

---

# PcWaveForm Script Subroutine Library

---

2010 年 04 月

Revision 1.01

## お断り

記載されている会社名および製品名はその会社の所有する商標です。  
記載された内容については事前のお断りなく変更させていただく場合がございます。  
記載された内容は 2010 年 4 月現在のものです。  
ご使用にあたっては、本取扱説明書の内容を十分お読みいただけますようお願い申し上げます。

本取扱説明書は、PDF 形式でプログラム CD の中に入っています。

## 株式会社 デイシー

〒205-0002

東京都羽村市栄町3-3-6

電話: 042-570-7121

メール: [info@deicy.co.jp](mailto:info@deicy.co.jp)

© Copyright 2006, 2007, 2008, 2009, 2010 DEICY Corporation

## 改定履歴

発行日	Revision	内容
2010年03月20日	1.00	初版
2010年04月26日	1.01	下記 Subroutine を追加 5.3. 範囲指定時間率頻度解析 5.4. 参照チャネル範囲指定時間率頻度解析 6. 振動暴露解析処理関連 6.1. 全身振動暴露・補正加速度演算 6.2. 全身振動暴露・補正加速度実効値演算 6.3. 全身振動暴露・区間最大過渡振動値演算 6.4. 全身振動暴露・区間振動値演算 6.5. 全身振動暴露・区間四乗曝露量値演算 6.6. 全身振動暴露・時間率頻度解析 6.7. 全身振動暴露・区間等価振動値演算 6.8. 全身振動暴露・相対累積時間率演算 6.9. 全身振動暴露・区間等価振動値/時間率相当振動値演算 6.10. 全身振動暴露・区間波高率演算 7. 快適性解析処理関連 7.1. 快適性・背もたれ補正加速度 7.2. 快適性・座面補正加速度 7.3. 快適性・座面補正角加速度 7.4. 快適性・足支持補正加速度 7.5. 快適性・背もたれ補正加速度実効値 7.6. 快適性・座面補正加速度実効値 7.7. 快適性・座面モーメント補正加速度実効値 7.8. 快適性・足支持補正加速度実効値 7.9. 快適性・全体合成補正加速度実効値

## 適用

- 本書は、下記 Version に適用します。

PcWaveForm Ver.7.01 ~

PcWaveFormFANA Ver.5.31 ~

PcWaveFormWBV+ Ver.2.11 ~

## ご注意

- 本書は万全を期して作成しておりますが、万一、ご不明なことや誤りなどお気づきのことがありましたらご連絡下さい。
- 本書の実行結果から生じるお客様の損害や不利益については、それが直接的、あるいは間接的を問わず一切責任を負いかねますのでご了承下さい。
- 本書は、改良のため予告なしに変更する場合があります。
- 本書の一部または全部を無断で複写または転載することは禁止されています。
- 本書に記載された会社名や製品名は、各社の登録商標である場合がございます。

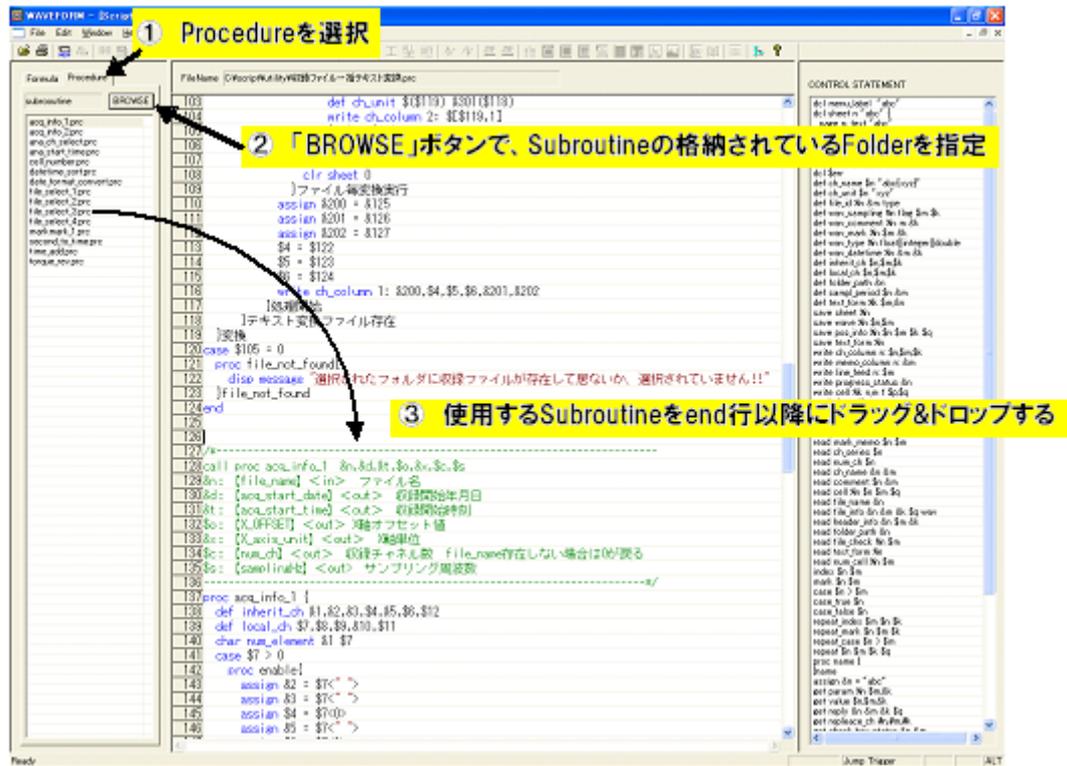
## 目次

Subroutine の組み込み方法	1
Subroutine の引数	1
1. 解析対象ファルの選択関連	2
1.1. フォルダ選択・波形ファイル選択1	2
1.2. フォルダ選択・波形ファイル選択2	3
1.3. カレント・フォルダ・同一サンプリング波形ファイル選択	5
1.4. カレント・フォルダ、同一拡張子ファイル選択	8
2. 解析対象波ファイルの収録情報取得関連	10
2.1. カレント・フォルダ・波形ファイル収録情報取得	10
2.2. カレント・波形ファイル収録情報取得	12
2.3. カレント・波形ファイル解析対象チャンネル選択	13
2.4. カレント・波形ファイル PAUSE 情報取得	14
3. 年月日時刻処理関連	15
3.1. ファイル名年月日時刻順並べ替え	15
3.2. 年月日時刻加算演算	17
3.3. 解析対象範囲開始年月日時刻演算	19
3.4. 年月日文字列形式相互変換	20
3.5. 秒数値から時分秒文字列変換	21
4. 解析対象範囲指定関連	22
4.1. MARK メモによる処理区間抽出	22
5. 頻度解析処理関連	24
5.1. 頻度処理結果・セル番号/中央値生成	24
5.2. トルク累積回転数頻度解析演算	25
5.3. 範囲指定時間率頻度解析	26
5.4. 基準チャンネル範囲指定時間率頻度解析	27
6. 振動暴露解析処理関連	28
6.1. 全身振動暴露解析・補正加速度演算	28
6.2. 全身振動暴露解析・補正加速度実効値演算	30
6.3. 全身振動暴露解析・区間最大過渡振動値演算	32
6.4. 全身振動暴露解析・区間振動値演算	34
6.5. 全身振動暴露解析・区間四乗暴露量値演算	36
6.6. 全身振動暴露解析・時間率頻度解析	38
6.7. 全身振動暴露解析・区間等価振動値演算	39
6.8. 全身振動暴露・相対累積時間率演算	41
6.9. 全身振動暴露・区間等価振動値/時間率相当振動値演算	42
6.10. 全身振動暴露・区間波高率演算	45

<b>7. 快適性解析処理関連</b> .....	47
7.1. 快適性・背もたれ補正加速度.....	47
7.2. 快適性・座面補正加速度.....	48
7.3. 快適性・座面補正角加速度.....	48
7.4. 快適性・足支持補正加速度.....	49
7.5. 快適性・背もたれ補正加速度実効値.....	50
7.6. 快適性・座面補正加速度実効値.....	51
7.7. 快適性・座面モーメント補正加速度実効値.....	52
7.8. 快適性・足支持補正加速度実効値.....	53
7.9. 快適性・全体合成補正加速度実効値.....	54

## Subroutine の組み込み方法

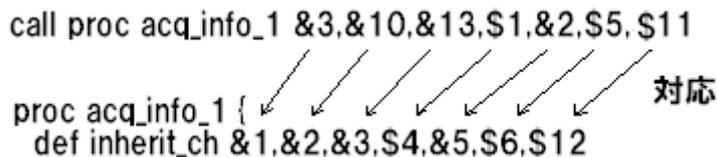
ScriptEditorWindow で subroutine を呼び出す Script 文を作成後、当該 Script の end 行以降に貼り付けます。



## Subroutine の引数

引数並び順、及び型式は、Subroutine 内で def inherit\_ch 文で定義されたチャンネル並び、及び型式と一致している必要があります。型式には、文字属性参照チャンネル、数値属性参照チャンネル、収録チャンネル、及びファイル番号の種別があります。チャンネル番号は、同じである必要はありません。

Subroutine 内では、引数として受け渡されるチャンネル番号に読み替えて実行されます。



尚、引数で受け渡されるチャンネル番号以外に、Subroutine 内で使用しているチャンネル番号はローカル定義されていますので、同じチャンネル番号を呼び出し元の Script 内で使用されていても問題ありません。

# 1. 解析対象ファルの選択関連

## 1.1. フォルダ選択・波形ファイル選択1

ProcFile 名	SS0101.pro
機能	フォルダを選択して、フォルダ内に格納されているファイルを選択します。 フォルダ選択ダイアログからフォルダを選択した後、選択されたフォルダに格納されている波形ファイル一覧リストが表示されますので、表示されたファイルリストから、ファイルを選択します。選択されたファイル名と個数を戻します。
呼び出し文	call file_select_1 &n,\$m
引数	&n:【FileName】<out><文字属性参照チャネル> 選択されたファイル名(拡張子無し)が戻る \$m:【num_file】<out><数値属性参照チャネル> 選択されたファイル個数が戻る
備考	<ul style="list-style-type: none"> <li>■ 指定したフォルダに指定した拡張子のファイルが存在していない場合は、ファイル個数が 0 として戻ります。ファイル個数が 0 の場合、再び関数を呼び出して再選択するか、以降の処理を行わず Script を終了させるかの処理が必要です。</li> <li>■ 選択されたファイルが現在使用中であるか否かはチェックされません。Open 可能か否かは、当該ファイルを処理する時点で、read file_check 文で行います。read file_check の戻り値が 0 の場合、Open 可能です。</li> <li>■ 他の拡張子を指定する場合は、Script の 10 行目 read file_info 文の最終引数を変更することにより可能です。</li> </ul>
記述例	<pre>def ch_name &amp;1 "ファイル名" def ch_name \$2 "ファイル個数" def ch_name \$3 "Open 可否フラグ" def ch_name \$4 "ファイルカウンタ" call proc file_select_1 &amp;1,\$2 case \$2 &gt; 0 /* ファイル個数&gt;0*/   proc ファイル選択済み{     \$4 = 0     repeat_case \$4 &lt; \$2       proc 処理{         def file_id %1 &amp;1(\$4) wav         read file_check %1 \$3 /*現在使用中か否か*/         case \$3 = 0 /* Open 可 */           proc 選択ファイル Open 可能 {             read wave %1             /* この演算処理ブロックに処理する内容を記述 */           }選択ファイル Open 可能           \$4 = \$4+1         }処理       }ファイル選択済み </pre>
Script	<pre>1 /* 2 call proc file_select_1 &amp;n,\$m 3 &amp;n : 【FileName】&lt;out&gt; 4 \$m : 【num_file】&lt;out&gt; 5 -----*/ 6 proc file_select_1{ 7   def inherit_ch &amp;3,\$2 8   def local_ch \$1,&amp;4,&amp;5 9   get folder_select 10  read file_info &amp;3 &amp;4 &amp;5 \$2 .hdr /* 拡張子指定行 */ 11  case \$2 &gt; 0 12    proc file_found{ 13      assign \$1 = \$2&lt;1&gt; 14      \$2 = 0 15      repeat_case \$2 = 0 16        proc file_select{ 17          get check_box_status &amp;3 \$1 18          \$2 = SUM(\$1) 19        }file_select 20      char recomposition &amp;3 \$1 0 &amp;3 21    }file_found </pre>

## 1.2. フォルダ選択・波形ファイル選択2

ProcFile 名	SS0102.prc
機能	フォルダを選択して、フォルダ内に格納されているファイルを選択します。 フォルダ選択ダイアログからフォルダを選択した後、選択されたフォルダに格納されている波形ファイル一覧リストが表示されますので、表示されたファイルリストからファイルを選択します。選択されたファイル名と個数及びファイル生成年月日時刻を戻します。
呼び出し文	<b>call proc file_select_2 &amp;n,&amp;d,&amp;t,\$m</b>
引数	&n:【FileName】<out><文字属性参照チャネル> 選択されたファイル名(拡張子無し)が戻る &d:【Date】<out><文字属性参照チャネル> 選択されたファイルの生成年月日が戻る yyyy/mm/dd &t :【time】<out><文字属性参照チャネル> 選択されたファイルの生成時刻が戻る hh:mm:ss \$m:【num_file】<out><数値属性参照チャネル> 選択されたファイル個数が戻る
備考	<ul style="list-style-type: none"> <li>■ 指定したフォルダに指定した拡張子のファイルが存在していない場合は、ファイル個数が 0 として戻ります。ファイル個数が 0 の場合、再び関数を呼び出して再選択するか、以降の処理を行わず Script を終了させるかの処理が必要です。</li> <li>■ 選択されたファイルが現在使用中であるか否かはチェックされません。 Open 可能か否かは、当該ファイルを処理する時点で、read file_check 文で行います。read file_check の戻り値が 0 の場合、Open 可能です。</li> <li>■ 他の拡張子を指定する場合は、Script の 12 行目 read file_info 文の最終引数を変更することにより可能です。</li> </ul>
記述例	<pre>def ch_name &amp;1 "ファイル名" def ch_name \$2 "ファイル個数" def ch_name \$3 "Open 可否フラグ" def ch_name \$4 "ファイルカウンタ" def ch_name &amp;5 "ファイル生成年月日" def ch_name &amp;6 "ファイル生成時刻" call proc file_select_2 &amp;1,&amp;5,&amp;6,\$2 case \$2 &gt; 0 /* ファイル個数&gt;0*/   proc ファイル選択済み{     \$4 = 0     repeat_case \$4 &lt; \$2       proc 処理{         def file_id %1 &amp;1(\$4) wav         read file_check %1 \$3 /*現在使用中か否か*/         case \$3 = 0 /* Open 可 */           proc 選択ファイル Open 可能 {             read wave %1             /* この演算処理ブロックに処理する内容を記述 */           }選択ファイル Open 可能           \$4 = \$4+1         }処理       }ファイル選択済み </pre>
Script	<pre>1 /*-----*/ 2 call proc file_select_2 &amp;n,&amp;d,&amp;h,\$m 3 &amp;n =&gt; 【FileName】&lt;out&gt; 4 &amp;d =&gt; 【date】&lt;out&gt; 5 &amp;t =&gt; 【time】&lt;out&gt; 6 \$m =&gt; 【num_file】&lt;out&gt; 7 /*-----*/ 8 proc file_select_2{ 9   def inherit_ch &amp;3,&amp;4,&amp;5,\$2 10  def local_ch \$1 11  get folder_select 12  read file_info &amp;3 &amp;4 &amp;5 \$2 .hdr /* 拡張子指定行 */ 13  case \$2 &gt; 0 14  proc file_found{ 15  assign \$1 = \$2&lt;1&gt; </pre>

---

```
16      $2 = 0
17      repeat_case $2 = 0
18      proc file_select{
19          get check_box_status &3 $1
20          $2 = SUM($1)
21      }file_select
22      char recomposition &3 $1 0 &3
23      char recomposition &4 $1 0 &4
24      char recomposition &5 $1 0 &5
25      }file_found
26      }file_select_2
```

