Volt.
- Eine Konstantstromquelle I1 speist die Basis des Transistors. Auch ihr Startwert ist „Null“

Das Simulation Command

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

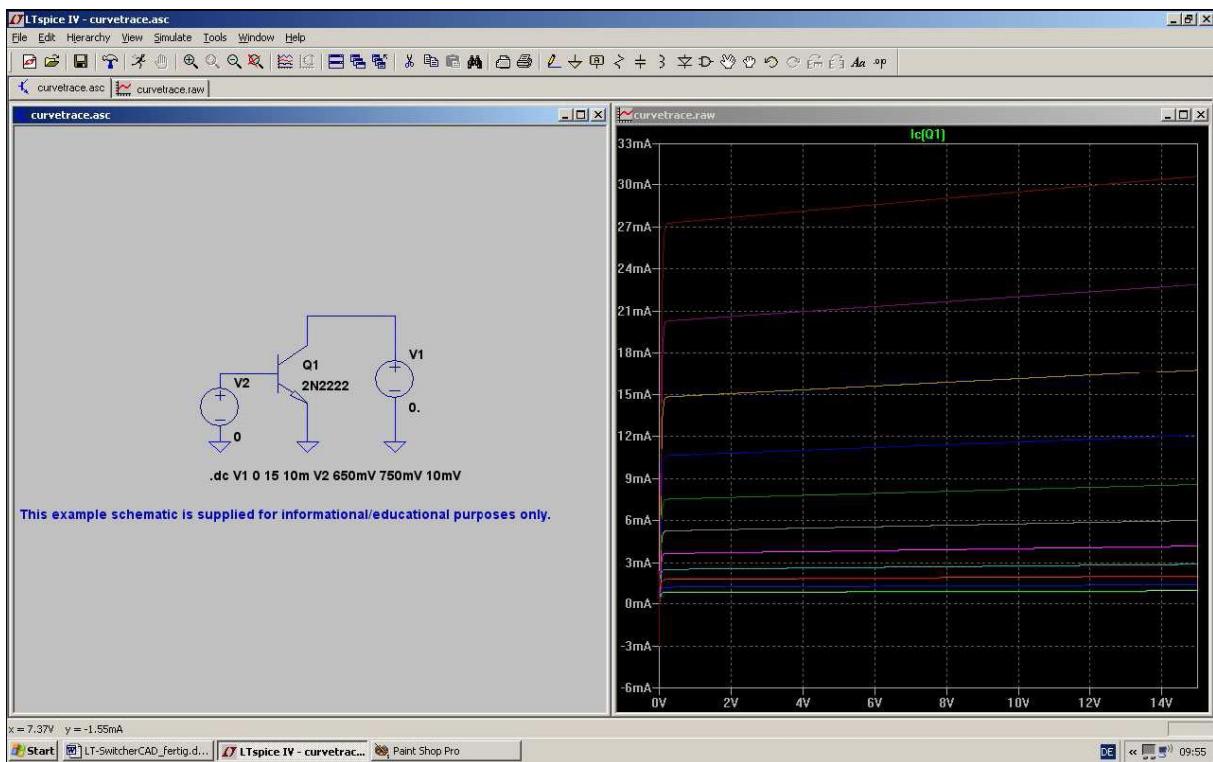
besagt:

**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:



- a) Die Stromquelle an der Basis wird durch eine Spannungsquelle ersetzt. Ihr Startwert ist „Null“
- b) 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.

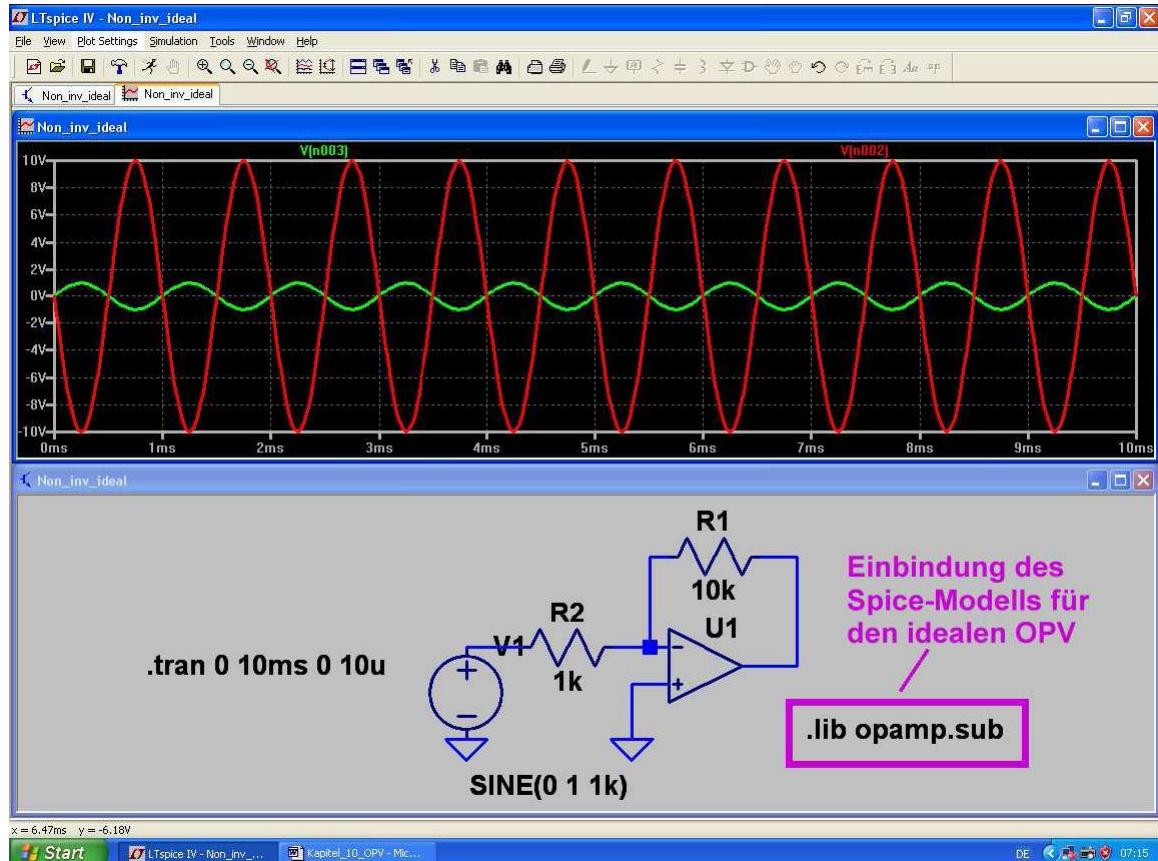
Ergebnis: Siehe oben...

## OPV-Schaltungen

### Einstieg: Umkehrender Verstärker

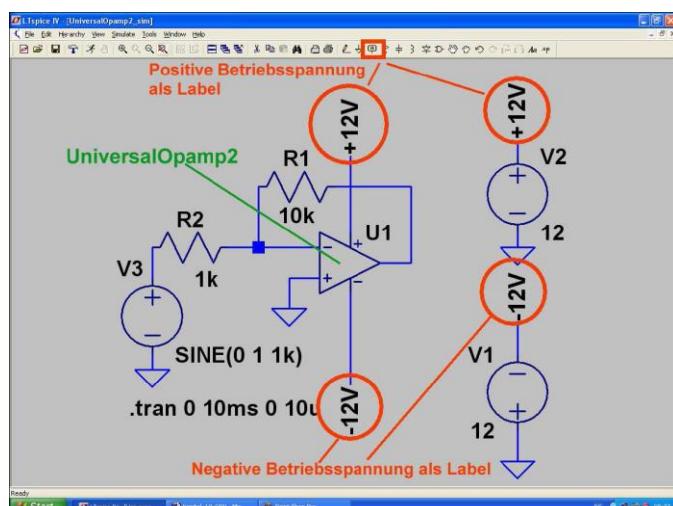
#### Verwendung des einfachsten OPV-Modells „opamp“

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 = 11V / Frequenz = 1kHz). Allerdings darf man die Spice-Direktive zur Einbindung der Modelldatei nicht vergessen:



### 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.

## 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.

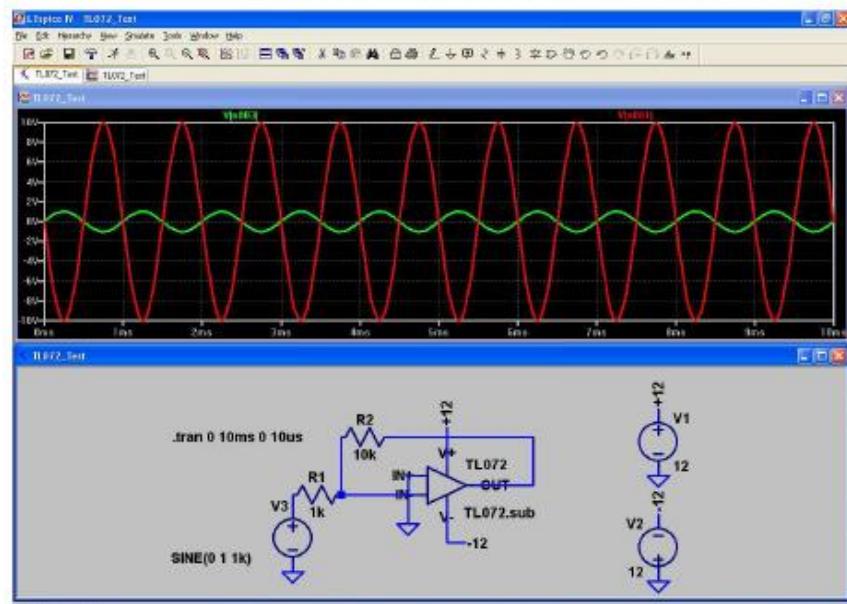
### 1. Schritt:

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.

**Oder: Sie benützen die Schritt für Schritt Anweisung im Anhang:**

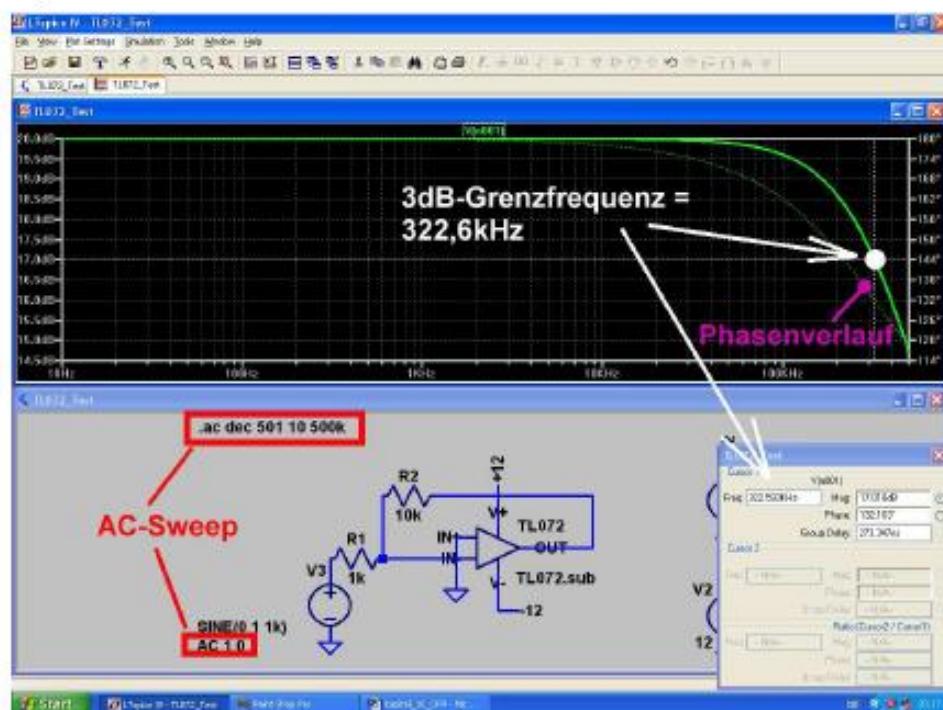
**Dies ist eine Zusatz-Aufgabe die Sie nur dann durchführen sollten, wenn Sie zu OPS schon eine Vorlesung hatten!**

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



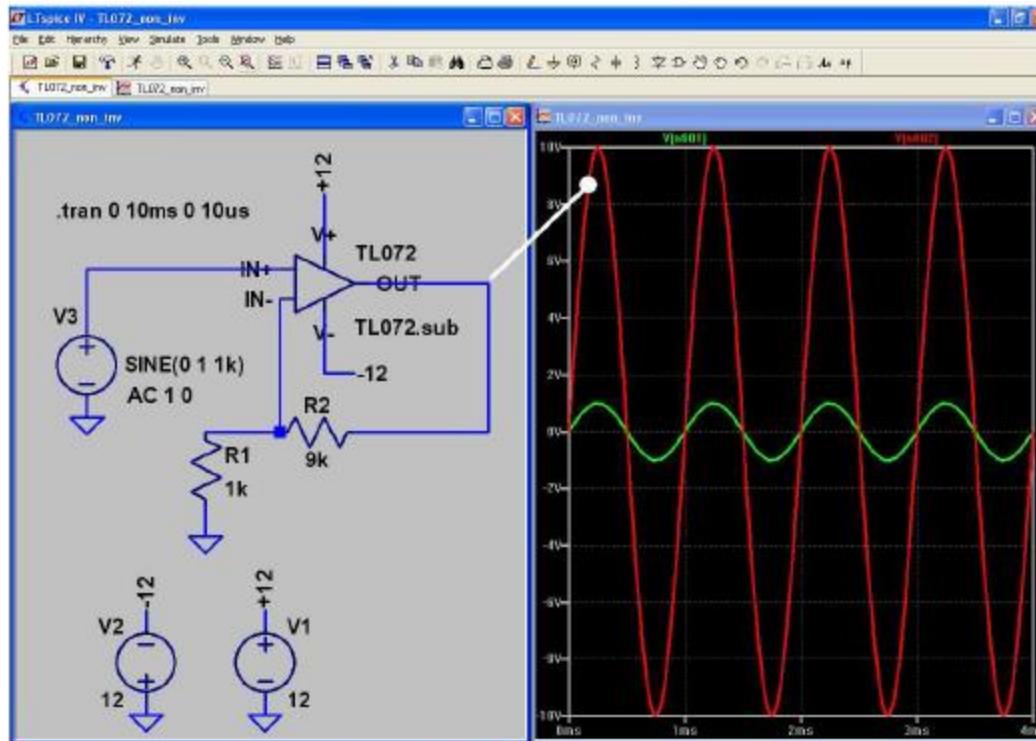
Interessant dürfte dagegen ein AC-Sweep sein, der das Verhalten bei höheren Frequenzen bzw. die obere Grenzfrequenz zeigt. Bitte mal simulieren!

Ergebnis:



## Nicht umkehrender Verstärker mit TL072

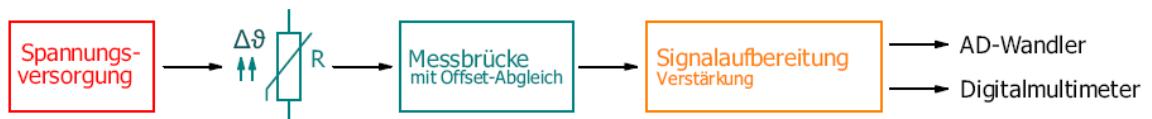
Gefordert wird wieder 10fache Verstärkung, aber der Einsatz einer nicht umkehrenden Schaltung. Dadurch erreicht man einen sehr hohen Eingangs-widerstand:



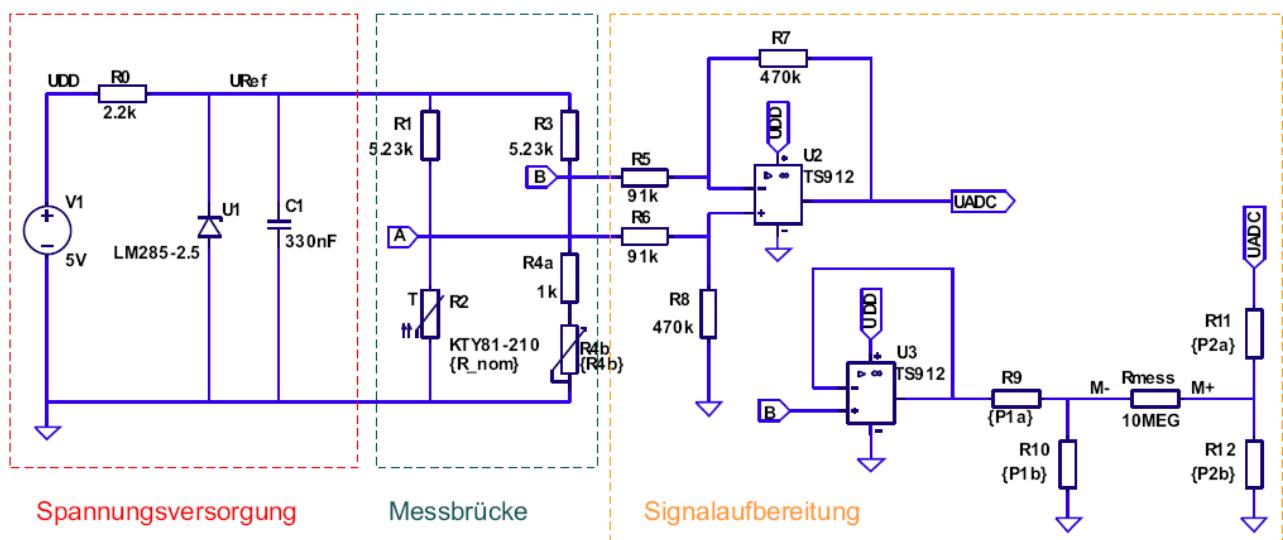
## Übung 2: Simulation Temperatur Messbrücke

