

Labor CAE - Einfuehrung in die Simulation elektrischer Schaltungen mit LTspice (8 UE)

Andreas Schaefer - Matthias Holetzko

2023

Duale Hochschule Baden-Württemberg

-

Fakultät Technik

Studiengang Mechatronik

Introduction to LTspice

Was ist LTspice - Historisches

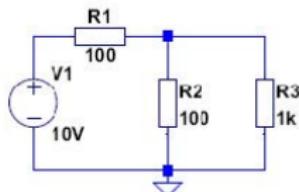
Schaltungssimulation mit LTspice

1. Viele Programme zur elektrischen Schaltungssimulation beruhen auf einem Spice Kernel
2. SPICE bedeutet: Simulation Program with Integrated Circuit Emphasis
3. SPICE wurde 1973 an der University of California at Berkeley entwickelt
4. SPICE ist eine “open source” Software (kostenlos)

Was ist LTspice - Wie funktioniert eine Simulation?

In einem klassischen SPICE Programm erfolgt die Eingabe der Schaltung über eine Netzliste.

Schaltung:



SPICE-Netzliste

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

Eine Netzliste repräsentiert eine Schaltung entsprechend Knoten- und Maschenregel.

Viele moderne SPICE Simulationsprogramme bieten dem Nutzer jedoch eine grafische Bedienoberfläche. So auch das in diesem Labor verwendete Programm, **LTspice XVII**.

Was ist LTspice - Installation & Features

Die folgenden Punkte sind eingebettete Hyperlinks:

1. Hier findet Ihr alle Infos vom Hersteller, **analog devices**
<http://ltspice.analog.com/software/LTspice.dmg>
2. Installer fuer MacOs
<http://ltspice.analog.com/software/LTspice.dmg>
3. Installer fuer Windows
<http://ltspice.analog.com/software/LTspiceXVII.exe>

Ziel dieser Laborübung

1. Erarbeiten der Grundlagen zur Simulation mit LTspice XVII
2. Eigenständige Simulation von Schaltungen für Studien-/Abschlussarbeiten
3. Verständnis theoretischer Inhalte mit empirischen Simulationen ergänzen

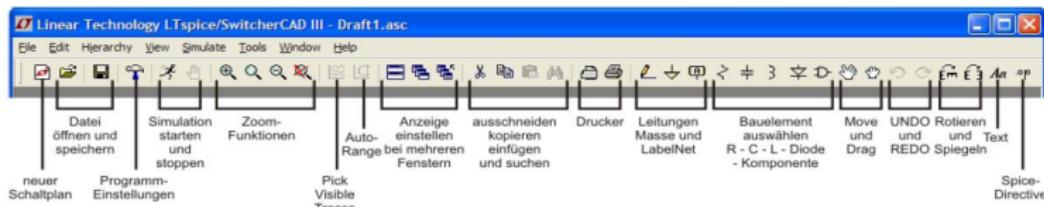
nächste Schritte

- Übungen zum Einstieg in die Simulationswelt
- Schrittweise Erarbeitung eines Kleinprojektes
Simulation einer Temperaturmessbrücke

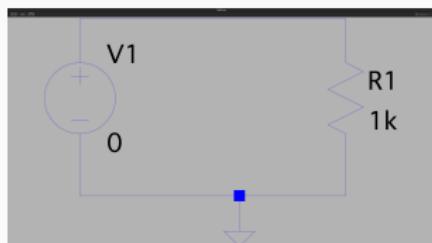
LTspice XVII - Bitte öffnet alle das Programm

Die Benutzeroberfläche von LTspice XVII besteht aus den folgenden Elementen

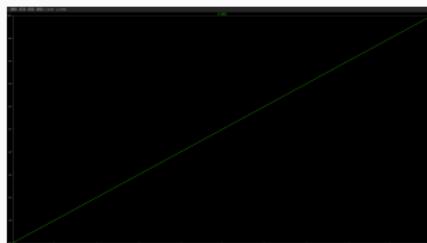
Hinweis: Die Benutzeroberfläche von LTspice auf MacOS und Windows unterscheidet sich leicht. Die Hauptmenuleiste ist für MacO nicht sichtbar.



(a) Hauptmenuleiste



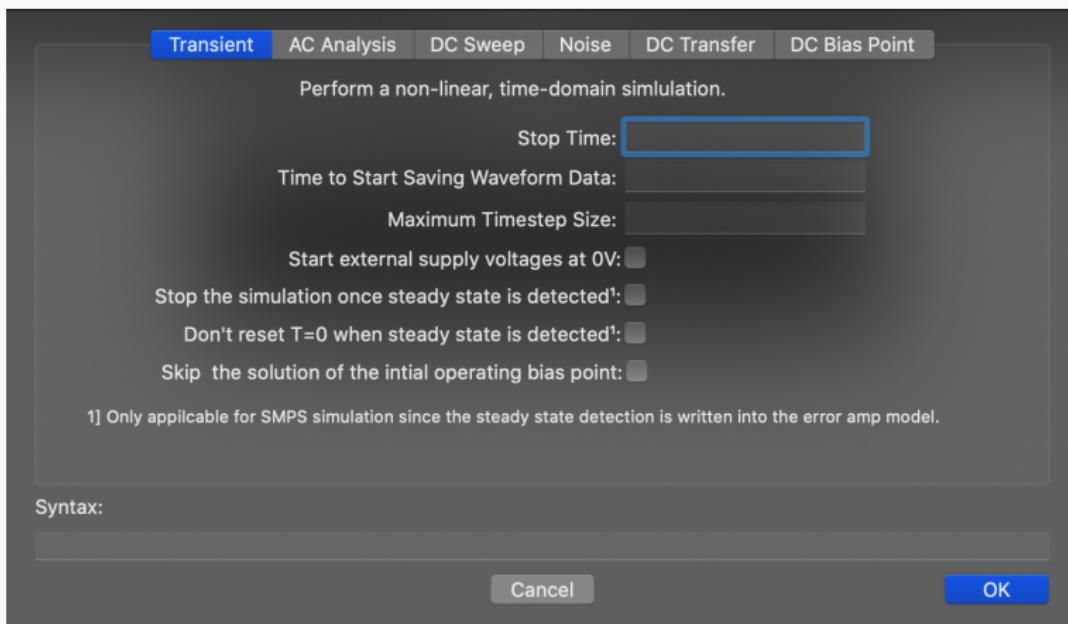
(b) schematic



(c) waveform

LTspice XVII - Überblick Simulationsarten

Die Simulationsart kann im Menu **Simulation Command** konfiguriert werden.



LTspice XVII - Überblick Simulationsarten

Transient

Analyse einer Schaltung **über die Zeit**, z.B. zur Analyse von Einschwingvorgängen

AC Analysis

Analyse einer Schaltung **unter Variation der Frequenz**,
z.B. zur Analyse von Grenzfrequenzen

DC Sweep

Analyse einer Schaltung **unter Variation der Spannung**,
z.B. zur Analyse von Bauteilkennlinien

Operation Point

Analyse des **Arbeitspunktes** einer Schaltung **ohne Variation**

Noise

Analyse von Rauschverhalten, Bauteilfehlern oder z.B. EMV Einflüssen

DC Transfer

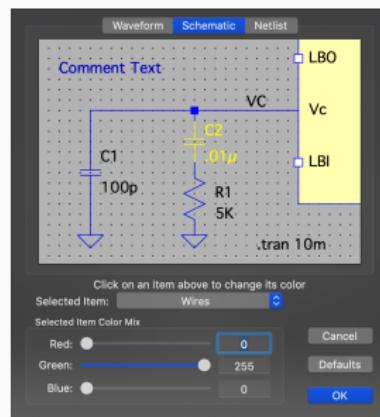
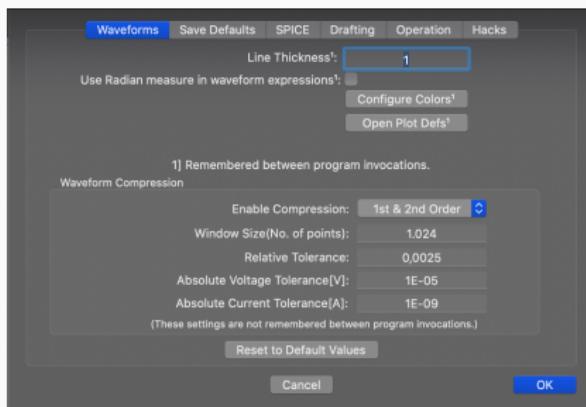
Analyse der DC-Übertragungsfunktion einer Schaltung

DC Bias Point

Analyse des Arbeitspunktes einer Schaltung ohne dynamisches Verhalten

Nützliche Voreinstellungen

Die Farbdarstellung von LTspice kann jederzeit konfiguriert werden. Geht dazu über das **control panel** Feld,  in die Farbeinstellungen (**configure colors**). Hier könnt ihr flexibel die Farbdarstellung konfigurieren.



Shortkeys

Es gibt eine gute Übersicht von HotKeys die die Arbeit mit LTspice deutlich effektiver macht (nachfolgend sind Hyperlinks embedded).

- Windows ShortKeys.
- MacOS ShortKeys.
- LTspice getting started guide.

- 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

Nützliche Hinweise

- Zahlenwerte immer mit **Dezimalpunkt statt Komma eingeben! 3.5 anstatt 3,5!**
- SPICE arbeitet nach dem Knotenpotenzialverfahren - **Es muss immer ein Bezugsknoten (Masse, Ground) angegeben werden**, sonst treten Simulationsfehler auf (Achtung: Es erfolgt keine Fehlermeldung!).

Achtet darauf die korrekten Suffixe für Einheiten zu verwenden.

Terra	Giga	Mega	Kilo	milli	micro	nano	pico	femto
T	G	Meg	K	m	u	n	p	f
e^{12}	e^9	e^6	e^3	e^{-3}	e^{-6}	e^{-9}	e^{-12}	e^{-15}

Hands-on LTspice - Das Labor beginnt

Vorgehensweise

Diese Übungen sind zur interaktiven Arbeit mit LTspice gedacht.
Sie führen von der Aufgabe über den Lösungsansatz bis hin zur fertigen Analyse.

Bitte geht nach den folgenden Schritten vor, um die Übungen strukturiert zu bearbeiten.

1. **Schematic modellieren**
2. **Simulationsart analysieren**
3. **Simulationsart konfigurieren**
4. **Ergebnis mit der Vorgabe abgleichen**

Sollte es zu Problemen kommen, bitten wir euch uns direkt einen Hinweis zu geben!

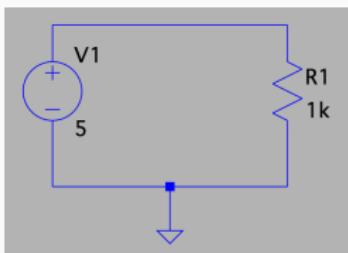
Übungen

ohmscher Widerstand

Ziel - Darstellung von Bauteilkennlinie

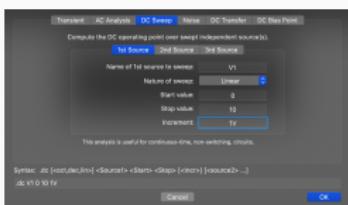
Das ohmsche Gesetz $U = RI$ lässt sich mit Hilfe eines kleinen Schaltungsexperimentes gut visualisieren und nachvollziehen. Hierzu werden wir die folgende Schaltung aufbauen und den Stromverlauf über dem Widerstand darstellen.

Erstellung des Schaltplans



- Erstellt ein neues schematic (File –> new schematic)
- Speichert es direkt als neue Datei ab (File –> save as)
- Öffnet den Bauteileditor (**F2**) und für eine Spannungsquelle hinzu (voltage)
- Öffnet erneut den Bauteileditor und für einen Widerstand hinzu (resistor bzw. EuropeanResistor fuer die bekannte Box-Darstellung)
- Fügt einen Masseknoten als Bezugsknoten hinzu (**F4**).
- Verdrahtet die Schaltung (**F3**)

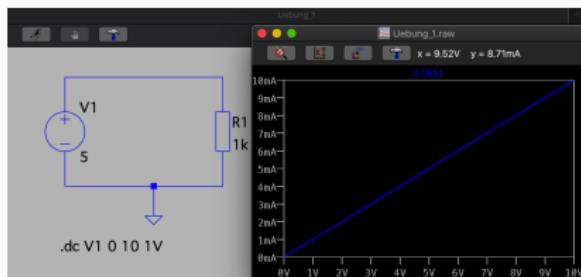
Konfiguration der Simulation



- Im Menu Simulation, wählt **Edit simulation command** und wählt den DC Sweep.
- Unser Ziel ist es den Stromverlauf über dem Widerstand zu simulieren
- Daher konfigurieren wir die Spannungsquelle **V1** mit einem **linearen Sweep** von 0 bis 10 V mit einer **Schrittweite** von 1V.
- Bestätigt mit **OK** und fügt die Sumlationsansweisung dem schematic hinzu

ohmscher Widerstand

Simulation und Analyse



- Klickt auf (run) und LTspice startet die Simulation
- Der waveform viewer öffnet sich - ihr könnt auf zwei Arten Strom, Spannung und Leistungen aus eurer Schaltung anzeigen lassen.

Im schematic könnt ihr über ein Bauteil mit der Maus fahren und es erscheint eine Stromzange

Im schematic könnt ihr über ein Bauteil mit der Maus fahren, Shift gedrückt halten und es erscheint eine Leistungsmessanzeige

Im schematic könnt ihr über einen Knoten mit der Maus fahren und es erscheint ein Spannungsmesser

Im waveform viewer könnt ihr über rechten Mausklick — > add trace die verfügbaren Messtellen direkt auswählen

- Wir wählen **I(R1)**, den Strom durch unseren Widerstand R1.