---

### 1.3. カレント・フォルダ・同一サンプリング波形ファイル選択

ProcFile 名	SS0103.prc
機能	カレント・フォルダ内に格納されている同一サンプリング周波数のファイルを選択します。選択するファイルリストには、ファイル名の他、収録開始年月日時刻とサンプリング周波数も表示されます。 サンプリング周波数の指定は引数で行い、引数が負数の場合は同じサンプリングの一致を無視し、引数が 0 の場合は選択されたファイルのサンプリング一致が検査され、一致していないとファイル選択を繰り返します。 戻り値は、ファイル名とファイル個数が戻り、指定した k が 0 の場合は、\$k に選択されたサンプリング周波数も戻ります。
呼び出し文	<b>call proc file_select_3 \$n, &amp;m \$k</b>
引数	\$n:【チャネル数】<out><数値属性参照チャネル> 選択されたファイル数が戻る &m:【ファイル名】<out><数値属性参照チャネル> 選択されたファイル名が拡張子無しで戻る \$k:【フラグ】<in/out> <In> フラグ<0> サンプリング周波数の一致を無視 フラグ = 0 ファイル List の中で、選択したファイル同士のサンプリング周波数一致 フラグ > 0 選択するサンプリング周波数 <out> フラグ = 0 の時、選択したサンプリング周波数が戻る
備考	<ul style="list-style-type: none"> <li>■ ファイルが存在していない場合は、ファイル個数が 0 として戻ります。尚、サンプリング周波数を指定した場合 (\$k&gt;0)は、当該サンプリング周波数のファイルが存在しない場合もファイル個数が 0 として戻ります。</li> <li>■ 現在使用中のファイルは存在しても選択から除外されます。</li> <li>■ カレント・フォルダ以外から選択したい場合、call proc 文の前に直接フォルダパスを指定する場合は、def folder_path “フォルダパス”を記述します。又、フォルダ選択ダイアログで設定する場合は、get folder_select 文を記述します</li> <li>■ 本 proc 内で、ファイル番号%1 を使用して波形ファイルを定義し、Header 情報を確認しています。ファイル番号はグローバル扱いとなりますので、本 proc を跨いでファイル番号%1 を使用する場合、定義内容が異なってくるため、その場合、ファイル番号%1 を避けて記述します。</li> </ul>
記述例	<pre>def ch_name &amp;1 "ファイル名" def ch_name \$2 "ファイル個数" def ch_name \$3 "サンプリング周波数:Hz" def ch_name \$4 "ファイルカウンタ" \$3 = 1000 call proc file_select_3 &amp;1,\$2,\$3 \$4 = 0 repeat_case \$4 &lt; \$2   proc 処理{     def file_id %1 &amp;1(\$4) wav     read wave %1     /* この演算処理ブロックに処理する内容を記述*/     \$4 = \$4+1   }処理</pre>
Script	<pre>1 /*-----*/ 2 call proc file_select_3 &amp;n,\$m 3 &amp;n =&gt; 【FileName】&lt;out&gt; 4 \$m =&gt; 【num_file】&lt;out&gt; 5 \$k =&gt; 【sampling】&lt;in/out&gt; in=&gt; k&lt;0:無視 k=0:一致 k&gt;0:Sampl_freq 6 /*-----*/ 7 proc file_select_3{ 8   def inherit_ch &amp;2,\$1,\$11 9   def local_ch &amp;3,&amp;4,\$5,\$6,\$7,&amp;8,\$9,\$10,\$11,\$12,\$13 10  read file_info &amp;2 &amp;3 &amp;4 \$1 .hdr 11  case \$1 &gt; 0 12    proc search{ 13      assign \$7 = \$1&lt;1&gt; 14      assign &amp;3 = \$1&lt;" "&gt; 15      assign &amp;4 = \$1&lt;" "&gt; 16      assign \$10 = \$1&lt;0&gt; 17      \$5 = 0 18      repeat_case \$5 &lt; \$1</pre>

```

19     proc file_header_read{
20         def file_id %1 &2($5) wav
21         read file_check %1 $6
22         case $6 > 0
23             proc status{
24                 $7($5) = 0
25             }status
26         case $6 = 0
27             proc header{
28                 read wave %1 0,0
29                 read header_info &8 $9 "DATE "
30                 assign &3($5) = &8
31                 read header_info &8 $9 "TIME "
32                 assign &4($5) = &8
33                 $10($5) = 1/PRD()
34             }header
35             $5 = $5+1
36         }file_header_read
37     char recomposition &2 $7 &2
38     char recomposition &3 $7 &3
39     char recomposition &4 $7 &4
40     $10 = ZSP($7,$10)
41     $7 = ZSP($7,$7)
42     assign &8 = &2|" "&3|" "&4|" "|$10(F1)|"Hz"
43     case $11 > 0
44         proc cmp{
45             $1 = SUM(EQU($10,$11))
46             case $1 > 0
47                 proc ext{
48                     $7 = AND(EQU($10,$11),$7)
49                     char recomposition &2 $7 &2
50                     char recomposition &3 $7 &3
51                     char recomposition &4 $7 &4
52                     char recomposition &8 $7 &8
53                     $10 = ZSP($7,$10)
54                     $7 = ZSP($7,$7)
55                 }ext
56             }cmp
57         case $1 > 0
58             proc sel{
59                 $1 = 0
60                 $12 = 0
61                 repeat_case $12 = 0
62                 proc loop{
63                     case $11 > 0
64                         proc 指定周波数{
65                             $12 = 1
66                         }指定周波数
67                     case $11 < 0
68                         proc 一致無視{
69                             $12 = 1
70                         }一致無視
71                     repeat_case $1 = 0
72                     proc select{
73                         get check_box_status &8 $7
74                         $1 = SUM($7)
75                     }select
76                     $13 = ZSP($7,$10)
77                     case $11 = 0
78                         proc 一致確認{
79                             $12 = EQU(SUM(EQU($13,$13(0))),$1)
80                         }一致確認
81                     proc warning{
82                         disp message "サンプリング周波数が一致していません!!"

```

---

```
83             $1 = 0
84             }warning
85             }一致確認
86             }loop
87             case $11 > 0
88                 proc assign_freq{
89                     $11 = $13(0)
90                 }assign_freq
91                 char recomposition &2 $7 &2
92             }sel
93         }search
94     }file_select_3
```

---

## 1.4. カレント・フォルダ・同一拡張子ファイル選択

ProcFile 名	SS0104.prc
機能	カレント・フォルダ内に格納されている同一拡張子のファイルを選択します。ファイル選択する前に、フォルダ内に格納されている全てのファイルの拡張子が検査され、選択可能な拡張子種別を生成します。拡張子を選択後、ファイルリストには、選択された拡張子が付くファイル名が表示されます。 戻り値は、選択されたファイル名とファイル個数が戻ります。尚、ファイル名は拡張子付きで戻ります。
呼び出し文	<b>call proc file_select_4 \$n, &amp;m</b>
引数	\$n:【チャンネル数】<out><数値属性参照チャンネル> 選択されたファイル数が戻る &m:【ファイル名】<out><数値属性参照チャンネル> 選択されたファイル名が拡張子付きで戻る
備考	<ul style="list-style-type: none"> <li>■ ファイルが存在していない場合は、ファイル個数が0として戻ります。</li> <li>■ ファイル拡張子は、取得したファイル名の後ろから3文字を拡張子と見なしています。</li> <li>■ カレント・フォルダ以外から選択したい場合 call proc 文の前に、直接フォルダパスを指定する場合は def folder_path “フォルダパス”を記述します。又、フォルダ選択ダイアログで設定する場合は get folder_select 文を記述します</li> </ul>
記述例	<pre>def ch_name &amp;1 “ファイル名” def ch_name \$2 “ファイル個数” def ch_name \$3 “ファイルカウンタ” \$3 = 1000 call proc file_select_4 &amp;1,\$2 \$4 = 0 repeat_case \$4 &lt; \$2   proc 処理{     /* この演算処理ブロックに処理する内容を記述*/     \$4 = \$4+1   }処理</pre>
Script	<pre>1 /*-----*/ 2 call proc file_select_4 &amp;n,\$m 3 &amp;n =&gt; 【FileName】&lt;out&gt; 4 \$m =&gt; 【num_file】&lt;out&gt; 5 -----*/ 6 proc file_select_4{ 7   def inherit_ch &amp;2,\$1 8   def local_ch &amp;3,&amp;4,\$5,\$6,&amp;7,\$8,\$9,&amp;10,\$11 9   def ch_name &amp;7 “選択するファイルの拡張子” 10  read file_info &amp;2 &amp;3 &amp;4 \$1 11  case \$1 &gt; 0 12    proc exist{ 13      char length &amp;2 \$6 14      \$8 = \$6-3 15      char extract &amp;2 \$8 \$6 &amp;7 16      assign &amp;10 = &amp;7(0) 17      \$5 = 1 18      repeat_case \$5 &lt; \$1 19        proc 登録 check{ 20          assign &amp;3 = &amp;7(\$5) 21          char find &amp;10 &amp;3 1 \$9 22          \$9 = SUM(\$9) 23          case \$9 = 0 24            proc regist{ 25              assign &amp;10 = &amp;10,&amp;3 26            }regist 27            \$5 = \$5+1 28          }登録 check 29          assign &amp;3 = &amp;10(0) 30          get value &amp;10:&amp;3 31          char find &amp;7 &amp;3 1 \$9 32          \$9 = GTE(\$9,1) 33          char recomposition &amp;2 \$9 &amp;2 34          \$9 = ZSP(\$9,\$9)</pre>

---

```
35         $1 = 0
36         repeat_case $1 = 0
37         proc file_sel[
38             get check_box_status &2 $9
39             $1 = SUM($9)
40         ]file_sel
41         char recomposition &2 $9 &2
42     ]exist
43 ]file_select_4
```

---

## 2. 解析対象波ファイルの収録情報取得関連

### 2.1. カレント・フォルダ・波形ファイル収録情報取得

ProcFile 名	SS0201.prc
機能	カレント・フォルダにある波形ファイルのファイル名を指定し、指定された個々の波形ファイルの Header 情報から、収録開始年月日時刻、X_OFFSET 値、X 軸単位、収録チャンネル数及びサンプリング周波数を読み出します。
呼び出し文	<code>call proc acq_info_1.prc &amp;n,&amp;d,&amp;t,\$o,&amp;x,\$c,\$s</code>
引数	<p>&amp;n:【波形ファイル名】&lt;in&gt;&lt;文字属性参照チャンネル&gt; 収録開始年月日時刻を読み出したい波形ファイル名、拡張子(hdr)は有っても無くても良い。</p> <p>&amp;d:【収録開始年月日】&lt;out&gt;&lt;文字属性参照チャンネル&gt; 読み出された収録開始年月日が戻る MM-DD-YYYY</p> <p>&amp;t:【収録開始時刻】&lt;out&gt;&lt;文字属性参照チャンネル&gt; 読み出された収録開始時刻が戻る hh:mm:ss</p> <p>\$o:【X_OFFSET 値】&lt;out&gt;&lt;文字属性参照チャンネル&gt; 読み出された X 軸 Offset 値が戻る</p> <p>&amp;x:【X 軸単位】&lt;out&gt;&lt;文字属性参照チャンネル&gt; 読み出された X 軸単位が戻る</p> <p>\$c:【収録チャンネル数】&lt;out&gt;&lt;数値属性参照チャンネル&gt; 収録チャンネル数が戻る。当該ファイルが存在しない場合 0 が戻る。 尚、存在しない場合、他の戻り引数は意味を持ちません。</p> <p>\$s:【サンプリング周波数】&lt;out&gt;&lt;数値属性参照チャンネル&gt; 読み出されたサンプリング周波数が戻る</p>
備考	<ul style="list-style-type: none"> <li>■ Header 情報を読み出す波形ファイルがカレント・フォルダに存在している事が前提です。</li> <li>■ カレント・フォルダに格納されている波形ファイル名をサブルーチン呼び出す前に、read file_info 文で拡張子をhdr で実行し、カレント・フォルダに格納されている波形ファイル名を取得して置く必要があります。</li> <li>■ 本 proc 内で、ファイル番号%1 を使用して波形ファイルを定義し、Header 情報を確認しています。ファイル番号はグローバル扱いとなりますので、本 proc を跨いでファイル番号%1 を使用する場合、定義内容が異なってくるので、その場合、ファイル番号%1 を避けて記述します。</li> <li>■ カレントファイルから読み出す場合は、wav_info_2 を使用します。</li> </ul>
記述例	<pre>def ch_name &amp;1 "ファイル名" def ch_name &amp;2 "収録開始年月日" def ch_name &amp;3 "収録開始時刻" def ch_name &amp;4 "X_OFFSET 値" def ch_name &amp;5 "X 軸単位" def ch_name &amp;6 "収録チャンネル数" def ch_name &amp;7 "サンプリング周波数:Hz" def ch_name &amp;8 "ファイル個数" read file_info &amp;1 &amp;2 &amp;3 &amp;4 &amp;5 &amp;6 &amp;7 case &amp;8 &gt; 0   proc header{ call proc acq_info_1 &amp;1,&amp;2,&amp;3,&amp;4,&amp;5,&amp;6,&amp;7 /* この演算処理ブロックに処理内容を記述*/ }header</pre>
Script	<pre>1 /*-----*/ 2 call proc acq_info_1 &amp;n,&amp;d,&amp;t,\$o,&amp;x,\$c,\$s 3 &amp;n:【file_name】&lt;in&gt; ファイル名 4 &amp;d:【acq_start_date】&lt;out&gt; 収録開始年月日 5 &amp;t:【acq_start_time】&lt;out&gt; 収録開始時刻 6 \$o:【X_OFFSET】&lt;out&gt; X 軸オフセット値 7 &amp;x:【X_axis_unit】&lt;out&gt; X 軸単位 8 \$c:【num_ch】&lt;out&gt; 収録チャンネル数 file_name 存在しない場合は 0 が戻る 9 \$s:【samplingHz】&lt;out&gt; サンプリング周波数 10 -----*/ 11 proc acq_info_1 { 12   def inherit_ch &amp;1,&amp;2,&amp;3,&amp;4,&amp;5,&amp;6,&amp;12 13   def local_ch \$7,\$8,\$9,&amp;10,\$11 14   char num_element &amp;1 \$7 15   case \$7 &gt; 0 16     proc enable{ 17       assign &amp;2 = \$7&lt;" "&gt; 18       assign &amp;3 = \$7&lt;" "&gt; 19       assign \$4 = \$7&lt;0&gt;</pre>

---

```

20     assign &5 = $7<" ">
21     assign $6 = $7<1>
22     assign $12 = $7<0>
23     $8 = 0
24     repeat_case $8 < $7
25         proc start{
26             def file_id %1 &1($8) wav
27             read wave %1 0,0
28             read file_check %1 $9
29             $6($8) = 0
30             case $9 < 2
31                 proc header_read{
32                     read header_info &10 $11 "DATE "
33                     assign &2($8) = &10
34                     read header_info &10 $11 "TIME "
35                     assign &3($8) = &10
36                     read header_info &10 $11 "X_OFFSET "
37                     assign &4($8) = &10
38                     read header_info &10 $11 "HORZ_UNITS "
39                     assign &5($8) = &10
40                     read num_ch $6($8)
41                     $12($8) = 1/PRD()
42                 }header_read
43                 case $9 = 2
44                     proc not_exist{
45                         assign &2($8) = " "
46                         assign &3($8) = " "
47                         $4($8) = 0
48                         assign &5($8) = " "
49                     }not_exist
50                 $8 = $8+1
51             }start
52         }enable
53     case $7 <= 0
54         proc disable{
55             assign &2 = " "
56             assign &3 = " "
57             $4 = 0
58             assign &5 = " "
59             $6 = 0
60             $12 = 0
61         }disable
62     }acq_info_1