### Kleinprojekt: Temperaturmessbrücke

#### Blockschaltbild

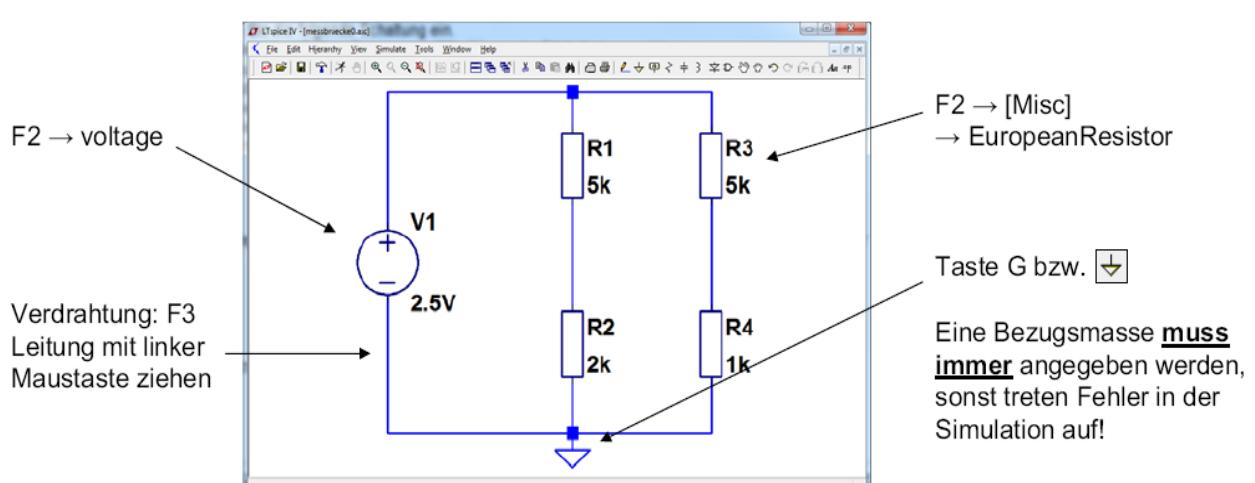


#### Schaltplan



### Entwurf der Brückenschaltung

- Starten Sie LTSpice IV
- Legen Sie eine neue Simulationsdatei mit dem Namen messbruecke0.asc an [File → New Schematic  , File → Save As ... → messbruecke0.asc]
- Geben Sie die folgende Schaltung ein.  
Bauteile mit → Edit → Component (F2) aus der Bibliothek auswählen  
Spannungsquelle: *voltage*  
Widerstände: *res (R)* oder → [Misc] → EuropeanResistor  
Massesymbol: → Edit → Place GND (G)
- Bauteilewerte editieren: Rechtsklick auf das Schaltsymbol (Dezimalpunkt statt Komma bei Zahlenwerten!)
- Verdrahtung mit → Edit → Draw Wire (F3) zeichnen



## Grundlegende Editierfunktionen

- Auswählen von Bauteilen aus der **Bibliothek**: F2
- **Verdrahtung** zeichnen: F3
- **Verschieben** von Bauteilen: Verschiebemodus aktivieren mit F7 oder F8
- **Rotieren** von Bauteilen: Strg + R (ggf. vorher mit F7 oder F8 + linker Mausklick auswählen)
- **Spiegeln** von Bauteilen: Strg + E (ggf. vorher mit F7 oder F8 + linker Mausklick auswählen)
- **Löschen** von Bauteilen: F5 (→ Scherensymbol ) + linker Mausklick auf Bauteil
- **Kopieren** von Bauteilen: F6 (→ Kopiersymbol ) + linker Mausklick auf Bauteil, dann verschieben
- Aktion **rückgängig** machen: F9 (Undo)
- Ansicht **vergrößern**: Strg + Z
- Ansicht **verkleinern**: Strg + B
- Schaltungsgröße an **Fenstergröße** anpassen: Leertaste
- **Verlassen** eines Bearbeitungsmodus: Esc-Taste

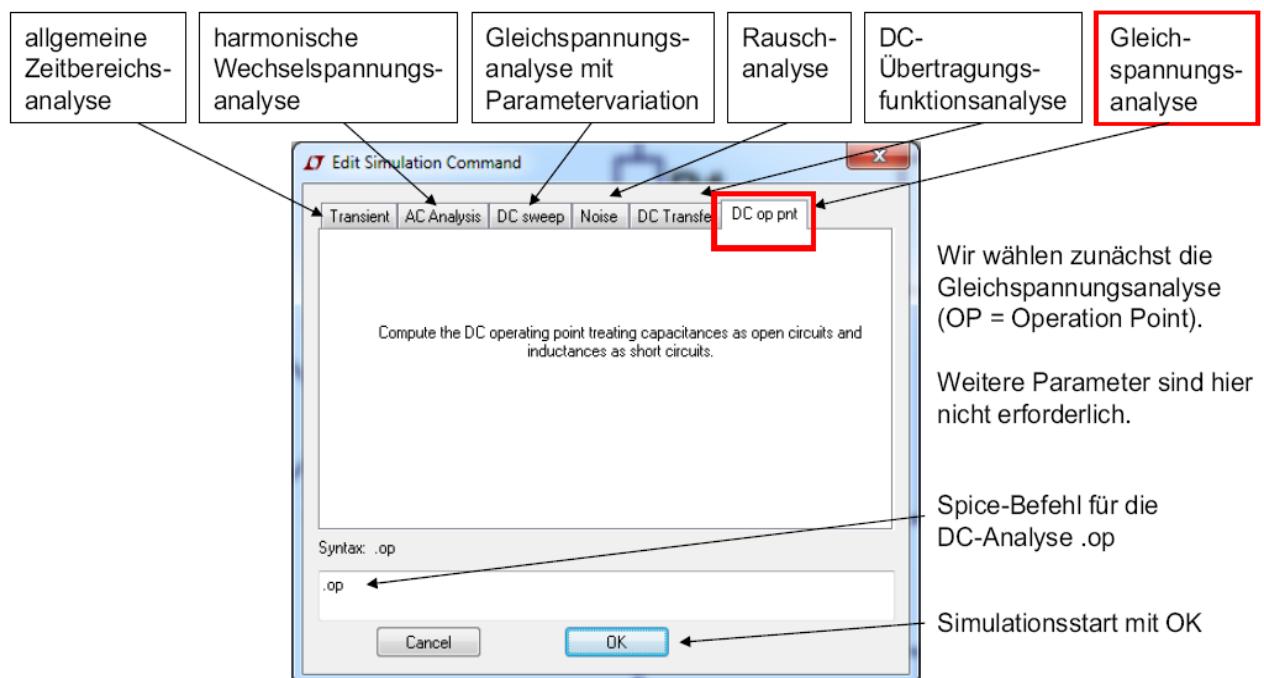
### Hinweise:

- Zahlenwerte immer mit Dezimalpunkt statt Komma eingeben!
- kein Leerzeichen Zwischen Zahlenwert und Einheit, also „2.5V“ statt „2.5 V“
- SPICE arbeitet nach dem **Knotenpotenzialverfahren** – Es **muss immer** ein Bezugsknoten (**Masse, Ground**) angegeben werden, sonst treten Simulationsfehler auf (Achtung: Es erfolgt keine Fehlermeldung!).

## Starten der Simulation (Gleichspannungsanalyse)

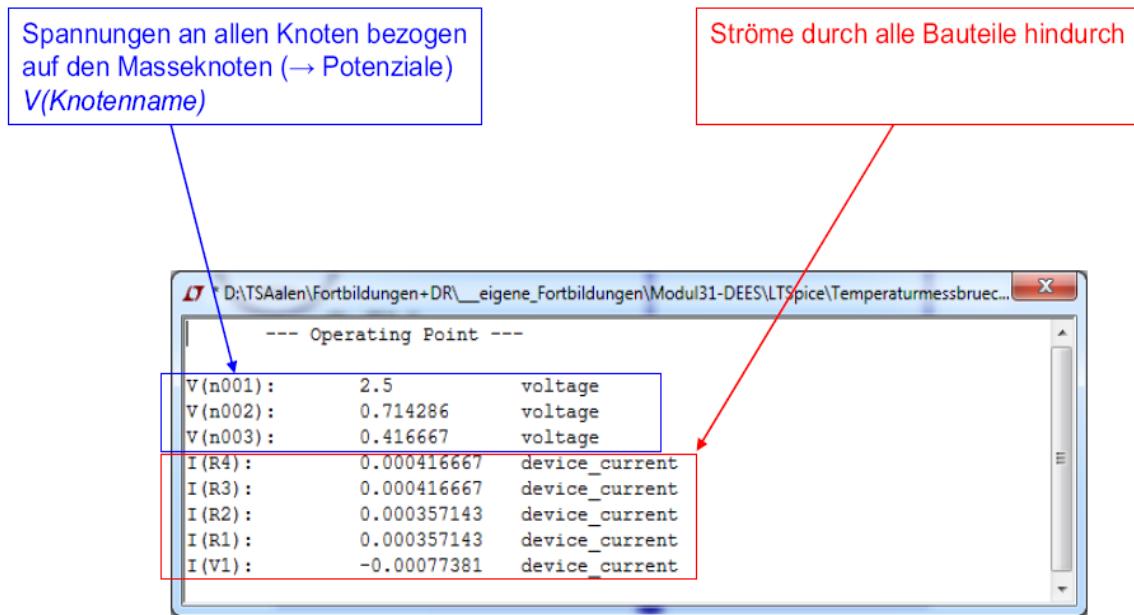
Eine Simulation wird mit → Simulate → Run (Tab Taste) gestartet.

(nur) beim erstmaligen Starten erscheint das Simulationsmenü:(→ Simulationsart, -parameter festlegen):



## Simulationsergebnisse

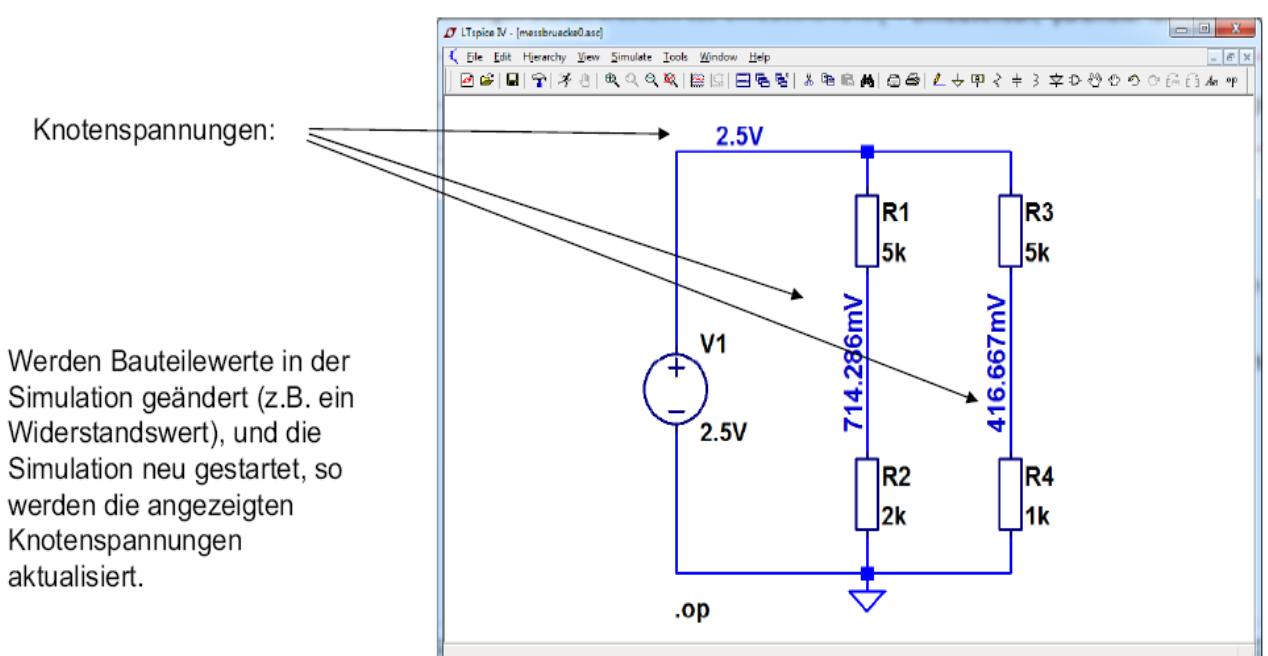
Nach der Simulation erscheint ein Ergebnisfenster, in dem alle Spannungen (Potenziale aller Knoten bezogen auf Masse) und alle Ströme angegeben sind:



## Simulationsergebnisse

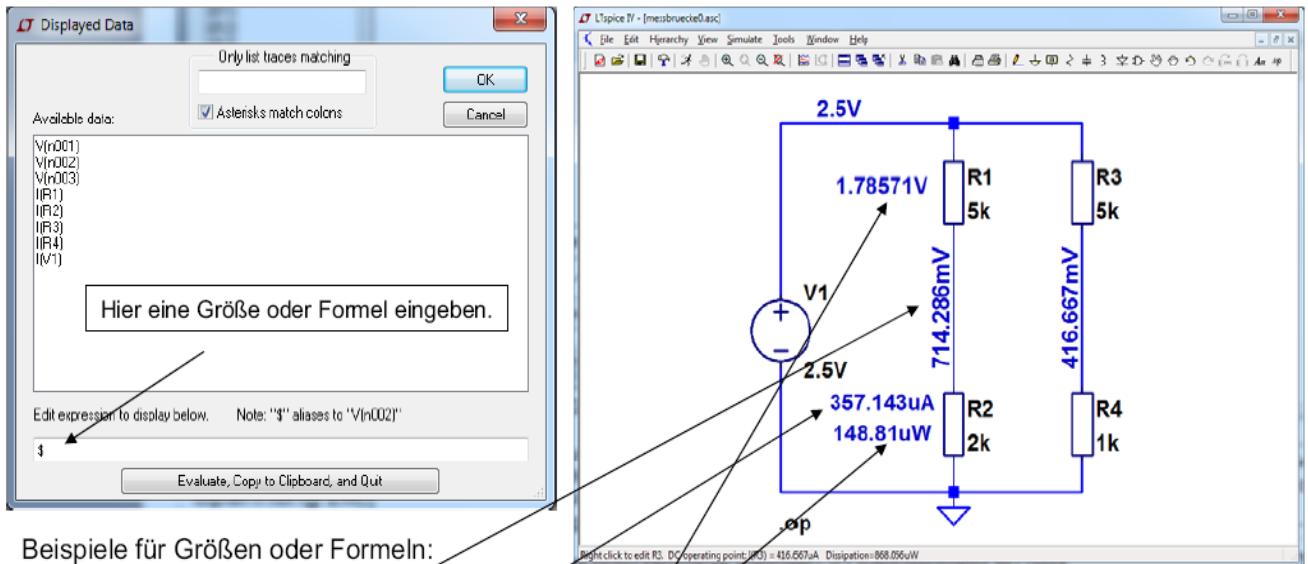
Alternativ können Simulationsergebnisse auch direkt im Schaltplan angezeigt werden.

Hierfür das Simulationsergebnisfenster **nach der Simulation** wieder schließen und im Schaltplan den gewünschten Knoten **zweimal** anklicken:



## Simulationsergebnisse

Sollen statt der Knotenspannungen andere Größen angezeigt werden (z.B. Ströme oder Verlustleistungen in einzelnen Bauteilen), so kann man mit der rechten Maustaste auf einen angezeigten Knotenspannungswert klicken und im „Display Data“ Menü eine beliebige Größe oder eine Formel eingeben:



Beispiele für Größen oder Formeln:

Knotenspannung:  $V(n002)$

Spannung zwischen zwei Knoten:  $V(n001, n002)$

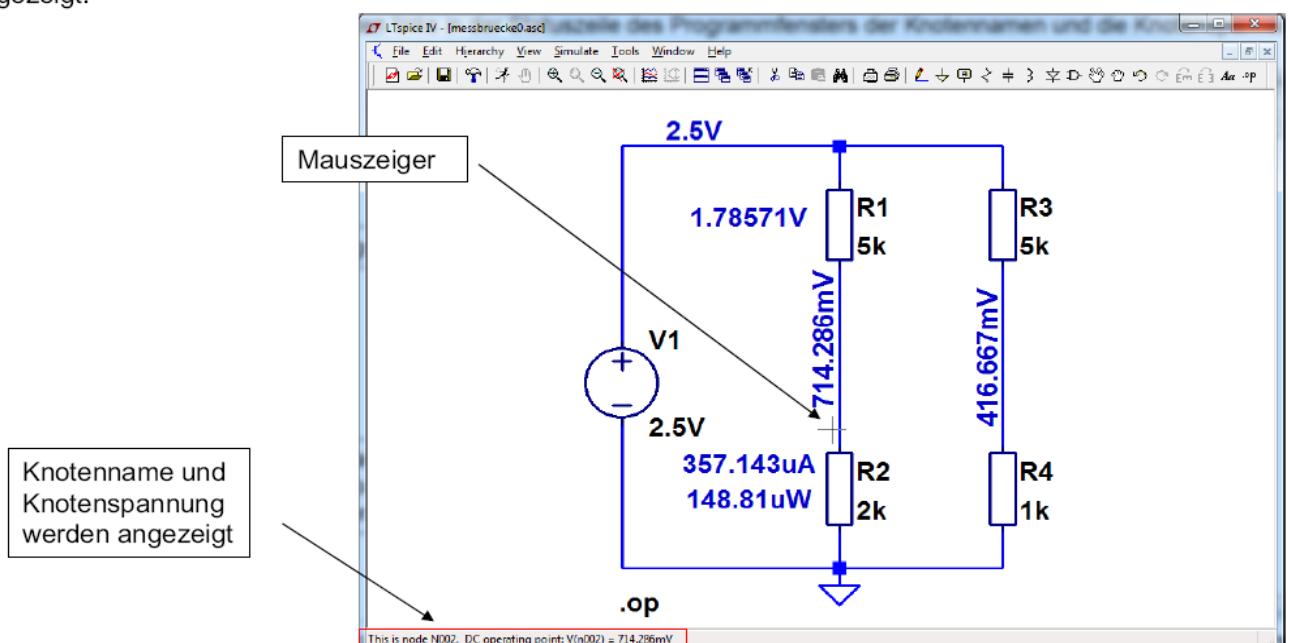
Strom (hier durch R2):  $I(R2)$

Verlustleistung (hier an R2):  $V(n002) * I(R2)$

Die simulierten Werte können anschließend im Schaltplan beliebig verschoben (F7) oder gedreht (Strg + R) werden.

## Anzeigen der Knotennamen

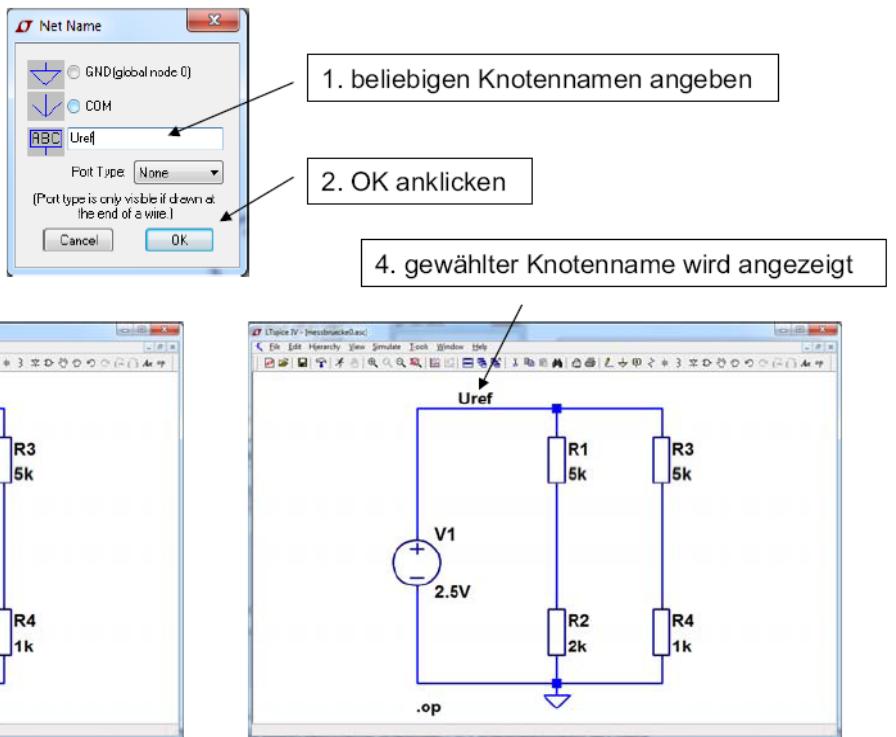
LTSpice nummeriert alle Knoten in einer Schaltung automatisch durch (n001, n002, ...). Der Masseknoten hat immer die Nummer 0 (nicht n000). Die Nummer bzw. den Namen eines Knotens erfährt man, indem man im Schaltplan mit der Maus (Fadenkreuz) an den jeweiligen Knoten geht (ohne Klicken). Dann wird unten links in der Statuszeile des Programmfensters der Knotennamen und die Knotenspannung angezeigt.