Ergebnis und Auswertung

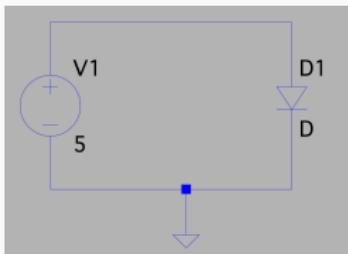
Wie zu erwarten liefert dieses einfache Beispiel den Zusammenhang zwischen Strom, Spannung und Widerstand. Probiert an dieser Stelle gerne die Spannung, Schrittweite zu variieren oder weitere Werte im waveform viewer anzueigen zu lassen.

Diode

Ziel - Darstellung von Bauteilkennlinie

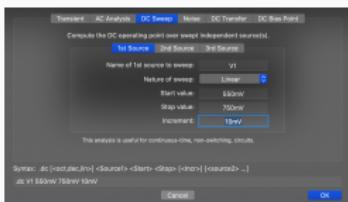
Eine Diode wird oft durch Ihre Durchlassspannung klassifiziert. Dies ist die Spannung ab der die Diode leitend wird. Leitend ist Sie, wenn ein Strom fließt. Dieses Verhalten möchten wir in einem kleinen simulativen Experiment herausarbeiten.

Erstellung des Schaltplans



- Speichert das Projekt direkt als neue Datei ab (File –> save as)
- Löscht den Widerstand heraus (**F5**)
- Öffnet erneut den Bauteileditor (**F2**) und für eine einfache Diode hinzufügen (sucht nach Diode ...)
- Verdrahtet die Schaltung wieder vollständig (**F3**)

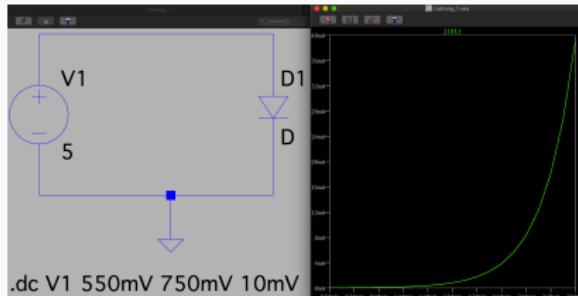
Konfiguration der Simulation



- Im Menu Simulation, wählt **Edit simulation command** und bleibt beim **DC Sweep**.
- Unser Ziel ist es den Stromverlauf durch die Diode zu messen. Dafür müssen wir die Spannung erhöhen. Wir wissen, dass die notwendige Spannung im Bereich 600 - 700 mV liegen muss.
- Daher konfigurieren wir die Spannungsquelle **V1** mit einem **linearen Sweep** von 550 bis 750 mV mit einer **Schrittweite** von 10 mV.
- Bestätigt mit **OK** und fügt die Sumulationsansweisung dem schematic hinzu

Diode

Simulation und Analyse



- Klickt auf (run) und LTspice startet die Simulation
- Wir wählen **I(D1)**, den Strom durch die Diode.

Ergebnis und Auswertung

Wie zu erwarten liefert dieses einfache Beispiel den Zusammenhang zwischen Strom, Spannung und Widerstand. Probiert den Spannungsbereich des DC-Sweep von 550 - 750 mV auf 550 mV - 2V zu erhöhen.

- Was fällt euch auf?
- Könnt ihr euch herleiten, warum man eine Diode immer mit einem Vorwiederstand betreiben sollte?

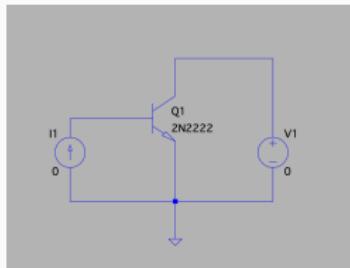


Mit einem rechten Mausklick auf die Diode könnte ihr die Dioden Typen variieren. (**pick new diode**) Zum Beispiel von einer idealen aus dem obigen Beispiel zu einer beliebigen realen entsprechend dem Modell des Herstellers, schaut euch die Unterschiede an.

Ziel - Darstellung des Ausgangskennlinienfeld

Das Ausgangskennlinienfeld eines npn Transistors beschreibt den Zusammenhang von Kollektorstrom I_c und der Spannung an der Kollektor-Emitter Strecke U_{ce} . Das Kennlinienfeld wird für verschiedene Basisströme I_b angegeben.

Erstellung des Schaltplans



- Startet mit einem neuen schematic
- Speichert das Projekt direkt als neue Datei ab (File –> save as)
- Fügt eine ideale Stromquelle (**F2 ... current**) hinzu und dreht dieser (**STRG+R**)
- Fügt einen npn-Transistor(**F2 ... npn**) hinzu.
- Per rechtem Mausklick könnt ihr vom idealen npn analog zur Diode im letzten Beispiel zum 2N2222 wechseln.
- Verdrahtet die Schaltung wieder vollständig (**F3**)

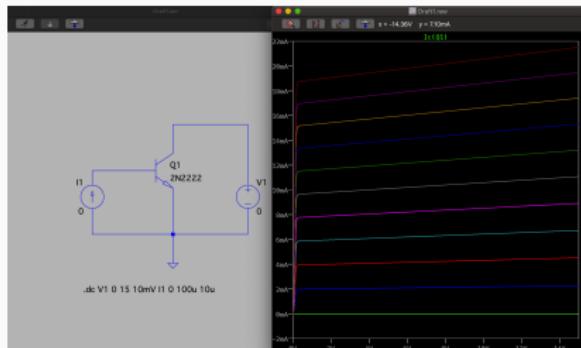
Konfiguration der Simulation



- Im Menu Simulation, wählt **Edit simulation command** und bleibt beim DC Sweep.
- Unser Ziel ist es das Ausgangskennlinienfeld des 2N2222 simlativ zu bestimmen. Dazu verwenden wir einen DC-Sweep mit einer ideal Stromquelle (I1) die verschiedene Basisströme I_b simuliert und eine Spannungsquelle (V1) die die Spannung U_{ce} simuliert.
- V1 soll von 0 - 15V in 10mV Schritten simuliert werden, I1 von 0 - 100u in 10u Schritten.
- Bestätigt mit **OK** und fügt die Simlationsansweisung dem schematic hinzu

NPN-Transistor

Simulation und Analyse



- Klickt auf (run) und LTspice startet die Simulation
- Wir wählen **IC(Q1)**, den Kollektorstrom.

Ergebnis und Auswertung

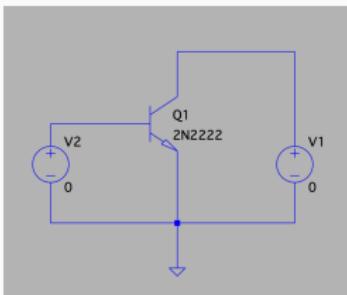
Jeder Graph steht für einen simulierten Basisstrom. Per rechtem Mausklick –> View –> Steps könnt ihr einzelne Graphen zur detaillierten Analyse auswählen.

Achtet darauf, welche Quelle ihr im `.dc...` simulation command zuerst wählt. **Quelle 1 ergibt im Diagramm die Abszisse, die Quelle 2 die Ordinate.**

NPN-Transistor

Wenn ihr den Zusammenhang zwischen der Basisspannung U_{be} und dem Kollektorstrom I_c simulativ herausfinden wollt, müsst ihr die Schaltung leicht variieren. Dazu werden wir das schematic wie folgt anpassen.

Erstellung des Schaltplans



- Speichert das Projekt direkt als neue Datei ab (File –> save as)
- Löscht die Stromquelle (**F5**) und fügt eine Spannungsquelle hinzu.
- Verdrahtet die Schaltung wieder vollständig (**F3**)
- Super simple dieses mal :)

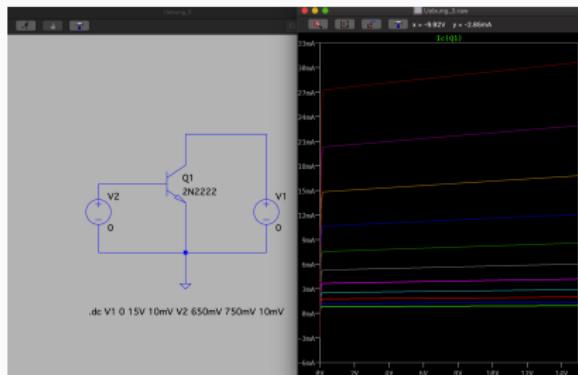
Konfiguration der Simulation



- Im Menu Simulation, wählt **Edit simulation command** und bleibt beim DC Sweep.
- Unser Ziel ist es das Ausgangskennlinienfeld des 2N2222 simlativ zu bestimmen. Dazu verwenden wir dieses mal einen DC-Sweep mit einer Spannungsquelle, die die Basis-Emitterspannung U_{be} simuliert und eine Spannungsquelle (V1) die die Spannung U_{ce} simuliert.
- V1 soll von 0 - 15V in 10mV Schritten simuliert werden, V2 von 650 - 750 mV in 10mV Schritten.
- Bestätigt mit **OK** und fügt die Simulationsansweisung dem schematic hinzu

NPN-Transistor

Simulation und Analyse



- Klickt auf (run) und LTspice startet die Simulation
- Wir wählen **IC(Q1)**, den Kollektorstrom.

Ergebnis und Auswertung

Jeder Graph steht für eine simulierte U_{be} . Per rechtem Mausklick –> View –> Steps könnt ihr einzelne Graphen zur detaillierten Analyse auswählen.

Achtet darauf, welche Quelle ihr im `.dc...` simulation command zuerst wählt. **Quelle 1 ergibt im Diagramm die Abszisse, die Quelle 2 die Ordinate.**

OPV Schaltungen - transient, ideal

Ziel - Simulation eines invertierenden Verstärkers

Wir wollen in einem einfachen simulativen Experiment die Funktionalität eines invertierenden Verstärkers nachvollziehen.



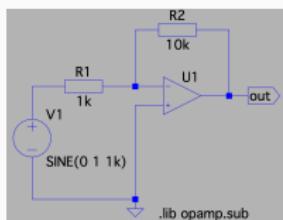
$$V_{out} = -\frac{R2}{R1} V_1 \quad (1)$$

Wenn man nur daran interessiert ist die grundsätzliche Funktionalität einer Schaltung zu verifizieren, kann man ideale Bauelemente aus der LTspice Bibliothek verwenden. Wie im obigen Beispiel benötigt z.B. der Operationsverstärker entgegen der Realität dann keine Spannungsversorgung. Für einen idealen OPV bietet LTspice das Bauelement **opamp** an, welches Ihr im Bauteileditor findet direkt über die Suche findet.

Wichtig: Ihr müsst als Spice-Direktive jedoch noch ein zu verwendendes Model einbinden. (.lib opamp.sub)

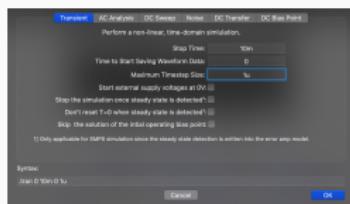
Erstellung des Schaltplans

- Baut den Schaltplan mit dem o.g. opamp als OPV auf.
- Über einen rechten Mausklick kommt ihr ins **Advanced** Menü der Spannungsquelle. Hier könnt ihr Sie als Signalgenerator konfigurieren. Wir wählen einen **Sinus mit 1V Amplitude und der Frequenz 1kHz**
- Die library kann direkt als spice directive eingebunden werden (siehe Folie 5, .op)
- Ihr könnt über die Funktion **Label Net (F4)** einen Knoten umbenennen und ihm mit einem Symbol für In-/Output versehen. Nennt den Ausgang der Schaltung z.B. *out*.
- **Hinweis:** Wenn ihr einen Knoten benennt, dann kann liegt unter diesem Namen überall im Schaltplan das selbe! Potential an. Dadurch könnt ihr den Plan übersichtlicher gestalten.



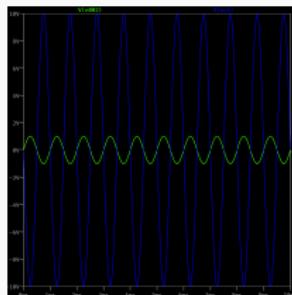
OPV Schaltungen - transient, ideal

Konfiguration der Simulation



- Im Menu Simulation, wählt **Edit simulation command** und wählt eine **Transient Analyse**.
- Unser Ziel ist es am Ausgang eine Verstärkte Spannung entsprechend des Verstärkungsfaktors der Schaltung zu messen.
- Da die Schaltung ideal kein Einschwingverhalten zeigt, starten wir die Simulation mit der Aufzeichnung und simulieren 10ms mit einer Schrittweite von 10us.
- Bestätigt mit *OK* und fügt die Sumlationsansweisung dem schematische hinzu

Simulation und Analyse



- Klickt auf (run) und LTspice startet die Simulation
- Fügt nun die Eingangsspannung sowie die Ausgangsspannung der Schaltung als Messpunkte hinzu.

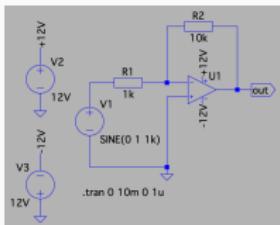
Ergebnis und Auswertung

Verifiziert, ob die Verstärkung sowie das invertierende Verhalten zu eurem Erwartungswert (10) passt.