```

---

## 2.2. カレント・波形ファイル収録情報取得

ProcFile 名	SS0202.prc
機能	カレント・波形ファイルの Header 情報から、収録開始年月日時刻、X_OFFSET 値、X 軸単位、収録チャンネル数、サンプリング周波数、及び収録チャンネル番号を読み出します。
呼び出し文	call proc acq_info_2 &d,&t,\$o,&x,\$c,\$n
引数	<p>&amp;d:【収録開始年月日】&lt;out&gt;&lt;文字属性参照チャネル&gt; 読み出された収録開始年月日に戻る MM-DD-YYYY</p> <p>&amp;t:【収録開始時刻】&lt;out&gt;&lt;文字属性参照チャネル&gt; 読み出された収録開始時刻に戻る hh:mm:ss</p> <p>\$o:【X_OFFSET 値】&lt;out&gt;&lt;文字属性参照チャネル&gt; 読み出された X 軸 Offset 値に戻る</p> <p>&amp;x:【X 軸単位】&lt;out&gt;&lt;文字属性参照チャネル&gt; 読み出された X 軸単位に戻る</p> <p>\$c:【収録チャンネル数】&lt;out&gt;&lt;数値属性参照チャネル&gt; 収録チャンネル数に戻る。尚、0 に戻る場合、カレントに波形ファイルが存在しないことを意味します。尚、存在しない場合、他の戻り引数は意味を持ちません。</p> <p>\$n:【収録チャンネル番号】&lt;out&gt;&lt;数値属性参照チャネル&gt; 収録チャンネル番号が戻ります。複数チャンネル収録されている場合は、数列で戻ります。</p>
備考	<p>■ Header 情報を読み出すことが可能である必要があります。 読み出すことが可能な場合は 2 通りあります。</p> <p>① 波形ファイルが、PcWaveForm 上で表示され、かつ、解析範囲が指定されている場合で、波形 Window から Script が実行されている場合。</p> <p>② Script 中で波形ファイルを読み出した場合。</p>
記述例	<pre>def ch_name &amp;2 "収録開始年月日" def ch_name &amp;3 "収録開始時刻" def ch_name \$4 "X_OFFSET 値" def ch_name &amp;5 "X 軸単位" def ch_name \$6 "収録チャンネル数" def ch_name \$7 "収録チャンネル番号" call proc wav_info_2 &amp;2,&amp;3,\$4,&amp;5,\$6,\$7</pre>
Script	<pre>1 /*----- 2 call proc acq_info_2 &amp;d,&amp;t,\$o,&amp;x,\$c,\$n 3 &amp;d:【acq_start_date】&lt;out&gt; 収録開始年月日 4 &amp;t:【acq_start_time】&lt;out&gt; 収録開始時刻 5 \$o:【X_OFFSET】&lt;out&gt; X 軸オフセット値 6 &amp;x:【X_axis_unit】&lt;out&gt; X 軸単位 7 \$c:【num_ch】&lt;out&gt; 収録チャンネル数 読み出せない場合は 0 に戻る 8 \$s:【ch_seris】&lt;out&gt; 収録チャンネル番号列 9 -----*/ 10 proc acq_info_2 { 11     def inherit_ch &amp;1,&amp;2,\$3,&amp;4,\$5,\$6 12     def local_ch &amp;7,\$8 13     read num_ch \$5 14     case \$5 &gt; 0 15     proc exist{ 16         read header_info &amp;7 \$8 "DATE " 17         assign &amp;1 = &amp;7 18         read header_info &amp;7 \$8 "TIME " 19         assign &amp;2 = &amp;7 20         read header_info &amp;7 \$8 "X_OFFSET " 21         assign \$3 = &amp;7 22         read header_info &amp;7 \$8 "HORZ_UNITS " 23         assign &amp;4 = &amp;7 24         read ch_series \$6 25     }exist 26 }acq_info_2</pre>

## 2.3. カレント・波形ファイル解析対象チャンネル選択

ProcFile 名	SS0203.prc
機能	カレント・波形ファイルの収録チャンネルリストから、解析対象チャンネル番号を選択します。
呼び出し文	<b>call proc ana_ch_select \$n,\$m,&amp;k,&amp;u</b>
引数	<p>\$n:【チャンネル数】&lt;out&gt;&lt;数値属性参照チャンネル&gt;          選択されたチャンネル数に戻る</p> <p>\$m:【チャンネル番号】&lt;out&gt;&lt;数値属性参照チャンネル&gt;          選択されたチャンネル番号に戻る</p> <p>&amp;k:【信号名】&lt;out&gt;&lt;文字属性参照チャンネル&gt;          チャンネルの信号名が、チャンネル番号に対応して戻る。チャンネル番号\$m(i)⇒信号名&amp;k(i)</p> <p>&amp;u:【単位】&lt;out&gt;&lt;文字属性参照チャンネル&gt;          チャンネルの単位がチャンネル番号に対応して戻る。チャンネル番号\$m(i)⇒単位&amp;u(i)</p>
備考	<p>■ Header 情報を読み出すことが可能である必要があります。          読み出すことが可能な条件は 2 通りあります。</p> <p>① 波形ファイルが PcWaveForm 上で表示され、かつ、解析範囲が指定されている場合で、波形 Window から Script が実行されている場合。</p> <p>② Script 中で波形ファイルを読み出した場合。</p>
記述例	<pre>def ch_name \$1 "選択されたチャンネル数" def ch_name \$2 "選択されたチャンネル番号列" def ch_name &amp;3 "選択されたチャンネルの信号名" def ch_name &amp;4 "選択されたチャンネルの単位" call proc ana_ch_select \$1,\$2,&amp;3,\$4</pre>
Script	<pre>1      /*----- ana_ch_select subroutine -----*/ 2      call porc \$n,\$m,&amp;k,&amp;u 3      \$n ⇒ &lt;out&gt; number of channel 4      \$m ⇒ &lt;out&gt; channel series 5      &amp;k ⇒ &lt;out&gt; channel name 6      &amp;u ⇒ &lt;out&gt; channel unit 7 8      proc ana_ch_select{ 9          def inherit_ch \$1,\$2,&amp;3,&amp;4 10         def local_ch &amp;5,\$6 11         read num_ch \$1 12         case \$1 &gt; 0 13             proc ch_select{ 14                 read ch_series \$2 15                 read ch_name &amp;3 &amp;4 16                 assign \$6 = \$1&lt;1&gt; 17                 assign &amp;5 = "ch" \$2(F0)" " &amp;3 "("&amp;4)" 18                 \$1 = 0 19                 repeat_case \$1 = 0 20                     proc check_box{ 21                         get check_box_status &amp;5 \$6 22                         \$1 = SUM(\$6) 23                     }check_box 24                     \$2 = ZSP(\$6,\$2) 25                     char recomposition &amp;3 \$6 &amp;3 26                     char recomposition &amp;4 \$6 &amp;4 27                 }ch_select 28             case \$1 = 0 29                 proc disable{ 30                     disp message "現在使用中か、波形ファイルが Open されていません!!" 31                 }disable 32             }ana_ch_select</pre>

## 2.4. カレント・波形ファイル PAUSE 情報取得

ProcFile 名	SS0204.prc
機能	カレント・波形ファイルに付けられている PAUSE 情報個数、及び対応する PAUSE 再開時刻とデータ番号を戻します。
呼び出し文	<b>call proc pause_pos \$n,\$m,&amp;k</b>
引数	<p>\$n:【pause 個数】&lt;out&gt;&lt;数値属性参照チャネル&gt; 解析対象範囲に付けられている PAUSE 個数が戻る。</p> <p>\$m:【index 値】&lt;out&gt;&lt;数値属性参照チャネル&gt; PAUSE の付けられている index が戻る。PAUSE 個数が 0 の場合-1 が戻る。</p> <p>&amp;k:【再開時刻】&lt;out&gt;&lt;文字属性参照チャネル&gt; PAUSE 解除した時刻が戻る。PAUSE 個数が 0 の場合、半角スペース1文字が戻る。</p>
備考	<p>■ Header 情報を読み出すことが可能である必要があります。 読み出すことが可能な条件は 2 通りあります。</p> <p>① 波形ファイルが、PcWaveForm 上で表示され、かつ、解析範囲が指定されている場合で、波形 Window から Script が実行されている場合</p> <p>② Script 中で波形ファイルを読み出した場合</p>
記述例	<pre>def ch_name \$1 "Pause 個数" def ch_name \$2 "Pause 位置 index" def ch_name &amp;3 "再開時刻" call proc pause_pos \$1,\$2,&amp;3</pre>
Script	<pre>1      /*-----*/ 2      call proc pause_pos \$n,\$m,&amp;k 3      \$n =&gt;【num_pause】&lt;out&gt;&lt;数値属性参照チャネル&gt; 4      \$m =&gt;【pause_index】&lt;out&gt;&lt;数値属性参照チャネル&gt; 5      &amp;k =&gt;【re-start_time】&lt;out&gt;&lt;文字属性参照チャネル&gt; 6      -----*/ 7      proc pause_pos{ 8          def inherit_ch \$1,\$3,&amp;2 9          read header_info &amp;2 \$1 10         char find &amp;2 "PAUSE " 1 \$3 11         \$1 = SUM(EQU(\$3,1)) 12         case \$1 &gt; 0 13             proc pause_detect{ 14                 \$3 = EQU(\$3,1) 15                 char recomposition &amp;2 \$3 &amp;2 16                 char extract &amp;2 6 30 &amp;2 17                 assign \$3 = &amp;2 18                 char find &amp;2 "." 6 \$4 19                 \$4 = \$4+1 20                 char extract &amp;2 \$4 8 &amp;2 21             }pause_detect 22         case \$1 &lt; 1 23             proc not_detect{ 24                 \$3 = -1 25                 assign &amp;2 = " " 26             }not_detect 27         }pause_pos</pre>

### 3. 年月日時刻処理関連

#### 3.1. ファイル名年月日時刻順並べ替え

ProcFile 名	SS0301.prc
機能	ファイル名と年月日及び時刻を入力し、年月日時刻順に並べ替えます。
呼び出し文	<b>call proc datetime_sort &amp;n,&amp;d,&amp;t,&amp;m</b>
引数	<p>&amp;n:【FileName】&lt;in/out&gt;&lt;文字属性参照チャネル&gt; 与えたファイル名が並び替えられて戻る</p> <p>&amp;d:【Date】&lt;in/out&gt;&lt;文字属性参照チャネル&gt; 与えた年月日が並び替えられて戻る</p> <p>&amp;t :【Time】&lt;in/out&gt;&lt;文字属性参照チャネル&gt; 与えた時刻が並び替えられて戻る hh:mm:ss</p> <p>\$m:【flag】&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 並び替え順、及び与える年月日フォーマットを指定します。</p> <p>flag = 0 ascending_order 年月日記述形式が YYYY/MM/DD の場合 flag = 1 descending_order 年月日記述形式が YYYY/MM/DD の場合 flag = 2 ascending_order 年月日記述形式が MM-DD-YYYY の場合 falg = 3 descending_oder 年月日記述形式が MM-DD-YYYY の場合</p> <p>戻り値は、正常終了⇒0、パラメタ不正⇒-1となります。</p>
備考	<ul style="list-style-type: none"> <li>■ 引数 FileName、Date、Time の要素数は同数で、1 個以上なければなりません。要素数が等しくない場合は、flag 戻り値が-1 となります。</li> <li>■ 時刻の秒は小数点以下の記述があっても問題ありません。</li> <li>■ 年月日形式が指定された形式と異なってもチェックされませんので、注意下さい。</li> </ul>
記述例	<pre>call proc file_select_2 &amp;1,&amp;2,&amp;3,\$4 /*ファイル名、生成年月日時刻取得*/ \$4 = 1 /* sort flag descending oder*/ call proc datetime_sort &amp;1,&amp;2,&amp;3,\$4 /*年月日時刻降順並べ替え*/</pre>
Script	<pre>1 /*-----*/ 2 call proc datetime_sort &amp;n,&amp;d,&amp;t,&amp;m 3 &amp;n =&gt; 【FileName】&lt;out&gt; 4 &amp;d =&gt; 【date】&lt;out&gt; 5 &amp;t =&gt; 【time】&lt;out&gt; 6 \$m =&gt; 【flag】&lt;in/out&gt; 7 flag = 0 ascending YYYY/MM/DD 8 flag = 1 descending YYYY/MM/DD 9 flag = 2 ascending MM-DD-YYYY 10 flag = 3 descending MM-DD-YYYY 11 return flag = 0 ok ==-1 error 12 -----*/ 13 proc datetime_sort{ 14 def inherit_ch &amp;3,&amp;4,&amp;5,\$2 15 def local_ch \$1,&amp;6,&amp;7,&amp;8,&amp;9,\$10,\$11,\$12 16 char num_element &amp;3 \$10 17 char num_element &amp;4 \$11 18 char num_element &amp;5 \$12 19 \$1 = AND(AND(EQU(\$10,\$11),EQU(\$11,\$12)),GTE(\$10,2)) 20 case_true \$1 21 proc enable{ 22 case \$2 &lt; 2 23 proc YYYY/MM/DD{ 24 char extract &amp;4 1 4 &amp;6 25 char extract &amp;4 6 2 &amp;7 26 char extract &amp;4 9 2 &amp;8 27 }YYYY/MM/DD 28 case \$2 &gt; 1 29 proc MM-DD-YYYY{ 30 char extract &amp;4 7 4 &amp;6 31 char extract &amp;4 4 2 &amp;7 32 char extract &amp;4 1 2 &amp;8 33 }MM-DD-YYYY 34 assign \$10 = &amp;6 &amp;7 &amp;8 35 char extract &amp;5 1 2 &amp;6 36 char extract &amp;5 4 2 &amp;7</pre>

---

```
37      char extract &5 7 10 &8
38      assign $11 = &6|&7|&8
39      $10 = $1*240000+$2
40      $11 = SRT(2,$10)
41      char recomposition &3 $11 1 &3
42      char recomposition &4 $11 1 &4
43      char recomposition &5 $11 1 &5
44      $2 = 0
45      }enable
46      case_false $1
47      proc error_return{
48          $2 = -1
49      }error_return
50      }datetime_sort
```

---

### 3.2. 年月日時刻加算演算

ProcFile 名	SS0302.prc
機能	与えられた年月日、時刻文字列に数値で与えられた秒数を加算した結果の年月日、時刻及び秒余り値を戻します。
呼び出し文	call proc time_add &d,&t,&m
引数	&d: [Date] <in/out> <文字属性参照チャネル> 与えた年月日に秒数が加算された結果が戻る MM-DD-YYYY &t : [Time] <in/out> <文字属性参照チャネル> 与えた時刻に秒数が加算された結果が戻る hh:mm:ss &m: [X_OFFSET] <in/out> <数値属性参照チャネル> 加算する秒数を与え、秒の小数点以下の値に更新される。
備考	<ul style="list-style-type: none"> <li>■ Date(&amp;d)、Time(&amp;t)、X_OFFSET(&amp;m) の要素数は同数でなければなりません。</li> <li>■ 年月日(Date)のフォーマットは、MM-DD-YYYY で無ければならず、同様に時刻(Time)は、hh:mm:ss で無ければなりません。</li> <li>■ 加算する秒数(&amp;m)は正数の必要があり、負数で与えられても絶対値化して加算されます。</li> </ul>
記述例	<pre>def ch_name &amp;1 "file name" def ch_name &amp;2 "date" def ch_name &amp;3 "time" def ch_name &amp;4 "X_OFFSET" def ch_name &amp;5 "xunit" read file_info &amp;1 &amp;2 &amp;3 &amp;4 .hdr /*カレントフォルダから波形ファイル名取得*/ call proc acq_info_1 &amp;1,&amp;2,&amp;3,&amp;4,&amp;5,&amp;6,&amp;7 /*収録開始年月日時刻取得*/ call proc time_add &amp;2,&amp;3,&amp;4 /* 収録年月日時刻に X_OFFSET を加算*/</pre>
Script	<pre>1 /*----- time_add subroutine -----*/ 2 call proc &amp;n,&amp;m,&amp;k 3 &amp;n =&gt; &lt;in/out&gt; MM-DD-YYYY 4 &amp;m =&gt; &lt;in/out&gt; hh:mm:ss 5 &amp;k =&gt; &lt;in&gt; add_sec &lt;out&gt; X_OFFSET 6 -----*/ 7 proc time_add[ 8   def inherit_ch &amp;9,&amp;10,&amp;2 9   def local_ch &amp;12,&amp;13,&amp;14,&amp;3,&amp;4,&amp;5,&amp;6,&amp;7,&amp;8,&amp;11,&amp;15 10  char extract &amp;10 1 2 &amp;12 /* hh */ 11  char extract &amp;10 4 2 &amp;13 /* dd */ 12  char extract &amp;10 7 2 &amp;14 /* ss */ 13  assign &amp;3 = &amp;12 14  assign &amp;4 = &amp;13 15  assign &amp;5 = &amp;14 16  &amp;3 = &amp;3*3600+&amp;4*60+&amp;5+ABS(&amp;2) /*total sec */ 17  &amp;8 = INT(&amp;3/86400) /*over_days*/ 18  &amp;3 = &amp;3-&amp;8*86400 /* remain_total_sec*/ 19  &amp;4 = INT(&amp;3/3600) /* hour */ 20  &amp;5 = INT((&amp;3-&amp;4*3600)/60) /* minute*/ 21  &amp;6 = INT(&amp;3-&amp;4*3600-&amp;5*60) /* second*/ 22  &amp;2 = &amp;3-&amp;4*3600-&amp;5*60-&amp;6 /* X_OFFSET*/ 23  assign &amp;12 = &amp;4(F0) 24  assign &amp;13 = &amp;5(F0) 25  assign &amp;14 = &amp;6(F0) 26  call proc Oadd &amp;12,&amp;4 27  call proc Oadd &amp;13,&amp;5 28  call proc Oadd &amp;14,&amp;6 29  assign &amp;10 = &amp;12 ":."&amp;13 ":."&amp;14 30  &amp;7 = LEN(&amp;2) 31  &amp;15 = 0 32  repeat_case &amp;15 &lt; &amp;7 33  proc loop[ 34  case &amp;8(&amp;15) &gt; 0 35  proc day_over[ 36  assign &amp;11 = 0,31,28,31,30,31,30,31,31,30,31 37  char extract &amp;9(&amp;15) 7 4 &amp;12 /* YYYY */ 38  char extract &amp;9(&amp;15) 4 2 &amp;13 /* DD */</pre>