## Festlegen/Umbenennen von Knotennamen

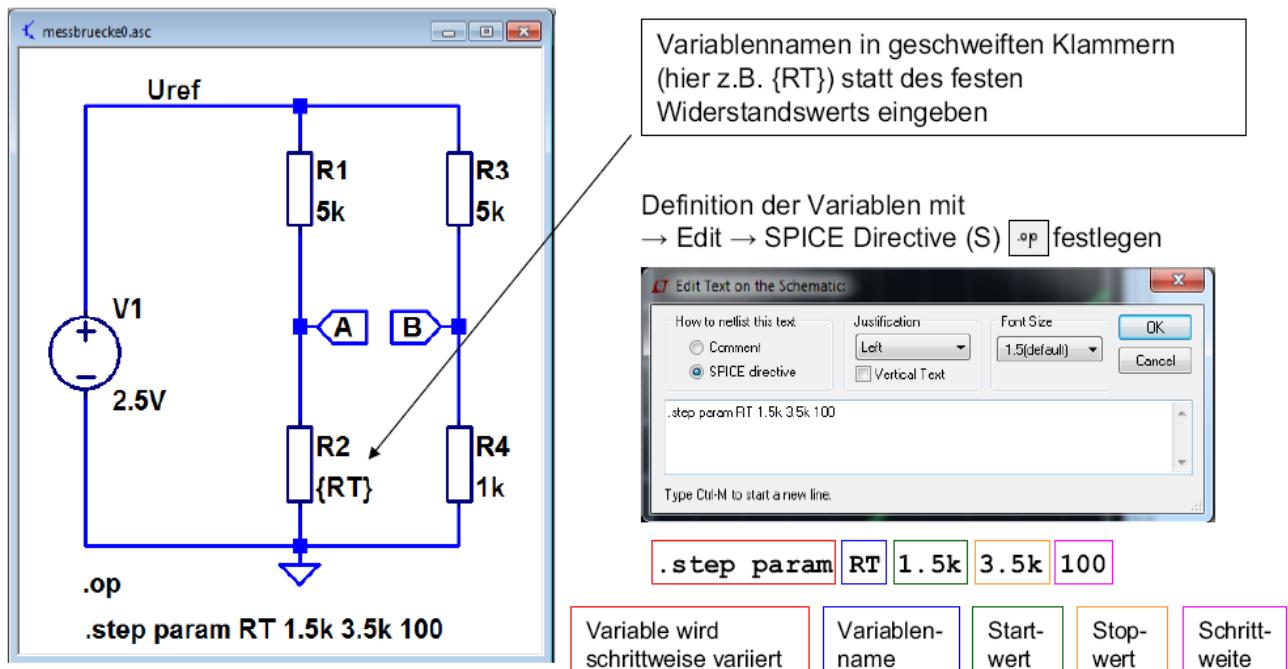
Namen von einzelnen Knoten können bereits vor der Simulation festgelegt werden. So können aussagekräftige Namen gewählt werden (z.B. Uref), die sich später leichter zuordnen lassen.

→ Edit → Label Net (F4)  :



## Variation von Bauteilwerten

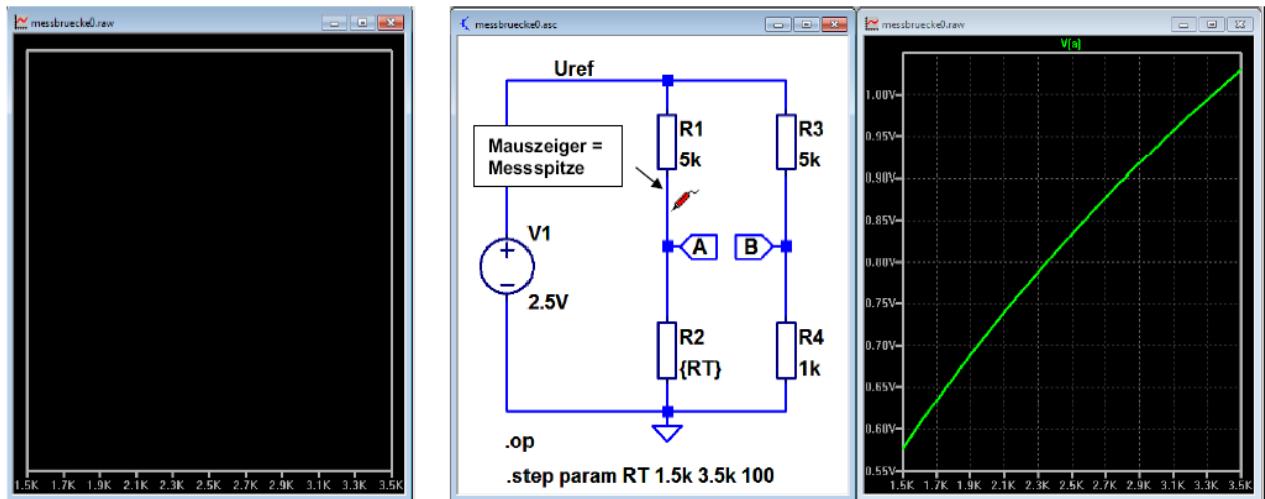
In der Messbrücke soll der Widerstand R2 durch einen temperaturabhängigen PTC-Widerstand ersetzt werden. Wir nehmen zunächst einmal an, dass der Widerstand des PTCs sich bei einer Temperaturänderung zwischen 0°C und 100°C von 1,5 kΩ auf 3,5 kΩ ändert. Um dies zu simulieren, geben wir für den Widerstand R2 keinen festen Wert ein, sondern legen einen Variablennamen fest. Variablen werden in (LT)Spice durch geschweifte Klammern {} gekennzeichnet:



## Knotenspannungen in Diagrammen darstellen

Nach der Simulation  
(mit variablem RT) erscheint ein  
leeres Diagrammfester:

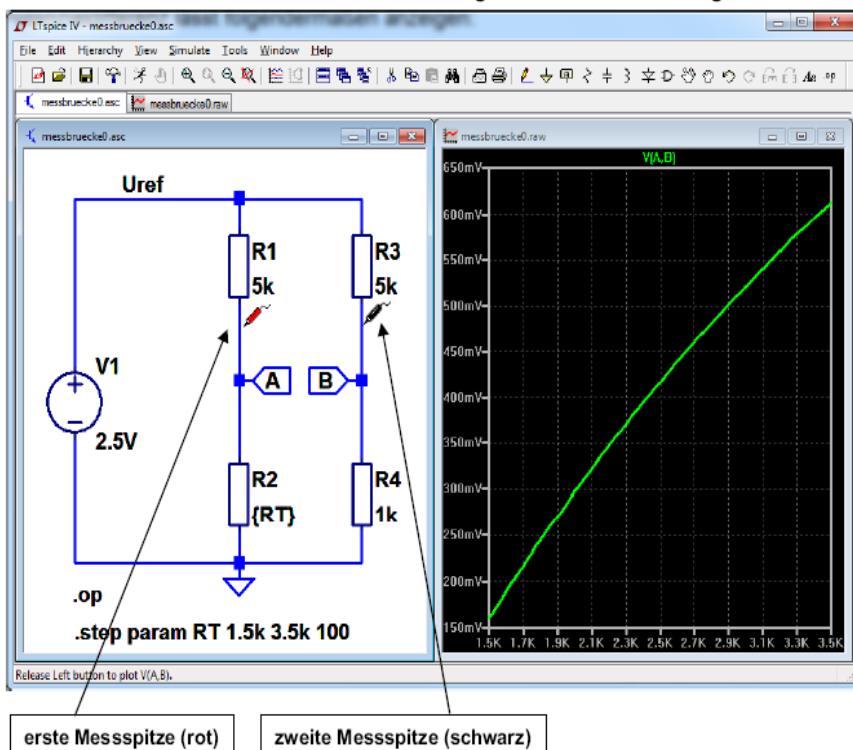
Bewegt man nun im Schaltplanfenster den Mauszeiger auf einen Knoten, so verwandelt sich der Zeiger in eine Messspitze. Beim Anklicken des Knotens erscheint die Knotenspannung als Kurve im Diagramm.



Knotenspannung  $V(A)$  wird in  
Abhängigkeit von  $RT$  als Kurve  
angezeigt

## beliebige Spannungen in Diagrammen darstellen

In der Messbrücke interessiert die Brückenspannung  $U_{AB}$ , die jedoch keine Knotenspannung zwischen einem Knoten und der Masse ist, sondern eine Spannung zwischen zwei Knoten (Potenzialdifferenz). Eine solche Potenzialdifferenz lässt folgendermaßen anzeigen:



1. Knoten im Schaltplan anklicken  
(→ rote Messspitze), dann linke  
Maustaste gedrückt halten und  
Messspitze zum zweiten Knoten  
ziehen (→ schwarze Messspitze).

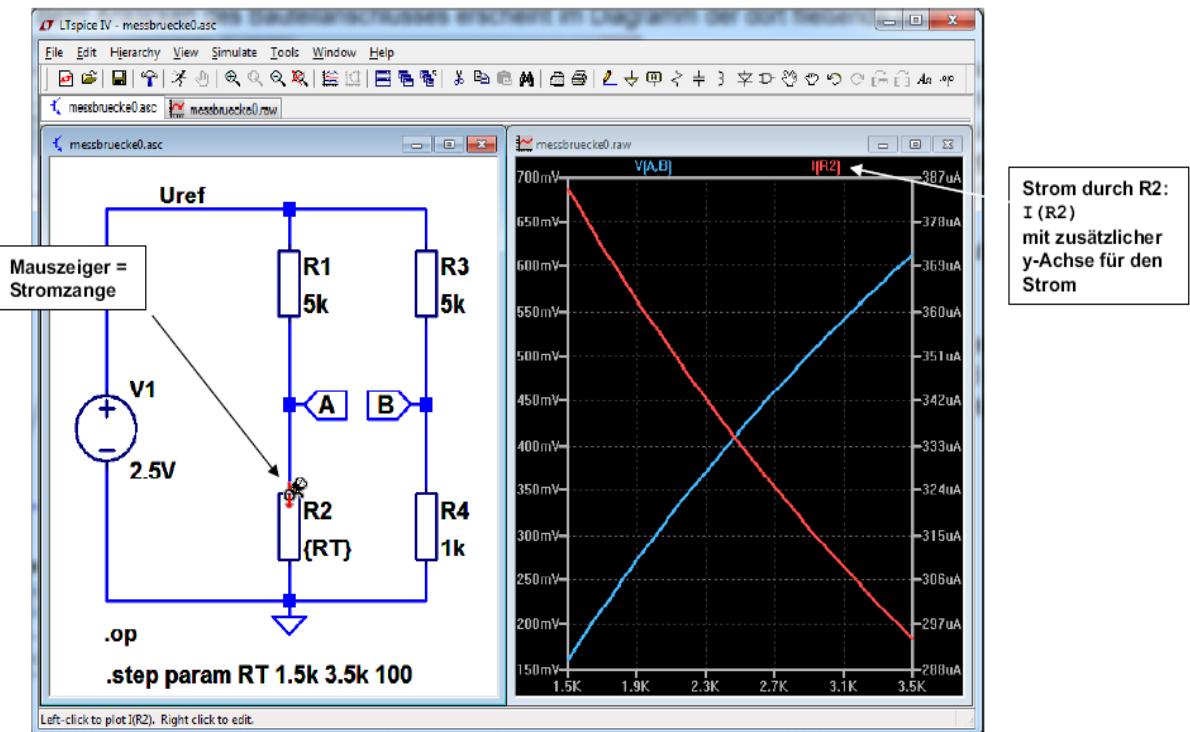
Am zweiten Knoten die Maustaste  
wieder loslassen.

Die Spannung (Potenzialdifferenz)  
zwischen dem ersten (A) und dem  
zweiten (B) Knoten wird angezeigt:

$V(A, B)$

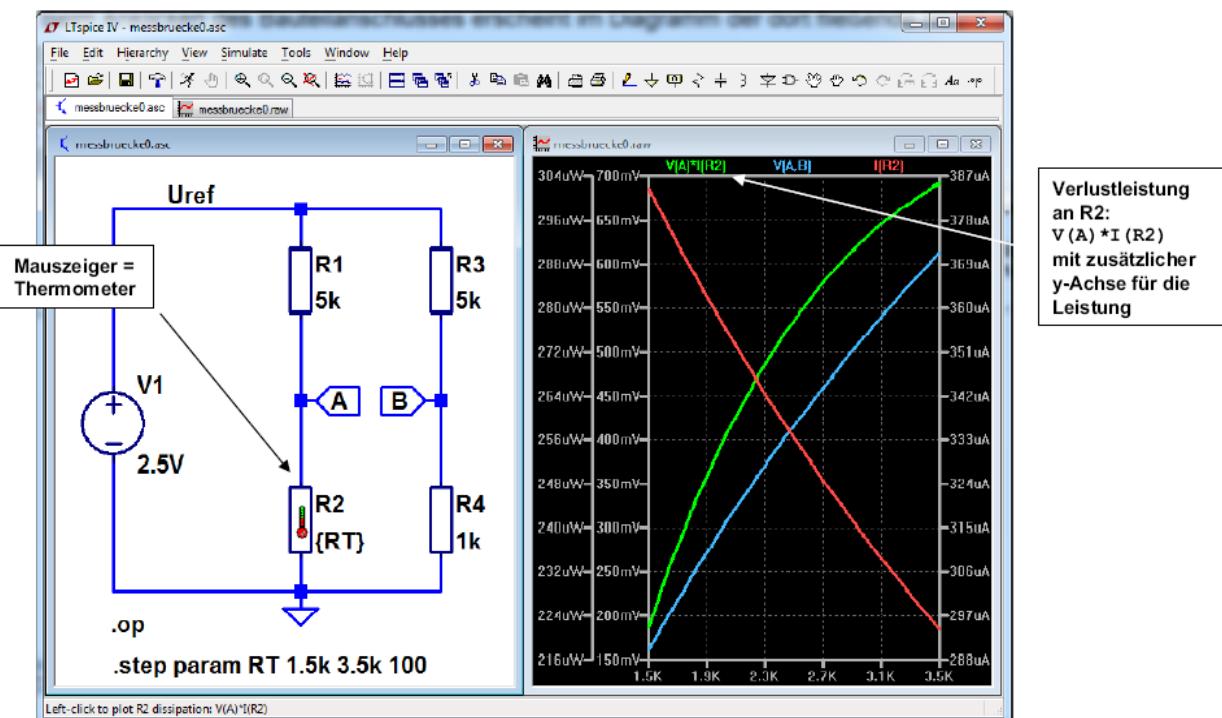
## Ströme in Diagrammen darstellen

Ströme können in Diagrammen dargestellt werden, indem man nach der Simulation mit dem Mauszeiger im Schaltplan auf einen Bauteilanschluss fährt. Dann verwandelt sich der Zeiger (Fadenkreuz) in eine Stromzange. Beim Anklicken des Bauteilanschlusses erscheint im Diagramm der dort fließende Strom.



## Leistungen in Diagrammen darstellen

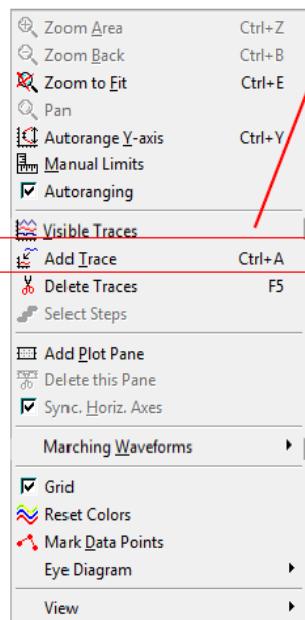
An einzelnen Bauteilen umgesetzte Leistungen können in Diagrammen dargestellt werden, indem man nach der Simulation mit dem Mauszeiger im Schaltplan **bei gleichzeitig gedrückter ALT-Taste** über das jeweilige Bauteil fährt. Dann verwandelt sich der Zeiger (Fadenkreuz) in ein Thermometer. Beim Anklicken des Bauteilanschlusses (ALT-Taste noch gedrückt) erscheint im Diagramm die umgesetzte Leistung.



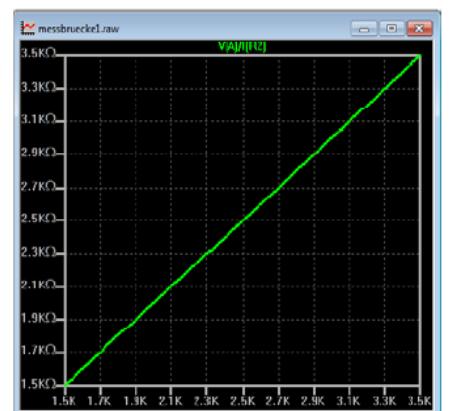
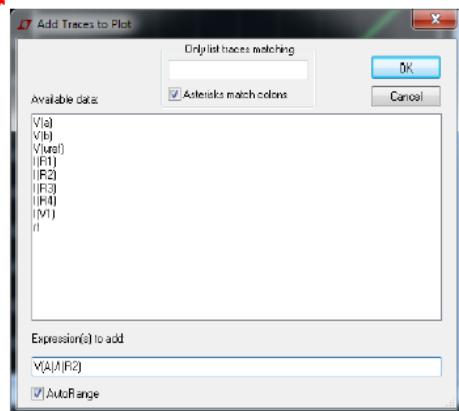
## Beliebige Größen in Diagrammen darstellen

Statt der Spannungen, Ströme oder Leistungen können auch beliebige andere Größen in einem Ergebnisdiagramm dargestellt werden. Mit einem Rechtsklick in der Diagrammfläche öffnet man das **Diagramm-Kontextmenü** und kann dort die Option **Add Trace** auswählen.

Diagramm-Kontextmenü



Add Traces to Plot Menü

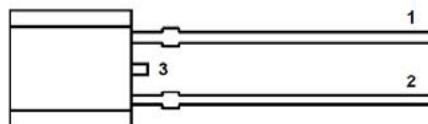


Eingabe von Formeln möglich, hier z.B.:  $V(A) / I(R2)$   
entspricht dem Widerstandswert von R2.