OPV Schaltungen - transient, nicht ideal

Im nächsten Experiment wollen wir die gleiche Schaltung mit einem "nicht idealen" Verstärker simulieren und die Verwendung von Labels verdeutlichen.

Erstellung des Schaltplans



- Tauscht den opamp gegen das Bauteil **UniversalOpamp2** aus dem Bauteileditor (**F2**).
- Fügt zwei neue Spannungsquellen für die Versorgung des OPV's hinzu.
- Achtet hierbei auf darauf, dass die Label wirklich auf dem Potential +12V und -12V liegen.
- Erstellt die Labels +12V und -12V (**Label Net (F4)**).

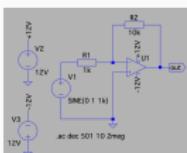
Ergebnis und Auswertung

Es muss das selbe Ergebnis herauskommen wie im vorherigen Experiment unter Verwendung des idealen OPV's **opamp**.

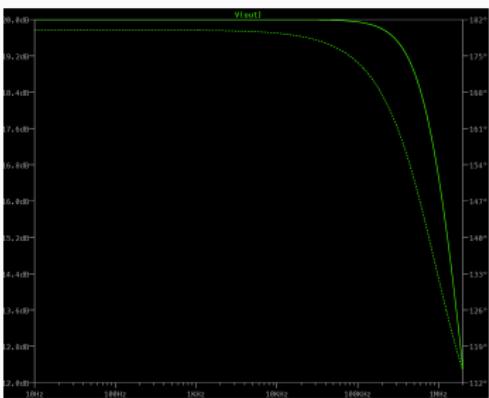
OPV Schaltungen - Analyse der Grenzfrequenz einer Schaltung

Eine Kennzahl von Tief-/Hochpassfiltern ist ihre 3dB Grenzfrequenz f_g . Die Grenzfrequenz kann man analytisch, jedoch auch simulativ über LTspice bestimmen. Hierzu bietet LTspice die Möglichkeit die Frequenz einer Schaltung zu variieren. Dies wird AC-Sweep genannt. Im Bode-Diagramm kann man den logarithmischen Verlauf der Amplitude über der Frequenz in LTspice darstellen und somit einfach grafisch die Grenzfrequenz bestimmen indem man den Punkt heraussucht, bei dem die Amplitude um 3dB abgefallen ist.

Erstellung des Schaltplans + Konfiguration der Simulation



Simulation und Analyse

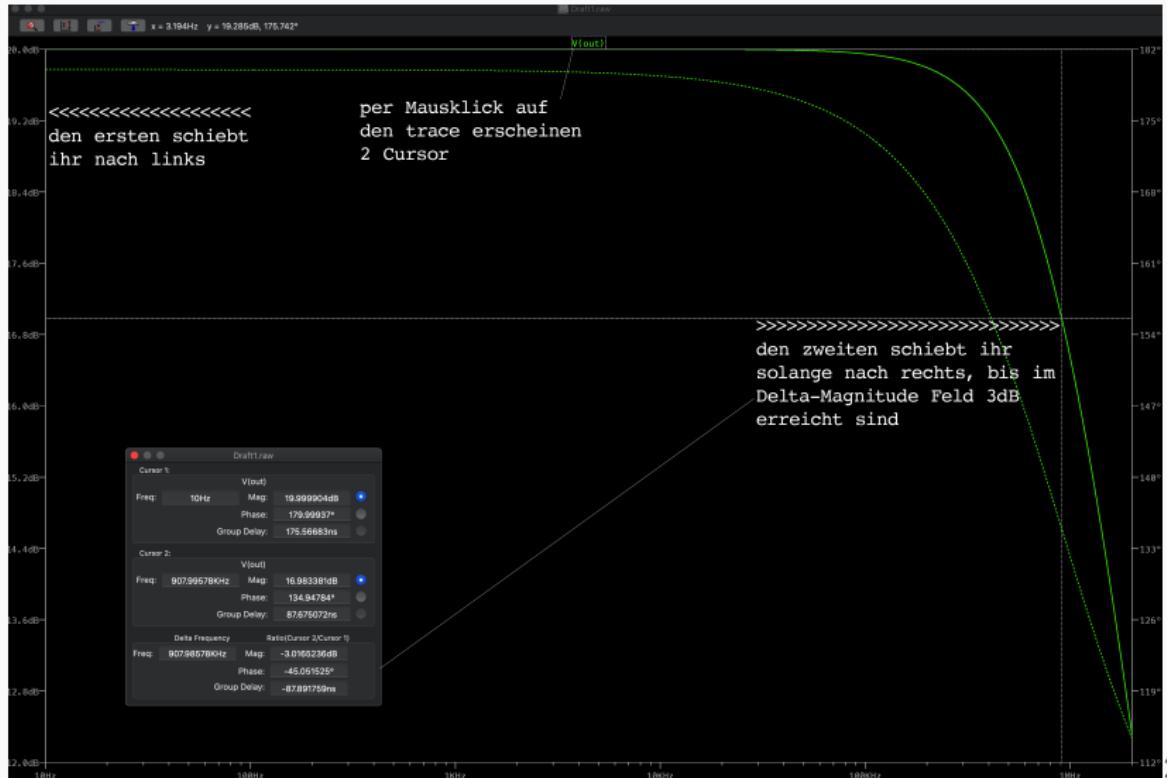


- Wir verwenden die Schaltung aus dem vorherigen Beispiel!
- Wichtig ist hierbei, dass wir Spannungsquelle für den AC-Sweep konfigurieren. Hierzu geht ihr per rechtem Mausklick in das advanced menu von V1 und stellt das Kleinsignal Verhalten (AC) auf **Amplitude 1V und Frequenz 1kHz**.

- Klickt auf (run) und LTspice startet die Simulation
- Fügt nun die Ausgangsspannung der Schaltung als Messpunkt hinzu.
- Im waveform viewer solltet ihr das Bode-Diagramm mit Amplitude und Phase sehen.

OPV Schaltungen - Analyse der Grenzfrequenz einer Schaltung

Simulation und Analyse

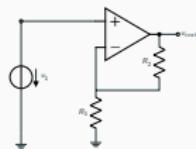


OPV Schaltungen - nicht-invertierender Verstärker

Ziel - Anwendung der Kenntnisse

Nachfolgend könnt ihr die Schaltung für einen nicht-invertierenden Verstärker in einer OPV Schaltung sehen.

1. Baut die Schaltung auf, dimensioniert sie so, dass Sie einen Verstärkungsfaktor
2. Verwendet die UniversalOpamp2 aus der vorherigen Übung mit einer Versorgungsspannung von +/-12V
3. Verifiziert eure Schaltung und das **nicht-invertierende** Verhalten mit einer transienten Simulation von 0 - 50ms.
4. Wählt dabei eine ausreichend kleine Schrittweite
5. Wechselt die Simulationsart zum AC-Sweep und ermittelt die Grenzfrequenz



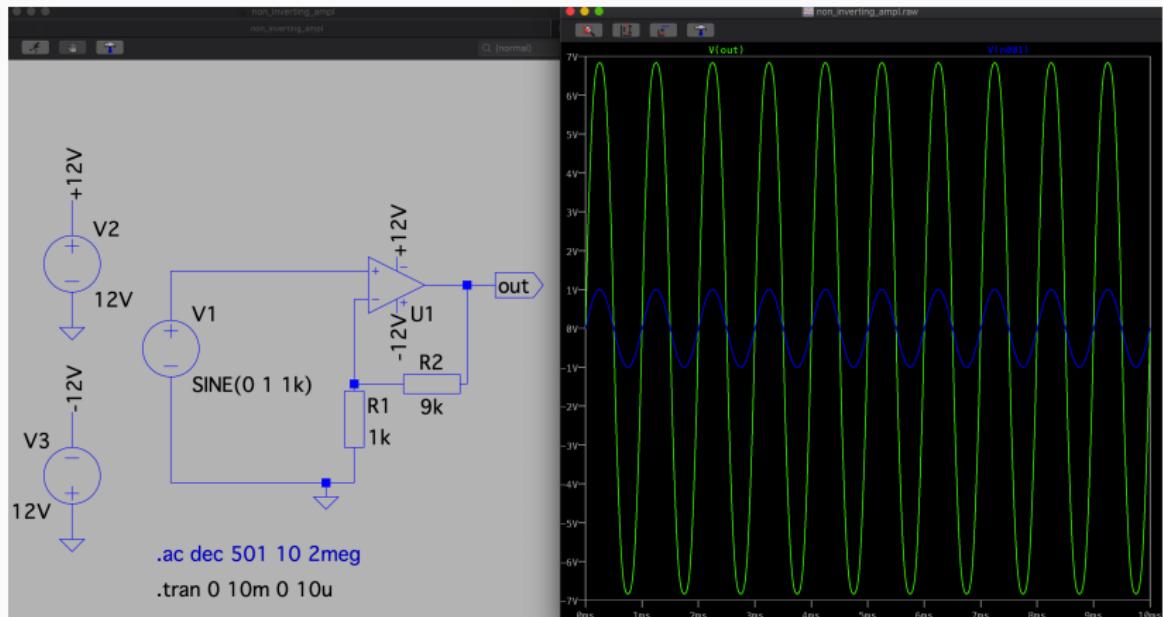
$$V_{out} = \left(1 + \frac{R2}{R1}\right)V_1 \quad (2)$$

Spoiler! - Die Lösungen folgen auf den folgenden Folien. Wenn sie sich nicht sicher sind, schauen Sie einfach nach.

Info! - Besonders die Analyse von Grenzfrequenzen ist für Ihre weitere Vorlesungszeit (Filter höherer Ordnung, EMV) hilfreich, da Sie neben der simulativen Bestätigung Ihrer Schaltungen auch theoretische Rechenaufgaben simulieren und so Ihre Rechnung verifizieren können.

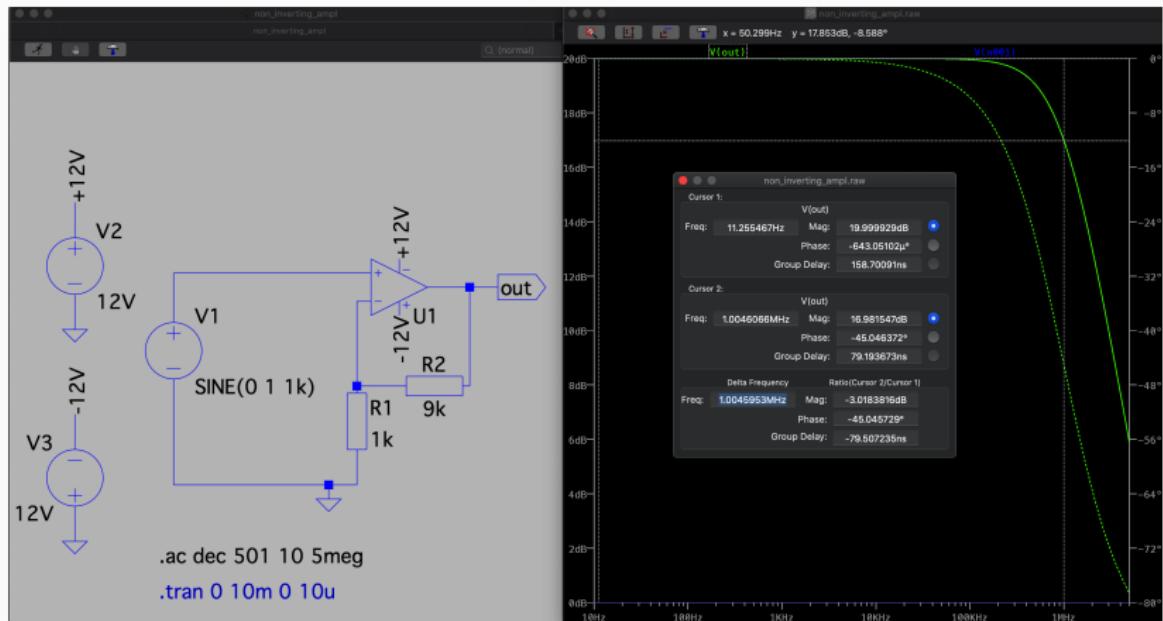
OPV Schaltungen - Lösung transient

Simulation und Analyse



OPV Schaltungen - Lösung Grenzfrequenz

Simulation und Analyse

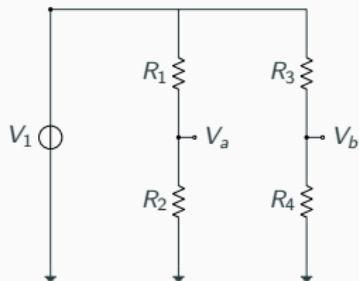


Simulation vs. Realität

Wheatstone Bridge

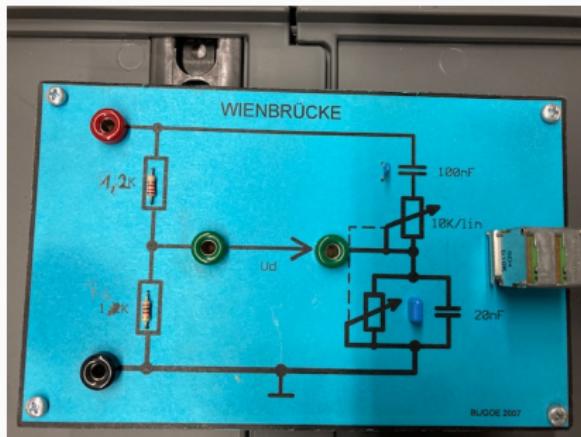
Die Wheatstone Bridge ist eine einfache Schaltung um einen Widerstand zu messen. Die Schaltung gilt als abgeglichen, wenn an V_{AB} keine Spannung anliegt.