---

```

39      char extract &9($15) 1 2 &14          /* MM */
40      assign $3 = &12                       /* year */
41      assign $4 = &13                       /* day */
42      assign $5 = &14                       /* month */
43      $6 = $3-INT($3/4)*4
44      case $6 = 0
45          proc 閏年{
46              $11(2) = 29
47          }閏年
48      $4 = $4+$8($15)                        /* 日加算 */
49      $6 = $4-$11($5)                       /* 当月 OVER 日数 */
50      repeat_case $6 > 0
51          proc month_up{
52              $4 = $6
53              $5 = $5+1                      /* month_up*/
54              case $5 > 12
55                  proc year_up{
56                      $5 = 1
57                      $3 = $3+1              /* year_up*/
58                  }year_up
59                  $6 = $4-$11($5)
60              }month_up
61          assign &12 = $3(F0)
62          assign &13 = $4(F0)
63          assign &14 = $5(F0)
64          call proc 0add &13,$4
65          call proc 0add &14,$5
66          assign &9($15) = &14|"-"|&13|"-"|&12
67      }day_over
68      $15 = $15+1
69  }loop
70  }time_add

```

---

### 3.3. 解析対象範囲開始年月日時刻演算

ProcFile 名	SS0303.prc
機能	解析対象範囲が指定された時の解析開始地点の年月日時刻戻します。
呼び出し文	<b>call proc ana_start_time &amp;d,&amp;t,&amp;m</b>
引数	<p>&amp;d:【Date】&lt;out&gt;&lt;文字属性参照チャネル&gt;          解析開始年月日が戻る MM-DD-YYYY</p> <p>&amp;t :【Time】&lt;out&gt;&lt;文字属性参照チャネル&gt;          解析開始時刻が戻る hh:mm:ss</p> <p>&amp;m:【X_OFFSET】&lt;out&gt;&lt;数値属性参照チャネル&gt;          解析開始時刻秒以下の値が戻る</p>
備考	<ul style="list-style-type: none"> <li>■ Header 情報を読み出すことが可能である必要があります。              読み出すことが可能な条件は 2 通りあります。             <ul style="list-style-type: none"> <li>① 波形ファイルが、PcWaveForm 上で表示され、かつ、解析範囲が指定されている場合で、波形 Window から Script が実行されている場合</li> <li>② Script 中で波形ファイルを読み出した場合</li> </ul> </li> <li>■ X 軸単位が秒の時は、解析開始年月日時刻は当該ファイルの X_OFFSET 値が参照され、開始時刻が秒単位の余り値が X_OFFSET として戻ります。</li> <li>■ X 軸単位が秒以外の時は、開始年月日及び時刻は変化せず、X_OFFSET 値が開始位置の値として戻ります。</li> <li>■ 本サブルーチン内で、年月日時刻加算演算サブルーチン time_add.prc が参照されますので、使用する場合、当該サブルーチン以外に time_add.prc も必要となります。</li> </ul>
記述例	<pre>def ch_name &amp;1 "解析開始年月日" def ch_name &amp;2 "解析開始時刻" def ch_name \$3 "X_OFFSET 値" call proc ana_start_time &amp;1,&amp;2,\$3</pre>
Script	<pre>1 /*----- ana_start_time1 subroutine -----*/ 2 call proc ana_start_time &amp;n,&amp;m,\$k 3 &amp;n =&gt; &lt;out&gt; MM-DD-YYYY 4 &amp;m =&gt; &lt;out&gt; hh:mm:ss 5 \$k =&gt; &lt;out&gt; X_OFFSET 6 -----*/ 7 proc ana_start_time{ 8   def inherit_ch &amp;1,&amp;2,\$3 9   def local_ch &amp;4,\$5,&amp;6,\$7,\$8 10  assign &amp;6 = "SEC","Sec","sec" 11  read header_info &amp;1 \$5 "DATE " 12  read header_info &amp;2 \$5 "TIME " 13  read header_info &amp;4 \$5 "HORZ_UNIT " 14  char find &amp;6 &amp;4 1 \$5 15  read header_info &amp;4 \$7 "X_OFFSET " 16  assign \$7 = &amp;4 17  \$3 = \$7+STA()*PRD() 18  \$5 = GTE(SUM(\$5),1) 19  case_true \$5 20    proc x_sec{ 21      call proc time_add &amp;1,&amp;2,\$3 22    }x_sec 23  }ana_start_time</pre>

### 3.4. 年月日文字列形式相互変換

ProcFile 名	SS0304.prc
機能	年月日文字列の形式を YYYY/MM/DD の時は MM-DD-YYYY に変換し、MM-DD-YYYY の時は YYYY/MM/DD に変換します。
呼び出し文	<b>call proc date_format_convert &amp;n</b>
引数	&n 【date】<in/out><文字属性参照チャネル> 形式は、YYYY/DD/MM、又は MM-DD-YYYY の何れかの必要があります。
備考	■ 年月日の形式が、YYYY/MM/DD、又は MM-DD-YYYY 以外の場合、結果は保証されません。
記述例	def ch_name &1 "file_name" def ch_name &2 "date" def ch_name &3 "time" def ch_name \$4 "num_file" call proc file_select_2 &1,&2,&3,\$4 call proc date_format_convert &2
Script	<pre> 1      /*-----*/ 2      call proc date_format_convert &amp;n 3      &amp;n =&gt; 【 date 】 &lt;in/out&gt; MM-DD-YYYY =&gt; YYYY/MM/DD  YYYY/MM/DD =&gt; MM-DD-YYYY 4      -----*/ 5      proc date_format_convert{ 6          def inherit_ch &amp;1 7          def local_ch &amp;3,&amp;4,&amp;5,\$7,\$8,&amp;10 8          char num_element &amp;1 \$7 9          char extract &amp;1 3 1 &amp;10 10         \$8 = 0 11         repeat_case \$8 &lt; \$7 12         proc format_conv{ 13             case &amp;10(\$8) = "-" 14                 proc MM-DD-YYYY_to{ 15                     char extract &amp;1 1 2 &amp;4 16                     char extract &amp;1 4 2 &amp;5 17                     char extract &amp;1 7 4 &amp;3 18                     assign &amp;1(\$8) = &amp;3"/"/&amp;4"/"/&amp;5 19                 }MM-DD-YYYY_to 20             case &amp;10(\$8) &lt;&gt; "-" 21                 proc YYYY/MM/DD_to{ 22                     char extract &amp;1 1 4 &amp;3 23                     char extract &amp;1 6 2 &amp;4 24                     char extract &amp;1 9 2 &amp;5 25                     assign &amp;1(\$8) = &amp;4"/"/&amp;5"/"/&amp;3 26                 }YYYY/MM/DD_to 27             \$8 = \$8+1 28         }format_conv 29     }date_format_convert </pre>

### 3.5. 秒数値から時分秒文字列変換

ProcFile 名	SS0305.prc
機能	数値属性で与えられた秒値を xx 時間 xx 分 xx.xxxx 秒の文字列形式に変換します。
呼び出し文	<b>call proc second_to_time \$n &amp;m</b>
引数	\$n 【second】<in/out> <数値属性参照チャネル> 変換する秒値を指定します。尚、flag 値が 1 の時、秒単位での余り数値が戻ります。 &m 【time】<out> <文字属性参照チャネル> 変換された結果の文字列が戻ります。
備考	■ 秒数の下 4 桁以下の値は切り捨てられます。
記述例	\$1 = 123456.789 call proc second_to_time \$1,&2
Script	<pre> 1      /*-----*/ 2      call proc second_to_time \$n,&amp;m 3      \$n =&gt;【second】&lt;in&gt; 4      &amp;m =&gt;【time】&lt;out&gt; hh 時間 mm 分 ss.ssss 秒 5      -----*/ 6      proc second_to_time{ 7          def inherit_ch \$1,&amp;2 8          def local_ch &amp;3,&amp;4,&amp;5,&amp;6,&amp;7,&amp;8,&amp;10 9          \$1 = \$1+1e-9 10         \$6 = INT(\$1/3600) 11         \$7 = INT((\$1-\$6*3600)/60) 12         \$8 = \$1-\$6*3600-\$7*60 13         assign &amp;3 = \$6(F0) 14         call proc Oadd &amp;3,&amp;6 15         assign &amp;4 = \$7(F0) 16         call proc Oadd &amp;4,&amp;7 17         assign &amp;5 = \$8(F4) 18         call proc Oadd &amp;5,&amp;8 19         assign &amp;2 = &amp;3"時間"&amp;4"分"&amp;5"秒" 20     }second_to_time 21     proc Oadd{ 22         def inherit_ch &amp;1,\$2 23         def local_ch \$2,\$3,\$4 24         \$3 = LEN(\$2) 25         \$4 = 0 26         repeat_case \$4 &lt; \$3 27             proc loop{ 28                 case \$2(\$4) &lt; 10 29                     proc Oadd{ 30                         assign &amp;1(\$4) = "0"&amp;1(\$4) 31                     }Oadd 32                     \$4 = \$4+1 33             }loop </pre>

## 4. 解析対象範囲指定関連

### 4.1. MARK メモによる処理区間抽出

ProcFile 名	SS0401.prc
機能	<p>現在の解析範囲に付けられた MARK 情報を読み出し、有効区間の開始 index 番号配列と対応したデータ個数配列、及び開始位置の MARK メモを戻します。(上図網掛け部分が処理対象区間)</p>
呼び出し文	<b>call proc markmark_1 \$n, \$m,k,&amp;s</b>
引数	<p>\$n ⇒【開始 index】&lt;out&gt;&lt;数値属性参照チャネル&gt; Mark メモ 1~98 の開始 index が戻ります。error が在った場合は -1 が戻ります。</p> <p>\$m ⇒【データ個数】&lt;out&gt;&lt;数値属性参照チャネル&gt; Mark 範囲のデータ個数が戻ります。</p> <p>&amp;k ⇒【マークメモ】&lt;out&gt;&lt;文字属性参照チャネル&gt; 開始 Index に付けられている Mark メモが戻ります。</p> <p>\$s ⇒【区間数】&lt;out&gt;&lt;数値属性参照チャネル&gt; 解析対象範囲から抽出した区間数が戻ります。</p>
備考	<ul style="list-style-type: none"> <li>■ 解析対象範囲に Mark が複数個付けられている必要があります。</li> <li>■ MARK が複数存在して居ない場合、及び MARK メモが数値でない場合は、開始 index に-1 が戻ります。</li> <li>■ MARK に対応した MARK メモは、全て数値で 0~99 範囲で記述されている必要があります。</li> <li>■ MARK メモ、0~99 の範囲内で、意味付けは 99:計測キャンセル、0:計測終了、1~98:試験コードとして使用されているものとします。</li> <li>■ 抽出される区間は、MARK メモ 1~98 で開始され、次の MARK メモが 0~98 までの区間です。</li> <li>■ 戻り値で収録データ処理範囲を制御する場合、制御を受ける演算処理ブロック内で記述する参照チャネルで、index の制御を禁止するチャネルは dcl exempt.ch 文で宣言してする必要があります。</li> </ul>
記述例	<pre>def ch_name \$1 "開始 index" def ch_name \$2 "データ個数" def ch_name \$3 "区間個数" call proc markmark_1 \$1,\$2,\$3 case \$1 &gt;= 0   proc calc{     index \$1 \$2 /*区間毎に繰り返し*/     proc calc_loop{       /* この演算処理ブロックに処理を記述*/     }calc_loop   }calc }</pre>
Script	<pre>1 /*-----*/ 2 call proc markmark_1 \$n,\$m,&amp;k,\$p 3 \$n ⇒【start_index】&lt;out&gt;&lt;数値属性参照チャネル&gt; 4 \$m ⇒【num of data】&lt;out&gt;&lt;数値属性参照チャネル&gt; 5 &amp;k ⇒【mark_memo】&lt;out&gt;&lt;文字属性参照チャネル&gt; 6 \$p ⇒【num_block】&lt;out&gt;&lt;数値属性参照チャネル&gt; 7 /*-----*/ 8 proc markmark_1{ 9   def inherit_ch \$7,\$8,&amp;9,\$10 10  def local_ch \$1,\$2,\$3,\$4,\$5,\$6 11  read num_mark \$1 /* MARK 個数取得 */ 12  case \$1 &gt; 1 13    proc mark_detect{ 14      read mark_memo 0 \$2 /* mark_memo 取得 */ 15      \$5 = EQU(SUM(AND(GTE(\$2,0),LTE(\$2,99))),LEN(\$2)) /* 無効 mark メモ検出*/ 16      case_true \$5 17        proc mark_memo_valid{ 18          read mark_pos 0 \$3 /* MARK 位置 Index 収録 */ 19          \$4 = ERC(0,\$1-2,\$2) /*区間先頭 mark メモ*/ 20          \$6 = AND(AND(GTE(\$4,1),LTE(\$4,98)),LTE(ERC(1,\$1-1,\$2),98)) /*区間除去*/ 21          \$4 = ZSP(\$6,\$4) /* MARK メモ配列を有効区間に再構成 */ 22          \$7 = ZSP(\$6,ERC(0,\$1-2,\$3)) /*区間開始 Index*/ 23          \$8 = ZSP(\$6,ERC(1,\$1-1,\$3))-7 /*データ個数*/ 24          assign &amp;9 = \$4(F0)</pre>

---

```
25         $10 = LEN($4)
26     }mark_memo_valid
27     case_false $5
28     proc mark_memo_invalid{
29         $7 = -1
30         $10 = 0
31     }mark_memo_invalid
32 }mark_detect
33 case $1 <= 1
34     proc not_detect{
35         $7 = -1
36         $10 = 0
37     }not_detect
38 }markmark_1
```

---

## 5. 頻度解析処理関連

### 5.1. 頻度処理結果・セル番号/中央値生成

ProcFile 名	SS0501.prc
機能	セルサイズ、セル個数と頻度解析処理手法が絶対値型か正負型かのフラグを与え、セル番号数列及びセル中央値数列を生成します。
呼び出し文	<code>call proc cell_number \$n,\$m,\$k,\$j,\$p</code>
引数	<p>\$n ⇒【セルサイズ】&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$m ⇒【セル個数】&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$k ⇒【領域】&lt;in&gt;&lt;数値属性参照チャネル&gt;          領域は、頻度解析した解析手法により異なります。          \$k : 0 絶対値型                Rainflow、PeakValley_ABS、PLS_MNS、Amplitude、Max、Min          \$k : 1 正負型                PeakValley、Peak、Valley、Min_Max、LevelCross、_UP、_DOWN、TimeRate          \$j ⇒【セル番号】&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$p ⇒【セル中央値】&lt;out&gt;&lt;数値属性参照チャネル&gt;</p>
備考	■ 領域指定で、正負型を指定した場合でセル個数が奇数の場合、+1し偶数化され生成されます。
記述例	<pre>\$1 "cell size:" = 0.5 \$2 "num_cell:" = 100 \$3 "flag:" = 1 def ch_name \$4 "cell number" def ch_name \$5 "cell value" call proc cell_number \$1,\$2,\$3,\$4,\$5</pre>
Script	<pre>1 /*-----*/ 2 call proc cell_number \$n,\$m,\$k,\$j,\$p 3 \$n ⇒【セルサイズ】&lt;in&gt;&lt;数値属性参照チャネル&gt; 4 \$m ⇒【セル個数】&lt;in&gt;&lt;数値属性参照チャネル&gt; 5 \$k ⇒【領域】&lt;in&gt;&lt;数値属性参照チャネル&gt; 6 頻度解析結果が絶対値型か正負型かを指定します。 7 \$k : 0 絶対値型 8 Rainflow、PeakValley_ABS、PLS_MNS、Amplitude、Max、Min 9 \$k : 1 正負型 10 PeakValley、Peak、Valley、Min_Max、LevelCross、_UP、_DOWN、TimeRate 11 \$j ⇒【セル番号】&lt;out&gt;&lt;数値属性参照チャネル&gt; 12 \$p ⇒【セル中央値】&lt;out&gt;&lt;数値属性参照チャネル&gt; 13 /*-----*/ 14 proc cell_number{ 15   def inherit_ch \$1,\$2,\$3,\$4,\$5 16   def local_ch \$6 17   case \$3 = 1 18     proc 全領域{ 19       \$6 = INT((\$2+0.5)/2) 20       assign \$4 = \$6&lt;1&gt; 21       \$5 = ACC(\$4)*\$1-\$1/2 22       \$4 = LNK(REV(0-ACC(\$4)),ACC(\$4)) /*cell number*/ 23       \$5 = LNK(0-REV(\$5),\$5) /* cell value*/ 24     }全領域 25   case \$3 = 0 26     proc 片領域{ 27       assign \$4 = \$2&lt;1&gt; 28       \$4 = ACC(\$4) /*cell number*/ 29       \$5 = \$4*\$1-\$1/2 /*cell value*/ 30     }片領域 31   }cell_number</pre>