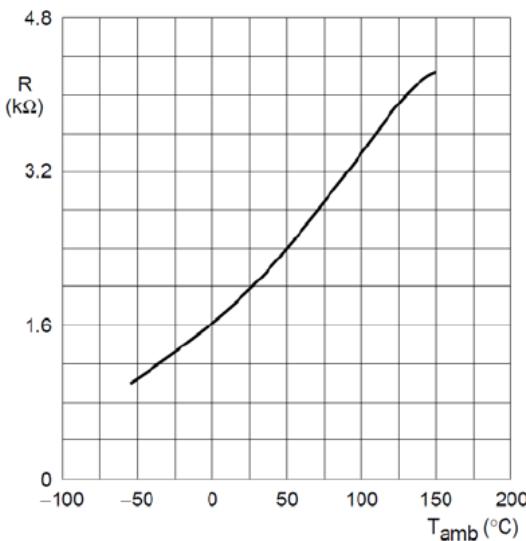
Die Formel wird oben im Diagrammfenster angezeigt und die y-Achse mit der passenden Größe/Einheit skaliert (hier in kΩ).

## Temperaturabhängiger Widerstand KTY81-210

Zur Temperaturmessung soll der Halbleiter-Temperatursensor KTY81-210 verwendet werden.



PIN	DESCRIPTION
1	electrical contact
2	electrical contact
3	not to be connected to a potential



AMBIENT TEMPERATURE (°C)	TEMP. COEFF. (%/K)	KTY81-210		
		MIN.	TYP.	MAX.
-55	-67	0.99	951	980
-50	-58	0.98	1000	1030
-40	-40	0.96	1105	1135
-30	-22	0.93	1218	1247
-20	-4	0.91	1338	1367
-10	14	0.88	1467	1495
0	32	0.85	1603	1630
10	50	0.83	1748	1772
20	68	0.80	1901	1922
25	77	0.79	1980	2000
30	86	0.78	2057	2080
40	104	0.75	2217	2245
50	122	0.73	2383	2417
60	140	0.71	2557	2597
70	158	0.69	2737	2785
80	176	0.67	2924	2980
90	194	0.65	3118	3182
100	212	0.63	3318	3392
110	230	0.59	3523	3607
120	248	0.53	3722	3817
125	257	0.49	3815	3915
130	266	0.44	3901	4008
140	284	0.33	4049	4166
150	302	0.20	4153	4280

## Modellierung des KTY81-210

### TEMPERATURE DEPENDENCY

$$R_T = R_{\text{ref}} \left[ 1 + A(T - T_{\text{ref}}) + B(T - T_{\text{ref}})^2 - C(T - T_I)^D \right]$$

where:

$R_T$  is resistance at temperature  $T$

$R_{\text{ref}}$  is the nominal resistance at the reference temperature ( $T_{\text{ref}}$ )

$T_{\text{ref}}$  is reference temperature (100 °C for the KTY84, 25 °C for all other types)

$A, B$  are type-dependent coefficients.

$T_I$  is temperature above which the slope of the characteristic curve starts to decrease (point of inflection).

$C, D$  are type-dependent coefficients.

$C$  is 0 for  $T < T_I$ .

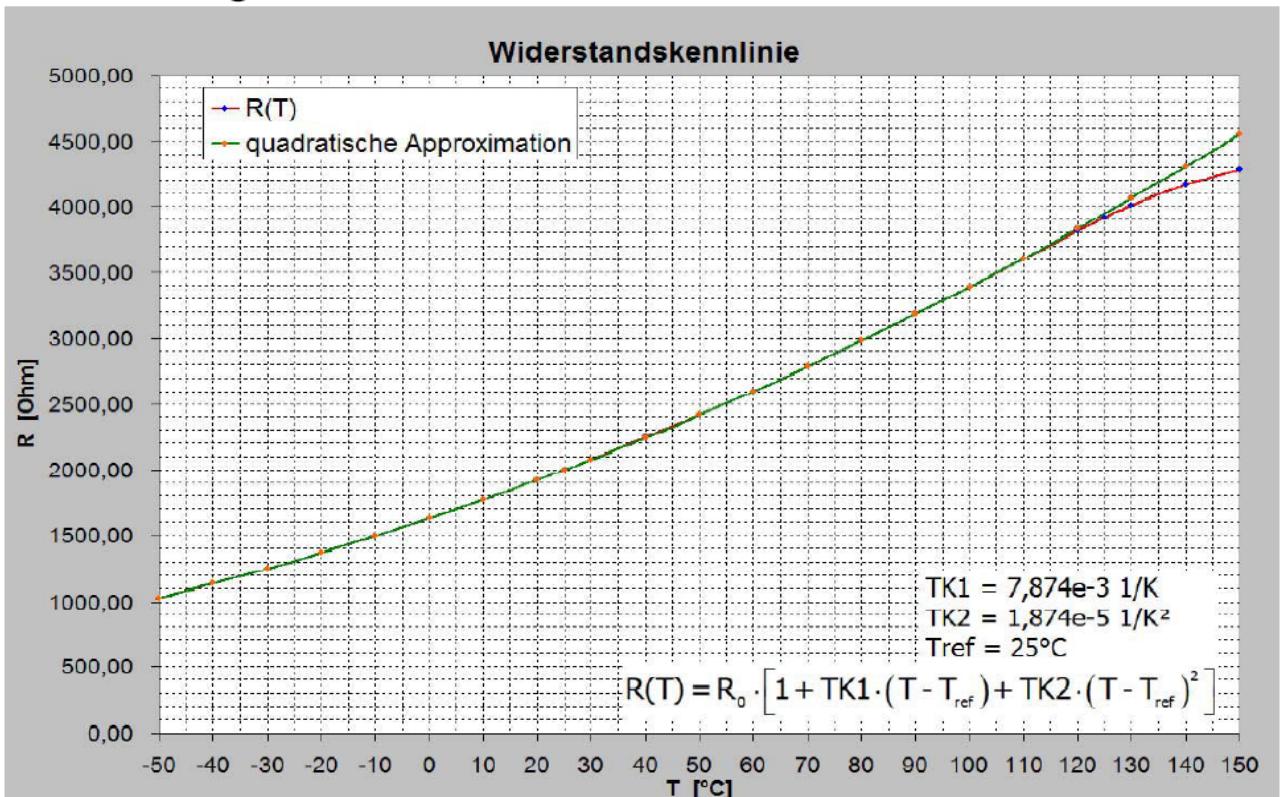
Hersteller-Datenblatt

SENSOR TYPE	A (K <sup>-1</sup> )	B (K <sup>-2</sup> )	C <sup>(1)</sup> (K <sup>-D</sup> )	D	T <sub>I</sub> (°C)
KTY81-1	7.874 × 10 <sup>-3</sup>	1.874 × 10 <sup>-5</sup>	3.42 × 10 <sup>-8</sup>	3.7	100
KTY81-2	7.874 × 10 <sup>-3</sup>	1.874 × 10 <sup>-5</sup>	1.096 × 10 <sup>-6</sup>	3.0	100

Note

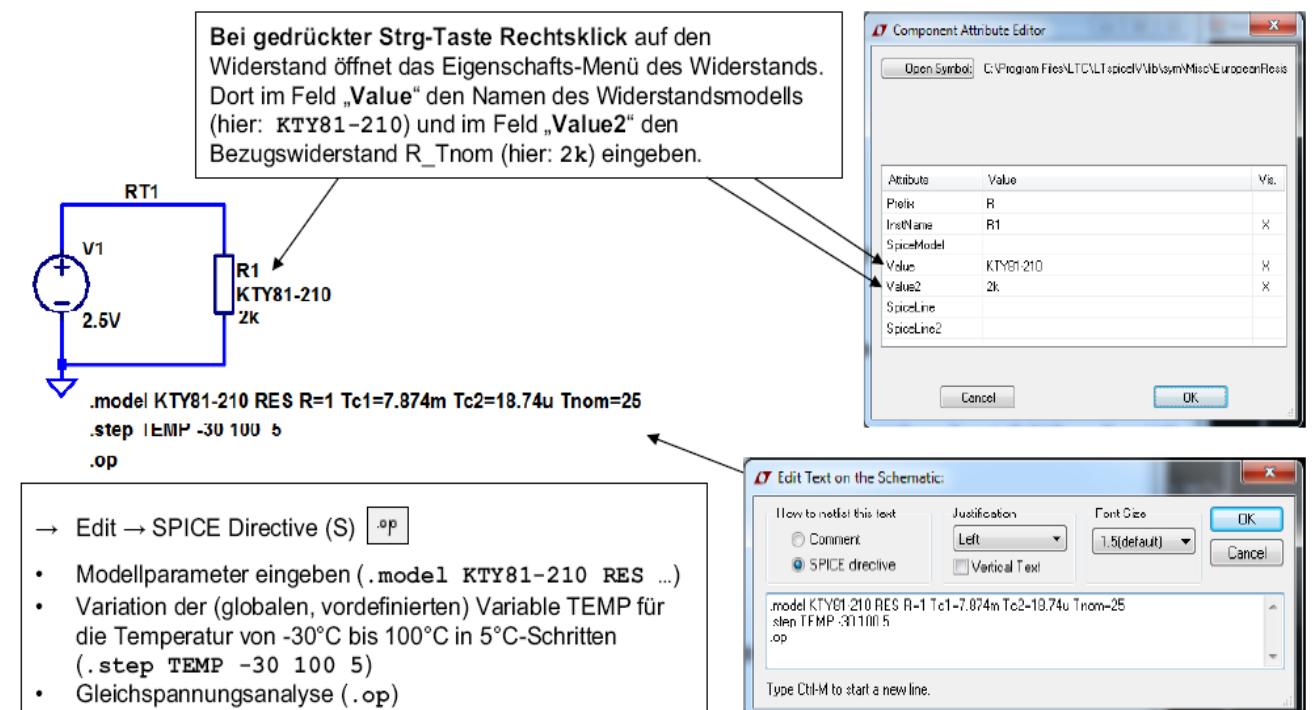
1. For  $T < T_I$ :  $C = 0$ .

## Modellierung des KTY81-210



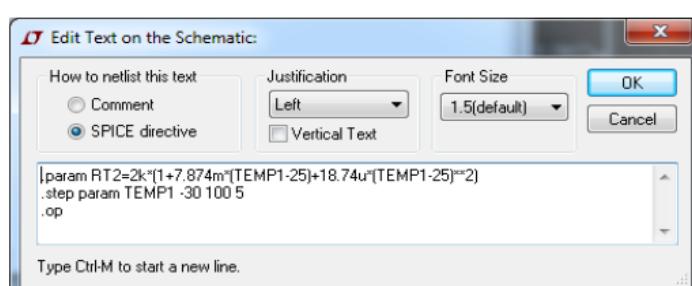
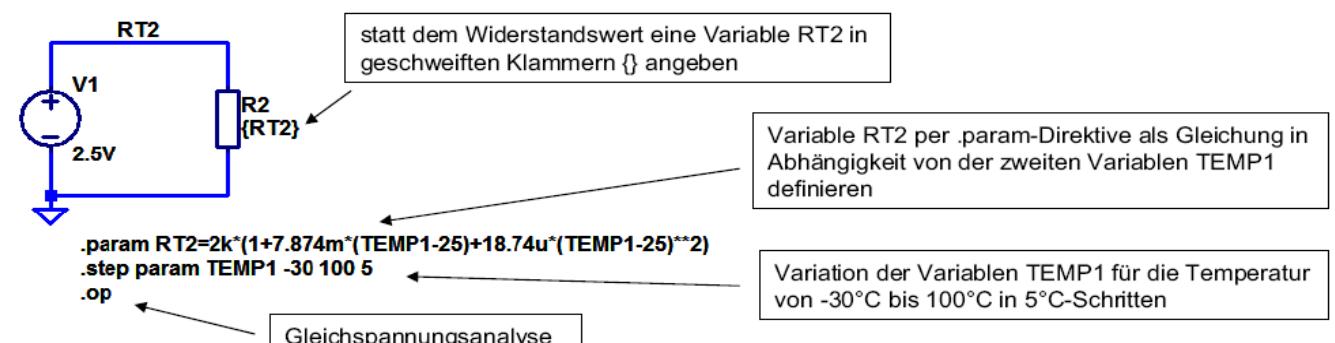
## Modellierung des KTY81-210 in LTSpice (1)

Da die Approximation der Temperaturkennlinie durch eine quadratische Funktion bei vielen Temperatursensoren verwendet wird, ist ein quadratischer Zusammenhang zwischen der Temperatur und dem Widerstand im SPICE-Modell für Widerstände bereits vordefiniert. Es müssen also nur noch die Temperaturkoeffizienten  $Tc1$ ,  $Tc2$ , die Bezugstemperatur  $T_{nom}$  und der Bezugswiderstand  $R_{T_{nom}}$  mithilfe einer .model-Direktive angegeben werden:



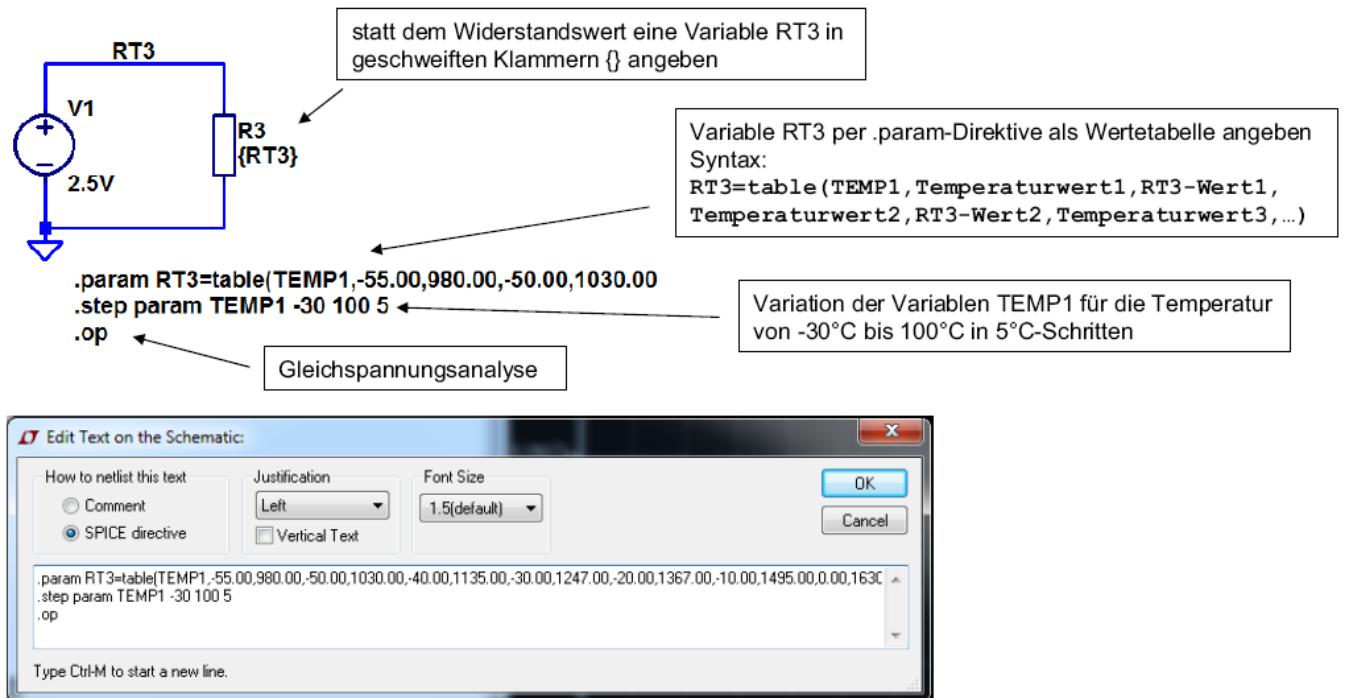
## Modellierung des KTY81-210 in LTSpice (2)

Der Nachteil der Modellierung über die .model-Direktive ist, dass für die Simulation die globale Variable TEMP für die Temperatur variiert werden muss. Alle anderen Bauteile in der Schaltung (später evtl. Operationsverstärker etc.) deren Eigenschaften auch temperaturabhängig sind, werden dann auch beeinflusst. Daher kann man die quadratische Gleichung für den Widerstandswert auch explizit mit einer anderen Variablen eingeben:



## Modellierung des KTY81-210 in LTSpice (3)

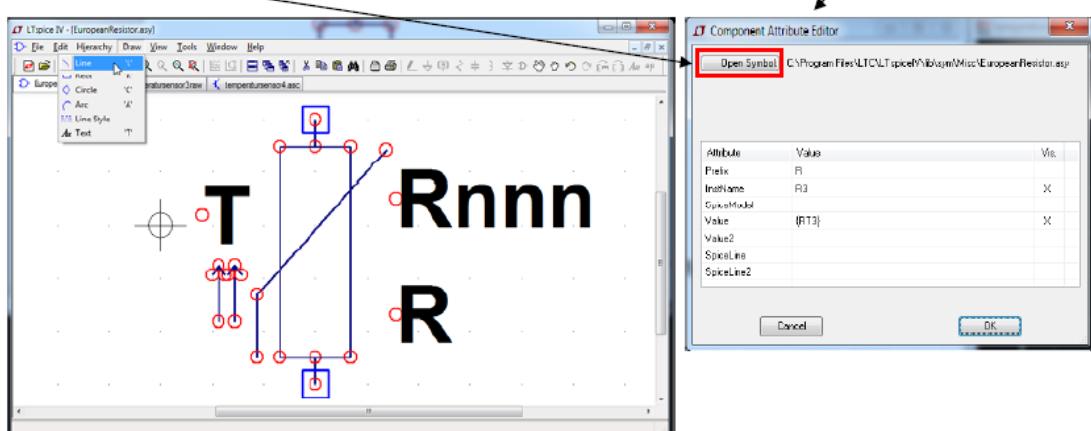
Statt der Approximation der Widerstandskennlinie durch eine quadratische Funktion, können Werte auf der Kennlinie auch direkt in Tabellenform angegeben werden. LTSpice interpoliert dann selbstständig zwischen den Tabellenwerten.



## Anpassen des Schaltzeichens mit dem Symboleditor

Der Widerstand beschreibt nun das Verhalten des Temperatursensors KTY81-210, aber für eine normgerechte Darstellung muss noch das Schaltzeichen angepasst werden. Dies lässt sich mit dem Symboleditor leicht bewerkstelligen.

- Bei gedrückter **Strg-Taste Rechtsklick** auf den Widerstand → Component Attribute Editor
- Open Symbol:** anklicken → Symboleditor wird gestartet



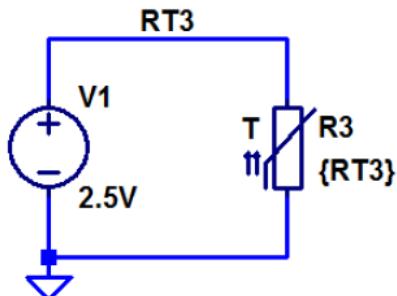
- Mit den Zeichenbefehlen (Line, Rect, Circle, Arc, Text) aus dem **Draw-Menü**, das Schalsymbol ändern.
- geändertes Schaltzeichen mit → File → Save As... im gleichen Verzeichnis wie die Simulationsdatei speichern (z.B. PTC.asy)  
[Achtung: nicht das ursprüngliche Schaltzeichen (EuropeanResistor.asy) überschreiben!]