Die theoretische Herleitung ist nicht ganz trivial, daher möchten wir an dieser Stelle die Realität und den Aufwand eines praktischen Aufbaus der Flexibilität einer Simulation gegenüberstellen.



Experiment

Wir werden jetzt den folgenden Aufbau "live" analysieren.
Beachtet, dass die Schaltung komplexer ist als in dem einleitenden schematischen Aufbau - wie immer.



Leitet aus der gemessenen Spannung den Widerstandswert her.

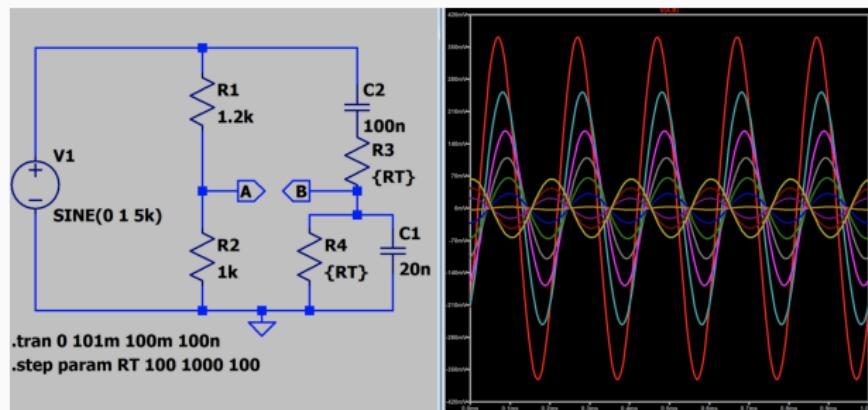
Simulation - Vorgehen

Wir wollen die Wienbrücke bei einer festen Spannungsversorgung $Sine(0 \ 1 \ 5k)$ abgleichen.

- Hierfür variieren wir den Trimmer bis die Differenzspannung U_{AB} null beträgt.
- Via LTspice können wir uns dem abgeglichenen Zustand annähern, indem wir den Trimmer als Variable betrachten.
- Hierfür nutzen wir die Variable RT mit der Spice directive `.step param RT Anfangswert Endwert Inkrement` und tasten uns an die ideale Trimmereinstellung heran.
- Mittels der Funktion `select steps` ermitteln wir die am besten abgeglichenen Widerstände und passen unsere Variable sukzessive an.

Simulation - Transient

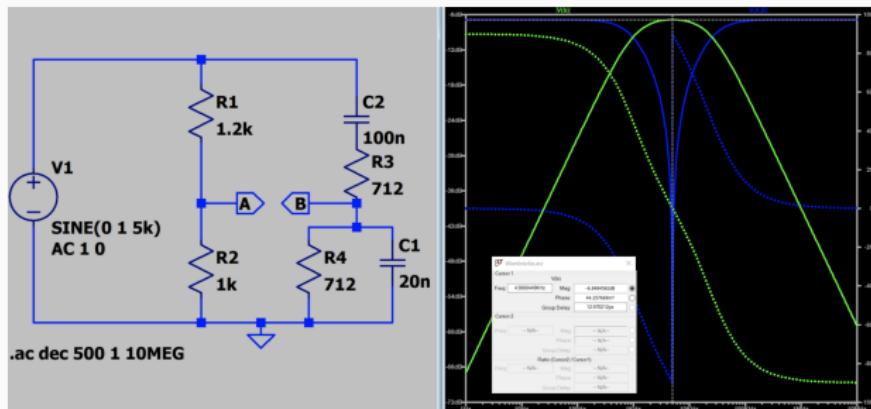
Bitte baut die unten stehende Schaltung nach und führt das Experiment durch.



Bei etwa 7120Ω ist die Brücke abgeglichen - wir ersetzen die Variable durch einen festen Widerstandswert.

Simulation - AC

Simulieren wir nun das Verhalten über die Frequenz via AC Analysis.



Simulation - AC Analyse

Betrachten wir zunächst die Spannung U_A , bzw. U_B .

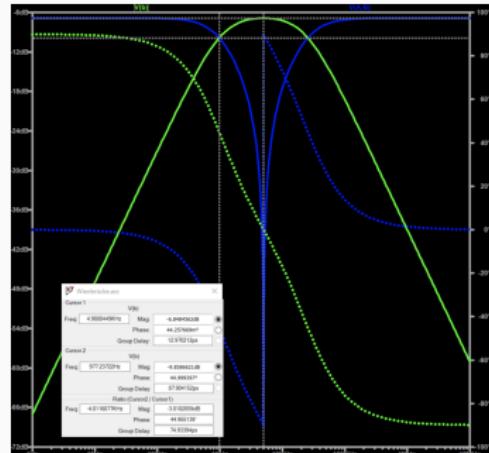
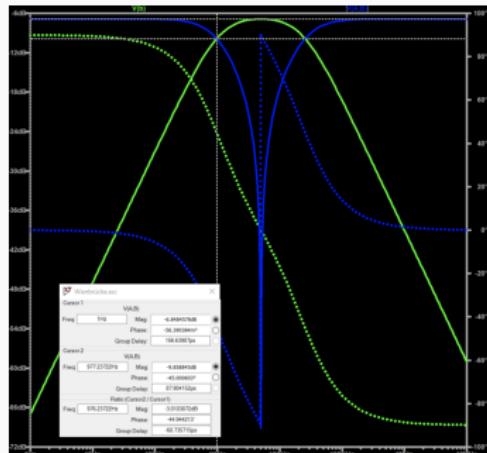
- Im rechten Teilzweig der Schaltung ergibt sich durch den in Reihe verbauten Tiefpass und Hochpass ein Bandpassverhalten.
- Lediglich in einem mittleren Frequenzbereich um 5kHz ergibt sich eine Spannung U_{AB} .

Hieraus resultiert ein umgekehrtes Verhalten der Differenzspannung U_{AB} . Hier zeigt sich das Verhalten einer Bandsperre. Die Differenzspannung U_{AB} ist bei 5kHz ideal abgeglichen und die Spannungen U_A und U_B heben sich gegenseitig auf.

Simulation - AC Grenzfrequenzen

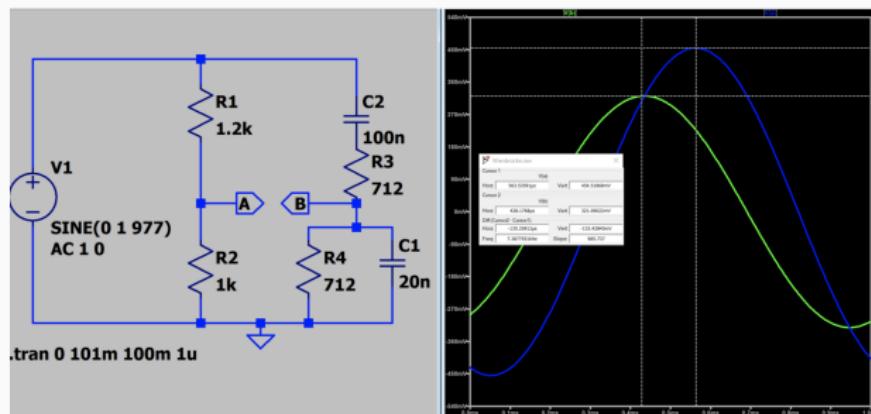
Die Schaltung hat ja nach Betrachtung ein Bandpass- bzw. Bandsperrverhalten. Wir bestimmen die Grenzfrequenzen für beiden Messpunkte, U_A und U_{AB} .

- U_A Bandpass - **977 kHz**
- U_{AB} Bandsperre - **977 kHz**



Simulation - Abgleich mit Theorie

Bei einer Simulation ist es immer wichtig zu überprüfen, ob das Ergebnis durch eine theoretische Annahme verifiziert werden kann. Übernehmen wir nun die untere Grenzfrequenz $f_g = 977\text{Hz}$ in unsere Spannungsquelle und betrachten nochmals den Zeitbereich über die transiente Simulation.



Das erwartete Verhalten bei der Grenzfrequenz f_g ist, dass die Leistung auf $\frac{1}{\sqrt{2}}$ sinkt.

Simulation - Abgleich mit Theorie

- Wir können mittels der Cursor die Spannung $U_A = 455mV$ und Spannung $U_B = 321mV$ ermitteln.
- Dies stimmt mit unserer Erwartungshaltung überein.

$$U_B = \frac{455mV}{\sqrt{2}} = 321mV$$

Verwendung & Modellierung realer Bauteile

Reale Bauteile modellieren

LTS spice ist eine wunderbare Simulationsumgebung, allerdings ist die Standardbibliothek auf die hauseigenen Bauteile von Linear Technology begrenzt.

Oftmals ist es jedoch notwendig in einer Simulation ein Bauteil zu verwenden, das einfach verfügbar ist. Als Beispiel dafür möchten wir gerne einen TS912 modellieren. Dieser Operationsverstärker ist leicht beschaffbar und günstig.

Dazu gehen wir in 3 Schritten vor:

1. Wir suchen im Internet nach dem **spice model** eines Herstellers
2. Wir erstellen einen **subcircuit** in unserer LTS spice Bibliothek
3. Wir erstellen ein LTS spice **Symbol**

TS912 - Recherche

Eine einfache Recherche für uns zu einem Hersteller, hier zum Beispiel ST.

<https://www.st.com/en/amplifiers-and-comparators/ts912.html>

The screenshot shows the product page for the TS912 operational amplifier. At the top, there are tabs for Overview, Sample & Buy, Documentation, CAD Resources (which is active), Tools & Software, and Quality & Reliability. Below the tabs, the product name 'TS912' is listed with an 'ACTIVE' status. A large heading says 'Low power with CMOS inputs'. There are two buttons: 'Download datasheet' and 'Order Direct'. A navigation bar below the heading includes 'Overview', 'Sample & Buy', 'Documentation', 'CAD Resources' (highlighted with a red box), 'Tools & Software', and 'Quality & Reliability'. A call-to-action button 'Find all our CAD and simulation models' is also present in this bar. To the right, there's a section titled 'EDA Symbols, Footprints and 3D Models' with a sub-section for 'STMicroelectronics - TSSOP'. It features icons for 'Symbols', 'Footprints', and '3D models'. A 'Choose CAD Format & Download' button is available. Below this, a 'All resources' section lists a single item: 'SPICE models (1)'. This item is a link to the 'TSSOP Spice model' version 1.0 from 11 Dec 2020.

Subcircuit erstellen

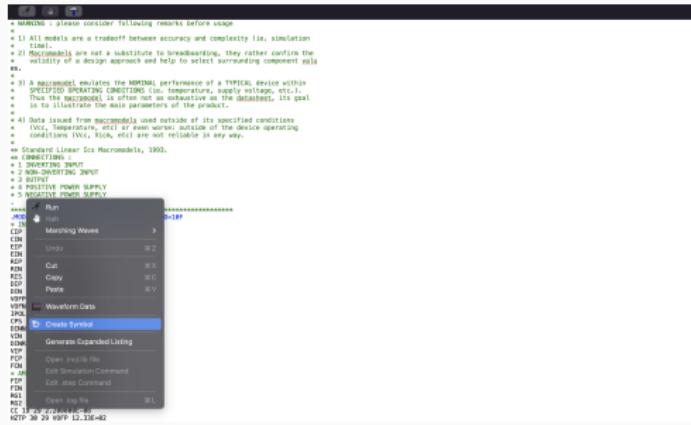
Ein spice Model ist eine einfache Text-Datei, die einfach in LTSpice eingebunden werden kann. Kopiert den **subcircuit** dazu in eine Text-Datei mit der Endung .sub **TS912.sub** und speichert diese im Ordner:

1. Windows C:/Users/"USER"/Documents/LTspiceXVII/lib/sub
2. MacOS /Users/"user"/Library/ApplicationSupport/LTspice/lib/sub.

```
* WARNING : please consider following remarks before usage
*
* 1) All models are a tradeoff between accuracy and complexity (ie. simulation
*    time).
* 2) Macromodels are not a substitute to breadboarding, they rather confirm the
*    validity of a design approach and help to select surrounding component values.
*
* 3) A macromodel emulates the NOMINAL performance of a TYPICAL device within
*    SPECIFIED OPERATING CONDITIONS (ie. temperature, supply voltage, etc.).
*    Thus the macromodel is often not as exhaustive as the datasheet, its goal
*    is to illustrate the main parameters of the product.
*
* 4) Data issued from macromodels used outside of its specified conditions
*    (Vcc, Temperature, etc) or even worse: outside of the device operating
*    conditions (Vcc, Vtcm, etc) are not reliable in any way.
*
** Standard Linear Ics Macromodels, 1993.
** CONNECTIONS :
* 1 INVERTING INPUT
* 2 NON-INVERTING INPUT
* 3 OUTPUT
* 4 POSITIVE POWER SUPPLY
* 5 NEGATIVE POWER SUPPLY
.SUBCKT TS912 1 2 3 4 5
*****
.MODEL MOTH D IS=1E-8 KF=6.563355E-14 CJO=10F
* INPUT STAGE
CIP 2 5 1.50000E-12
```