## 5.2. トルク累積回転数頻度解析演算

ProcFile 名	SS0502.prc
機能	トルク累積回転数頻度解析演算を行います。 トルクデータの時間率頻度処理を行い、計数されるトルクデータの位置での回転数データを cell に加算し、結果を rpm 単位から秒単位に換算します。
呼び出し文	<b>call proc torque_rev \$n,\$m,\$f,\$j,\$p,\$k</b>
引数	\$n ⇒【セルサイズ】<in><数値属性参照チャネル> \$m ⇒【セル個数】<in><数値属性参照チャネル> \$f ⇒【領域】<in><数値属性参照チャネル> 時間率頻度解析対象のトルクデータの解析領域を指定します。 \$f = 0 : 全領域    \$f = 1 : 正領域のみ    \$f = -1 負領域のみ \$j ⇒【トルクデータ】<in><数値属性参照チャネル> \$p ⇒【回転数データ】<in><数値属性参照チャネル> \$k ⇒【累積回転頻度】<out><数値属性参照チャネル>
備考	<ul style="list-style-type: none"> <li>■ 回転数データの単位は rpm と見なして処理します。</li> <li>■ トルクデータの全領域を演算範囲とした場合で、セル個数が奇数で設定された場合、+1し偶数化されたセル個数となります。</li> </ul>
記述例	<pre> \$1 "cell_size:" = 0.5 \$2 "num_cell:" = 100 \$3 "ana_flag:" = 0 \$4 "torque:" = #1      /*トルクチャネル*/ \$5 "revo:rpm" = #2    /*回転数チャネル*/ def ch_name \$6 "累積回転頻度:sec" def ch_name \$7 "cell No." def ch_name \$8 "cell value" call proc torque_rev \$1,\$2,\$3,\$4,\$5,\$6 call proc cell_number \$1,\$2,,\$3,\$7,\$8 </pre>
Script	<pre> 1      /*----- 2      call torque_rev \$n,\$m,\$f,\$j,\$p,\$k 3      \$n ⇒【セルサイズ】&lt;in&gt;&lt;数値属性参照チャネル&gt; 4      \$m ⇒【セル個数】&lt;in&gt;&lt;数値属性参照チャネル&gt; 5      \$f ⇒【領域】&lt;in&gt;&lt;数値属性参照チャネル&gt; 6      \$f = 0 : 全領域  \$f = 1 : 正領域  \$f = -1 負領域 7      \$j ⇒【トルクデータ】&lt;in&gt;&lt;数値属性参照チャネル&gt; 8      \$p ⇒【回転数データ】&lt;in&gt;&lt;数値属性参照チャネル&gt; 9      \$k ⇒【累積回転頻度】&lt;out&gt;&lt;数値属性参照チャネル&gt; 10     -----*/ 11     proc torque_rev{ 12         def inherit_ch \$1,\$2,\$3,\$4,\$5,\$6 13         def local_ch \$7 14         case \$3 = 0 15             proc 全領域{ 16                 \$7 = INT((\$2+0.5)/2)*2 17                 \$6 = TRX(\$1,\$7,\$4,\$5.5)/60*PRD() 18             }全領域 19         case \$3 = 1 20             proc 正領域{ 21                 \$7 = \$2*2 22                 \$6 = TRX(\$1,\$7,\$4,\$5.5)/60*PRD() 23                 \$6 = ERC(\$7/2-1,\$7-1,\$6) 24             }正領域 25         case \$3 = -1 26             proc 負領域{ 27                 \$7 = \$2*2 28                 \$6 = TRX(\$1,\$7,\$4,\$5.5)/60*PRD() 29                 \$6 = REV(ERC(0,\$7/2-1,\$6)) 30             }負領域 31     }torque_rev </pre>

### 5.3. 範囲指定時間率頻度解析

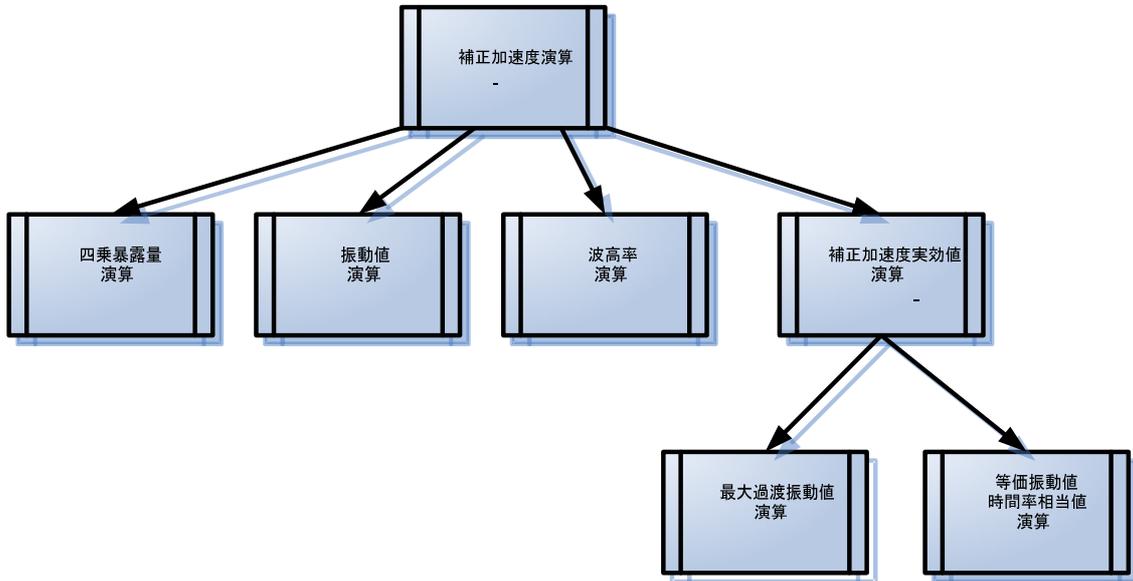
ProcFile 名	SS0503.prc
機能	時間率頻度解析を行う上下限範囲を指定して時間率頻度を求めます。
呼び出し文	<b>call proc ScopeTimerate \$1,\$2,\$3,\$4,\$5,\$6,\$7</b>
引数	<p>\$1:【セルサイズ】&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$2:【上限値】&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$3:【下限値】&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$4:【解析対象チャネル】&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$5:【度数】&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$6:【セル中央値】&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$7:【セル番号】&lt;out&gt;&lt;数値属性参照チャネル&gt;</p>
備考	<ul style="list-style-type: none"> <li>■ セル個数は、与えた上限値、下限値をセルサイズで刻んで自動的に設定します。</li> <li>■ セルの上限値は、未満を意味し、上限値自身は解析範囲に含まれません。</li> <li>■ 結果のセル個数を知る場合は、演算関数 LEN(\$5)を使用して求めます。</li> </ul>
記述例	<pre> \$1 "cell_size:" = 0.02 \$2 "UpperLimit:" = 10 \$3 "LowerLimit:" = 2.5 \$4 = #1 def ch_name \$5 "度数" def ch_name \$6 "cell 中央値" def ch_name \$7 "cell.No." call proc ScopeTimeRate \$1,\$2,\$3,\$4,\$5,\$6,\$7 </pre>
Script	<pre> 1 /*-----*/ 2 \$1:【セルサイズ】&lt;in&gt;&lt;数値属性参照チャネル&gt; 3 \$2:【上限値】&lt;in&gt;&lt;数値属性参照チャネル&gt; 4 \$3:【下限値】&lt;in&gt;&lt;数値属性参照チャネル&gt; 5 \$4:【解析対象チャネル】&lt;in&gt;&lt;数値属性参照チャネル&gt; 6 \$5:【度数】&lt;out&gt;&lt;数値属性参照チャネル&gt; 7 \$6:【セル中央値】&lt;out&gt;&lt;数値属性参照チャネル&gt; 8 \$7:【セル番号】&lt;out&gt;&lt;数値属性参照チャネル&gt; 9 /*-----*/ 10 proc ScopeTimeRate[ 11     def inherit_ch \$[1,7] 12     def local_ch \$20,\$22,\$23,\$24 13     \$20 = (\$2-\$3)/2 14     case \$20 &gt; 0 15         proc valid[ 16             \$23 = ((\$2-\$3)/\$1)*2+2 17             \$22 = \$4-\$2 /* オフセット処理*/ 18             \$5 = TRC(\$1,\$23,\$22) 19             \$5 = ERC(1,\$23/2-1,\$5) 20             \$24 = \$23/2-1 21             assign \$7 = \$24&lt;1&gt; 22             \$7 = ACC(\$7) 23             \$6 = \$7*\$1-\$1/2+\$3 24         ]valid 25     case \$20 &lt;= 0 26         proc invalid[ 27             \$5 = 0 28             \$7 = 0 29             \$6 = 0 30         ]invalid 31     ]ScopeTimerate </pre>

## 5.4. 基準チャンネル範囲指定時間率頻度解析

ProcFile 名	SS0504.prc
機能	基準チャンネルの上下限指定範囲のデータに対応した解析チャンネルの時間率頻度を求めます。
呼び出し文	<b>call proc RefChTimeRate \$1,\$2,\$3,\$4,\$5,\$6,\$7</b>
引数	\$1:【上限値】<in><数値属性参照チャンネル> \$2:【下限値】<in><数値属性参照チャンネル> \$3:【基準チャンネル】<in><数値属性参照チャンネル> \$4:【セルサイズ】<in><数値属性参照チャンネル> \$5:【セル個数】<inj><数値属性参照チャンネル> \$6:【解析対象チャンネル】<in><数値属性参照チャンネル> \$7:【度数】<out><数値属性参照チャンネル>
備考	■ 特になし
記述例	\$1 "UpperLimit." = 250 \$2 "LowerLimit." = 80 \$3 = #1 /*基準チャンネル*/ \$4 "cell_size." = 0.02 \$5 "cals:" = 120 \$6 = #2 /*解析対象チャンネル*/ def ch_name \$7 "度数" call proc ScopeTimeRate \$1,\$2,\$3,\$4,\$5,\$6,\$7 \$8 "領域:" = 1 def ch_name \$9 "cell.No." def ch_name \$10 "cell_value" call proc cell_number \$4,\$5,\$8,\$9,\$10
Script	1 /*----- 2 \$1:【上限値】<in><数値属性参照チャンネル> 3 \$2:【下限値】<in><数値属性参照チャンネル> 4 \$3:【基準チャンネル】<in><数値属性参照チャンネル> 5 \$4:【セルサイズ】<in><数値属性参照チャンネル> 6 \$5:【セル個数】<inj><数値属性参照チャンネル> 7 \$6:【解析対象チャンネル】<in><数値属性参照チャンネル> 8 \$7:【度数】<out><数値属性参照チャンネル> 9 -----*/ 10 proc RefChTimeRate[ 11 def inherit_ch \$1,\$2,\$3,\$4,\$5,\$6,\$7 12 def local_ch \$10 13 \$10 = AND(NOT(GTE(\$3,\$1)),GTE(\$3,\$2)) 14 \$6 = TRC(\$4,\$5,ZSP(\$10,\$6)) 15 ]RefChTimeRate

## 6. 振動暴露解析処理関連

サブルーチン ツリー



### 6.1. 全身振動暴露解析・補正加速度演算

ProcFile 名	SS0601.prc
機能	X,Y,Z 軸加速度のそれぞれに ISO2631-1 補正フィルタ処理を行い、補正加速度に変換します。 X,Y 軸の信号名には元の信号名に "Wd" を付加し、Z 軸の信号名には "Wk" を付加します。
呼び出し文	call proc 補正加速度 \$1,\$2,\$3
引数	\$1:【座位立位 X 軸加速度:m/s <sup>2</sup> 】<in/out><数値属性参照チャネル> \$2:【座位立位 Y 軸加速度:m/s <sup>2</sup> 】<in/out><数値属性参照チャネル> \$3:【座位立位 Z 軸加速度:m/s <sup>2</sup> 】<in/out><数値属性参照チャネル>
備考	<ul style="list-style-type: none"> <li>■ 収録チャネルを直接引数として記述出来ません。数値属性参照チャネルに代入して引数として下さい。</li> <li>■ X,Y 軸は Wd、Z 軸は Wk のウェイトイング・フィルタを掛けます。</li> <li>■ チャネルの単位がチェックされ"G"または"g"の場合、m/s<sup>2</sup>に変換します。</li> </ul>
記述例	<pre> read ch_name &amp;1 &amp;2 #1 def ch_name \$1 &amp;1 def ch_unit \$1 &amp;2 \$1 = #1 read ch_name &amp;1 &amp;2 #2 def ch_name \$2 &amp;1 def ch_unit \$2 &amp;2 \$2 = #2 read ch_name &amp;1 &amp;2 #3 def ch_name \$3 &amp;1 def ch_unit \$3 &amp;2 \$3 = #3 call proc 補正加速度 \$1,\$2,\$3 </pre>
Script	<pre> 1 /*----- 2 \$1:【座位立位 X 軸加速度:m/s<sup>2</sup>】&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 3 \$2:【座位立位 Y 軸加速度:m/s<sup>2</sup>】&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 4 \$3:【座位立位 Z 軸加速度:m/s<sup>2</sup>】&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 5 -----*/ 6 proc 補正加速度[ 7     def inherit_ch \$1,\$2,\$3 8     def local_ch &amp;1,&amp;2,&amp;3 9     assign &amp;3 = "G","g" 10    read ch_name &amp;1 &amp;2 \$1 11    call proc unit_check &amp;3,&amp;2,\$1 </pre>

---

```
12     assign &1 = &1|"_Wd:"|&2
13     def ch_name $1 &1
14     read ch_name &1 &2 $2
15     call proc_unit_check &3,&2,$2
16     assign &1 = &1|"_Wd:"|&2
17     def ch_name $2 &1
18     read ch_name &1 &2 $3
19     call proc_unit_check &3,&2,$3
20     assign &1 = &1|"_Wk:"|&2
21     def ch_name $3 &1
22     $1 = WBD($1)
23     $2 = WBD($2)
24     $3 = WBK($3)
25 } 補正加速度
26 proc unit_check{
27     def inherit_ch &3,&2,$4
28     def local_ch $1
29     char find &3 &2 1 $1
30     $1 = SUM($1)
31     case $1 > 0
32     proc unit_change{
33         assign &2 = "m/s^2"
34         $4 = $4*9.80665
35     }unit_change
36 }unit_check
```