## Einfügen des geänderten Schaltzeichens

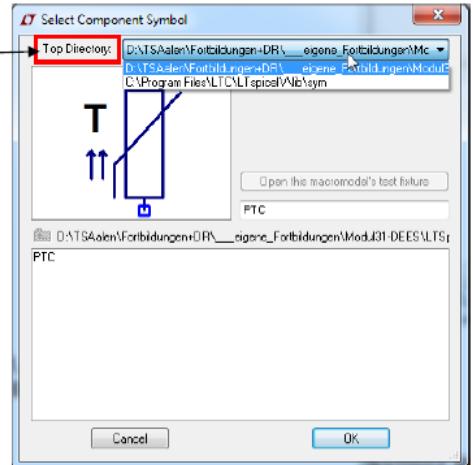
Das Bauelement mit dem geänderten Schaltzeichen wie ein neues Bauelement mit → Edit → Component (F2)  in die Schaltung einfügen.

Dabei unter **Top Directory:** zwischen der Bauteilbibliothek und dem Verzeichnis, in dem die Simulationsdatei und das neue Schaltzeichen abgespeichert ist, umschalten, **PTC.asy** auswählen und in die Schaltung einfügen.

Name und Wert (bzw. Variable {RT3}) ggf. anpassen und den alten Widerstand durch den PTC ersetzen:

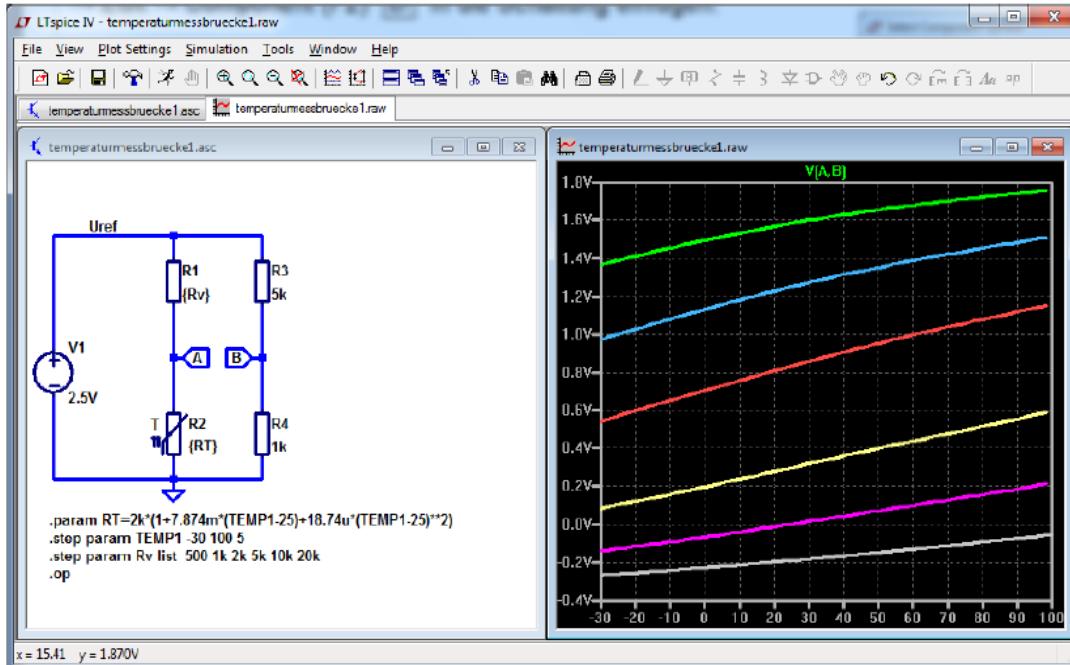


```
.param RT3=table(TEMP1,-55.00,980.00,-50.00,1030.00,-40.00,1135.00,-30.00,1247.00,
.step param TEMP1 -30 100 5
.op
```



## Linearisierung der Temperaturkennlinie

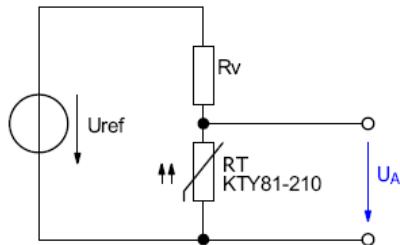
Wir bauen nun den PTC-Temperatursensor in unsere ursprüngliche Brückenschaltung ein.



Rv =  
500 Ω  
1 kΩ  
2 kΩ  
5 kΩ  
10 kΩ  
20 kΩ

Zunächst ist der Zusammenhang zwischen der Brückenspannung  $U_{AB}$  und der Temperatur nichtlinear. Durch geeignete Wahl von  $R_1$  kann der Zusammenhang aber linearisiert werden. Dazu variiert man in der Simulation neben der Temperatur zusätzlich den Wert des Vorwiderstands (Variable  $R_v$ ) so lange, bis der Verlauf von  $U_{AB}(T)$  möglichst linear wird. [Details zu Linearisierung siehe Excel-Datei bzw. Application Note]

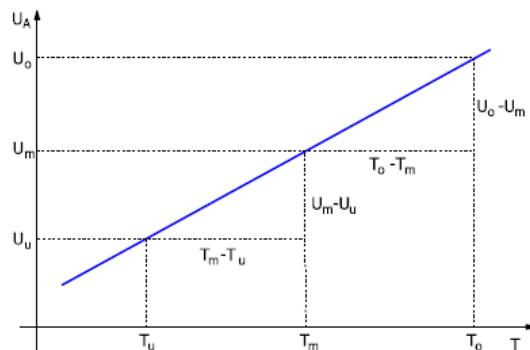
## Linearisierung der Temperaturkennlinie



Kennlinie des KTY81-210:

$$U_A = \frac{R_T}{R_v + R_T} \cdot U_{\text{ref}}$$

T	R <sub>T</sub>	
-20°C	1367 Ω	= R <sub>u</sub>
0°C	1630 Ω	
25°C	2000 Ω	
40°C	2245 Ω	= R <sub>m</sub>
100°C	3392 Ω	= R <sub>o</sub>



Annahme: Messbereich von  $T_u$  bis  $T_o$   
 $T_m$  liegt genau in der Mitte des Messbereich

→ Linearitätsbedingung:  $U_o - U_u = U_m - U_u$   
 (gleiche Steigung)

$$\rightarrow \frac{R_o}{R_v + R_o} - \frac{R_m}{R_v + R_m} = \frac{R_m}{R_v + R_m} - \frac{R_u}{R_v + R_u}$$

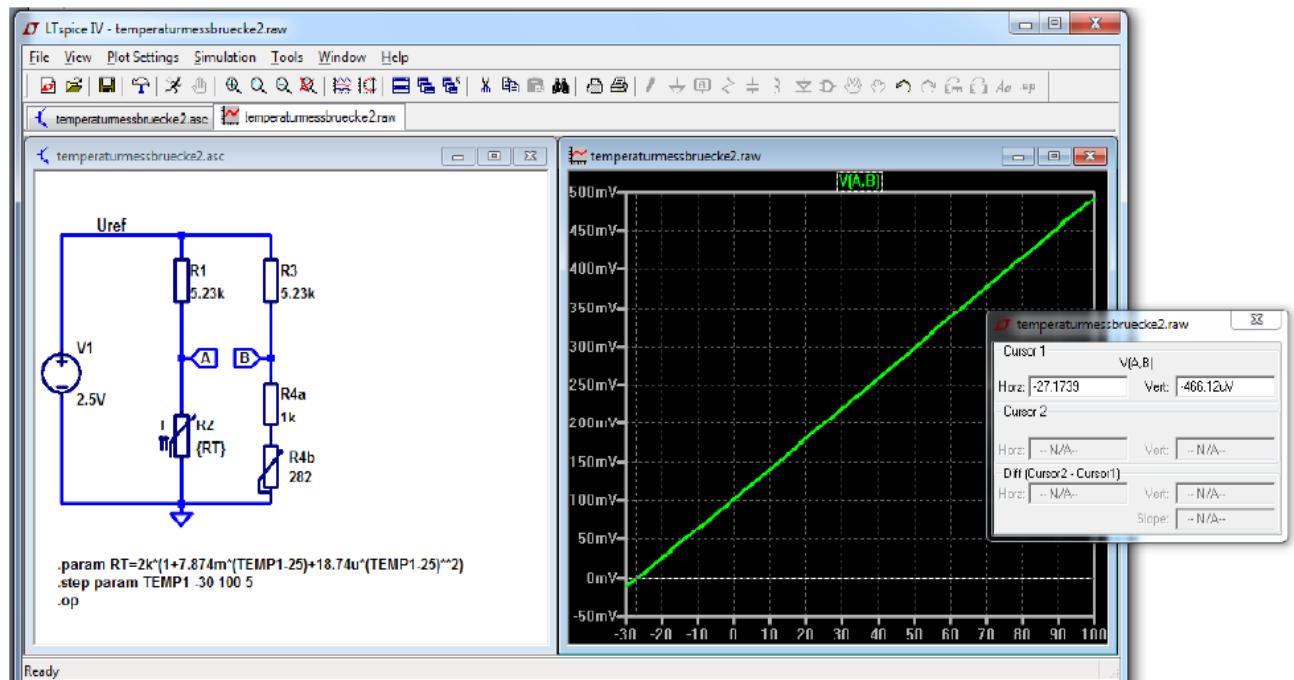
optimaler Linearisierungswiderstand:

$$R_v = \frac{R_m(R_o + R_u) - 2R_oR_u}{R_o + R_u - 2R_m} = \frac{2245 \Omega(3392 \Omega + 1367 \Omega) - 23392 \Omega \cdot 1367 \Omega}{3392 \Omega + 1367 \Omega - 2245 \Omega} = 5,24 \text{ k}\Omega \quad \rightarrow 5,23 \text{ k}\Omega \text{ (E96)}$$

## Abgleich der Messbrücke

Damit die Brückenspannung  $U_{AB}$  später mit einem **8-Bit-AD-Wandler** gemessen und eine **Temperaturauflösung von  $0,5^\circ\text{C}$**  erreicht werden kann, wird die Brücke über  $R_4$  so abgeglichen, dass der **Messbereich von  $-27^\circ\text{C}$  bis  $+100,5^\circ\text{C}$**  geht und die Brückenspannung bei  $-27^\circ\text{C}$  genau 0 V beträgt.

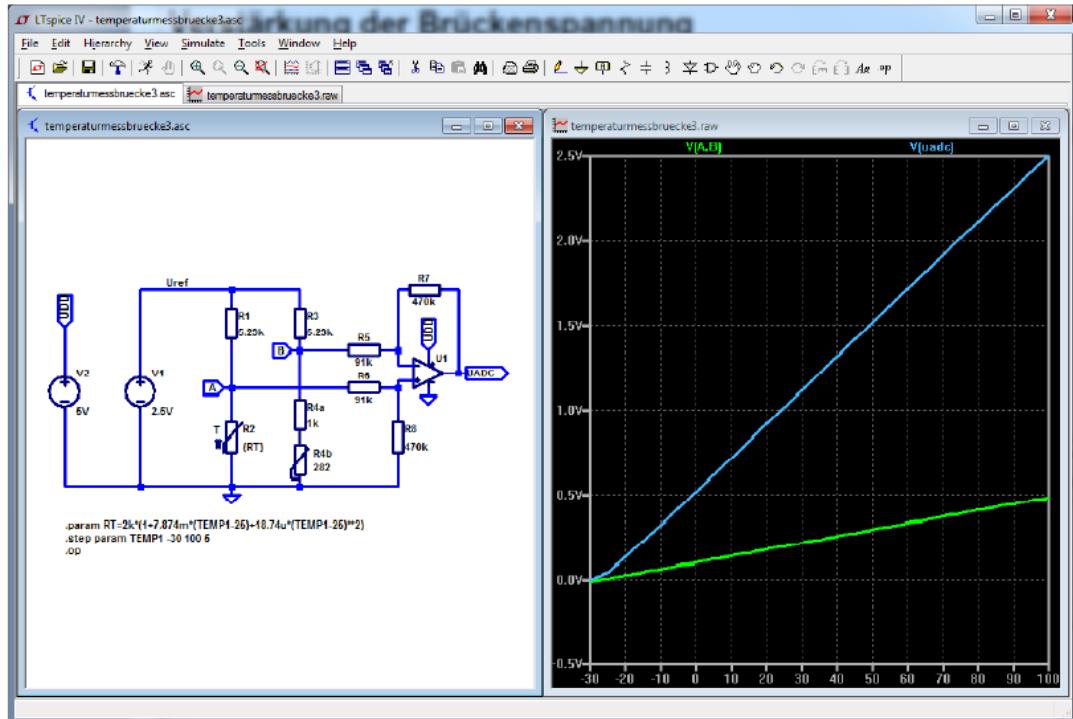
$R_2(-27^\circ\text{C}) = 1282 \Omega \rightarrow R_4 = 1282 \Omega \rightarrow R4a = 1 \text{ k}\Omega, R4b = 282 \Omega$  (500  $\Omega$ -Trimmer auf 282  $\Omega$  abgleichen)



## Verstärkung der Brückenspannung

Annahme: Der **8-Bit-AD-Wandler** hat eine Referenzspannung  $U_{ref} = 2,5 \text{ V}$   
 → Brückenspannung  $U_{AB}$  muss bei  $T = 100,5^\circ\text{C}$  auf 2,5 V verstärkt werden.

→ Differenzverstärker mit  $V = U_{ref} / U_{AB}(100,5^\circ\text{C}) = 2,5 \text{ V} / 493,2 \text{ mV} = 5,1 (\approx 470 \text{ k}\Omega / 91 \text{ k}\Omega)$

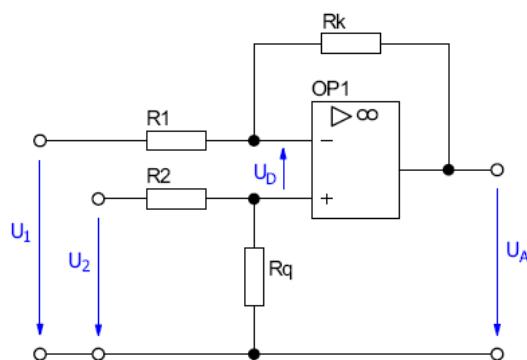


Zunächst Universal-Operationsverstärker aus der Bibliothek auswählen

- Edit
- Component
- [Opamps]
- UniversalOpamp2

## Exkurs: Dimensionierung des Differenzverstärkers

allgemeine Schaltung  
des Differenzverstärkers:



$$U_A = \frac{R_q}{R_2 + R_q} \cdot \left[ 1 + \frac{R_k}{R_q} \right] \cdot U_2 - \frac{R_k}{R_q} \cdot U_1$$

Verstärkungsformel für den allgemeinen, unsymmetrischen Differenzverstärker

Symmetribedingung:  $V_D = \frac{R_k}{R_1} = \frac{R_q}{R_2}$

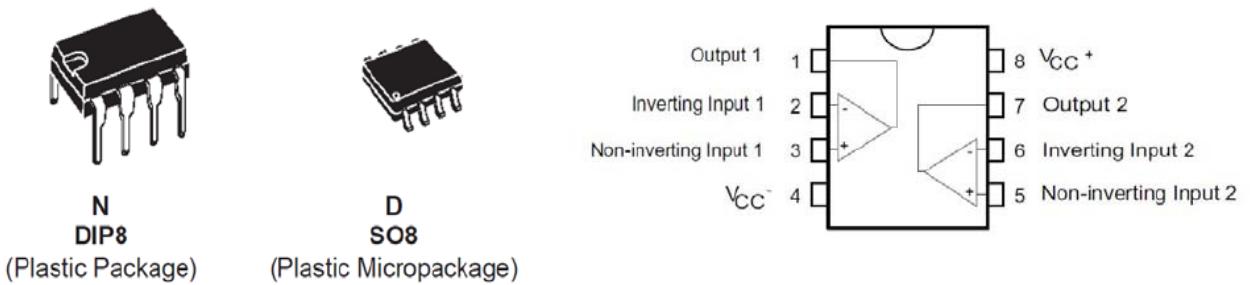
Für den symmetrischen Differenzverstärker gilt dann:

$$U_A = \frac{R_k}{R_1} \cdot [U_2 - U_1] = \frac{R_q}{R_2} \cdot [U_2 - U_1] = V_D \cdot [U_2 - U_1]$$

# Verwendung des Operationsverstärkers TS912

Datenblattangaben:

- Rail-to-rail CMOS Dual Operational amplifier
- Versorgungsspannung: 2,7 ... 16 V unsymmetrisch
- sehr geringer Eingangsstrom: 1 pA
- Ausgangsspannung:  $V_{CC^-} + 40 \text{ mV} \dots V_{CC^+} - 50 \text{ mV}$  ( $R_L = 10 \text{ k}\Omega$ )
- Ausgangstrom: 40 mA (kurzschlussfest)



## Einbinden des TS912 in LTSpice

In der Bibliothek von LT-Spice sind zwar alle Operationsverstärker der Fa. Linear Technology verfügbar, aber der TS912 gehört nicht zur Produktpalette der Firma. Daher muss dieser OP separat eingebunden werden.

Im Herstellerdatenblatt der Fa. ST Microelectronics ist ein vollständiges SPICE-Modell angegeben, das sich in LTSpice einbinden lässt:

```
Applies to : TS912 (VCC = 5V)
** Standard Linear Ics Macromodels, 1993.
** CONNECTIONS :
* 1 INVERTING INPUT
* 2 NON-INVERTING INPUT
* 3 OUTPUT
* 4 POSITIVE POWER SUPPLY
* 5 NEGATIVE POWER SUPPLY
* 6 STANDBY
.SUBCKT TS912_5 1 2 3 4 5
*****
.MODEL MDTH D IS=1E-8 KF=6.564341E-14 CJO=10F
* INPUT STAGE
CIP 2 5 1.000000E-12
CIN 1 5 1.000000E-12
EIP 10 5 2 5 1
EIN 16 5 1 5 1
RIP 10 11 6.500000E+00
...
...
```

Das SPICE-Modell besteht aus einer eigenen „Unterschaltung“ (Subcircuit), die in SPICE mit dem Befehl .SUBCKT <Name> <Pinnummem> definiert wird.

Pinbelegung

Definition der „Subcircuit“

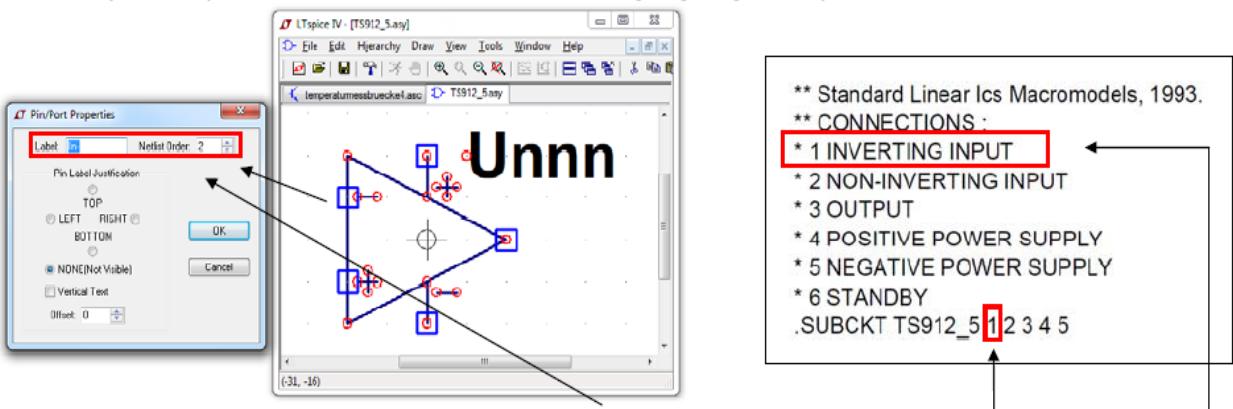
Beschreibung des internen Aufbaus der „Subcircuit“ (SPICE-Syntax)

Zur Verwendung in LTSpice speichert man das SPICE-Modell einfach als Text-Datei mit der Endung \*.lib im gleichen Verzeichnis, wie die Simulationsdatei (oder im Bibliothekspfad der LTSpice-Installation), also hier z.B. TS912\_5.lib.

