

.TEXT コマンドで文字列を定義する (LTspice 新コマンド)

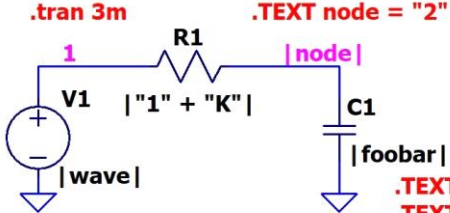
任意の文字列を変数に置き換えることができる。
本ドキュメントでは、このコマンドの基本文法の
解説と、信号源の定義する文字列を変数として
取り扱う例を示す。

.TEXT の文法

`.TEXT <変数名>=<"文字列"> [<変数名>=<"文字列"> ...]
定義される文字列は「”」(ダブルクォーテーション)でくくる`

変数名を回路図 (ネットリスト) 中で使うときは「`|`」(縦線...プログラミングの経験者では「パイプ記号」で通じるでしょう) で始めと終わりをくくる。

使用例



`.tran 3m`
`.TEXT node = "2"`

`|\"1\" + \"K\"|`
`|node|`
`|wave|`
`|foobar|`

`.TEXT foo="0.1" bar="u"`
`.TEXT foobar = foo + bar`

`;` `.TEXT wave="PWL(0 0 1m 1 1.5m 0.8 2m 0.2 2.2m 0.5 2.6m 1 2.8m 0.1 3m 0)"`
`.TEXT wave="PULSE(0 1 0 1u 1u 0.5m 1m)"`
`;` `.TEXT wave="SINE(0 1 1k)"`

<.TEXT コマンドの評価>
TEXT で定義した `|node|` のノード名(実質的な名称は「2」)だが、`|node|` とラベルを付けたノードの電圧は、マウス・カーソルを利用した「電圧プローブ」を使って直接プローブすることはできない。「Add Trace」を利用し、`V(2)` を選択し、プロットする。
`|\"1\" + \"K\"|` は何らかの「.TEXT=...」の行がスパス・ディレクティブとして少なくとも1つが有効であれば、「1K」と等価になる。

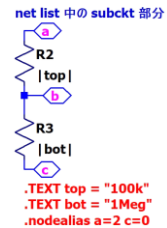
この例では、信号源「V1」の波形の切り換えに、複数の信号形式(PULSE, SINE, PWLなど)を文字列としてあらかじめ用意し、その文字列を変数「wave」で定義しておくことで、不要な行をコメントにすれば簡単に信号のパターンを切り換えることができる。

Helpファイルの例題のネット・リスト

前頁の回路図は、このネットリストをもとにキャプチャーしたものだ。
SUBCKT部分は別途用意した（実際の回路図ではフラットにしてある）。

V1 1 0 PULSE(0 1 0 1u 1u .5m 1m)	← V1 は1と0につながっており、その値はPULSEで定義されている	←
R1 node 1 "1" + "K"	← R1 は変数node(実は2)と1につながっており、その値は 1K	
C1 node 0 foobar	← C1 は変数node(実は2)と0につながっており、その値は 変数foobar(実は0.1u)	
X1 2 div 0 NIX	← サブ・サーキットX1は divと0につながっており、サブ・サーキット名はNIXである	
.text node="2"	← node という変数名に 文字「2」を定義する	
.text foo=".1" bar="u"	← foo という変数名に文字「0.1」、bar という変数名に文字「u」を定義する	
.text foobar = foo + bar	← foobar という変数名を foo(実は0.1)とbar(実はu)をつないだ文字列(0.1u)と定義する	
.tran 3m	← シミュレーション実行コマンド・・・トランジェント解析を3m秒まで行う	
.text top="100k"	← top という変数名に 文字「100k」を定義する	
.text bot="100k"	← bot という変数名に 文字「100k」を定義する	
.subckt nix a b c	← サブ・サーキットの宣言: 名称は「nix」でa,b,cの3端子	
.text bot="1Meg"	← bot という変数名に 文字「1Meg」を定義する(サブ・サーキットでの再定義)	
R1 a b top ; uses global scope definition of top	← R1 はaとbにつながっており値は「top」で定義される	
R2 b c bot ; uses locally scope definition of bot	← R2 はbとcにつながっており値は「bot」で定義される	
.ends nix	← サブ・サーキット「nix」の終了宣言	
.end	← ネット・リスト全体の終了宣言	

「bot」の値はメインの回路中で定義されているが、それをサブ・サーキット内でover-ride(無効化)している



3

Helpファイルの関連情報

- 変数名が、foo、bar・・・というあまり聞きなれない文字列が例題に登場するが、最近のプログラミングにおける仮の（とりあえず・・・といった場合の）変数に対し、1番目が「foo」、2番目が「bar」、3番目が「baz」・・・という名称がよく利用されている。ここでは、その慣習に従ったものと考えられる。
- SUBCKT に「nix」という名称があたえられているが、これは直訳すると「無（皆無）」という意味で、サブルーチン名を仮につける場合に使われる慣習がある。
 - 宗教的というよりは、哲学的・精神的な願望・期待・・・のような心理と関連しているのかもしれない。

さて、最大の関心事は・・・このコマンドはどのような場面でその真価を発揮するのか？
回路要素の定数(抵抗値・容量など)をシミュレーション中に変化させるときは「.STEP」を使うほうが理にかなっている(文字列を「.STEP」の変数の値としては使えない)。

トランジスタのnative SPICE Model(特性)をそっくり変化させてシミュレーションするには、「AKO」が利用できる。

そこで思いあたるのが・・・シミュレーション条件を決める要素の長い文字列を短い変数名に置き換え、いくつかを切り替えて使うような、今回の例として示した電源設定に応用することが、ひとつのアイデアである。

4