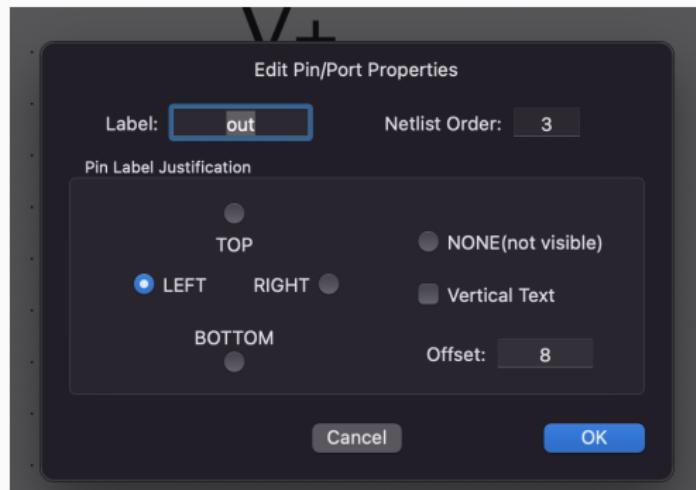
LTSpice Symbol erstellen

1. Öffnet LTspice und öffnet das soeben erstellte File.
 2. Mac: More Choices - Open Other Type of File
 3. Windows: Drag and Drop der Datei in LTspice
 4. Jetzt markiert ihr die Zeile mit dem **.subckt** klickt ihr mit der rechten Maustaste
 5. Klickt nun auf "Create Symbol"



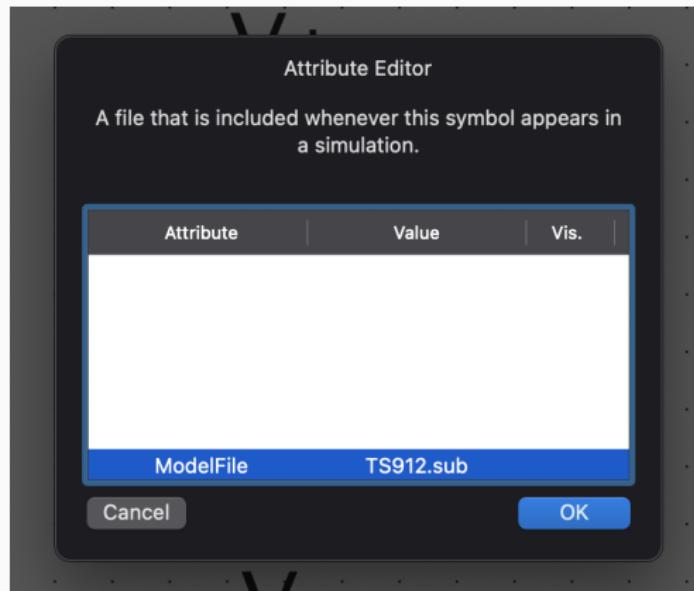
LTS defense Symbol modellieren 1

Nun könnt ihr hier über das Zeichentool kreativ werden. Wichtig ist, dass ihr die Symbol Attribute korrekt einstellt. Ihr kommt in die Attribut Dialog über (Edit - Attributes) . Die Pin Nummerierung muss mit der im subcircuit übereinstimmen, klickt mit der rechten Maustaste auf einen Pin. Die Benamung der Pins könnt ihr natürlich auch ändern.



LTSpice Symbol modellieren 2

Zudem müssen wir in LTspice den Bezug zum subcircuit definieren. Dazu stellt ihr sicher, dass die Variable ModelFile auf unser zuvor gespeichertes File zeigt. (TS912.sub)



Bauteil verwenden

Wir haben nun:

1. Einen subcircuit heruntergeladen
2. Ihn unter dem Namen TS912.sub im **sub** Ordner von LTspice gespeichert
3. Aus dem subscircuit ein Symbol erstellt
4. Dieses Symbol angepasst und zum subcircuit verlinkt

Jetzt können wir über den Bauteileditor (F2) im Menü **AutoGenerated**, unseren TS912 laden und in LTspice verwenden.

Hint: Bei einem Mac müsst ihr nach dem Erstellen einmal neu starten.

Projekt - Temperaturmessbrücke

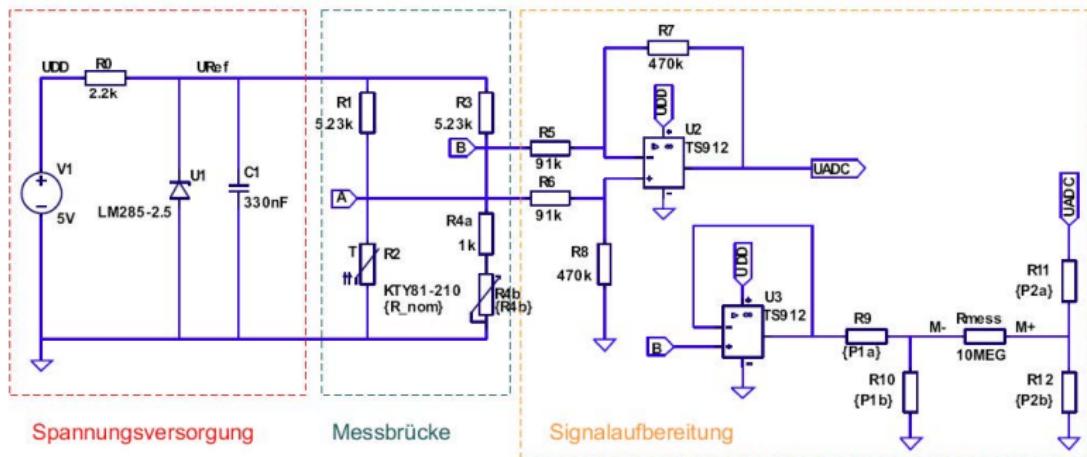
Überblick

Kleinprojekt: Temperaturmessbrücke

Blockschaltbild



Schaltplan



Vorgehensweise

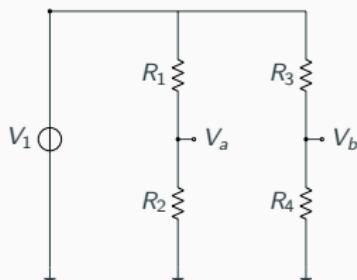
Zu diesem Zeitpunkt sollten wir alle in der Lage sein einfache Simulationen in DC-, AC-Sweep sowie transient durchführen zu können. Sie sollten den Bauteileditor sowie die grundlegenden Schematic-Funktionen (Rotate, Cut, ...) sicher beherrschen.

Wir werden nun die Brücke und die vorgestellten Bestandteile Stück für Stück aufbauen. **Bitte beachtet Folgendes:**

1. Bitte speichert alle Zwischenschritte ab, wir werden Teile später wiederverwenden
2. Bei Fragen bitten wir euch uns direkt im MS Teams zu benachrichtigen, sodass wir euren Fortschritt so gut wie möglich unterstützen können.

Die Messbrücke - Arbeitspunktanalyse

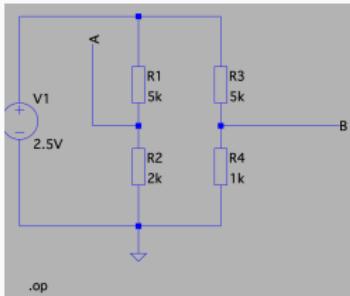
Die Grundlage für unsere Messbrücke bildet eine einfache Wheatstonesche Brückenschaltung. Zur Wiederholung findet ihr unten den Schaltplan sowie die Formel zur Berechnung der Brückenspannung. V_{AB} .



$$V_{AB} = \frac{R_2 R_3 - R_4 R_1}{(R_1 + R_2)(R_3 + R_4)} V_1 \quad (3)$$

Die Messbrücke - Arbeitspunktanalyse

Simulation



- Ihr baut die Brückenschaltung entsprechend auf und verdrahtet Sie
- Wir wählen eine neue Analyseart, die Arbeitspunktanalyse (.op == operation point)
- Ihr fügt .op als spice-Direktive hinzu (oder wählt im simulation cmd DC Bias Point)
- Ihr klickt auf (run)

Exkurs zur Netzliste

```
+ /Users/mhainz/Downloads/E2_projects/E2_vorlesung/cap_script/schematics/bridge_op_1.asc  
bridge_op_1.net  
V1 N001 0 2.5V  
R1 N001 A 5k  
R2 A B 2k  
R3 N001 B 5k  
R4 B 0 1k  
.op  
.backanno  
.end
```

This is node N002. DC operating point: V(n002) = 714.286mV

- Der Spice Algorithmus mit einer Netzliste.
- Jeder Knoten im Schaltplan erhält einen Namen.
- Ihr könnt euch diese Netzliste per rechtem Mausklick –> View –> SPICE Netlist anzeigen lassen.
- Die Masse hat immer den Knotennamen 0
- Wird ein Knoten nicht explizit mit einem Label (**F4**) versehen, numeriert LTspice sie aufsteigend durch (N001, N002, ...)
- Weiterhin folgt eine Reihe in der Netzliste dem Schema <Bauteil> <Knoten1> <Knoten2> <Bauteilwert>
- z.B. Widerstand R2 geht von Knoten A (wir haben ein Label vergeben) zur Knoten 0 (Masse)
- Wenn ihr mit der Maus über einen Knoten fahrt, wird euch unten links der Name des Knotens angezeigt.

Die Messbrücke - Arbeitspunktanalyse

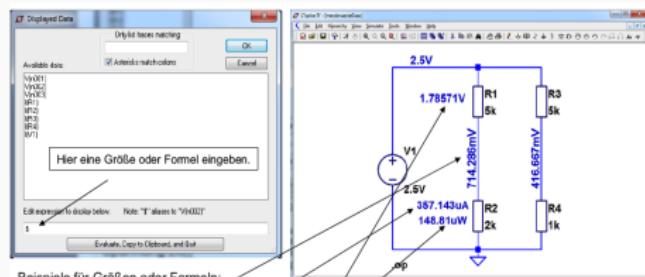
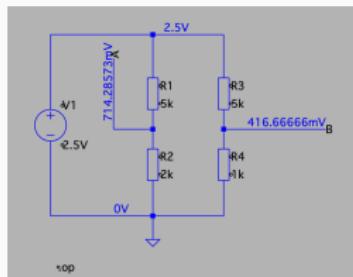
Analyse der Spannungswerte

```
Circuit: * /Users/mholetz/opt/02_projects/07_vor  
Direct Newton iteration for .op point succeeded.  
Operating Bias Point Solution:  
V(n001)      2.5    voltage  
V(a)         0.714286   voltage  
V(b)         0.416667   voltage  
I(R4)        0.000416667 device_current  
I(R3)        0.000416667 device_current  
I(R2)        0.000357143 device_current  
I(R1)        0.000357143 device_current  
I(V1)       -0.00077381 device_current  
  
Date: Fri May  8 21:14:21 2020  
Total elapsed time: 0.016 seconds.
```

Um die Spannungswerte abzulesen habt mehrere Möglichkeiten.

- Ihr könnt euch den log anzeigen lassen - er sollte direkt bei Start der Analyse erscheinen
- Ihr könnt nach der **Simulation** den gewünschten Knoten **zweimal** anklicken
- **V(KNOTENNAME)** beschreibt das Potential an einem Knoten
- **I(BAUTEILNAME)** beschreibt den Strom durch ein Bauteil hindurch
- Per rechtem Mausklick auf einen Wert, öffnet sich ein Dialog über den ihr den Wert frei wählen könnt.

Unter Verwendung der Potentiale und Ströme könnt ihr auch Formeln wie z.B. die Verlustleistung über einem Bauteil anzeigen. Per rechtem Mausklick auf einen Knotenspannungswert kommt ihr in das Display Data Menü.



Beispiele für Größen oder Formeln:

Knotenspannung: $V(n002)$

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

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

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

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

Die Messbrücke - Modellierung eines variablen Widerstandes

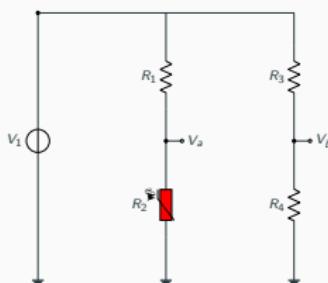
Das Messprinzip

Ist eine Messbrücke abgeglichen, so gilt:

$$\frac{R_1}{R_2} = \frac{R_3}{R_4} \quad (4)$$

$$V_{AB} = 0 \quad (5)$$

Um die Temperatur messen zu können, ersetzen wir den Widerstand R_2 durch einen PTC.
Verändert sich der R_2 ist die Brücke nicht mehr abgeglichen und die Spannung V_{AB} ungleich 0.



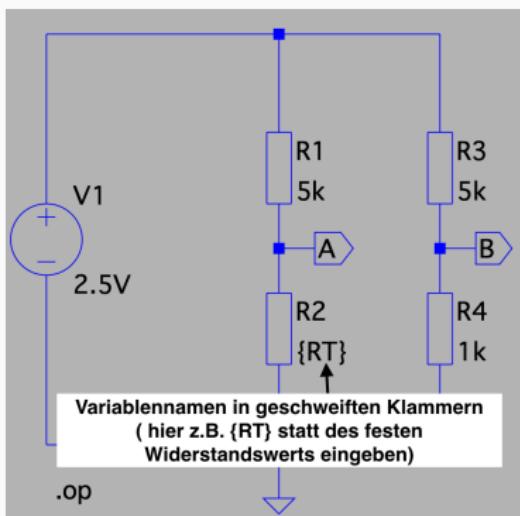
Um dieses Prinzip zu simulieren werden wir nun in LTspice den Widerstand R_2 modellieren und variieren.