---

## 6.2. 全身振動暴露解析・補正加速度実効値演算

ProcFile 名	SS0602.prc
機能	X,Y,Z 軸補正加速度に与えられた時定数で移動実効値演算を行い、各軸の補正加速度実効値及び、合成軸加速度実効値を求めます。
呼び出し文	<b>call proc 補正加速度実効値 \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8</b>
引数	<p>\$1:【積分時定数】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$2:【X 軸補正加速度:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$3:【Y 軸補正加速度:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$4:【Z 軸補正加速度:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$5:【X 軸補正加速度実効値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;            \$6:【Y 軸補正加速度実効値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;            \$7:【Z 軸補正加速度実効値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;            \$8:【合成軸補正加速度実効値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;</p>
備考	<ul style="list-style-type: none"> <li>■ 積分時定数が 0 又は与える補正加速度の収録時間より長い場合、及び与える各軸の加速度要素数が異なる場合は、解析範囲全体の実効値として戻ります。</li> <li>■ 全体の実効値として求められたか否かの確認は、戻り補正加速度実効値の要素数を LEN(X)関数を使用して求め、その値が 1 個の場合は、解析範囲全体の実効値が求められた事になります。</li> </ul>
記述例	<pre> \$1"X 加速度:m/s^2" = #1 \$2"Y 加速度:m/s^2" = #2 \$3"Z 加速度:m/s^2" = #3 call proc 補正加速度\$1,\$2,\$3 \$4"時定数:sec" = 0.25 def ch_name \$5 "X 軸補正加速度実効値:m/s^2" def ch_name \$6 "Y 軸補正加速度実効値:m/s^2" def ch_name \$7 "Z 軸補正加速度実効値:m/s^2" def ch_name \$8 "合成軸補正加速度実効値:m/s^2" call proc 補正加速度実効値 \$4,\$1,\$2,\$3,\$5,\$6,\$7,\$8           </pre>
Script	<pre> 1 /*-----*/ 2 \$1:【時定数:sec】&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 3 \$2:【X 軸補正加速度:m/s^2】&lt;in&gt;&lt;数値属性参照チャネル&gt; 4 \$3:【Y 軸補正加速度:m/s^2】&lt;in&gt;&lt;数値属性参照チャネル&gt; 5 \$4:【Z 軸補正加速度:m/s^2】&lt;in&gt;&lt;数値属性参照チャネル&gt; 6 \$5:【X 軸補正加速度実効値:m/s^2】&lt;out&gt;&lt;数値属性参照チャネル&gt; 7 \$6:【Y 軸補正加速度実効値:m/s^2】&lt;out&gt;&lt;数値属性参照チャネル&gt; 8 \$7:【Z 軸補正加速度実効値:m/s^2】&lt;out&gt;&lt;数値属性参照チャネル&gt; 9 \$8:【合成補正加速度実効値:m/s^2】&lt;out&gt;&lt;数値属性参照チャネル&gt; 10 -----*/ 11 proc 補正加速度実効値[ 12     def inherit_ch \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8 13     def local_ch \$9,\$10 14     \$1 = ABS(\$1) 15     \$9 = AND(EQU(LEN(\$2),LEN(\$3)),EQU(LEN(\$3),LEN(\$4))) 16     \$9 = NOT(LTE(\$9*\$1,0)) 17     \$10 = LEN(\$2)*PRD() 18     case \$1 &gt; \$10 19     proc timeconst_invalid[ 20         \$9 = 0 21     ]timeconst_invalid 22     case_true \$9 23     proc move_effective[ 24         \$5 = RRV(\$1,\$2) 25         \$6 = RRV(\$1,\$3) 26         \$7 = RRV(\$1,\$4) 27     ]move_effective 28     case_false \$9 29     proc allover[ 30         \$5 = EFF(\$2) 31         \$6 = EFF(\$3) 32         \$7 = EFF(\$4) 33     ]allover 34     \$8 = SQR((1.4*\$5)^2+(1.4*\$6)^2+\$7^2)           </pre>



### 6.3. 全身振動暴露解析・区間最大過渡振動値演算

ProcFile 名	SS0603.prc
機能	X,Y,Z 軸及び合成軸の補正加速度実効値から与えた区間毎の最大過渡振動値を演算します。
呼び出し文	<b>call proc 区間最大過渡振動値 \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8,\$9,\$10</b>
引数	<p>\$1:【開始 index】&lt;in/out&gt;&lt;数値属性参照チャネル&gt;          解析開始 index 列を記述します。解析範囲は、ここで指定した index から\$2 で記述したデータ個数範囲を一回の解析範囲として、\$1 の要素数分行われます。</p> <p>\$2:【解析対象データ個数】&lt;in&gt;&lt;数値属性参照チャネル&gt;          データ個数の要素数は、開始 Index の要素数と等しくない場合は、index0 のみ繰り返して参照されます。</p> <p>\$3:【X 軸補正加速度実効値:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$4:【Y 軸補正加速度実効値:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$5:【Z 軸補正加速度実効値:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$6:【合成軸補正加速度実効値:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$7:【X 軸最大過渡振動値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$8:【Y 軸最大過渡振動値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$9:【Z 軸最大過渡振動値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$10:【合成軸最大過渡振動値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;</p>
備考	<ul style="list-style-type: none"> <li>■ 演算データ個数に 0 が指定された場合、対応する開始 index から終端までを演算範囲に置き換えます。</li> <li>■ 与えた各軸の補正加速度の個数は等しい必要があります。等しくない場合は、何も演算せず、開始 index に -1 が戻ります。</li> </ul>
記述例	<pre> \$1"X 加速度:m/s^2" = #1 \$2 "Y 加速度:m/s^2" = #2 \$3 "Z 加速度:m/s^2" = #3 call proc 補正加速度 1\$1,\$2,\$3 \$4 "時定数:sec" = 0.25 def ch_name \$5 "X 軸補正加速度実効値:m/s^2" def ch_name \$6 "Y 軸補正加速度実効値:m/s^2" def ch_name \$7 "Z 軸補正加速度実効値:m/s^2" def ch_name \$8 "合成補正加速度実効値:m/s^2" call proc 補正加速度実効値 1\$4,\$1,\$2,\$3,\$5,\$6,\$7,\$8 def ch_name \$9 "開始 index" def ch_name \$10 "データ個数" def ch_name \$22 "区間個数" call proc markmark_1 \$9,\$10,\$22 /*mark-mark 間を解析区間とする*/ \$9 = LNK(0,\$9) /*最初の解析範囲は全範囲を指定する*/ \$10 = LNK(0,\$10) def ch_name \$11 "X 軸最大過渡振動値:m/s^2" def ch_name \$12 "Y 軸最大過渡振動値:m/s^2" def ch_name \$13 "Z 軸最大過渡振動値:m/s^2" def ch_name \$14 "合成軸最大過渡振動値:m/s^2" call proc 区間最大過渡振動値 \$9,\$10,\$5,\$6,\$7,\$8,\$11,\$12,\$13,\$14         </pre>
Script	<pre> 1 /*-----*/ 2 \$1:【開始 index】&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 3 \$2:【データ個数】&lt;in&gt;&lt;数値属性参照チャネル&gt; 4 \$3:【X 軸振動値:m/s^2】&lt;in&gt;&lt;数値属性参照チャネル&gt; 5 \$4:【Y 軸振動値:m/s^2】&lt;in&gt;&lt;数値属性参照チャネル&gt; 6 \$5:【Z 軸振動値:m/s^2】&lt;in&gt;&lt;数値属性参照チャネル&gt; 7 \$6:【合成軸振動値:m/s^2】&lt;in&gt;&lt;数値属性参照チャネル&gt; 8 \$7:【X 軸最大過渡振動値:m/s^2】&lt;out&gt;&lt;数値属性参照チャネル&gt; 9 \$8:【Y 軸最大過渡振動値:m/s^2】&lt;out&gt;&lt;数値属性参照チャネル&gt; 10 \$9:【Z 軸最大過渡振動値:m/s^2】&lt;out&gt;&lt;数値属性参照チャネル&gt; 11 \$10:【合成軸最大過渡振動値:m/s^2】&lt;out&gt;&lt;数値属性参照チャネル&gt; 12 -----*/ 13 proc 区間最大過渡振動値{ 14     def inherit_ch \$[1,10] 15     def local_ch \$11,\$12,\$13,\$14,\$15,\$30 16     \$12 = EQU(LEN(\$3)+LEN(\$4)+LEN(\$5),LEN(\$6)*3)-1 17     case \$12 = 0 18         proc len_valid{ 19             \$11 = EQU(LEN(\$1),LEN(\$2)) 20         case_false \$11         </pre>

---

```

21      proc count_mdf{
22          $2 = $1*0+$2(0)
23      }count_mdf
24      $30 = $2+EQU($2,0)*(LEN($3)-$1-1) /* ゼロの場合、終端までの個数に置換*/
25      $30 = $1+$30
26      $15 = 0
27      $14 = LEN($1)
28      assign $7 = $14<0>
29      assign $8 = $14<0>
30      assign $9 = $14<0>
31      assign $10 = $14<0>
32      repeat_case $15 < $14
33          proc calc_loop{
34              $7($15) = MAX(ERC($1($15),$30($15),$3))
35              $8($15) = MAX(ERC($1($15),$30($15),$4))
36              $9($15) = MAX(ERC($1($15),$30($15),$5))
37              $10($15) = MAX(ERC($1($15),$30($15),$6))
38              $15 = $15+1
39          }calc_loop
40      }len_valid
41      case $12 = -1
42          proc len_invalid{
43              $1 = -1
44          }len_invalid
45      }区間最大過渡振動値

```

---

## 6.4. 全身振動暴露解析・区間振動値演算

ProcFile 名	SS0604.prc
機能	X,Y,Z 軸の補正加速度から与えた区間毎の振動値(実効値)を求めます。
呼び出し文	<b>call proc 区間振動値 \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8,\$9</b>
引数	<p>\$1:【開始 index】&lt;in/out&gt;&lt;数値属性参照チャネル&gt;            解析開始 index 列を記述します。解析範囲は、ここで指定した index から\$2 で記述したデータ個数範囲を一回の解析範囲として、\$1 の要素数分行われます。</p> <p>\$2:【解析対象データ個数】&lt;in&gt;&lt;数値属性参照チャネル&gt;            データ個数の要素数は、開始 Index の要素数と等しくない場合は、index0 のみ繰り返して参照されます。</p> <p>\$3:【X 軸補正加速度:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$4:【Y 軸補正加速度:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$5:【Z 軸補正加速度:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$6:【区間 X 軸振動値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;            \$7:【区間 Y 軸振動値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;            \$8:【区間 Z 軸振動値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;            \$9:【区間合成軸振動値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;</p>
備考	<ul style="list-style-type: none"> <li>■ 振動値は補正加速度実効値と等価ですが、指定された範囲内で 1 個となる点が異なります。</li> <li>■ 演算データ個数に 0 が指定された場合、対応する開始 index から終端までを演算範囲に置き換えます。</li> <li>■ 与えた各軸の補正加速度の個数は等しい必要があります。等しくない場合は、何も演算せず、開始 index に -1 が戻ります。</li> </ul>
記述例	<pre> \$1"X 加速度:m/s^2" = #1 \$2 "Y 加速度:m/s^2" = #2 \$3 "Z 加速度:m/s^2" = #3 call proc 補正加速度\$1,\$2,\$3 def ch_name \$9 "開始 index" def ch_name \$10 "データ個数" def ch_name \$22 "区間個数" call proc markmark_1 \$9,\$10,\$22 /*mark-mark 間を解析区間とする*/ \$9 = LNK(0,\$9) /*最初の解析範囲は全範囲を指定する*/ \$10 = LNK(0,\$10) def ch_name \$15 "区間 X 軸振動値:m/s^2" def ch_name \$16 "区間 Y 軸振動値:m/s^2" def ch_name \$17 "区間 Z 軸振動値:m/s^2" def ch_name \$18 "区間合成軸振動値:m/s^2" call proc 区間振動値 \$9,\$10,\$1,\$2,\$3,\$15,\$16,\$17,\$18           </pre>
Script	<pre> 1 /* 2 \$1:【開始 index】&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 3 \$2:【データ個数】&lt;in&gt;&lt;数値属性参照チャネル&gt; 4 \$3:【X 軸補正加速度:m/s^2】&lt;in&gt;&lt;数値属性参照チャネル&gt; 5 \$4:【Y 軸補正加速度:m/s^2】&lt;in&gt;&lt;数値属性参照チャネル&gt; 6 \$5:【Z 軸補正加速度:m/s^2】&lt;in&gt;&lt;数値属性参照チャネル&gt; 7 \$6:【区間 X 軸振動値:m/s^2】&lt;out&gt;&lt;数値属性参照チャネル&gt; 8 \$7:【区間 Y 軸振動値:m/s^2】&lt;out&gt;&lt;数値属性参照チャネル&gt; 9 \$8:【区間 Z 軸振動値:m/s^2】&lt;out&gt;&lt;数値属性参照チャネル&gt; 10 \$9:【区間合成軸振動値:m/s^2】&lt;out&gt;&lt;数値属性参照チャネル&gt; 11 -----*/ 12 proc 区間振動値[ 13     def inherit_ch \$[1,9] 14     def local_ch \$11,\$12,\$13,\$14,\$15,\$30 15     \$12 = EQU(LEN(\$3)+LEN(\$4),LEN(\$5)*2)-1 16     case \$12 = 0 17         proc len_valid[ 18             \$11 = EQU(LEN(\$1),LEN(\$2)) 19             case false \$11 20                 proc count_mdff[ 21                     \$2 = \$1*0+\$2(0) 22                 ]count_mdff 23                 \$30 = \$2+EQU(\$2,0)*(LEN(\$3)-\$1-1) /* ゼロの場合、終端までの個数に置換*/ 24                 \$30 = \$1+\$30 25                 \$15 = 0 26                 \$14 = LEN(\$1)           </pre>

---

```
27      assign $6 = $14<0>
28      assign $7 = $14<0>
29      assign $8 = $14<0>
30      assign $9 = $14<0>
31      repeat_case $15 < $14
32          proc calc_loop{
33              $6($15) = EFF(ERC($1($15),$30($15),$3))
34              $7($15) = EFF(ERC($1($15),$30($15),$4))
35              $8($15) = EFF(ERC($1($15),$30($15),$5))
36              $9($15) = SQR((1.4*$6($15))^2+(1.4*$7($15))^2+$8($15)^2)
37              $15 = $15+1
38          }calc_loop
39      }len_valid
40      case $12 = -1
41          proc len_invalid{
42              $1 = -1
43          }len_invalid
44      }区間振動値
```

---

## 6.5. 全身振動暴露解析・区間四乗暴露量値演算

ProcFile 名	SS0605.prc
機能	X,Y,Z 軸の補正加速度から与えた区間毎の四乗暴露量値を求めます。
呼び出し文	<b>call proc 区間四乗暴露量 \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8</b>
引数	<p>\$1:【開始 index】&lt;in/out&gt;&lt;数値属性参照チャネル&gt;            解析開始 index 列を記述します。解析範囲は、ここで指定した index から\$2 で記述したデータ個数範囲を一回の解析範囲として、\$1 の要素数分行われます。</p> <p>\$2:【解析対象データ個数】&lt;in&gt;&lt;数値属性参照チャネル&gt;            データ個数の要素数は、開始 Index の要素数と等しくない場合は、index0 のみ繰り返して参照されます。</p> <p>\$3:【X 軸補正加速度:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$4:【Y 軸補正加速度:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$5:【Z 軸補正加速度:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$6:【区間 X 軸四乗暴露量値:m/s<sup>1.75</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;            \$7:【区間 Y 軸四乗暴露量値:m/s<sup>1.75</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;            \$8:【区間 Z 軸四乗暴露量値:m/s<sup>1.75</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;</p>
備考	<ul style="list-style-type: none"> <li>■ 演算データ個数に 0 が指定された場合、対応する開始 index から終端までを演算範囲に置き換えます。</li> <li>■ 与えた各軸の補正加速度の個数は等しい必要があります。等しくない場合は、何も演算せず、開始 index に -1 が戻ります。</li> </ul>
記述例	<pre> \$1"X 加速度:m/s^2" = #1 \$2 "Y 加速度:m/s^2" = #2 \$3 "Z 加速度:m/s^2" = #3 call proc 補正加速度 \$1,\$2,\$3 def ch_name \$9 "開始 index" def ch_name \$10 "データ個数" def ch_name \$22 "区間個数" call proc markmark_1 \$9,\$10,\$22 /*mark-mark 間を解析区間とする*/ \$9 = LNK(0,\$9) /*最初の解析範囲は全範囲を指定する*/ \$10 = LNK(0,\$10) def ch_name \$19 "区間 X 軸四乗暴露量値:m/s^2" def ch_name \$20 "区間 Y 軸四乗暴露量値:m/s^2" def ch_name \$21 "区間 Z 軸四乗暴露量値:m/s^2" call proc 区間四乗暴露量 \$9,\$10,\$1,\$2,\$3,\$19,\$20,\$21           </pre>
Script	<pre> 1 /* 2 \$1:【開始 index】&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 3 \$2:【データ個数】&lt;in&gt;&lt;数値属性参照チャネル&gt; 4 \$3:【X 軸補正加速度:m/s^2】&lt;in&gt;&lt;数値属性参照チャネル&gt; 5 \$4:【Y 軸補正加速度:m/s^2】&lt;in&gt;&lt;数値属性参照チャネル&gt; 6 \$5:【Z 軸補正加速度:m/s^2】&lt;in&gt;&lt;数値属性参照チャネル&gt; 7 \$6:【区間 X 軸四乗暴露量値:m/s^1.75】&lt;out&gt;&lt;数値属性参照チャネル&gt; 8 \$7:【区間 Y 軸四乗暴露量値:m/s^1.75】&lt;out&gt;&lt;数値属性参照チャネル&gt; 9 \$8:【区間 Z 軸四乗暴露量値:m/s^1.75】&lt;out&gt;&lt;数値属性参照チャネル&gt; 10 */ 11 proc 区間四乗暴露量[ 12   def inherit_ch \$[1,8] 13   def local_ch \$11,\$12,\$13,\$14,\$15,\$30 14   \$12 = EQU(LEN(\$3)+LEN(\$4),LEN(\$5)*2)-1 15   case \$12 = 0 16     proc len_valid[ 17       \$11 = EQU(LEN(\$1),LEN(\$2)) 18       case_false \$11 19         proc count_mdf[ 20           \$2 = \$1*0+\$2(0) 21         ]count_mdf 22       \$30 = \$2+EQU(\$2,0)*(LEN(\$3)-\$1-1) /* ゼロの場合、終端までの個数に置換*/ 23       \$30 = \$1+\$30 24       \$15 = 0 25       \$14 = LEN(\$1) 26       assign \$6 = \$14&lt;0&gt; 27       assign \$7 = \$14&lt;0&gt; 28       assign \$8 = \$14&lt;0&gt; 29       repeat_case \$15 &lt; \$14           </pre>