## Schaltzeichen und Pinbelegung festlegen

Für den neuen OP ist noch ein Schaltzeichen erforderlich. Hierfür kann das Schaltzeichen des UniversalOpamps verwendet und angepasst werden.

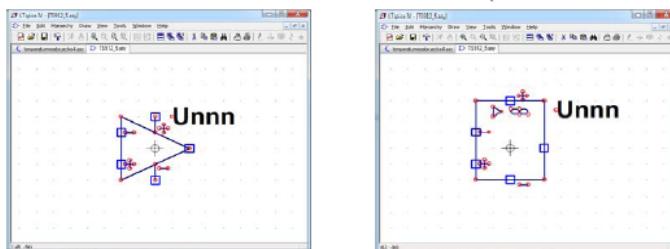
1. Im Schaltplan bei gedrückter Strg-Taste Rechtsklick auf den UniversalOpamp2  
→ Component Attribute Editor Menü öffnet sich
2. Im Component Attribute Editor Menü auf **Open Symbol** klicken. Es öffnet sich der Symbol Editor mit dem Schaltzeichen des UniversalOpamp2.
3. Das Schaltzeichen zunächst unter einem neuen Namen (z.B. TS912\_5.asy) im gleichen Verzeichnis wie die Simulationsdatei speichern.
4. Im Symbol Editor zunächst die Pinbelegung überprüfen: Rechtsklick auf die einzelnen Pins des Schaltsymbols (im Bild wurde der invertierende Eingang angeklickt).



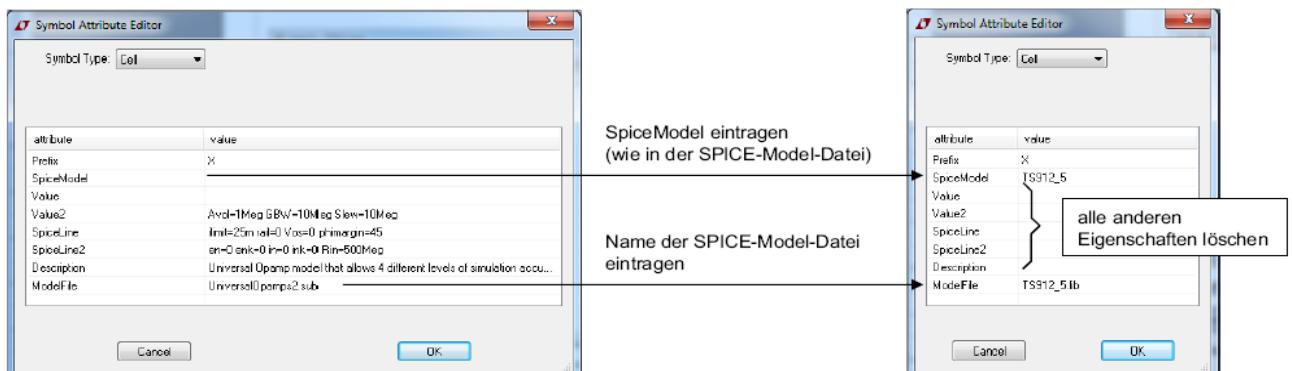
Der invertierende Eingang (In-) hat beim UniversalOpamp die „Netlist Order“ 2, im Modell des TS912 hat der invertierende Eingang jedoch die Pin-Nr. 1, d.h. die „Netlist Order“ muss auf 1 umgestellt werden. Ebenso müssen die anderen fünf Pins angepasst werden.

## Schaltzeichen anpassen und Zuordnung zum SPICE-Modell festlegen

Nun kann das Schaltzeichen noch beliebig angepasst werden (z.B. Symbol nach DIN 40900 Teil 13). Dazu lassen sich die Editierfunktionen aus dem → Edit-Menü (verschieben, kopieren, löschen, ...) und die Zeichenfunktionen aus dem Draw-Menü (Line, Rechteck, Kreis, ...) verwenden.

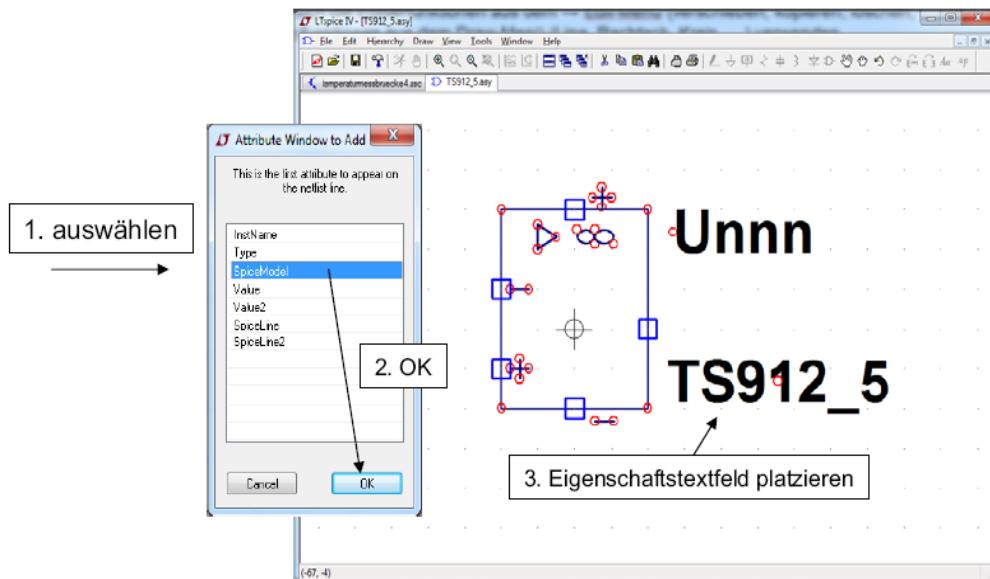


Nun müssen dem neuen Schaltzeichen noch die Eigenschaften des OP zugeordnet werden. Dies geschieht über den Attribute Editor (→ Edit → Attributes → Edit Attributes bzw. Strg+A).



## Schaltzeichenbeschriftung ergänzen

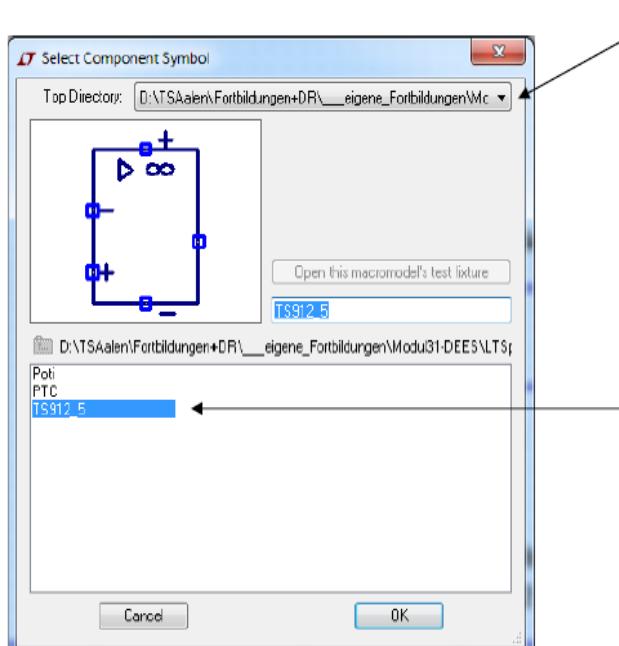
Über → Edit → Attributes → Attribute Window (Strg+W) können nun noch Bauelementeeigenschaften (die gerade eingetragen wurden) als Beschriftung zum Schaltzeichen hinzugefügt werden. Dazu in der Liste die anzuseigende Eigenschaft auswählen, auf OK klicken und das Textfeld an der passenden Stelle neben dem Schaltzeichen absetzen. Diese Eigenschaft (z.B. der Name des Bauelements) wird nun immer mit den Schaltzeichen zusammen angezeigt.



Abschließend das erneute Speichern des neuen Bauelements TS912\_5.asy nicht vergessen!

## Neue erstelltes Bauelement einfügen

Nun kann das neue Bauelement – hier der OP TS912 wie jedes andere Bauelement in die Schaltung eingefügt werden (→ Edit → Component (F2) ).

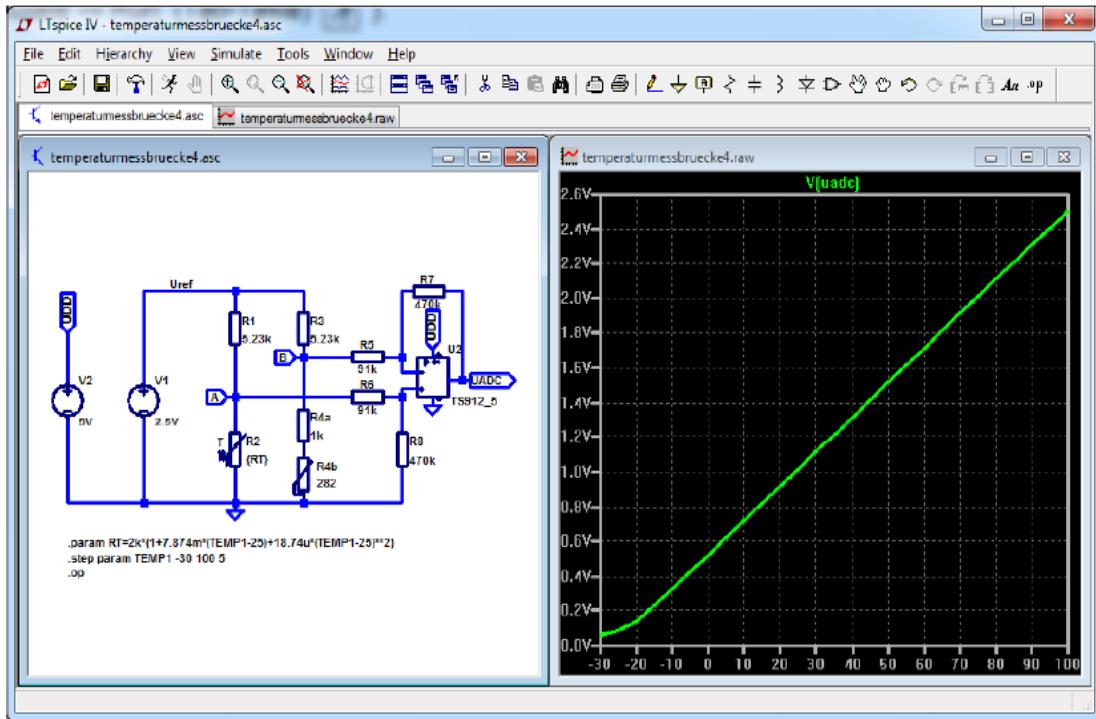


Verzeichnis auswählen, in dem das neue Bauelement gespeichert wurde (hier das Verzeichnis, in dem auch der Schaltplan (\*.asc) und die SPICE-Model-Datei (\*.lib) liegen).

neues Bauelement (TS912\_5) auswählen und mit OK einfügen.

## Simulation starten mit neuem Bauelement

Nun kann – genauso wie zuvor mit dem UniversalOpamp2 – die Simulation gestartet werden  
 (→ Simulate → Run (Tab-Taste) .

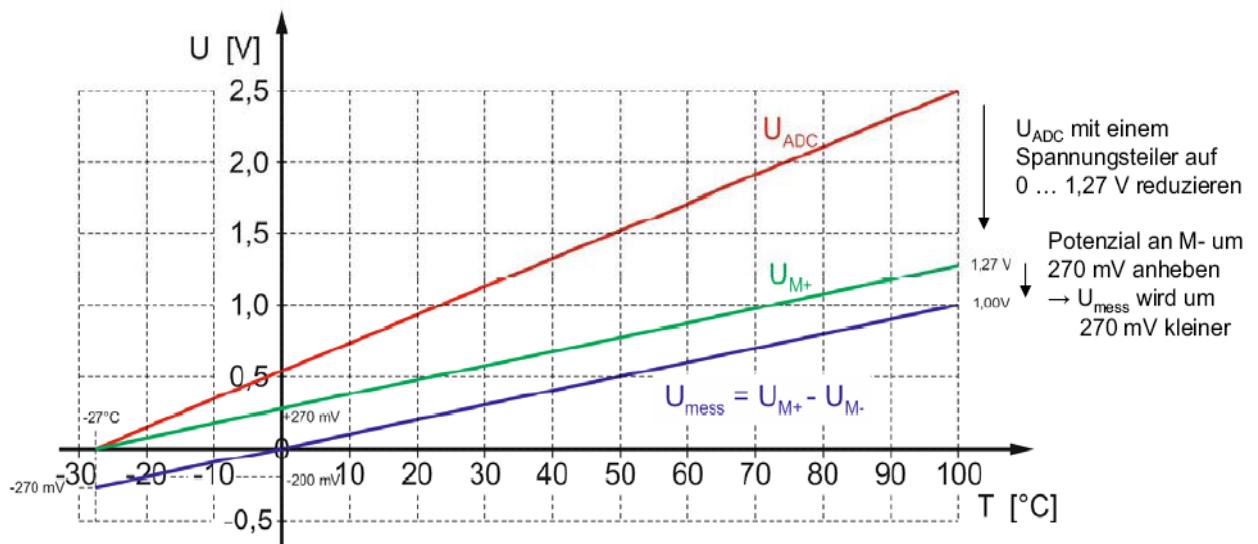


Da der OP TS912 bei unsymmetrischer Versorgungsspannung mit seiner Ausgangsspannung nicht ganz bis auf 0 V herunterkommt (→ Kurve flacht bei -20°C ab) ist nur ein Temperaturmessbereich von -20°C bis +100°C sinnvoll nutzbar.

## Temperaturanzeige am Digitalmultimeter

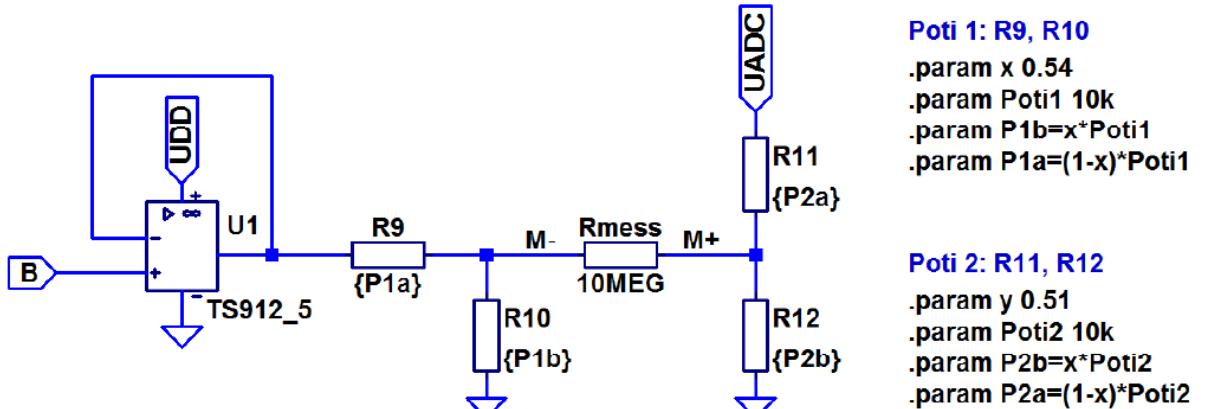
Nun soll die Temperatur nicht nur über einen AD-Wandler erfasst, sondern auch direkt mit einem Digitalmultimeter angezeigt werden. Der **Messbereich** soll unverändert bei **-20°C ... 100°C** bleiben.