Die Messbrücke - Modellierung eines variablen Widerstandes

Ansatz

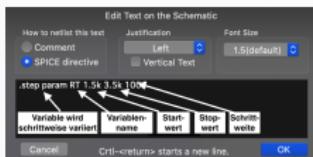
- Der Widerstand soll durch einen PTC ersetzt werden
- Wir nehmen zunächst vereinfacht an, dass sich der Widerstand zwischen 0C - 100C von $1.5\text{k}\Omega$ linear auf $3.5\text{k}\Omega$ erhöht.

Variablen in LTspice



Die Messbrücke - Modellierung eines variablen Widerstandes

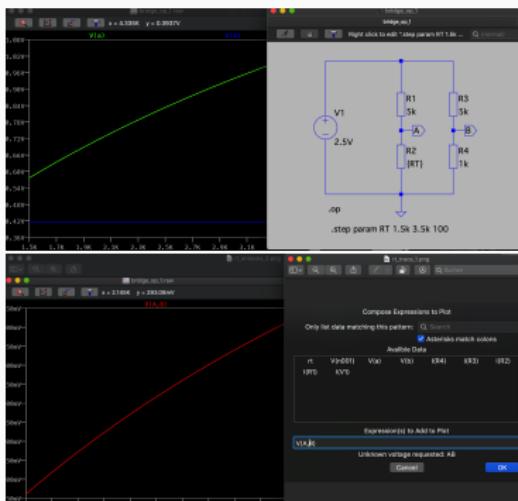
Definition von Variablen



Variablen können auf verschiedene Weisen variiert werden.

- Eine Möglichkeit ist die spice-Direktive `.step param`
- Fügt einfach eine spice-Direktive hinzu
(Edit → Spice Directive (oder drückt S))
- `.step param < varname >< start >< stop >< step >`

Simulieren in Abhängigkeit von RT

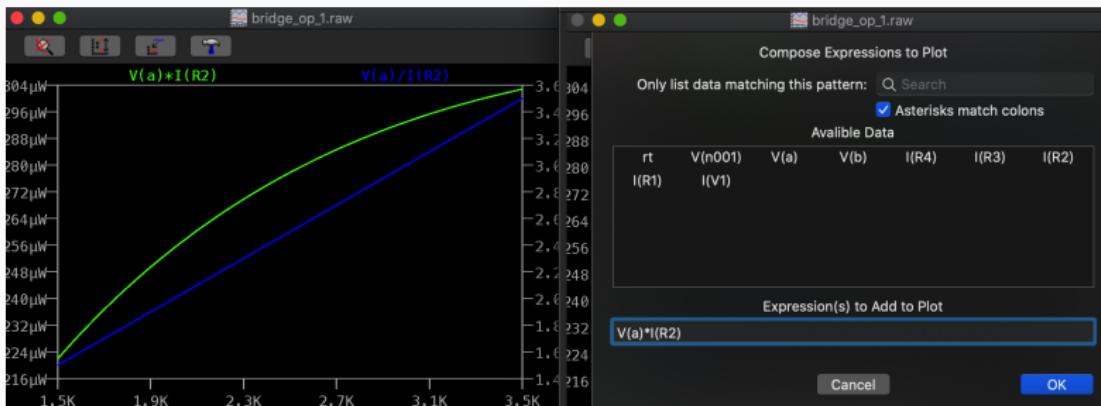


- Es wird die **Spannung** über dem variablen Widerstand auf der x-Achse aufgetragen
- Ihr könnt die Spannung zwischen den A und B anzeigen lassen:
 - ... über den waveform viewer — > add traces — > V(A,B) (Spannung von Knoten zu Knoten)
 - ... im schematic, indem ihr mit der roten Messspitze (red probe) auf ein Potential klickt,
 - die linke Maustaste gedrückt hältet und mit der erscheinenden schwarzen Messspitze (black probe) auf der zweite Potential klickt.
- Der **Strom** kann angezeigt werden, indem ihr mit der Maus über ein Bauteil fahrt und die Strommesszange (ampere probe) anklickt.
- Die **Verlustleistung** kann angezeigt werden, indem ihr mit der Maus über ein Bauteil fahrt, Shift gedrückt hältet und das Temperatursymbol (thermometer probe) anklickt.

Die Messbrücke - Modellierung eines variablen Widerstandes

Formeln visualisieren

Weiterhin könnt ihr im waveform viewer auch beliebige Formeln und somit Kennwerte eurer Schaltung berechnen und visualisieren.



Ihr solltet die folgenden Messwerte der Brückenschaltung ermitteln, bzw. ermittelt haben:

- Brückenspannung V_{AB}
- Strom durch $I(R_2)$
- Verlustleistung $P(R_2) = V_a I(R_2)$
- Verifiziert den Widerstandswert $R_2 = \frac{V(a)}{I(R_2)}$

Check - Was haben wir bisher gelernt

Bis jetzt solltet ihr die folgenden Elemente beherrschen

- Verwendung des Bauteileditors (**F2**)
- DC-Sweep verstehen und verwenden
- Transient verstehen und verwenden
- AC-Sweep verstehen und verwenden
 - Das Kleinsignal Verhalten einer Spannungsquelle im AC-Sweep verstehen und verwenden
 - Die Grenzfrequenz einer Filterschaltung simulativ ermitteln
- Eine Arbeitspunktanalyse durchführen
- Den waveform viewer verwenden
 - traces hinzufügen und anpassen
- Variablen im schematic einführen und variieren
- Verwendung von Labels (**F4**)

Die Messbruecke - realer PTC KTY81-210

Das Bauteil KTY81-210

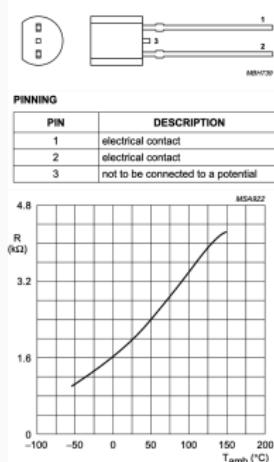
Der KTY81-210 ist ein PTC und wir wollen ihn als Temperaturfühler in unserer Brückenschaltung verwenden. Dazu müssen wir ihn in LTspice modellieren.

Um das entsprechende Verhalten nachzubilden benötigen wir die Spezifikation des Sensors aus seinem Datenblatt.

- Ihr könnt das Hersteller-Datenblatt hier << [Hyperlink zum PDF](#) >> herunterladen
- Ihr könnt die Hersteller-Formelreferenz hier << [Hyperlink zum PDF](#) >> herunterladen

Analyse des Datenblattes

Das Datenblatt enthält eine Tabelle die die Widerstandswerte über die Temperatur beschreibt.



AMBIENT TEMPERATURE		TEMP. COEFF. (%/K)	KTY81-210		
(°C)	(°F)		MIN.	TYP.	MAX.
-55	-67	0.99	951	980	1009
-60	-58	0.98	1000	1030	1059
-40	-40	0.96	1105	1135	1165
-30	-22	0.93	1218	1247	1277
-20	-4	0.91	1338	1367	1396
-10	14	0.88	1467	1495	1523
0	32	0.85	1603	1630	1656
10	50	0.83	1748	1772	1797
20	68	0.80	1901	1922	1944
25	77	0.79	1980	2000	2020
30	86	0.78	2057	2080	2102
40	104	0.75	2217	2245	2272
50	122	0.73	2383	2417	2451
60	140	0.71	2557	2597	2637
70	158	0.69	2737	2785	2832
80	176	0.67	2924	2980	3035
90	194	0.65	3118	3182	3246
100	212	0.63	3316	3392	3466
110	230	0.59	3523	3607	3691
120	248	0.53	3722	3817	3912
125	257	0.49	3815	3915	4016
130	266	0.44	3901	4008	4114
140	284	0.33	4049	4166	4283
150	302	0.20	4153	4280	4407

Die Messbrücke - Approximation eines PTC

Quadratische Approximationsformel

Der Hersteller gibt in seinem Datenblatt eine Approximationsformel zur Modellierung des Verhaltens an.

$$R_T = R_{Ref} [1 + A(T - T_{Ref}) + B(T - T_{Ref})^2 + C(T - T_i^D)] \quad (6)$$

SENSOR TYPE	A (K ⁻¹)	B (K ⁻²)	C ⁽¹⁾ (K ^{-D})	D	T _i (°C)
KTY81-1	7.874 × 10 ⁻³	1.874 × 10 ⁻⁵	3.42 × 10 ⁻⁸	3.7	100
KTY81-2	7.874 × 10 ⁻³	1.874 × 10 ⁻⁵	1.096 × 10 ⁻⁶	3.0	100

TEMPERATURE DEPENDENCY

For the KTY83 series of temperature sensors, the mathematical expression for the sensor resistance 'R_T' as a function of temperature is given by:

$$R_T = R_{ref}[1 + A(T - T_{ref}) + B(T - T_{ref})^2] \quad (1)$$

where:

R_T is resistance at temperature T

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

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

A, B are type-dependent coefficients.

For the KTY81/82/84 series, the slope of the characteristic curve decreases slightly in the upper temperature range above a certain temperature T_i (point of inflection).

Therefore, an additional term in equation (1) becomes necessary:

$$R_T = R_{ref}[1 + A(T - T_{ref}) + B(T - T_{ref})^2 - C(T - T_i^D)]$$

where:

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.

For the types previously mentioned, the type-dependent constants 'A', 'B', 'C' and 'D', as well as 'T_i', are given in Table 3.

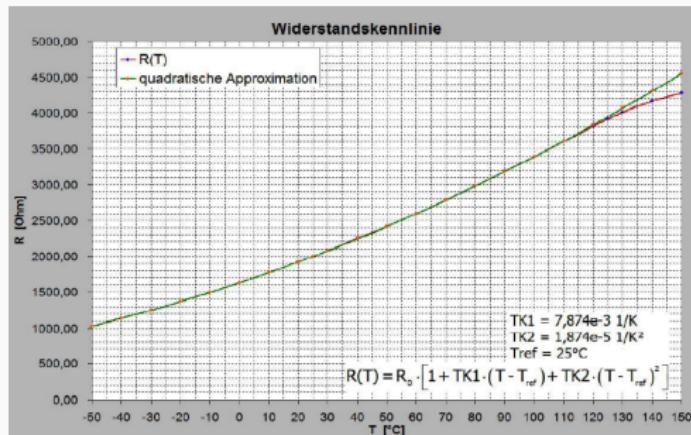
For high-precision applications, e.g. microcontroller-based control systems, the above expressions and the values in Table 3 can be used to generate a calibration table to store in a ROM for look-up and linear interpolation. Data for maximum expected temperature error is supplied separately in the related data sheets. The calculations are based on both specified resistance ratios (R₂₅/R₁₀₀ and R₂₅/R₋₅₅) and the basic resistance spread at 25 °C.

If a microcontroller is not used, the slight deviation from linearity can easily be compensated using a parallel resistor (if a constant current source is used), a series resistor (if a constant voltage source is used) or a suitable combination of both. This is discussed in the Section "Linearization".

Die Messbrücke - Approximation eines PTC

Genauigkeit Approximation

Wenn wir die Tabelle aus dem Datenblatt mit der quadratischen Approximationsformel (Formel 6, $C = 0$) vergleichen ergibt sich folgender Verlauf.



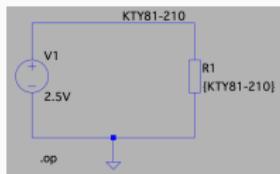
Aus Formel 6 wissen wir, dass die quadratische Approximation ab einer Temperatur $T > T_i$ nicht mehr vollständig ist. Diesen Effekt sehen wir hier. Ab $T > T_i = 100$ weicht der reale Verlauf ab. Wenn wir unseren Messbereich jedoch auf -50C - 100C begrenzen, können wir die quadratische Formel verwenden.

Dies werden wir im nächsten Schritt tun.

Die Messbrücke - Modellierung des KTY81-210

Widerstandsformel als Variable

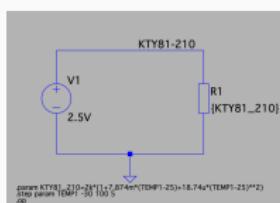
Zur Modellierung starten wir mit einem einfachen schematic.



- Baut den Schaltplan auf
- Gebt dem Widerstand R_1 eine Variable $KTY81 - 210$ als Wert
- Erstellt ein Label mit dem Namen $KTY81 - 210$

Um den Widerstand abhängig von der Temperatur und seiner Formel zu simulieren gehen wir wie folgt vor:

- Schrittweise Steigerung der Temperatur
- Simulation entsprechend Formel 6 mit $R_{Ref} = 2k$ $A = 7.874^{-3}$ $B = 1.874^{-5}$ $T_{Ref} = 25$
- $R_{KTY81-210} = 2k[1 + 7.874^{-3}(T - 25) + 1.874^{-5}(T - 25)^2]$

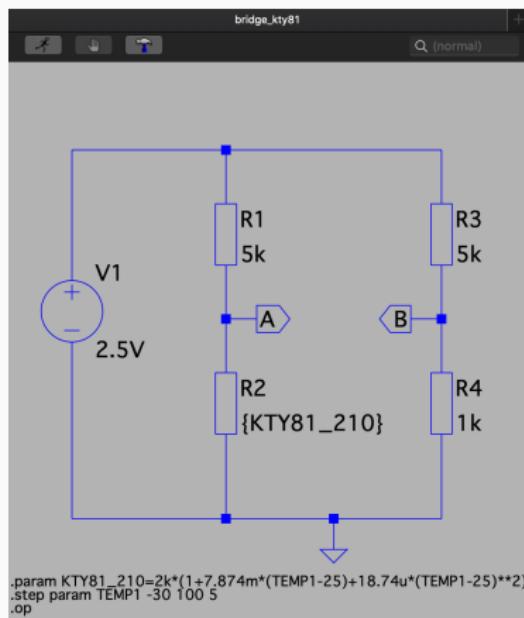


- Ihr könnt für die Variable eine Formel über die Spice-Direktive `.param` hinterlegen
- Über die Spice-Direktive `.step` könnt ihr einen Parameter $TEMP1$ (nicht $TEMP$ dieser beschreibt die globale Temperatur) für die Temperatur einführen und variieren.
- In einer Arbeitspunktanalyse können wir uns nun den trace von $R_1 = \frac{V(KTY81-210)}{I(R_1)}$ anzeigen lassen.

