



# ASCII出力結果

出力ファイル名は<シミュレーション回路名.txt>のように、もとの「\*.asc」ファイル名の拡張子を「txt」にしたものになる。

1行目はデータのインデックス。  
この場合、出力の第1カラムは「time」、続いてV(out)、I(L1)の順になっている。  
次の行からは、それぞれの時間におけるV(out)・I(L1)が改行しながら、並んでいく。

結果の有効数字に対して .options plotwinsize=0 は影響を与えないが、 .options numdgt=15 を使うことで、倍精度計算の影響がある。

time	V(out)	I(L1)	
0.0000000000000000e+000	0.000000e+000	0.000000e+000	0.000000e+000
9.765624940649698e-012		2.167075e-021	3.905735e-015
1.953124988129940e-011		5.779764e-021	7.813008e-015
3.427805714679055e-009		2.060089e-016	1.457102e-012
6.836080179476811e-009		5.808017e-016	2.800883e-012
1.024435464427457e-008		1.124384e-015	4.039155e-012
1.365262910907232e-008		1.836756e-015	5.171918e-012
...		...	...
9.999198228278466e-004		2.524333e+000	9.464988e-001
9.999952485598694e-004		2.523963e+000	8.565262e-001
1.0000000000000000e-003		2.523931e+000	8.508697e-001

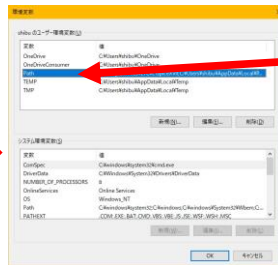
time	V(out)	I(L1)	
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
9.765624940649698e-012		2.167075139958051e-021	3.905735086196636e-015
1.953124988129940e-011		5.779763426864272e-021	7.813007813265502e-015
3.906249976259879e-011		1.907692961314187e-020	1.562929821010931e-014

## バッチ処理でシミュレーション結果をTEXTに・・・

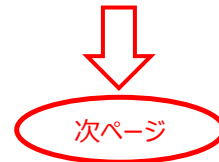
LTspice を Windows のコマンド・プロンプトから利用するには、Windows の「環境変数」に「PATH」を追加し、回路図がどのワーキング・ディレクトリ（フォルダ）にあっても、LTspice を起動できるようにしておく。

Windowsの変数設定に精通していない方は、十分に注意していただいた上で、<自己責任>にて操作していただくようお願いいたします。

Windowsの「スタート」ボタンの右にある、「検索窓」に「環境変数」と入力し Enter すると・・・

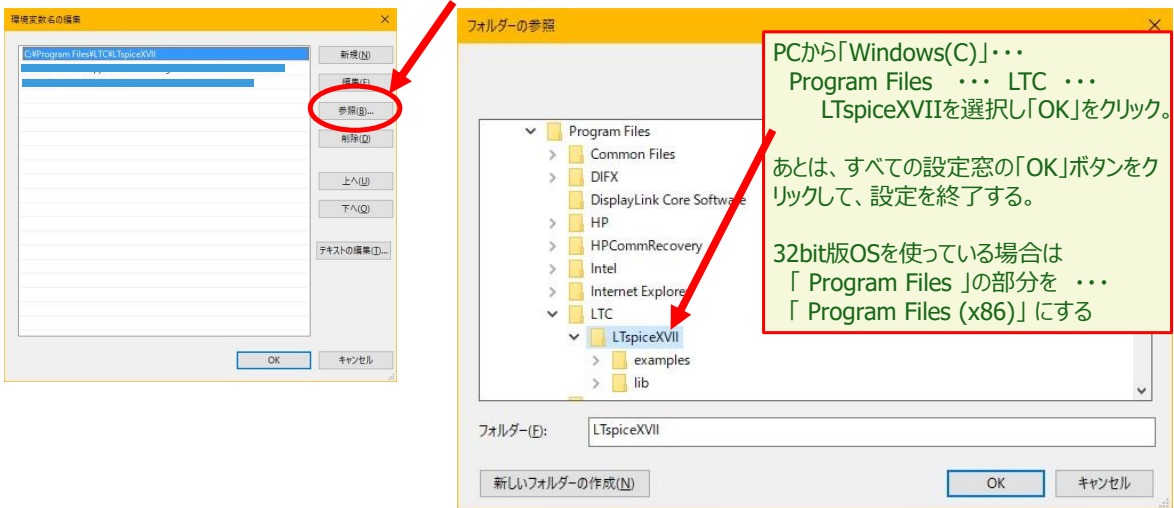


左側の「Path」の項目をアクティブにして・・・「編集」ボタンをクリック



## 環境変数の編集

右側に並んだボタンの中から「参照」をクリック・・・

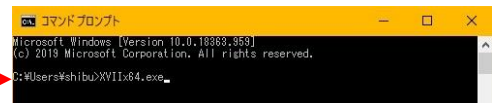


## Pathが通っているかを確認する

Windows の「コマンド・プロンプト」を起動する。

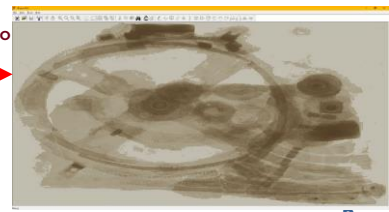
PCの設定によっては「アクセサリ」のフォルダが見えなかったり、ショートカットが表示されていない場合がある。そのときは、Windows の右下の検索窓で「コマンドプロンプト」と入力してEnter を押す。