Dimensionierung:  $T = -20^{\circ}\text{C} \dots +100^{\circ}\text{C} \rightarrow U_{\text{mess}} = -0,20 \text{ V} \dots +1,00 \text{ V} \rightarrow 10 \text{ mV}/^{\circ}\text{C}$



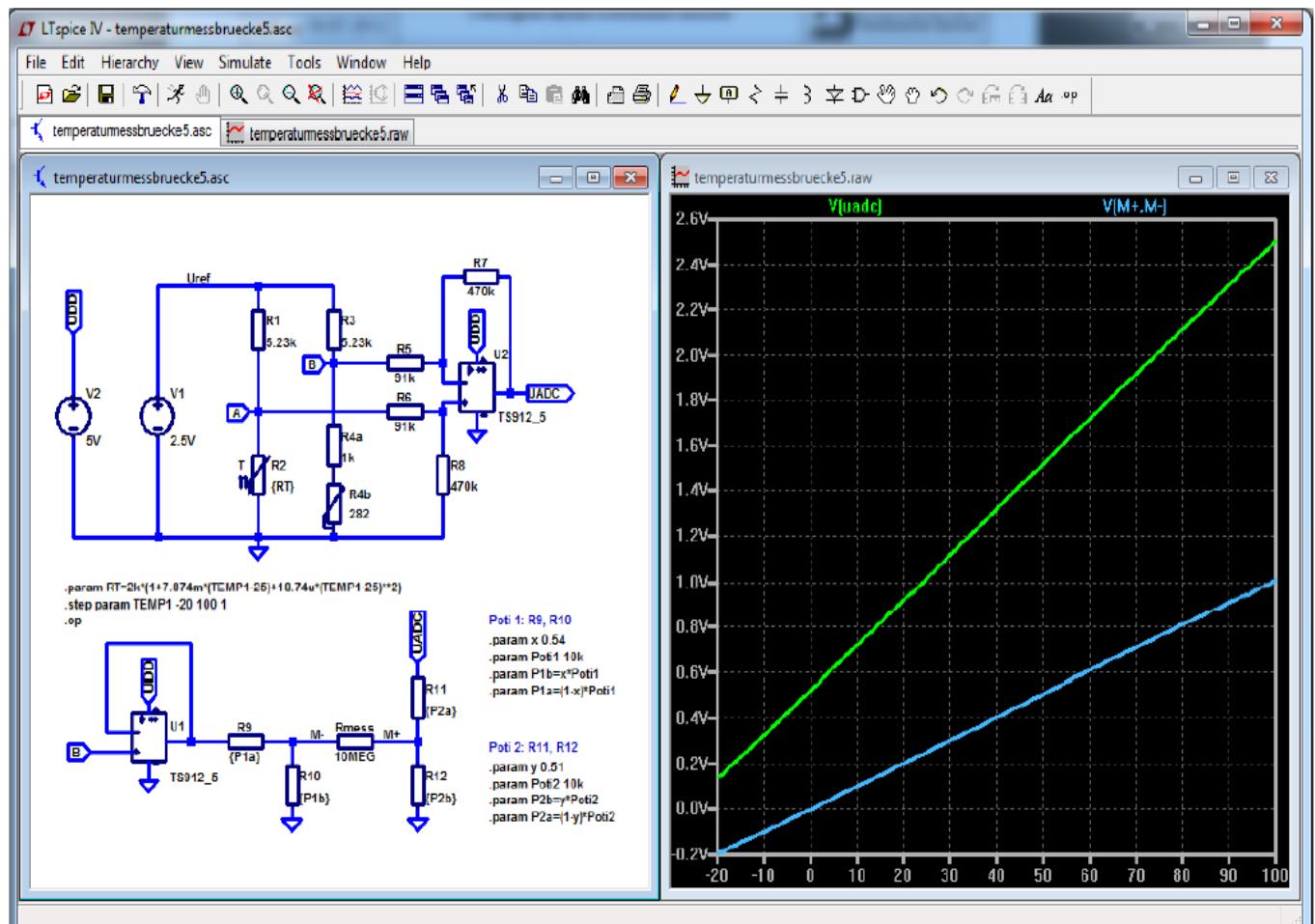
## Temperaturanzeige am Digitalmultimeter

Schaltung zur Messbereichsanpassung für das Digitalmultimeter:



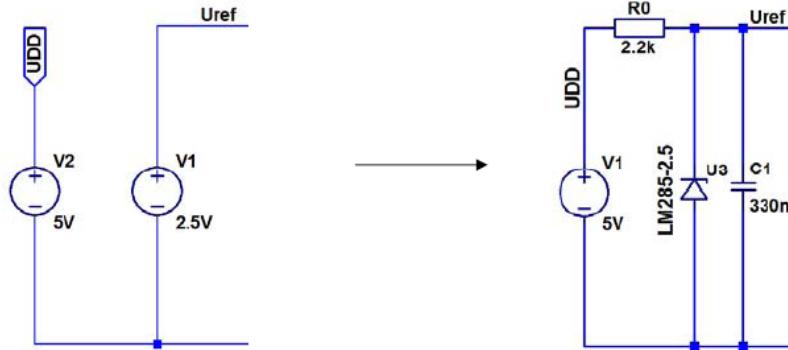
- Die Schaltung wird im gleichen Schaltplan neben die Messbrücke gezeichnet.
- Gleiche Knotennamen wie in der Messbrückenschaltung (B, UDD, UADC) bedeuten, dass es sich für SPICE um die gleichen Knoten handelt (direkter Anschluss ohne, dass eine Leitung gezeichnet werden muss).
- Die Spannungsteiler R9, R10 und R11,R12 können als Potis realisiert werden. Über die Variablen x und y lässt sich der Mittelabgriff der Potis verstellen (x, y = 0 ... 1).

## Simulation der Gesamtschaltung



## Referenzspannungsquelle

Beim praktischen Schaltungsaufbau lässt sich die Referenzspannung  $U_{ref}$  mit einer Referenzdiode LM285-2.5 aus der Versorgungsspannung erzeugen. Somit kann die gesamte Schaltung aus einer einzigen 5V-Quelle versorgt werden, die Referenzspannung wird sehr genau eingehalten und dient gleichzeitig als Referenzspannung für den AD-Wandler.



Datenblatt Temperatursensor (Vollständiges Datenblatt ist beim Hersteller erhältlich)

Philips Semiconductors	Product specification								
<b>Silicon temperature sensors</b>									
<b>KTY81-2 series</b>									
<b>DESCRIPTION</b>									
<p>The temperature sensors in the KTY81-2 series have a positive temperature coefficient of resistance and are suitable for use in measurement and control systems. The sensors are encapsulated in the SOD70 leadless plastic package.</p> <p>Tolerances of 0.5% or other special selections are available on request.</p>									
<b>PINNING</b>									
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>PIN</th> <th>DESCRIPTION</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>electrical contact</td> </tr> <tr> <td>2</td> <td>electrical contact</td> </tr> <tr> <td>3</td> <td>not to be connected to a potential</td> </tr> </tbody> </table>		PIN	DESCRIPTION	1	electrical contact	2	electrical contact	3	not to be connected to a potential
PIN	DESCRIPTION								
1	electrical contact								
2	electrical contact								
3	not to be connected to a potential								
Fig.1 Simplified outline (SOD70).									

Philips Semiconductors	General
<b>Temperature sensors</b>	
<p>(a) With a resistor '<math>R_L</math>' shunted across the sensor.      (b) With a resistor '<math>R_L</math>' in series with the sensor and system powered by a constant-voltage source.      (c) With a series '<math>R_S</math>' and parallel resistor '<math>R_P</math>' and system powered by a constant-voltage source.</p>	
Fig.11 Linearization of sensor characteristics.	

## Übung: 3: Digitale Schaltungssimulation

**Bemerkung:** Leider wird dieses Thema sowohl von der Online-Hilfe wie auch von den mitgelieferten Beispielen her etwas stiefmütterlich behandelt -- man muss also etwas Wissen und Erfahrung mitbringen und dann alle möglichen Mosaiksteine zusammentragen, um Erfolg zu haben. [Aber eigentlich haben die SPICE Programme in der Analogen Simulation ihre Stärken und haben dort auch ihre Berechtigung.](#)

### Einschränkungen:

Die mitgelieferte Symbolbibliothek „**[Digital]**“ enthält nur eine Reihe von „**idealen Grundbausteinen**“. An ihnen fällt auf, dass sie alle **8 Anschlüsse** („**Pins**“) aufweisen. Dahinter steckt folgendes System:

- a) Es gibt eine ganze Reihe Eingänge (z. B. beim AND-Gatter deren 5...). Nicht benötigte Eingänge werden einfach offen gelassen, denn dadurch sind sie automatisch abgeschaltet und nicht in die Simulation einbezogen.
- b) Meist ist nicht nur der zugehörige logische Ausgang, sondern zusätzlich auch die Invertierung vorhanden.
- c) Die logischen Pegel betragen „**Null Volt**“ für die „**Logisch Null**“ und „**+1 Volt**“ für die „**Logische Eins**“. Die intern programmierte Umschaltschwelle zwischen beiden Zuständen ist +0,5V.

**Achtung:** Wer auf andere Werte (z. B. den TTL-Pegel von 0V / +5V) umstellen will, der klickt mit der rechten Maustaste auf das Gatter-Symbol und trägt dann in der Zeile „value“ ein: **Vhigh=5V Vlow=0V**

- d) Ausgänge sollten nicht unbedingt „frei in der Luft hängen“. Sie können entweder mit einem Label versehen oder über einen Widerstand (empfehlenswert:  $R = 10k_{\Omega}$ ) an Masse gelegt werden.
- e) Man findet **keine Betriebsspannungsanschlüsse**, da mit „idealen Bausteinen“ simuliert wird.

**Als Eingangssignal verwendet man entweder das „PULSE-“ oder das PWL-Signal. Minimaler Spannungswert ist Null Volt, maximaler Spannungswert = +1Volt (...andere Werte: siehe oben...)**

**Als Anstiegs- und Abfallzeit reicht eine Zeit von 1 Nanosekunde.**

Bei der Darstellung der Simulationsergebnisse sollte man sofort -- bevor überhaupt etwas im Ergebnisdiagramm zu sehen ist! -- mit der rechten Maustaste draufklicken und dann **Add Plot Pane** wählen. Dadurch taucht ein weiteres Diagramm auf. Falls nötig, wiederholt man diese Prozedur solange, bis man genügend Diagramme für die Darstellung aller gewünschten Signale zur Verfügung hat. Wer das nicht tut, wird ganz schnell die Lust an digitalen Simulationen verlieren.

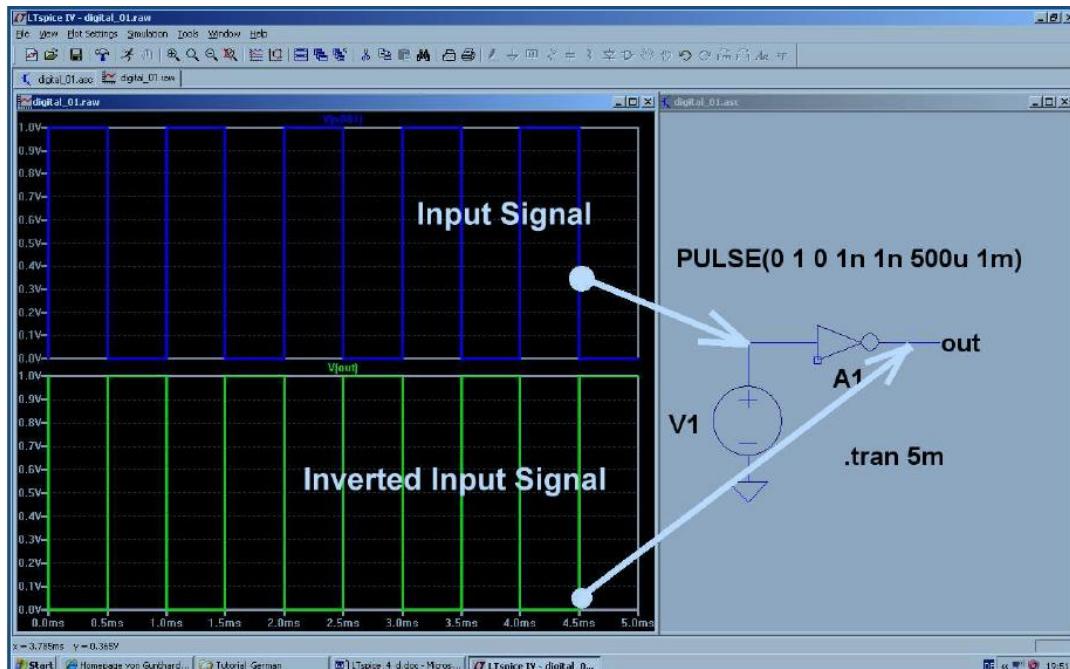
Und die Darstellung einer Kurve in einem dieser Diagramme ist sehr einfach: rechts auf das Diagramm klicken, „Add Trace“ aktivieren und das gewünschte Signal auswählen.

**Aufgabe : Die Umkehrstufe (= NOT oder Inverter)**

Den Baustein holen wir uns als „**Digital / INV**“ aus der Library und zur Ansteuerung verwenden wir eine symmetrische Rechteckspannung ( $U_{min} = 0V / U_{max} = +1V /$

Anstiegszeit = Abfallzeit = 1ns / Pulslänge = 500 Mikrosekunden / Periodendauer = 1ms).  
Simuliert wird von 0....5ms.

Am oberen Eingang wird angesteuert, der andere wird offen gelassen. Der Ausgang erhält den Label „out“ .Wie erwähnt, öffnen wir über „Add Plot Plane“ zwei Ergebnisdigramme und stellen darin die Eingangs- bzw. die Ausgangsspannung dar. So sieht das Ergebnis aus, wenn man „Tile vertically“ im Menü „Window“ wählt:

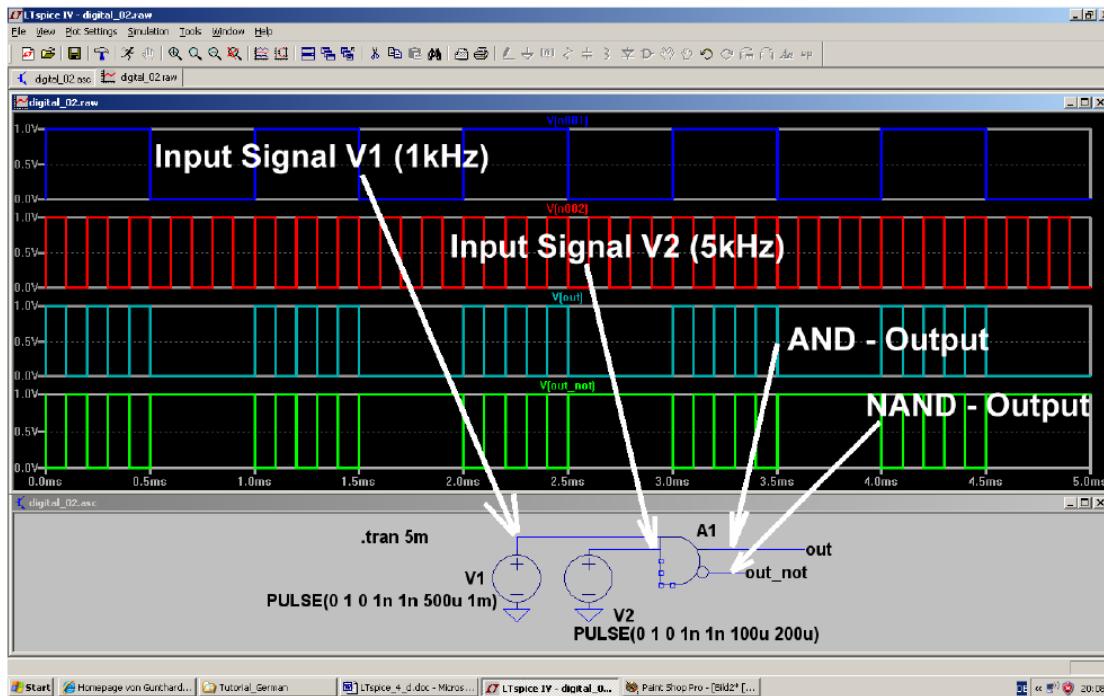


### Aufgabe: AND-Baustein

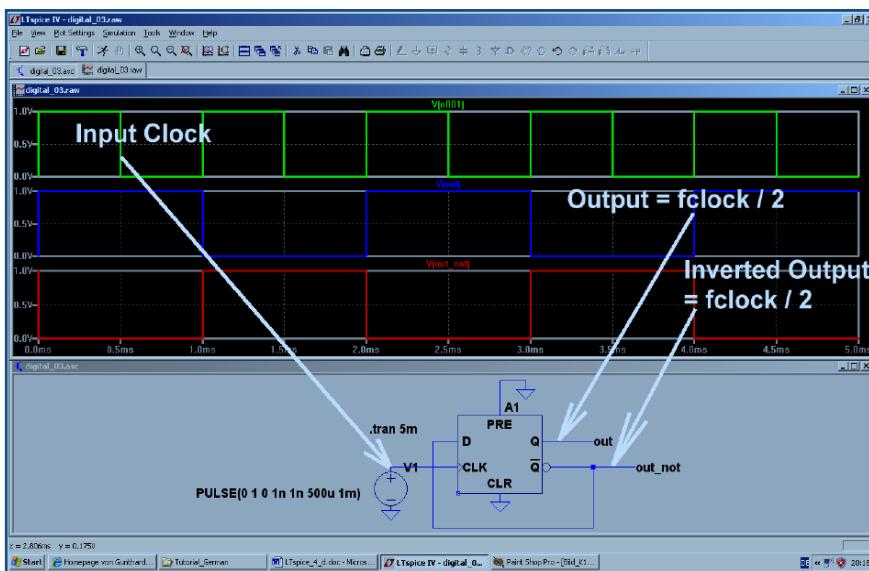
Wir wollen eine UND-Verknüpfung mit 2 Eingängen realisieren. Am einen Eingang liegt ein Digitalsignal mit  $f = 1\text{kHz}$ , am anderen dagegen eines mit  $f = 5\text{kHz}$ . Simuliert wird von 0....5ms.

Wie zu Beginn des Kapitels erwähnt, müssen wir dazu die drei zusätzlich vorhandenen, aber nicht benötigten Eingänge offen lassen. Den vorhandenen beiden Ausgängen wurden wieder **Label zugewiesen**.

So sollte die Simulation aussehen:



## Aufgabe: D-Flipflop



Die Schaltung stellt den bekannten Binär-Frequenzteiler dar, bei dem am Ausgang die halbe Eingangsfrequenz beobachtet werden kann. Dazu wird der invertierte Q Ausgang mit dem D-Eingang verbunden. Die Input Clock Frequenz beträgt 1 kHz. Die Signaleingänge „Preset“ und

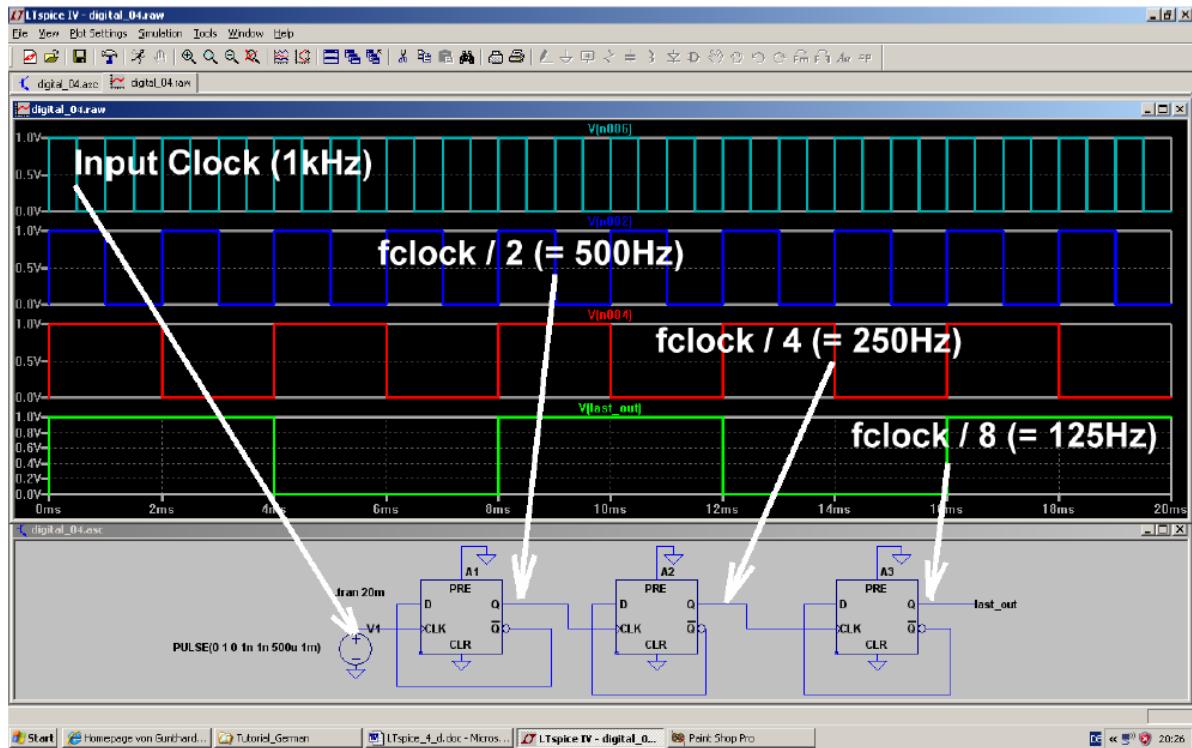
„Clear“ dürfen nicht aktiv werden und liegen deshalb an Masse. Bei jeder positiven Flanke des Clocksignals wechseln die Ausgänge ihre Zustände.