Entspricht der Verlauf dem des Datenblattes und euren Erwartungen?
Wenn ja, dann können wir diesen Widerstand nun in unsere Messbrücke übernehmen.

Die Messbrücke - mit KTY81

Schaltungsaufbau



- Ihr baut den KTY81-210 inklusive der als spice-Direktive modellierten Formel in eure Messbreuckenschaltung ein
- Prüft ob das Model immer noch simulierbar ist und es zu keinem Fehler kommt.

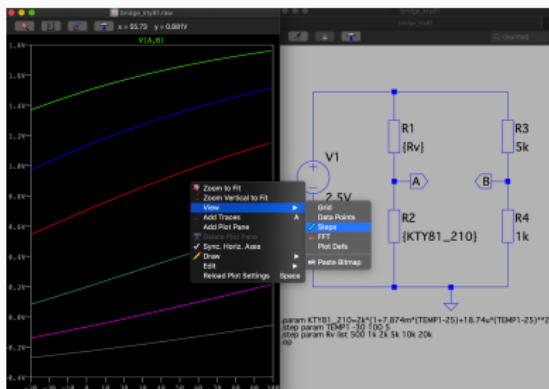
Die Messbrücke - Linearisierung der Kennlinie

Problemstellung

- Die Formel zur Approximation des PTC ist quadratisch
- Dadurch sind wir unsere Messung "nicht-linear" und wir werden bei einer **konstanten Temperaturerhöhung keine konstante Spannungserhöhung** erwarten können

Simulatives Experiment

Zur Annäherung werden wir die Vorwiderstand R_1 ebenfalls variieren, um experimentell einen linearen Verlauf der Spannung V_{AB} zu ermitteln.



- Variiert den Vorwiderstand R_1 als Parameter R_V
- Über die Spice-Direktive `.step param < Var > list < Wert1 > ... < Wertn >` könnt ihr eine Liste von Werten hinterlegen
- Wir wählen einen Wertebereich von $500\Omega - 20k\Omega$
- Positioniert die Spice-Direktive `.stepparamRv` nach der Temperatur $TEMP1$, da der erste Wert auf der Abszisse dargestellt wird
- Im waveform viewer könnt ihr über *rechter Mausklick* –> *View* –> *Steps* die einzelnen Werte auswählen und eigenständig bewerten.

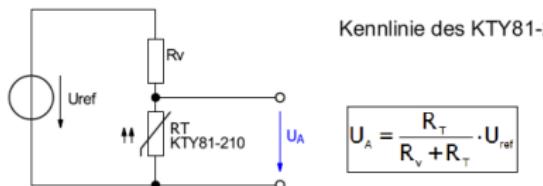
Hinweis zur Simulationsschrittweite

Wählt die Schrittweite immer möglichst klein, da zu viele Werte die Übersichtlichkeit erschweren!

Die Messbrücke - Linearisierung der Kennlinie

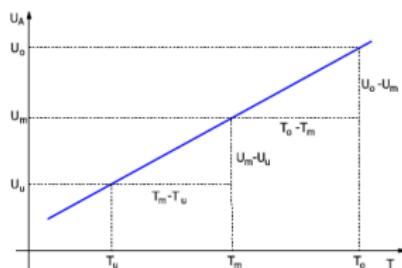
Analytische Herleitung

Den exakten Wert kann man auch analytisch Herleiten. Dazu schauen wir uns den Spannungsteiler des PTCs genauer an.



Kennlinie des KTY81-210:

T	R _T	
-20°C	1367 Ω	= R _u
0°C	1630 Ω	
25°C	2000 Ω	
40°C	2245 Ω	= R _m
100°C	3392 Ω	= R _o



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

→ Linearitätsbedingung: $U_o - U_m = U_m - U_u$
(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:

$$R_v = \frac{R_m(R_o + R_u) - 2R_oR_u}{R_o + R_u - 2R_m} = \frac{2245 \Omega(3392 \Omega + 1367 \Omega) - 23392 \Omega \cdot 1367 \Omega}{3392 \Omega + 1367 \Omega - 22245 \Omega} = 5,24 \text{ k}\Omega \quad \rightarrow 5,23 \text{ k}\Omega \text{ (E96)}$$

Die Messbrücke - Abgleich der Messbrücke

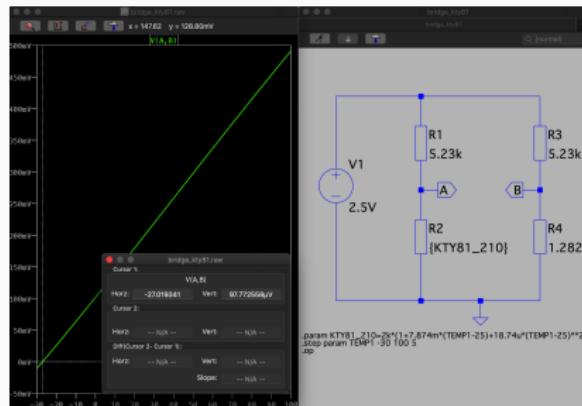
Definition des Messbereichs

- Unsere Temperaturmessbrücke mit einem 8Bit AD-Wandler digitalisiert werden
- 254 Schritte
- quadratische Approximation bis 100C genau
- wir wählen eine Schrittweite von 0,5C

Somit ergibt sich ein Temperaturbereich von -27C bis $100,5\text{C}$ und die folgende Anforderung an unser Schaltung.

Die Messbrücke soll bei -27C abgeglichen sein.

- Der KTY81-210 hat laut Datenblatt einen bei -27C Widerstand von 1282Ω
- R_4 muss daher 1282Ω betragen
- In der Realität würde man einen $1k\Omega$ und einen auf 282Ω getrimmten 500Ω -Trimmer verwenden
- In unserer Simulation reicht ein einfacher Widerstand mit 1282Ω aus.



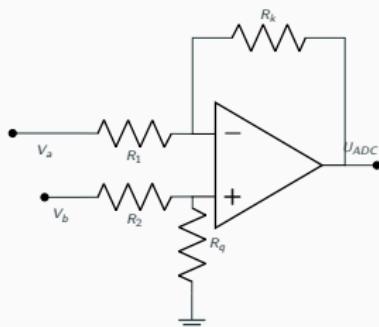
Die Messbrücke - Anpassung an AD-Wandler

Ausnutzung des Quantisierungsbereiches des AD-Wandlers

- Der AD-Wandler hat eine Referenzspannung von 2,5V
- In der vorherigen Folie sehen wir, dass die Spannung der Messbrücke von 0 bis etwa 500mV reicht
- Zur Ausnutzung des kompletten Quantisierungsbereiches des AD-Wandlers muss die Brückenspannung auf 2,5V bei 100,5C verstärkt werden

Hierzu werden wir einen Differenzverstärker verwenden

Exkurs Differenzverstärker



Wenn gilt:

$$V_D = \frac{R_k}{R_1} = \frac{R_q}{R_2} \quad (7)$$

Dann ist der Differenzverstärker symmetrisch und kann mit der folgenden Formel dimensioniert werden.

$$U_{ADC} = V_D(V_b - V_a) \quad (8)$$

Um die folgenden Berechnungen zu vereinfachen verwenden wir einen symmetrischen Differenzverstärker.

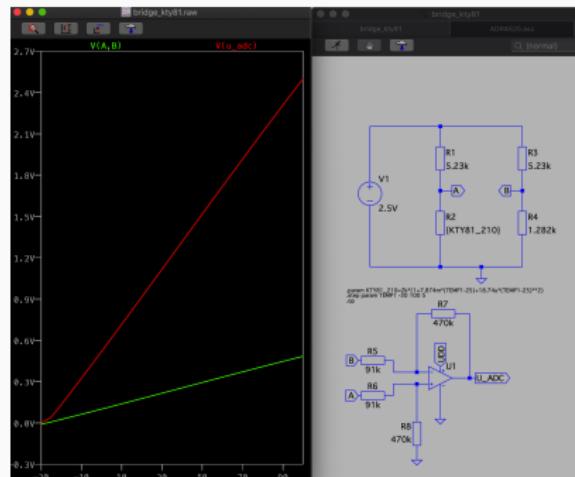
Die Messbrücke - Anpassung an AD-Wandler

Dimensionierung Differenzverstärker

- Aus unserer Simulation (Folie 45) wissen wir, dass die Spannung V_{AB} bei 100C 493mV entspricht
- Die Referenzspannung des AD-Wandler ist 2,5V
- Das ergibt einen Verstärkungsfaktor $V_D = \frac{2,5}{0,493} = \sim 5,1$
- Wir wählen daher $R_k = R_q = 480\text{k}\Omega$ sowie $R_1 = R_2 = 91\text{k}\Omega$

Aufbau des Differenzverstärkers und Integration

Zur einfacheren Versorgung (eine 5V Versorgung für alles anstatt einer zusätzlichen +/-12V Versorgung) verwenden wir den UniversalOpamp2 im unsymmetrischen Modus von 5V bis 0V. Dies reicht aus, da unser Messbereich von 0-2,5V reicht.

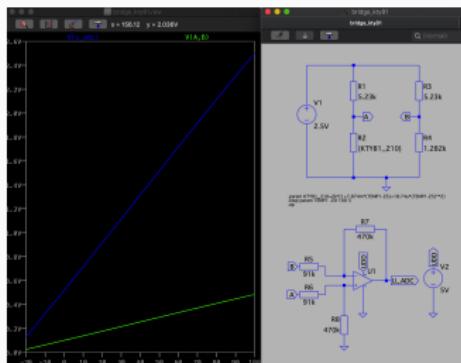


Die Messbrücke - Alternative Anzeige am Digitalmultimeter

Potentialanpassung für das Multimeter

Alternativ kann die Spannung direkt an einem Digitalmultimeter angezeigt werden.

Das Digitalmultimeter soll jedoch idealerweise direkt die Temperatur ohne Umrechnungsfaktor anzeigen. Wir haben einen nahezu idealen unsymmetrischen OPVs verwendet. Hier sehen wir, dass (wie auch bei einem realen TS912-5) die Kurve am Beginn des Messbereichs leicht abflacht. Daher limitieren wir den Messbereich bei -20C.



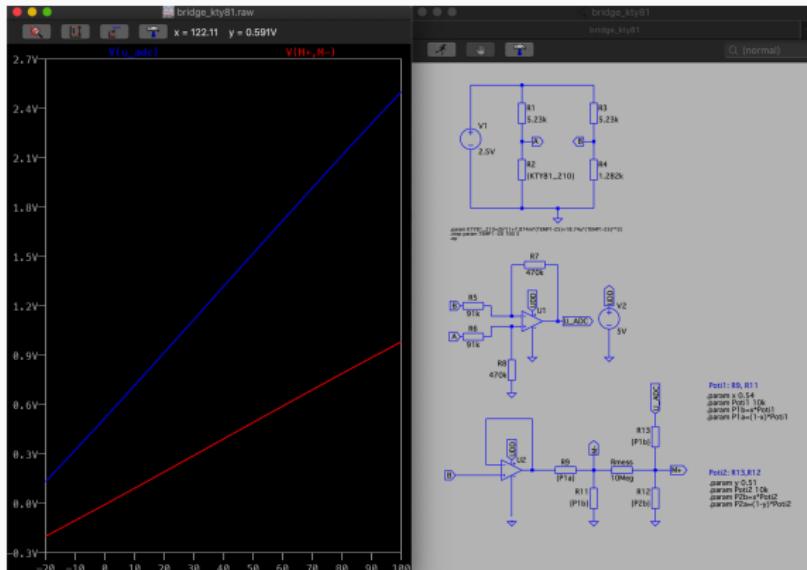
Um die Anpassung durchzuführen werden wir die folgenden Schritte abarbeiten um die Spannung der Temperatur anzugelichen:

- Spannungsteiler reduziert das Potential U_{ADC}
- Wir nutzen die temperaturkonstante Spannung V_B mit einem weiteren Spannungsteiler um das Potential am negativen Messpunkt um 270mV zu senken
- Für die Spannungsteiler müssen Potis zur initialen Einstellung verwendet werden, da es keine Widerstände in beliebigen Werten gibt. Jedenfalls nicht um unsere Spannungsteiler exakt zu dimensionieren.