---

```
30         proc calc_loop{
31             $6($15) = SUM(ERC($1($15),$30($15),$3)^4)^0.25
32             $7($15) = SUM(ERC($1($15),$30($15),$4)^4)^0.25
33             $8($15) = SUM(ERC($1($15),$30($15),$5)^4)^0.25
34             $15 = $15+1
35         }calc_loop
36     }len_valid
37     case $12 = -1
38     proc len_invalid{
39         $1 = -1
40     }len_invalid
41 } 区間四乗暴露量
```

---

## 6.6. 全身振動暴露解析・時間率頻度解析

ProcFile 名	SS0606.prc
機能	補正加速度実効値の軸毎に時間率頻度解析を行います。
呼び出し文	<b>call proc 振動時間率頻度 \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8,\$9,\$10</b>
引数	<p>\$1:[X 軸補正加速度実効値:m/s<sup>2</sup>]&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$2:[Y 軸補正加速度実効値:m/s<sup>2</sup>]&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$3:[Z 軸補正加速度実効値:m/s<sup>2</sup>]&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$4:[合成軸補正加速度実効値:m/s<sup>2</sup>]&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$5:[X 軸時間率頻度:sec]&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$6:[Y 軸時間率頻度:sec]&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$7:[Z 軸時間率頻度:sec]&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$8:[合成軸時間率頻度:sec]&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$9:[cell 中央値:m/s<sup>2</sup>]&lt;out&gt;&lt;数値属性参照チャネル&gt;</p>
備考	<ul style="list-style-type: none"> <li>■ セルサイズ及びセル個数は、Script 内で固定しています。セルサイズは 0.005m/s<sup>2</sup>、セル個数は 1250 です。</li> <li>■ 最大計数範囲は、0~6.25m/s<sup>2</sup> です。溢れた値は、最上位に計数されます。</li> <li>■ セルサイズ及び個数を修正する場合、line15、16 を修正して使用下さい。</li> </ul>
記述例	<pre> \$1 "X 加速度:m/s^2" = #1 \$2 "Y 加速度:m/s^2" = #2 \$3 "Z 加速度:m/s^2" = #3 call proc 補正加速度 \$1,\$2,\$3 \$4 "時定数:sec" = 0.25 def ch_name \$5 "X 軸補正加速度実効値:m/s^2" def ch_name \$6 "Y 軸補正加速度実効値:m/s^2" def ch_name \$7 "Z 軸補正加速度実効値:m/s^2" def ch_name \$8 "合成軸補正加速度実効値:m/s^2" call proc 補正加速度実効値 \$4,\$1,\$2,\$3,\$5,\$6,\$7,\$8 def ch_name \$22 "X 軸度数:sec" def ch_name \$23 "Y 軸度数:sec" def ch_name \$24 "Z 軸度数:sec" def ch_name \$25 "セル中央値" call proc 振動時間率頻度 \$5,\$6,\$7,\$8,\$22,\$23,\$24,\$25,\$26         </pre>
Script	<pre> 1 /*-----*/ 2 \$1:[X 軸補正加速度実効値:m/s^2]&lt;in&gt;&lt;数値属性参照チャネル&gt; 3 \$2:[Y 軸補正加速度実効値:m/s^2]&lt;in&gt;&lt;数値属性参照チャネル&gt; 4 \$3:[Z 軸補正加速度実効値:m/s^2]&lt;in&gt;&lt;数値属性参照チャネル&gt; 5 \$4:[合成軸補正加速度実効値:m/s^2]&lt;in&gt;&lt;数値属性参照チャネル&gt; 6 \$5:[X 軸時間率頻度:sec]&lt;out&gt;&lt;数値属性参照チャネル&gt; 7 \$6:[Y 軸時間率頻度:sec]&lt;out&gt;&lt;数値属性参照チャネル&gt; 8 \$7:[Z 軸時間率頻度:sec]&lt;out&gt;&lt;数値属性参照チャネル&gt; 9 \$8:[合成軸時間率頻度:sec]&lt;out&gt;&lt;数値属性参照チャネル&gt; 10 \$9:[cell 中央値:m/s^2]&lt;out&gt;&lt;数値属性参照チャネル&gt; 11 -----*/ 12 proc 振動時間率頻度[ 13     def inherit_ch \$[1,9] 14     def local_ch \$11,\$12,\$13,\$14,\$15 15     \$11 "cell_size:m/s^2" = 0.005 16     \$12 "class:" = 2500 17     \$5 = ERC(1250,2499,TRC(\$11,\$12,\$1)*PRD()) 18     \$6 = ERC(1250,2499,TRC(\$11,\$12,\$2)*PRD()) 19     \$7 = ERC(1250,2499,TRC(\$11,\$12,\$3)*PRD()) 20     \$8 = ERC(1250,2499,TRC(\$11,\$12,\$4)*PRD()) 21     assign \$9 = 1250&lt;\$11&gt; 22     \$9 = ACC(\$9)-\$11/2 23 ]振動時間率頻度         </pre>

## 6.7. 全身振動暴露解析・区間等価振動値演算

ProcFile 名	SS0607.pro
機能	指定した区間毎に補正加速度実効値の時間率頻度解析を行い、その結果から等価振動値を求めます。
呼び出し文	<b>call proc 区間等価振動値 \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8,\$9,\$10</b>
引数	<p>\$1:【開始 index】&lt;in/out&gt;&lt;数値属性参照チャネル&gt;          解析開始 index 列を記述します。解析範囲は、ここで指定した index から\$2 で記述したデータ個数範囲を一回の解析範囲として、\$1 の要素数分行われます。</p> <p>\$2:【解析対象データ個数】&lt;in&gt;&lt;数値属性参照チャネル&gt;          データ個数の要素数は、開始 Index の要素数と等しくない場合は、index0 のみ繰り返して参照されます。</p> <p>\$3:【X 軸補正加速度実効値:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$4:【Y 軸補正加速度実効値:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$5:【Z 軸補正加速度実効値:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$6:【合成軸補正加速度実効値:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$7:【X 軸等価振動値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$8:【Y 軸等価振動値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$9:【Z 軸等価振動値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$10:【合成軸等価振動値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;</p>
備考	<ul style="list-style-type: none"> <li>■ 本 Subroutine は、内部にて”振動時間率頻度”を呼び出し使用しますので、timerate.2631.1.pro も同時に Script に挿入して下さい。</li> <li>■ 演算データ個数に 0 が指定された場合、対応する開始 index から終端までを演算範囲に置き換えます。</li> <li>■ 与えた各軸の振動値の個数は等しい必要があります。等しくない場合は、何も演算せず、開始 index に-1 が戻ります。</li> </ul>
記述例	<pre> \$1"X 加速度:m/s^2" = #1 \$2"Y 加速度:m/s^2" = #2 \$3"Z 加速度:m/s^2" = #3 call proc 補正加速度 \$1,\$2,\$3 \$4"時定数:sec" = 0.25 def ch_name \$5 "X 軸補正加速度実効値:m/s^2" def ch_name \$6 "Y 軸補正加速度実効値:m/s^2" def ch_name \$7 "Z 軸補正加速度実効値:m/s^2" def ch_name \$8 "合成軸補正加速度実効値:m/s^2" call proc 補正加速度実効値 \$4,\$1,\$2,\$3,\$5,\$6,\$7,\$8 def ch_name \$9 "開始 index" def ch_name \$10 "データ個数" def ch_name \$11 "区間個数" call proc markmark_1 \$9,\$10,&amp;11,11 /*mark-mark 間を解析区間とする*/ \$9 = LNK(0,\$9) /*最初の解析範囲は全範囲を指定する*/ \$10 = LNK(0,\$10) def ch_name \$12 "X 軸等価振動値:m/s^2" def ch_name \$13 "Y 軸等価振動値:m/s^2" def ch_name \$14 "Z 軸等価振動値:m/s^2" def ch_name \$15 "合成軸等価振動値:m/s^2" call proc 区間等価振動値 \$9,\$10,\$5,\$6,\$7,\$8,\$12,\$13,\$14,\$15 </pre>
Script	<pre> 1 /*- 2 \$1:【開始 index】&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 3 \$2:【解析対象データ個数】&lt;in&gt;&lt;数値属性参照チャネル&gt; 4 \$3:【X 軸補正加速度実効値:m/s^2】&lt;in&gt;&lt;数値属性参照チャネル&gt; 5 \$4:【Y 軸補正加速度実効値:m/s^2】&lt;in&gt;&lt;数値属性参照チャネル&gt; 6 \$5:【Z 軸補正加速度実効値:m/s^2】&lt;in&gt;&lt;数値属性参照チャネル&gt; 7 \$6:【合成軸補正加速度実効値:m/s^2】&lt;in&gt;&lt;数値属性参照チャネル&gt; 8 \$7:【X 軸等価振動値:m/s^2】&lt;out&gt;&lt;数値属性参照チャネル&gt; 9 \$8:【Y 軸等価振動値:m/s^2】&lt;out&gt;&lt;数値属性参照チャネル&gt; 10 \$9:【Z 軸等価振動値:m/s^2】&lt;out&gt;&lt;数値属性参照チャネル&gt; 11 \$10:【合成軸等価振動値:m/s^2】&lt;out&gt;&lt;数値属性参照チャネル&gt; 12 -----*/ 13 proc 区間等価振動値[ 14     def inherit_ch \$[1,10] 15     def local_ch \$[11,15],\$30 16     \$12 = EQU(LEN(\$3)+LEN(\$4)+LEN(\$5),LEN(\$6)*3)-1 17     case \$12 = 0 18     proc len_valid[ </pre>

---

```

19      $11 = EQU(LEN($1),LEN($2))
20      case_false $11
21          proc count_mdf{
22              $2 = $1*0+$2(0)
23          }count_mdf
24      $30 = $2+EQU($2,0)*(LEN($3)-$1-1) /* ゼロの場合、終端までの個数に置換*/
25      $30 = $1+$30
26      $15 = 0
27      $14 = LEN($1)
28      assign $7 = $14<0>
29      assign $8 = $14<0>
30      assign $9 = $14<0>
31      assign $10 = $14<0>
32      repeat_case $15 < $14
33          proc calc_loop{
34              $16 = ERC($1($15),$30($15),$3)
35              $17 = ERC($1($15),$30($15),$4)
36              $18 = ERC($1($15),$30($15),$5)
37              $19 = ERC($1($15),$30($15),$6)
38              call proc 振動時間率頻度 $16,$17,$18,$19,$20,$21,$22,$23,$24
39              $7($15) = SQR(SUM($20*$24^2)/SUM($20))
40              $8($15) = SQR(SUM($21*$24^2)/SUM($21))
41              $9($15) = SQR(SUM($22*$24^2)/SUM($22))
42              $10($15) = SQR(SUM($23*$24^2)/SUM($23))
43              $15 = $15+1
44          }calc_loop
45      }len_valid
46      case $12 = -1
47          proc len_invalid{
48              $1 = -1
49          }len_invalid
50      }区間等価振動値

```

---

## 6.8. 全身振動暴露・相対累積時間率演算

ProcFile 名	SS0608.prc
機能	振動時間率頻度演算結果度数を相対累積度数に変換します。
呼び出し文	<b>call proc 相対累積時間率 \$1,\$2,\$3,\$4</b>
引数	\$1:[X 軸時間率度数:sec]<in/out><数値属性参照チャネル> \$2:[Y 軸時間率度数:sec]<in/out><数値属性参照チャネル> \$3:[Z 軸時間率度数:sec]<in/out><数値属性参照チャネル> \$4:[合成軸時間率度数:sec]<in/out><数値属性参照チャネル>
備考	■ 累積時間率相当値を求める為の事前処理で、時間率頻度処理結果の度数の合計を 100%とした値に各度数を変換し、セル番号昇順に累積加算した結果が戻ります。
記述例	<pre> \$1 "X 加速度:m/s^2" = #1 \$2 "Y 加速度:m/s^2" = #2 \$3 "Z 加速度:m/s^2" = #3 call proc 補正加速度\$1,\$2,\$3 \$4 "時定数:sec" = 0.25 def ch_name \$5 "X 軸補正加速度実効値:m/s^2" def ch_name \$6 "Y 軸補正加速度実効値:m/s^2" def ch_name \$7 "Z 軸補正加速度実効値:m/s^2" def ch_name \$8 "合成軸補正加速度実効値:m/s^2" call proc 補正加速度実効値 \$4,\$1,\$2,\$3,\$5,\$6,\$7,\$8 def ch_name \$22 "X 軸度数:sec" def ch_name \$23 "Y 軸度数:sec" def ch_name \$24 "Z 軸度数:sec" def ch_name \$25 "セル中央値" call proc 振動時間率頻度 \$5,\$6,\$7,\$8,\$22,\$23,\$24,\$25,\$26 \$27 "X 軸相対累積度数:%" = \$22 \$28 "Y 軸相対累積度数:%" = \$23 \$29 "Z 軸相対累積度数:%" = \$24 \$30 "合成軸相対累積度数:%" = \$25 call proc 相対累積頻度 \$22,\$23,\$24,\$25 </pre>
Script	<pre> 1 /*----- 2 \$1:[X 軸時間率度数:sec]&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 3 \$2:[Y 軸時間率度数:sec]&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 4 \$3:[Z 軸時間率度数:sec]&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 5 \$4:[合成軸時間率度数:sec]&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 6 -----*/ 7 proc 相対累積時間率[ 8   def inherit_ch \$1,\$2,\$3,\$4 9     \$1 = ACC(\$1)/SUM(\$1)*100 10    \$2 = ACC(\$2)/SUM(\$2)*100 11    \$3 = ACC(\$3)/SUM(\$3)*100 12    \$4 = ACC(\$4)/SUM(\$4)*100 13  ]相対累積時間率 </pre>