## Aufgabe: Dreistufiger Frequenzteiler mit D-Flipflops (Zusatzaufgabe)

Das ist jetzt nur noch reine Arbeit, aber kein Problem mehr, denn es werden einfach zwei weitere identische Stufen nachgeschaltet.

**Anmerkung:** Es fehlen einfach die vielen, vielen Bausteine der TTL – oder CMOS-Serie, die das alles interessanter oder praxisnäher machen.

Abhilfe: <http://tech.groups.yahoo.com/group/LTspice/files/%20Lib/Digital%2074HCTxxx/>



Die Simulationszeit wurde auf 20ms erhöht, der Eingangs-Clock (= Input Clock Signal) besteht weiterhin aus einem Rechtecksignal mit der Frequenz  $f = 1\text{kHz}$ .

### **Wichtiger Hinweis:**

### **Bauteiletoleranzen berücksichtigen (*Montecarlo-Analyse*)**

Sehr wichtig ist es auch, den Einfluss der Bauteiletoleranzen zu untersuchen. Um diese Toleranzen zu simulieren, gibt man folgende Anweisung über das Spice-Kommandofenster ein: `.step param a -0.1 0.1 0.05`

Der Befehl ändert nacheinander den Parameter  $a$  von -0,1 bis +0,1 in Intervallen von 0,05.

Nun muss noch der Bauteilwert mithilfe des obigen Parameters „ $a$ “ variabel gemacht werden. Bei jedem Bauteil muss normalerweise der Bauteilewert, bei einem Widerstand z.B.  $100\text{k}\ \Omega$  angegeben werden. In unserem Fall ersetzen wir den Wert durch eine Funktion. Diese Funktion muss in geschweiften Klammern angegeben werden.

**z.B.: statt  $100\text{k} \Rightarrow \{100\text{k} * (1+a)\}$**

Vorher hatte der Widerstand einen Wert von  $100\text{k}\ \Omega$ , jetzt hängt der Bauteilwert von der Funktion mit dem Parameter „ $a$ “ ab.

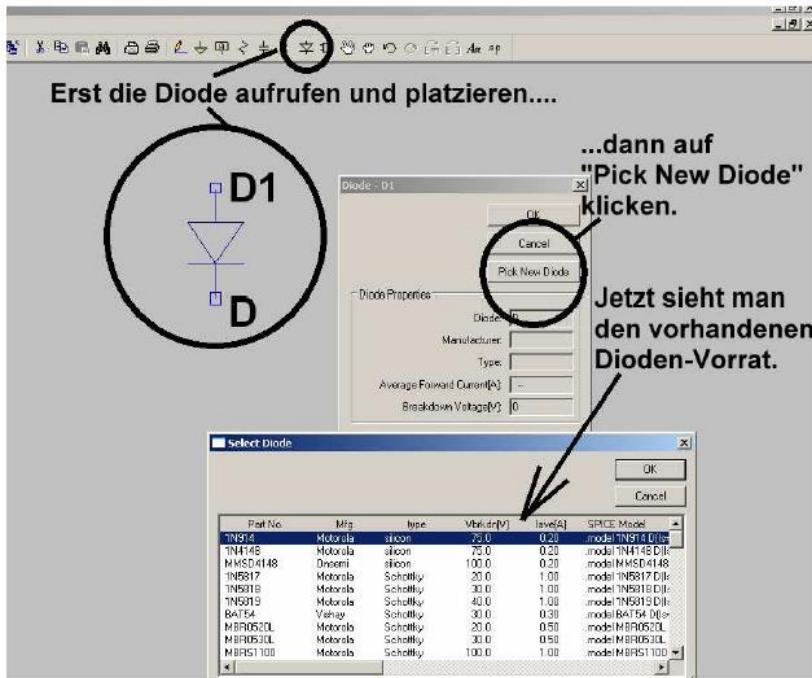
Das Untersuchen dieses Verhaltens mit zufällig verteilten Werten nennt man Monte-Carlo-Analyse.

### **Anhang:**

### **Beispiel: Verwendung der Diode 1N4007**

SPICE Modelle einbinden: LTSpice hat ein recht umfangreiche Library, aber oft fehlen gerade die einfachen Bauteile wie die Diode **1N4007** oder den **OP TL072**. Die Sie im Regelfall einsetzen und auch in unseren Laboren vorrätig sind.

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 verfahre nach der nebenstehenden Methode.

## 1. Schritt:

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

Aber Vorsicht:

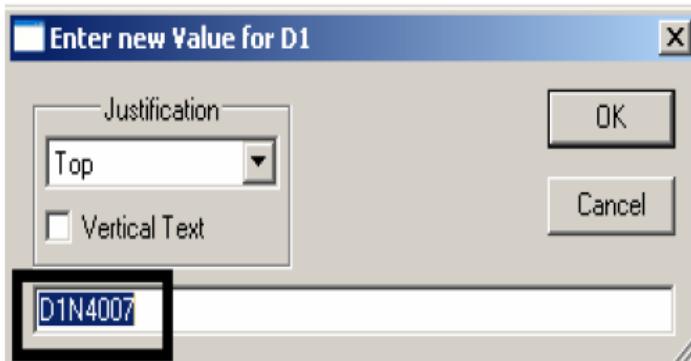
Diese Library kommt bei uns als HTML-File an und ist in dieser Form noch nicht einsetzbar. Vorgehensweise: erst den Inhalt mit **<STRG> + <a>** komplett markieren und mit **<STRG> + <c>** in die Zwischenablage kopieren. Dann den Text-Editor (z.B. Notepad) öffnen und die Zwischenablage mit **<STRG> + <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 zusätzlich die Endung „**.txt**“ angehängt!)

## 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!

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



### 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.

## **Beispiel: Verwendung der 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 „**LTS defense / lib / sub**“ abgelegt. So sieht das am Ende aus:

**<http://www.ti.com/product/tl072>**

```
* 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
```

```

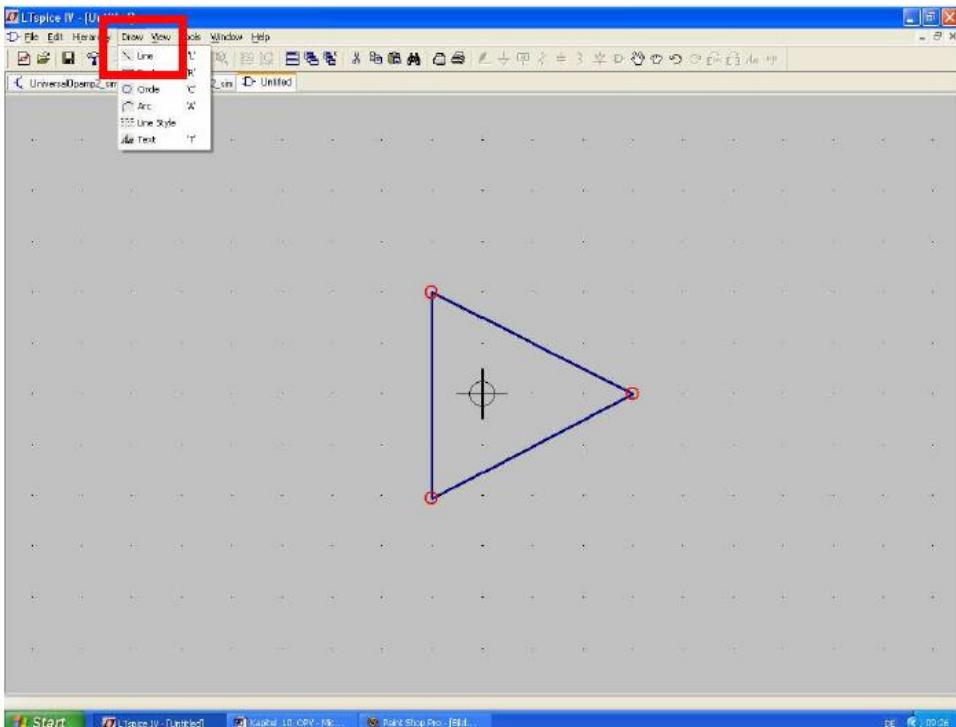
C2 6 7 15.00E-12
DC 5 53 DX
DE 54 5 DX
DLP 90 91 DX
DLN 92 90 DX
DP 4 3 DX
EGND 99 0 POLY(2) (3,0) (4,0) 0 .5 .5
FB 7 99 POLY(5) VB VC VE VLP VLN 0 4.715E6 -5E6 5E6 5E6 -5E6
GA 6 0 11 12 282.8E-6
GCM 0 6 10 99 8.942E-9
ISS 3 10 DC 195.0E-6
HLIM 90 0 VLIM 1K
J1 11 2 10 JX
J2 12 1 10 JX
R2 6 9 100.0E3
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:**

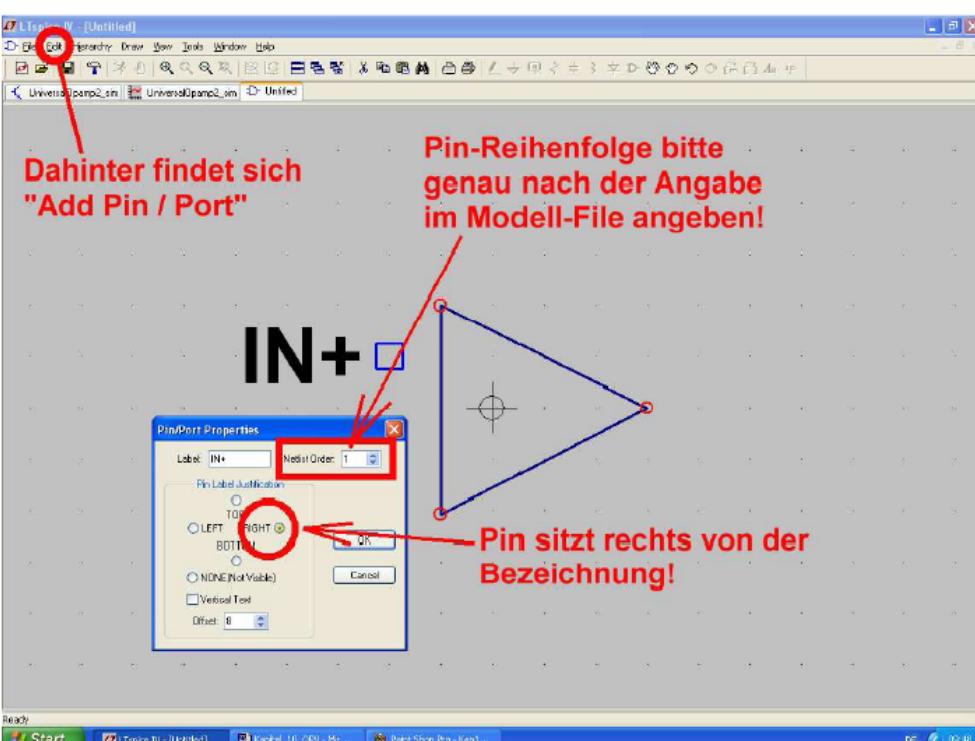
LTSpice wird gestartet, aber unter „**File**“ die Option „**New Symbol**“ angeklickt. Dadurch öffnet sich ein neues Blatt in einem etwas helleren Grau, um es von einem Schaltbild unterscheiden zu können. Über dem Bildschirm liegt ein feines Punktraster und in der Mitte ist das Zentrum für das Symbol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

**Siehe nächste Seite!**



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“.



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

Im nebenstehenden Menü muss man dann auf 2 Dinge achten:

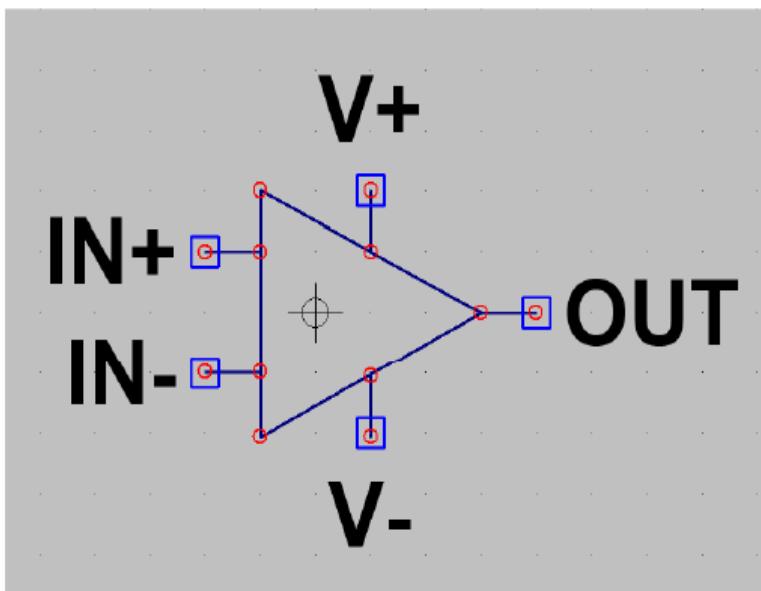
**Erstens** muss die richtige Pin-Nummer gemäß den Angaben im Modell-File („TL072.sub“) eingetragen werden.

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.

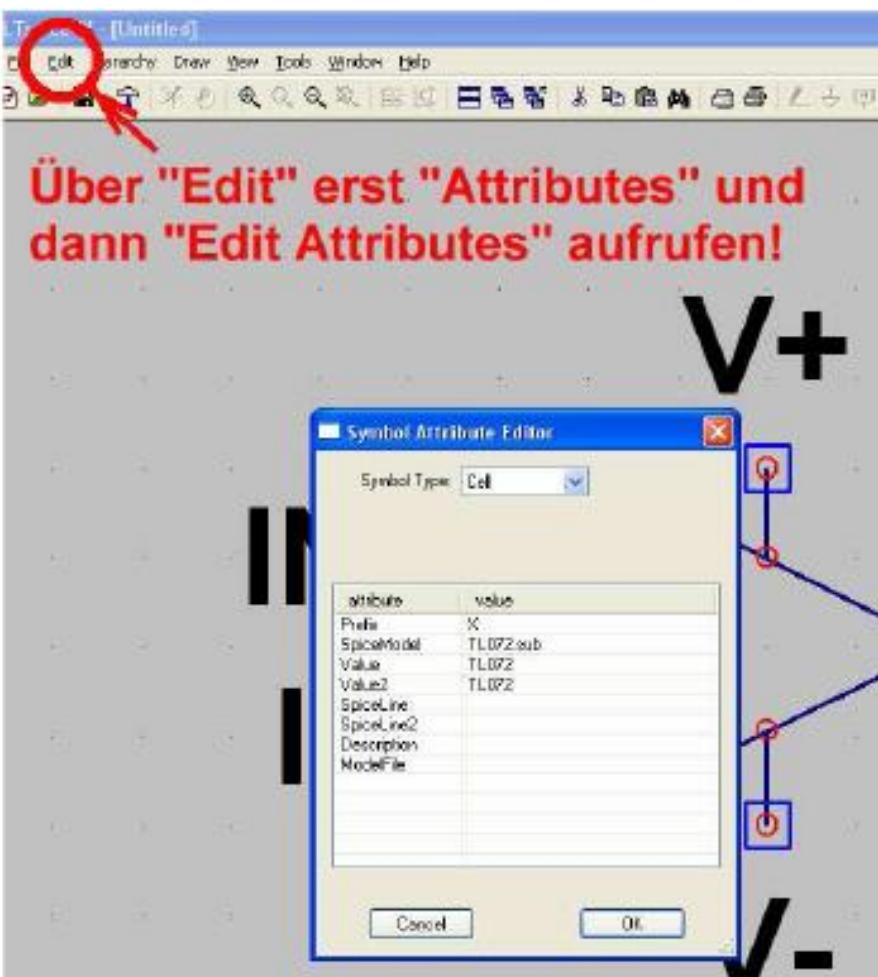
**Zweitens** muss man angeben, wo sich der Pin räumlich gegenüber der Bezeichnung befinden soll (hier: rechts).

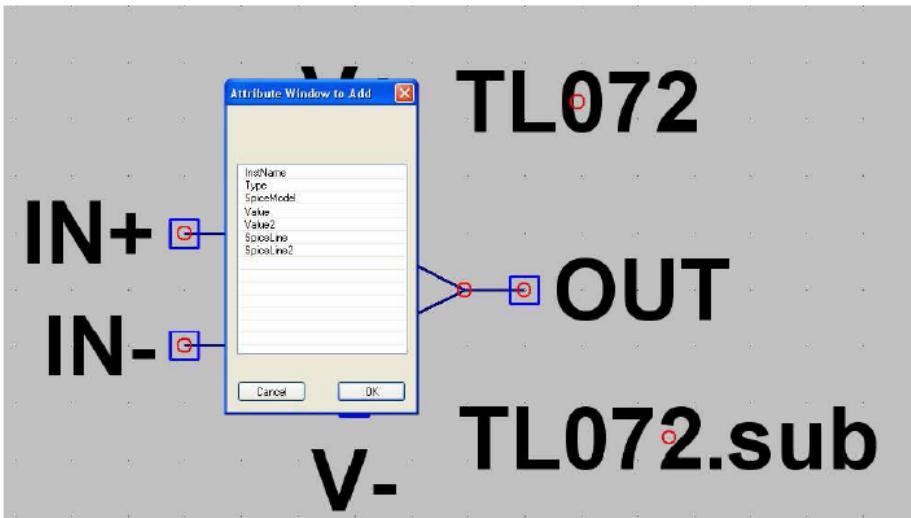


Sind alle Pins platziert, dann sorgt man im Anschluss wieder mit „Draw Lines“ für Verbindungsleitungen von jedem Pin zum Schaltzeichen.

Das ist das Ziel...

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

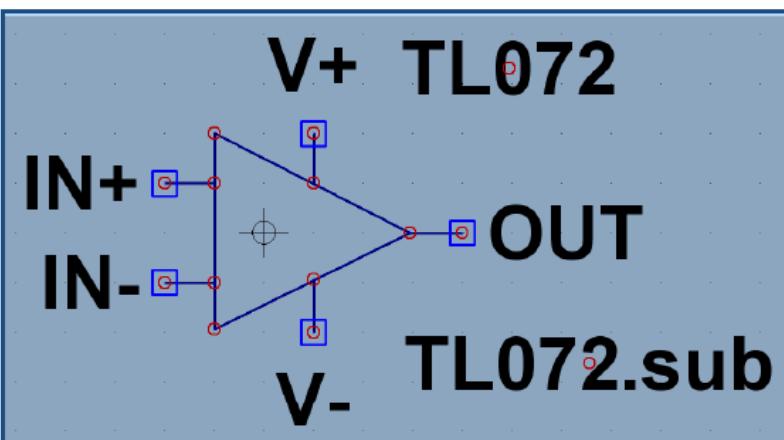




Nun wiederholt sich das Spiel: über „Edit“ und „Attributes“ kommen wir nun an „Attribute Window“ heran.

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.

ENDE

Mit Erlaubnis verwendete Datenquellen:

Reinhold Birk, Gottlieb-Daimler-Schule 2, Sindelfingen

Jürgen Richter, Technische Schule Aalen