Die Messbrücke - Alternative Anzeige am Digitalmultimeter

Schematic mit Modellierung von Potis

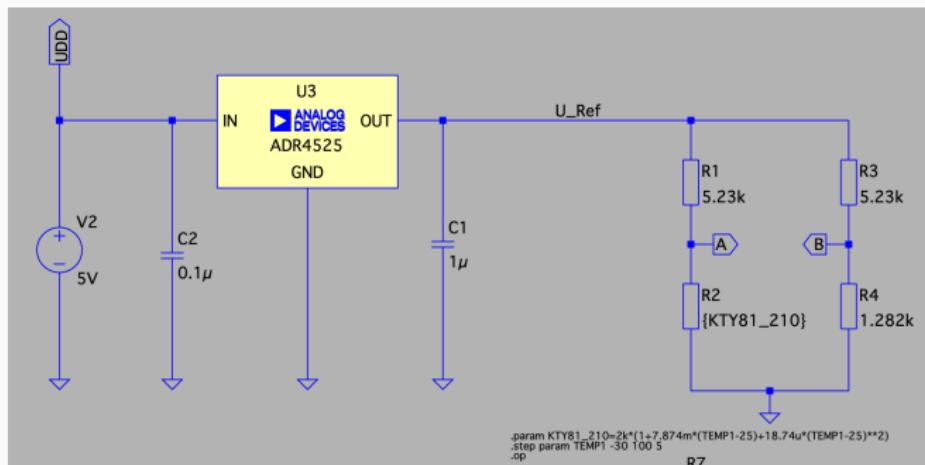
- Spannungsteiler $U_{UAC} \frac{1,27}{2,5} = \sim 0,51$
 - $U_B = 2,5 \left(\frac{5,23k\Omega}{1,28k\Omega + 5,23k\Omega} \right) = 0,492V = \text{konstant}$
 - Spannungsteiler $U_B = \frac{0,27}{0,492} = \sim 0,54$
 - R_{Mess} sollte möglichst hochohmig dimensioniert werden.



Die Messbrücke - Die Referenzspannungsquelle

Verwendung eines realen Bauelementes

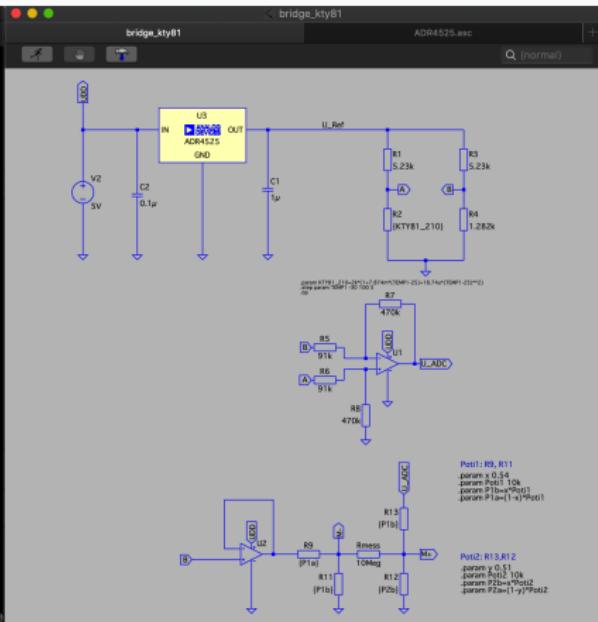
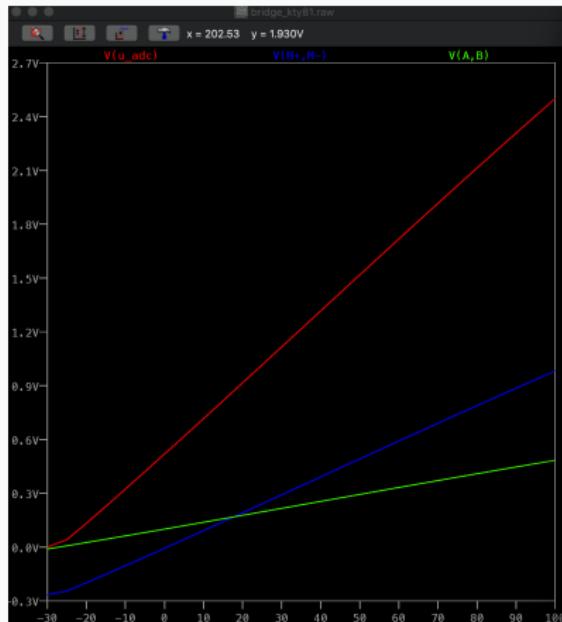
Beim praktischen Schaltungsaufbau lässt sich die Referenzspannung mit einer Referenzdiode (z.B. direkt von analog devices ADR4525) aus der Versorgungsspannung erzeugen. Wir haben hier bisher eine ideale Spannungsquelle in der Simulation verwendet!



Im Vergleich zur Übersicht auf Folie 28, ist der LM285-2.5 nicht mehr in der Bauteilbibliothek verfügbar. Daher haben wir hier zur realen Modellierung der Spannungsversorgung ein Bauteil von analog devices verwendet.

Die Messbrücke - Alles zusammen

-0,2 bis 1V bei Abgriff von M+ nach M-



Appendix

Reale Bauteile modellieren mit einem subcircuit

Hier findet ihr eine Übersicht wie man ein reales Bauteil im Internet findet, den subcircuit einbindet und ein Modellfile dazu erzeugt. Diese Dateien sind Übernahmen aus dem alten Skript CAE Labor von Franz Smagacz-Allramseder.

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 hauseligen Typen mitgeliefert werden, wollen wir uns mal Schritt für Schritt anschauen, 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:

<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
```

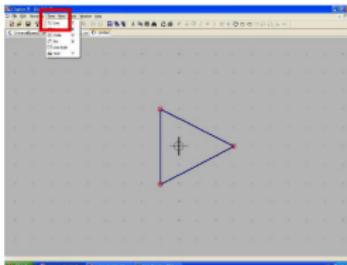
```
C2  6 7 15.00E-12
DC  5 53 DX
DE  54 5 DX
DLP 90 91 DX
DLN 92 90 DX
DP  4 3 DX
EGND 99 0 POLY(2)(3,0)(4,0) 0 .5 .5
FB  7 99 0 VBV VC VE VLP VLN 0 4.715E6 -5E6 5E6 5E6 -5E6
G  0 11 12 252 0 -6
GCM 0 6 10 99 8.942E-9
ISS 3 10 DC 195.0E-6
HUM 90 0 VLIM 1K
J1 11 2 10 JX
J2 12 1 10 JX
R2 6 9 100.0E3
RD1 4 11 3.536E3
RD2 4 12 3.536E3
RP1 7 8 5 150
RP2 7 9 5 150
RP 3 4 2.143E3
RSS 10 99.1 0.026E6
VB 9 0 DC 0
VC 3 53 DC 2.200
VE 54 4 DC 2.200
VLIM 7 8 DC 0
VLP 9 0 DC 25
VLN 0 92 DC 25
MODEL DX D(I=800.0E-18)
MODEL JX PJF(I=15.00E-12 BETA=270.1E-6 VTO=-1)
.ENDS
```

3. Schritt:

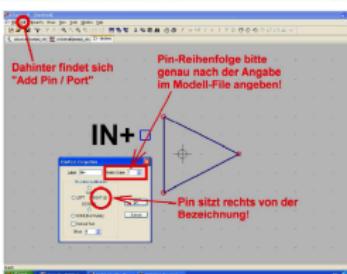
LTSpice wird gestartet, aber unter „File“ die Option „New Symbol“ angeklickt. Dadurch öffnet sich ein neues Blatt in einem etwas helleren Grau, um es von einem Schaltbild unterscheiden zu können. Über dem Bildschirm liegt ein feines Punktraster und in der Mitte ist das Zentrum für das Symbol durch ein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

Siehe nächste Seite!

Reale Bauteile modellieren mit einem subcircuit



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:

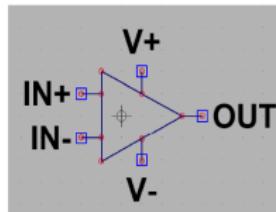
Erstens muss die richtige Pin-Nummer gemäß den Angaben im Modell-File („TL072-auf“) eingetragen werden.

Da gilt für den TL072 folgende Zuordnung:

Pin 1 = Nicht invertierender Eingang = IN+
Pin 2 = 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.

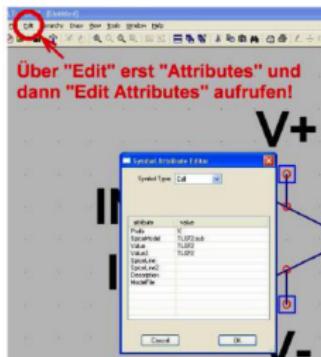
Zweitens muss man angeben, wo sich der Pin räumlich gegenüber der Bezeichnung befinden soll (hier: rechts).



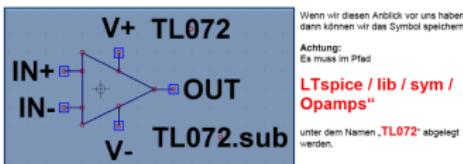
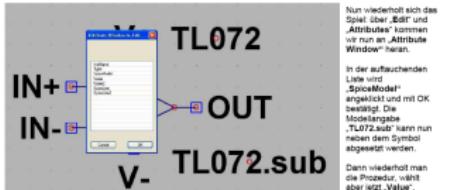
Sind alle Pins platziert, dann sorgt man im Anschluss wieder mit „Draw Lines“ für Verbindungsleitungen von jedem Pin zum Schaltzeichen.

Das ist das Ziel...

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



Reale Bauteile modellieren mit einem subcircuit



ENDE

Mit Erlaubnis verwendete Datenquellen:

Reinhold Birk, Gottlieb-Daimler-Schule 2, Sindelfingen
Jürgen Richter, Technische Schule Aalen

Digitale Schaltungssimulation

Übung: 3: Digitale Schaltungssimulation

Bemerkung: Leider wird dieses Thema sowohl von der Online-Hilfe wie auch von den mitgelieferten Beispielen hier etwas stiefmütterlich behandelt – man muss also etwas Wissen und Erfahrung mitbringen und dann alle möglichen Mosaiksteine zusammenfragen, 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:

- 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.
- Meist ist nicht nur der zugehörige logische Ausgang, sondern zusätzlich auch die Invertierung vorhanden.
- 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

- 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.
- Man findet keine Betriebsspannungsanschlüsse, da mit „idealen Bausteinen“ simuliert wird.

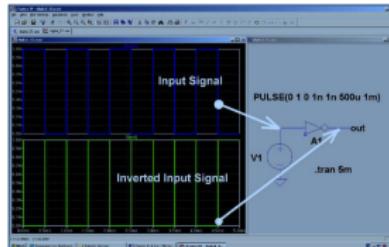
Als Eingangssignal verwendet man entweder das „PULSE“- oder das PWL-Signal. Minimaler Spannungswert ist Null Volt, maximaler Spannungswert = $+1V$ olt (...andere Werte: siehe oben...)

Als Anstiegs- und Abfallzeit reicht eine Zeit von 1 Nanosekunde.
Bei der Darstellung der Simulationsergebnisse sollte man sofort – bevor überhaupt etwas im Ergebnisdigramm zu sehen ist! – mit der rechten Maustaste draufklicken und dann **Add Plot Plane** 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 Mikroseunden / Periodendauer = 1ms). Simuliert wird von 0...5ms.
Am oberen Eingang wird angeleert, der andere wird offen gelassen. Der Ausgang erhält den Label „out“. Wie erahnt, 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:

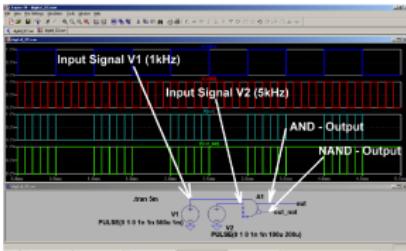


Aufgabe: AND-Baustein

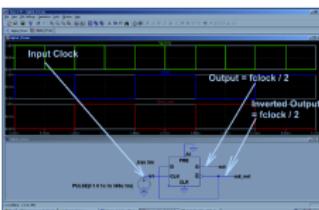
Wir wollen eine UND-Verknüpfung mit 2 Eingängen realisieren. Am einen Eingang liegt ein Digitalsignal mit $f = 1kHz$, am anderen dagegen eines mit $f = 5kHz$. Simuliert wird von 0...5ms.
Wie zu Beginn des Kapitels erwähnt, müssen wir dazu die drei zusätzlich vorhandenen, aber nicht benötigten Eingänge offen lassen. Den vorhandenen beiden Ausgängen wurden wieder **Label zugewiesen**.

So sollte die Simulation aussehen:

Reale Bauteile modellieren aus einem model file



Aufgabe: D-Flipflop



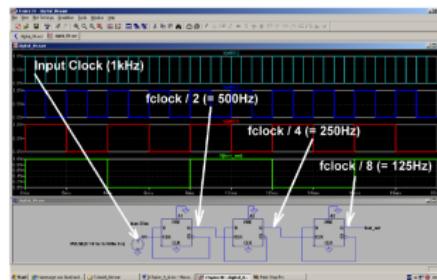
Die Schaltung stellt den bekannten Binär-Frequenzzeller dar, bei dem am Ausgang die halbe Eingangs frequenz beobachtet werden kann. Dazu wird der invertierte Q Ausgang mit dem D-Eingang verbunden. Die Input Clock Frequenz beträgt 1 kHz. Die Signaleingänge „Preset“ und „Clear“ dürfen nicht aktiv werden und liegen deshalb an Masse.

Bei jeder positiven Flanke des Clocksignals wechseln die Ausgänge ihre Zustände.

Aufgabe: Dreistufiger Frequenzzeller 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://www.ti.com/lit/ug/tst027/tst027.pdf>



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