「 XVIIx64.exe 」と入力しEnter を押す。



これで、LTSpiceが起動すれば、Pathは通っている。

起動時の背景画像は設定によってこの図とは異なる場合がある



## バッチ・ファイル(\*.bat) を作る

テキスト・エディタ（Windows付属のメモ帳で可）を開き、以下のテキストを入力し、保存（ファイル名の拡張子は「\*.bat」にする・・・「\*.bat.txt」にならないように注意）。

```
XVIIx64.exe -netlist TEST_3561.asc ← 回路図ファイル(*.asc) をネットリスト (*.net) に変換する  
XVIIx64.exe -b -ascii TEST_3561.net ← ネットリスト(*.net) をシミュレーションし、  
結果をASCIIファイル (*.raw)で出力する。
```

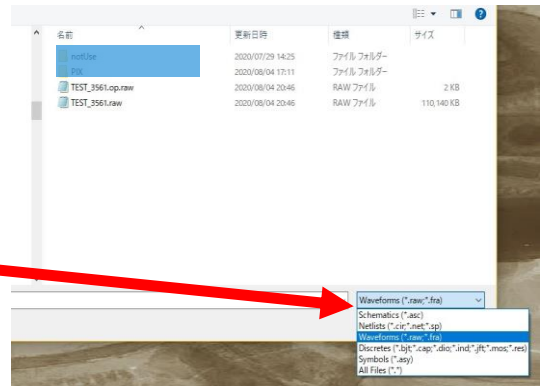
この例題では、このファイル名を「TEST\_3561.bat」として「TEST\_3561.asc」を保存した同じフォルダーの中に保存する。

ほかの回路も続けて実行する場合には、同一のフォルダーの中に「\*.asc」の形式の回路図ファイルを用意するか、あらかじめネット・リストになっているものを保存しておく（ネットリストになっていれば、上記の1行目は不要である）。

**Control Panel の Operation タブにある「 Automatically delete .raw files 」のチェックを外しておくことを強く勧めます。**

## 出力ファイルの確認

シミュレーション結果をグラフで確認する場合には、LTspiceを起動し、File → Open でシミュレーションを実行したフォルダーの中で、「\*.raw」を開く。このとき、ファイル拡張子の選択窓が「Waveforms」か「All Files」になっていないと、目的のファイル名が表示されない。



あるいは、デスクトップ上のLTspiceのショートカットキーの上に、「\*.raw」ファイルをドラッグ&ドロップしてもよい。同じフォルダーにシミュレーションと同じ名前のプロット・ファイル (\*.plt) があれば、その信号をグラフ表示する（通常のシミュレーション結果の表示の場合よりも、ASCII出力の場合は、表示するまでに時間がかかる）。pltファイルがなければ、グラフペインの上で、右クリックし、Add Traces から目的の信号を選択する。

## ASCII 出力の確認 (1)

```
Title: * G:YTOP_20200618Y_6_M_LTspice-TIPS_MonthlyLTspice-TIPS_2020-08YASCII-
OutputYTEST_3561.asc
Date: Wed Jul 29 14:47:57 2020
Plotname: Transient Analysis
Flags: real forward
No. Variables: 25
No. Points: 176741
Offset: 0.0000000000000000e+000
Command: Linear Technology Corporation LTSpice XVII
Backannotation: u1 1 2 3 4 5 6 7 8 9 10
Variables:
```

```
0          time          time
1          V(n004)        voltage
2          V(in)         voltage
3          V(n002)        voltage
4          V(n003)        voltage
5          V(n005)        voltage
6          V(n001)        voltage
7          V(out)         voltage
8          I(C2)          device_current
9          I(C1)          device_current
10         I(L1)          device_current
11         I(Rload)       device_current

.
.

22         Ix(u1:7)       subckt_current
23         Ix(u1:9)       subckt_current
24         Ix(u1:10)      subckt_current
```

シミュレーションのヘッダー部分。

信号名の数。

シミュレーションの全ポイント数 (ポイント番号が0から始まるので最後のポイント番号は176740である)。

Offset はシミュレーションコマンドの中の「Time to start saving data」に設定された時間が入る。設定がなければ「0」。

信号名の一覧表。

シミュレーション結果は、データ・ポイントごとに、そのポイントの時間と各信号の値を、この信号名の順序で改行しながら出力する。

## ASCII 出力の確認 (2)

176740

```
1.0000000000000000e-003
8.087365730002839e-001
4.0000000000000000e+000
9.142528801755030e-001
-9.359566962407984e-002
8.002115740325824e-001
9.149777235248591e-001
2.523930576758237e+000
-1.587044418774894e-001
-5.575718070582880e-008
8.508697321402495e-001
1.009572230703295e+000
1.942261102020831e-006
1.943313419081910e-006
-5.575718071969805e-008
2.496100533951494e-006
-6.450429182297657e-005
-2.496100533951541e-006
-6.205680948945618e-005
-8.508697321402499e-001
8.508697239489608e-001
8.191289028513453e-009
6.449610053395168e-005
1.052317061072360e-009
5.575718071971129e-008
```

シミュレーション結果の最後のデータポイント。

176740番目。時間は $1.00 \cdot 10^{-3}$ 、すなわち1m秒を示している。以下、25個の信号の値が並んでいる。

例えば、V(out)は、信号番号7番・・・すなわち上から8番目なので・・・  
2.5239e+0 ... 約2.52Vであることが読み取れる。

このようにバッチ処理をすると、すべてのノードの電圧と、素子 (または端子) に流れる電流のすべてをASCII (テキスト) 形式で出力する。シミュレーション結果を、他のプログラムで利用する場合には、必要なものを抽出する手間が必要になるが、データの取りそこないがないので、活用の自由度が広がるともいえる。

シミュレーション後、手動で必要な信号だけをASCIIファイルにするか、バッチ処理をするかは、ユーザーの使用目的で選択することになる。