

試験波形を作る その1

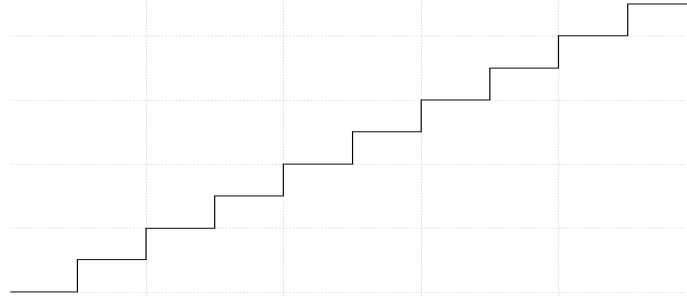
Script 記述した処理手続きを debug する場合に、実際の収録データが無い時に Script 記述で仮の波形を生成して行うことがあります。ここでは、階段状波形を作る方法について説明します。

<階段位置を明示的にデータ番号で指定して生成する方法>

X 位置（データ番号）と Y 値（階段の高さ）のテーブルを作成し、補間関数(ITP 関数)を使用して直線補間処理にて生成します。例では、次の条件で作成しています。

サンプリング周期=1ms、生成データ個数=10000 個（10 秒分）

階段値=50、階段数=10 段、階段幅=1000 点(1 秒)、繰り返し数=10 回



step1： 波形のサンプリング周期を定義します

サンプリング周期の定義は、サンプリング周期定義文を使用して定義します。

```
def sampl_period 1e-3 "sec" /* サンプリング周期を 1ms として定義する場合*/
```

step2： 補間テーブルを作成します。

補間テーブルは、Y 値と X 値 (index) のペアで生成します。

```
assign $2 "Table_y:" = 10<1> /* Y 値の元となる 10 要素の配列を用意*/
```

```
$2 = MGR((ACC($2)-1)*50,(ACC($2)-1)*50) /* 階段 1 段当たりの開始値と終了値を生成*/
```

又は、直接代入文で生成します。

```
assign $2 "table_y:" = 1<0,0,50,50,100,100,150,150,200,200,250,250,300,300,350,350,400,400,450,450>
```

※ \$2 は、0,0,50,50,100,100,150,150,200,200.....と並んだ要素 20 個の数列となります。

```
assign $3 "Table_X:" = 10<1> /*X 値(index)の元となる 10 要素の配列を用意*/
```

```
$3 = MGR((ACC($3)-1)*1000,ACC($3)*1000-1) /*階段の始点と終了点のデータ番号を生成*/
```

※ \$3 は、0,999,1000,1999,2000,2999,3000,3999...と並んだ要素 20 個の数列となります。

step3： 補間テーブルを使用する為の X 軸値(index)配列を生成します。

```
assign $6 = 10000<1> /* 要素数 10000 点の要素値 1 の配列を用意*/
```

```
$6 = ACC($6)-1 /* 0 から昇順に index を生成*/
```

※ \$6 は、0,1,2,3,4,5,6,7,8,9,10,11....と並んだ要素 10000 個の数列となります。

step4： 補間テーブルを使用して階段状波形を作成します。

作成した補間テーブル X 軸値 (index) と Y 軸値を使用して直線補間処理を行い、階段状波形を生成します。

```
$5 "階段波形:" = ITP($3,$2,$6) /* $5 に階段状波形が生成されます*/
```

```
assign $5 = 10<$5> /* 繰り返し数 10 回として生成します*/
```

step5： 生成した波形を格納します。

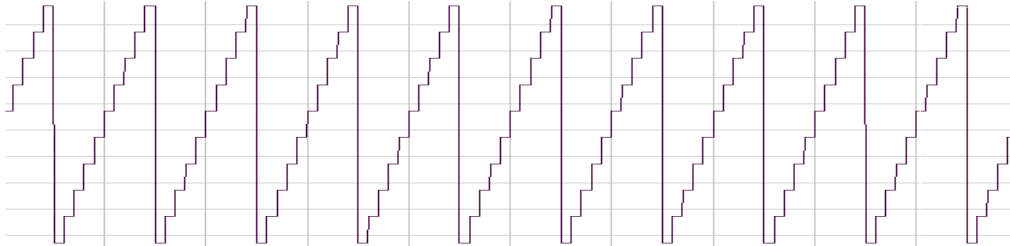
```
def file_id %1 "階段状波形" wav /*ファイル ID を定義します*/
```

```
save wave %1 $5 /*格納します*/
```

この結果、0~450 を範囲の 10 段 Step の上り階段状波形が格納されます。

<波形生成関数を使用して生成する方法>

波形生成関数(DAG 関数)を使用して、2種の鋸歯状波を生成して合成して生成します。
この時、留意する点は、生成される波は1周期を360°として演算される為、定義するサンプリング周波数を360の倍数の周波数で定義します。例では、次の条件で作成しています。
サンプリング周期=1/3.6ms、生成データ個数=3600個(1秒分)
階段値=2、階段数=10段、繰り返し数=10回



- step1: 波形のサンプリング周期を定義します
サンプリング周期の定義は、サンプリング周期定義文を使用して定義します。
`$4 = 1/3600`
`def sampl_period $4 "sec" /* 36の逆数は割り切れない為、倍精度演算で周期を求めて定義*/`
- step2: 繰り返し周期の基本鋸歯状波を生成します。
`$1 = DAG(3600,3,10,10,0)`
- step3: 階段生成用鋸歯状波を生成します。
`$2 = DAG(3600,3,100,1,180)+0.5 /* 基本鋸歯状波の振幅値の1/10の振幅値、位相180°で生成*/`
※ DAG関数で生成される波形は±の振幅を持ちますので、正の領域にシフトさせます。
- step4: 基本鋸歯状波と階段生成用鋸歯状波を合成します。
合成時に、階段生成用鋸歯状波を時間軸反転してから加算します。
`$3 = $1+REV($2) /* $3に階段状波形が生成されます*/`
- step5: 生成した波形を格納します。
`def file_id %1 "wave_test" wav`
`save wave %1 $3`

但し、波形生成関数で鋸歯状波を生成した場合は同じ時刻に別の値を持つ為(最大値と最小値が同じ時刻)厳密には、最大値を生成することが出来ません。サンプリング周波数を上げれば、最大値に限りなく近づきます。