テキスト形式収録データファイルを PcWaveForm 形式ファイルに変換する

テキスト形式で格納されている収録ファイルを読み出し.PcWaveForm で読み出し可能な hdr/.dat 形式のファイルに変換する例について説明します。

例では、テキスト形式ファイルは一般的な項目区切り半角カンマの拡張子 csv 形式としています。
記述形式は、1 行目=信号名行、2 行目=単位行、2 行目以降=データ行
尚、記述規則は下記の規則であることを前提としています。
※　1 列目は時刻列、２列目以降はチャネルに対応したデータ列が記録されている。
※　データはチャネル毎に 1 列を占有し、行方向(縦方向)に時刻歴毎に記録されている。
※　チャネル列は特に制限しないが、最低 1ch は記録されている。
※　時刻歴は等分刻みでサンプリングされている。

テキストファイルに記載されている内容サンプルを示します。

	1 列目	2 列目	3 列目	4 列目	5 列目	6 列目	7 列目
1 行目	時刻歴	油圧１	油圧２	油圧３	油音	トルク	角度
2 行目	sec	k Pa	MPa	MPa	degC	Nm	deg
3 行目	0	0.098	17.88	16.23	68	30	0.9
4 行目	0.01	0.098	17.88	18.01	68	30	1,2
5 行目	0.02	0.098	17.88	16.23	68	30	0.9
6 行目	0.03	0.098	17.88	16.23	68	30	0.9
7 行目	0.04	0.098	17.88	16.23	68	30	0.9
8 行目	0.04	0.099	17.88	16.23	68	30	0.9
9 行目	0.05	0.099	17.88	16.23	68	30	0.9
10 行目	0.06	0.099	17.88	16.23	68	30	0.9
11 行目	0.07	0.099	17.88	16.23	68	30	0.8

step1：　変換対象ファイルを選択する

```
get file_name &1 $1 "csv"      /*ファイル読み出しダイヤログを表示させ、変換するファイル名を取得*/
```

※　&1 に選択されたファイル名、$1 に選択されたファイル個数が戻ります。尚、"csv"を付けた場合、ダイヤログに表示されるファイルは拡張子 csv のファイルのみ表示され、選択されたファイル名には拡張子は付きません。もし、拡張子が csv 以外の場合は、読み出す対象ファイルの拡張子を記述するか、又は、拡張子指定を付けないでおきます。拡張子指定されない場合は、&1 への戻りファイル名は選択されたファイル名に拡張子が付いて戻ります。

step2：　変換対象ファイルを定義する

```
def file_id %1 &1 csv            /*ファイル名&1 で取扱属性項目区切り文字半角カンマで ID を定義*/
```

※　区切り文字が半角カンマの場合は取扱属性を csv とし、tab の場合は txt として定義します。

step3：　信号名を読み出す

```
read cell %1 1,2 0 &2    /*1 行 2 列目から横方向に信号名を一括して読み出し、&2 に格納する*/
```

step4：　単位名を読み出す

```
read cell %1 2,2 0 &3     /*2 行 2 列目から横方向に単位名を一括して読み出し、&3 に格納する*/
```

step5：　サンプリング周期を取得し定義する

サンプリング周期の取得は時刻歴列を読み出し、差分値をサンプリング周期として定義します。

```
read cell %1 3,1 1 $10                /*3 行 1 列目から縦方向に時刻歴を読み出し、$10 に格納する*/
$10 = $10(1)-$10(0)                   /*データ番号 1 の値からデータ番号 0 の値を引き周期を求める*/
def sampl_period $10 "sec"             /*サンプリング周期の定義*/
```

step6：　データを読み出す

データを読み出す前に収録チャネル数（列数）を取得します。

```
read num_cell %1 $3                   /*変換対象ファイルの列数を読み出し$3 に格納する*/
```

※　変換対象ファイルの行毎の列数が$3 に格納されています。

```
$3 = MAX($3)-1                        /*$3 にチャネル数を求めます*/
```

※　チャネル数は、1 列目が時刻歴の為、列数の最大値から-1 して求めます。

```
read cell %1 3,2 1 $[100,$3]   /*3 行 2 列目から縦方向にチャネルデータを読み出す*/
```

※　格納チャネルは、$100 から連続して$3 に取得したチャネル数分のチャネル番号に格納されます。

$100 に 2 列目のデータが、$101 に 3 列目のデータが格納され、例えば$3 が 6 の場合は、$106 に 7 列目のデータが格納されることを意味します。

step7： データ格納先チャネルに信号名及び単位を付ける

```
assign &5 = &2|":"|&3              /*信号名と単位を連結する*/
$6 = 100                           /*チャネル番号カウンタ初期値*/
$5 = 0                             /*チャネル数カウンタ初期値*/
repeat_case $5 < $3                /*  チャネル数分の繰り返し*/
  proc name_unit_define{
    def ch_name $($6) &5($5)       /*チャネル毎に信号名単位を定義*/
    $6 = $6+1                      /*チャネル番号カウンタ更新*/
    $5 = $5+1                      /*チャネル数カウンタ更新*/
  }name_unit_define
```

※ 信号名、単位は一度に定義出来ないので、LOOP を形成しチャネル毎に付けます。

step8： ファイルを格納します。

```
def file_id %2 &1 wav              /*格納先ファイル ID を定義*/
```

※ ファイル名は変換対象ファイルと同じ名前で、拡張子が.hdr と.dat の 2 種のファイルが生成されます。
尚、&1 には拡張子無しのファイル名が格納されているものとしています。

```
def wav_type %2 float              /*格納データ形式は、4byte 浮動小数点形式を指定*/
save wave %2 $[100,$3]             /*データの格納*/
```