## 6.9. 全身振動暴露-区間等価振動値/時間率相当振動値演算

ProcFile 名	SS0609.prc
機能	指定した区間毎に補正加速度実効値の時間率頻度解析を行い、その結果から等価振動値及び時間率相当値を求めます。
呼び出し文	<b>call proc 区間等価振動値 \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8,\$9,\$10,\$11,\$12,\$13,\$14,\$15</b>
引数	<p>\$1:【開始 index】&lt;in/out&gt;&lt;数値属性参照チャネル&gt;          解析開始 index 列を記述します。解析範囲は、ここで指定した index から\$2 で記述したデータ個数範囲を一回の解析範囲として、\$1 の要素数分行われます。</p> <p>\$2:【解析対象データ個数】&lt;in&gt;&lt;数値属性参照チャネル&gt;          データ個数の要素数は、開始 Index の要素数と等しくない場合は、index0 のみ繰り返して参照されます。</p> <p>\$3:【X 軸補正加速度実効値:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$4:【Y 軸補正加速度実効値:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$5:【Z 軸補正加速度実効値:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$6:【合成軸補正加速度実効値:m/s<sup>2</sup>】&lt;in&gt;&lt;数値属性参照チャネル&gt;          \$7:【X 軸等価振動値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$8:【Y 軸等価振動値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$9:【Z 軸等価振動値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$10:【合成軸等価振動値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$11:【パーセンタイル値:%】&lt;in&gt;&lt;数値属性参照チャネル&gt;          設定する値は、0&lt; パーセンタイル &lt;100 の範囲でなければなりません。          \$12:【X 軸時間率相当値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$13:【Y 軸時間率相当値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$14:【Z 軸時間率相当値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;          \$15:【合成軸時間率相当値:m/s<sup>2</sup>】&lt;out&gt;&lt;数値属性参照チャネル&gt;</p>
備考	<ul style="list-style-type: none"> <li>■ 本 Subroutine は、内部にて“振動時間率頻度”及び“相対累積時間率”を呼び出し使用しますので、timerate_2631_1.prc 及び ACtimerate.prc も同時に Script に挿入して下さい。</li> <li>■ 演算データ個数に 0 が指定された場合、対応する開始 index から終端までを演算範囲に置き換えます。</li> <li>■ 与えた各軸の振動値の個数は等しい必要があります。等しくない場合は、何も演算せず、開始 index に-1 が戻ります。</li> </ul>
記述例	<pre> \$1 "X 加速度:m/s^2" = #1 \$2 "Y 加速度:m/s^2" = #2 \$3 "Z 加速度:m/s^2" = #3 call proc 補正加速度 \$1,\$2,\$3 \$4 "時定数:sec" = 0.25 def ch_name \$5 "X 軸補正加速度実効値:m/s^2" def ch_name \$6 "Y 軸補正加速度実効値:m/s^2" def ch_name \$7 "Z 軸補正加速度実効値:m/s^2" def ch_name \$8 "合成軸補正加速度実効値:m/s^2" call proc 補正加速度実効値 \$4,\$1,\$2,\$3,\$5,\$6,\$7,\$8 def ch_name \$9 "開始 index" def ch_name \$10 "データ個数" def ch_name \$11 "区間個数" call proc markmark_1 \$9,\$10,&amp;11,\$11 /*mark-mark 間を解析区間とする*/ \$9 = LNK(0,\$9) /*最初の解析範囲は全範囲を指定する*/ \$10 = LNK(0,\$10) def ch_name \$12 "X 軸等価振動値:m/s^2" def ch_name \$13 "Y 軸等価振動値:m/s^2" def ch_name \$14 "Z 軸等価振動値:m/s^2" def ch_name \$15 "合成軸等価振動値:m/s^2" \$16 "パーセンタイル:%" = 75 assign &amp;1 = "軸時間率" \$16(F1) "相当値:m/s^2" assign &amp;2 = "X" &amp;1 def ch_name \$17 &amp;2 assign &amp;2 = "Y" &amp;1 def ch_name \$18 &amp;2 assign &amp;2 = "Z" &amp;1 def ch_name \$19 &amp;2 assign &amp;2 = "合成" &amp;1 def ch_name \$20 &amp;2 call proc 区間等価振動値時間率相当値 \$9,\$10,\$5,\$6,\$7,\$8,\$12,\$13,\$14,\$15,\$16,\$17,\$18,\$19,\$20 </pre>

```

Script 1      /*-----
2      $1:【開始 index】<in/out><数値属性参照チャネル>
3      $2:【解析対象データ個数】<in><数値属性参照チャネル>
4      $3:【X 軸補正加速度実効値:m/s^2】<in><数値属性参照チャネル>
5      $4:【Y 軸補正加速度実効値:m/s^2】<in><数値属性参照チャネル>
6      $5:【Z 軸補正加速度実効値:m/s^2】<in><数値属性参照チャネル>
7      $6:【合成軸補正加速度実効値:m/s^2】<in><数値属性参照チャネル>
8      $7:【X 軸等価振動値:m/s^2】<out><数値属性参照チャネル>
9      $8:【Y 軸等価振動値:m/s^2】<out><数値属性参照チャネル>
10     $9:【Z 軸等価振動値:m/s^2】<out><数値属性参照チャネル>
11     $10:【合成軸等価振動値:m/s^2】<out><数値属性参照チャネル>
12     $11:【パーセンタイル値:%】<in><数値属性参照チャネル>
13     $12:【X 軸時間率相当値:m/s^2】<out><数値属性参照チャネル>
14     $13:【Y 軸時間率相当値:m/s^2】<out><数値属性参照チャネル>
15     $14:【Z 軸時間率相当値:m/s^2】<out><数値属性参照チャネル>
16     $15:【合成軸時間率相当値:m/s^2】<out><数値属性参照チャネル>
17     -----*/
18     proc 区間等価振動値時間率相当値(
19         def inherit_ch $[1,15]
20         def local_ch $[16,20]
21         $16 = EQU(LEN($3)+LEN($4)+LEN($5),LEN($6)*3)-1
22         case $16 = 0
23             proc len_valid[
24                 $17 = EQU(LEN($1),LEN($2))
25                 case_false $17
26                     proc count_mdf[
27                         $2 = $1*0+$2(0)
28                     ]count_mdf
29                 $30 = $2+EQU($2,0)*(LEN($3)-$1-1) /* ゼロの場合、終端までの個数に置換*/
30                 $30 = $1+$30 /* データ個数を終了 index に変換*/
31                 $18 "counter:" = 0
32                 $19 = LEN($1)
33                 assign $7 = $19<0> /* 結果個数配列生成*/
34                 assign $8 = $19<0>
35                 assign $9 = $19<0>
36                 assign $10 = $19<0>
37                 assign $12 = $19<0>
38                 assign $13 = $19<0>
39                 assign $14 = $19<0>
40                 assign $15 = $19<0>
41                 repeat_case $18 < $19
42                     proc calc_loop[
43                         $20 = ERC($1($18),$30($18),$3) /*解析範囲切り出し*/
44                         $21 = ERC($1($18),$30($18),$4)
45                         $22 = ERC($1($18),$30($18),$5)
46                         $23 = ERC($1($18),$30($18),$6)
47                         call proc 振動時間率頻度 $20,$21,$22,$23,$24,$25,$26,$27,$28
48                         $7($18) = SQR(SUM($24*$28^2)/SUM($24)) /*等価振動値演算*/
49                         $8($18) = SQR(SUM($25*$28^2)/SUM($25))
50                         $9($18) = SQR(SUM($26*$28^2)/SUM($26))
51                         $10($18) = SQR(SUM($27*$28^2)/SUM($27))
52                         call proc 相対累積時間率 $24,$25,$26,$27
53                         $10 = ABS($10)
54                         $29 = AND(GTE($10,0.01),LTE($10,99.99)) /* パーセンタイル check*/
55                         case_true $29
56                             proc 相当頻度演算[
57                                 $12($18) = PTV(DTD(1,$10,1.0,$24),$28) /*時間率相当頻度値演算*/
58                                 $13($18) = PTV(DTD(1,$10,1.0,$25),$28)
59                                 $14($18) = PTV(DTD(1,$10,1.0,$26),$28)
60                                 $15($18) = PTV(DTD(1,$10,1.0,$27),$28)
61                             ]相当頻度演算
62                             $18 = $18+1 /* 区間 counterUp*/
63                         ]calc_loop
64                 ]len_valid

```

---

```
65     case $12 = -1
66         proc len_invalid{
67             $1 = -1
68         }len_invalid
69     }区間等価振動値時間率相当値
```

---

## 6.10. 全身振動暴露・区間波高率演算

ProcFile 名	SS0610.pro
機能	各軸の補正加速度から指定した区間毎のクレストファクタを求めます。
呼び出し文	<b>call proc クレストファクタ \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8</b>
引数	<p>\$1:【開始 index】&lt;in/ou&gt; &lt;数値属性参照チャネル&gt;          \$2:【データ個数】&lt;in&gt; &lt;数値属性参照チャネル&gt;          \$3:【X 軸補正加速度:m/s<sup>2</sup>】&lt;in&gt; &lt;数値属性参照チャネル&gt;          \$4:【Y 軸補正加速度:m/s<sup>2</sup>】&lt;in&gt; &lt;数値属性参照チャネル&gt;          \$5:【Z 軸補正加速度:m/s<sup>2</sup>】&lt;in&gt; &lt;数値属性参照チャネル&gt;          \$6:【区間 X 軸クレストファクタ】&lt;out&gt; &lt;数値属性参照チャネル&gt;          \$7:【区間 Y 軸クレストファクタ】&lt;out&gt; &lt;数値属性参照チャネル&gt;          \$8:【区間 Z 軸クレストファクタ】&lt;out&gt; &lt;数値属性参照チャネル&gt;</p>
備考	<p>■ 演算データ個数に 0 が指定された場合、対応する開始 index から終端までを演算範囲に置き換えます。          ■ 与えた各軸の振動値の個数は等しい必要があります。等しくない場合は、何も演算せず、開始 index に-1 が戻ります。</p>
記述例	<pre> \$1 "X 加速度:m/s^2" = #1 \$2 "Y 加速度:m/s^2" = #2 \$3 "Z 加速度:m/s^2" = #3 call proc 補正加速度 \$1,\$2,\$3 def ch_name \$9 "開始 index" def ch_name \$10 "データ個数" def ch_name \$11 "区間個数" call proc markmark_1 \$9,\$10,&amp;11,\$11 /*mark-mark 間を解析区間とする*/ \$9 = LNK(0,\$9) /*最初の解析範囲は全範囲を指定する*/ \$10 = LNK(0,\$10) def ch_name \$12 "X 軸クレストファクタ" def ch_name \$13 "Y 軸クレストファクタ" def ch_name \$14 "Z 軸クレストファクタ" call proc クレストファクタ \$9,\$10,\$1,\$2,\$3,\$12,\$13,\$14         </pre>
Script	<pre> 1 /*-----*/ 2 \$1:【開始 index】&lt;in/out&gt; &lt;数値属性参照チャネル&gt; 3 \$2:【データ個数】&lt;in&gt; &lt;数値属性参照チャネル&gt; 4 \$3:【X 軸補正加速度:m/s^2】&lt;in&gt; &lt;数値属性参照チャネル&gt; 5 \$4:【Y 軸補正加速度:m/s^2】&lt;in&gt; &lt;数値属性参照チャネル&gt; 6 \$5:【Z 軸補正加速度:m/s^2】&lt;in&gt; &lt;数値属性参照チャネル&gt; 7 \$6:【区間 X 軸クレストファクタ】&lt;out&gt; &lt;数値属性参照チャネル&gt; 8 \$7:【区間 Y 軸クレストファクタ】&lt;out&gt; &lt;数値属性参照チャネル&gt; 9 \$8:【区間 Z 軸クレストファクタ】&lt;out&gt; &lt;数値属性参照チャネル&gt; 10 /*-----*/ 11 proc クレストファクタ{ 12     def inherit_ch \$[1,8] 13     def local_ch \$11,\$12,\$13,\$14,\$15,\$30 14     \$12 = EQU(LEN(\$3)+LEN(\$4),LEN(\$5)*2)-1 15     case \$12 = 0 16     proc len_valid{ 17         \$11 = EQU(LEN(\$1),LEN(\$2)) 18         case_false \$11 19         proc count_mdf{ 20             \$2 = \$1*0+\$2(0) 21         }count_mdf 22         \$30 = \$2+EQU(\$2,0)*(LEN(\$3)-\$1-1) /* ゼロの場合、終端までの個数に置換*/ 23         \$30 = \$1+\$30 24         \$15 = 0 25         \$14 = LEN(\$1) 26         assign \$6 = \$14&lt;0&gt; 27         assign \$7 = \$14&lt;0&gt; 28         assign \$8 = \$14&lt;0&gt; 29         repeat_case \$15 &lt; \$14 30         proc calc_loop{ 31             \$6(\$15) = MAX(ABS(ERC(\$1(\$15),\$30(\$15),\$3)))/EFF(ERC(\$1(\$15),\$30(\$15),\$3)) 32             \$7(\$15) = MAX(ABS(ERC(\$1(\$15),\$30(\$15),\$4)))/EFF(ERC(\$1(\$15),\$30(\$15),\$4))         </pre>

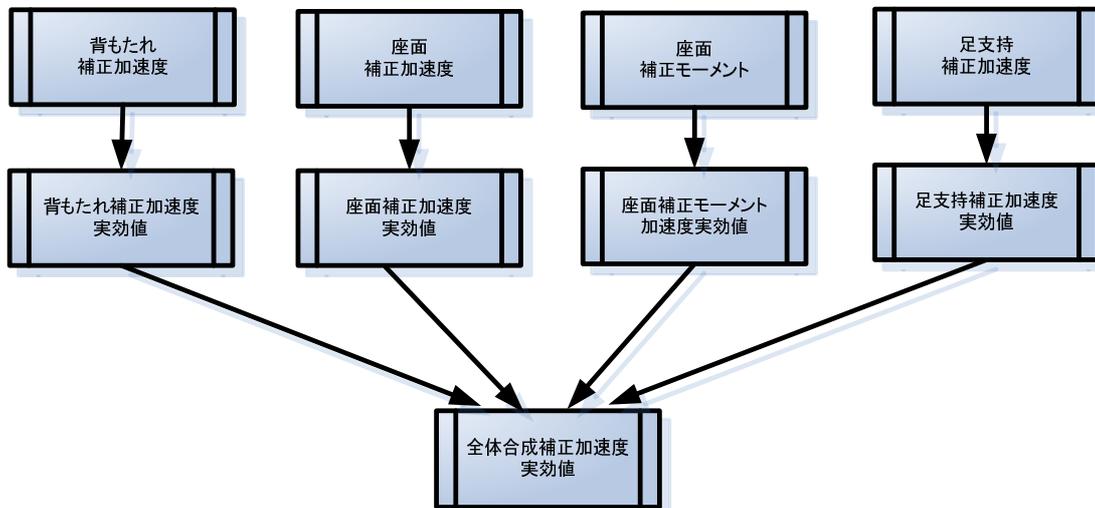
---

```
33          $8($15) = MAX(ABS(ERC($1($15),$30($15),$5)))/EFF(ERC($1($15),$30($15),$5))
34          $15 = $15+1
35          }calc_loop
36          }len_valid
37          case $12 = -1
38          proc len_invalid{
39          $1 = -1
40          }len_invalid
41          }クレストファクタ
```

---

## 7. 快適性解析処理関連

サブルーチン・ツリー



### 7.1. 快適性・背もたれ補正加速度

ProcFile 名	SS0701.prc
機能	背もたれ部 3 軸加速度データに ISO2631-1 補正フィルタ処理を行います。
呼び出し文	call proc 背もたれ補正加速度 \$1,\$2,\$3
引数	\$1: [X 軸背もたれ加速度] <in/out> <数値属性参照チャネル> \$2: [Y 軸背もたれ加速度] <in/out> <数値属性参照チャネル> \$3: [Z 軸背もたれ加速度] <in/out> <数値属性参照チャネル>
備考	<ul style="list-style-type: none"> <li>■ 背もたれ部は身体の主要支持範囲、困難な場合は、感度軸を 15° までずらして良い。</li> <li>■ チャネルは、数値属性参照チャネルで与えます。</li> <li>■ X 軸(前後)Wc、Y 軸(左右)Wd、Z 軸(上下)Wd のウエイティングフィルタを掛けます。</li> </ul>
記述例	<pre> \$1 = #1 \$2 = #2 \$3 = #3 call proc 背もたれ補正加速度 \$1,\$2,\$3           </pre>
Script	<pre> 1      /*-----*/ 2      \$1: [X 軸背もたれ加速度] &lt;in/out&gt; &lt;数値属性参照チャネル&gt; 3      \$2: [Y 軸背もたれ加速度] &lt;in/out&gt; &lt;数値属性参照チャネル&gt; 4      \$3: [Z 軸背もたれ加速度] &lt;in/out&gt; &lt;数値属性参照チャネル&gt; 5      /*-----*/ 6      proc 背もたれ補正加速度{ 7          def inherit_ch \$1,\$2,\$3 8          \$5 = WBC(\$1) 9          \$6 = WBD(\$2) 10         \$7 = WBD(\$3) 11         }背もたれ補正加速度           </pre>

## 7.2. 快適性・座面補正加速度

ProcFile 名	SS0702.prc
機能	座面部 3 軸加速度データに ISO2631-1 補正フィルタ処理を行います。
呼び出し文	<b>call proc 座面補正加速度 \$1,\$2,\$3</b>
引数	\$1:[X 軸座面加速度]<in/out><数値属性参照チャネル> \$2:[Y 軸座面加速度]<in/out><数値属性参照チャネル> \$3:[Z 軸座面加速度]<in/out><数値属性参照チャネル>
備考	<ul style="list-style-type: none"> <li>■ 座面部は、座骨結節の直下</li> <li>■ チャネルは、数値属性参照チャネルで与えます。</li> <li>■ X 軸(前後)Wd、Y 軸(左右)Wd、Z 軸(上下)Wk のウエイティングフィルタを掛けます。</li> </ul>
記述例	\$1 = #4 \$2 = #5 \$3 = #6 call proc 座面補正加速度 \$1,\$2,\$3
Script	<pre> 1      /*----- 2      \$1:[X 軸座面加速度]&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 3      \$2:[Y 軸座面加速度]&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 4      \$3:[Z 軸座面加速度]&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 5      -----*/ 6      proc 座面補正加速度{ 7          def inherit_ch \$1,\$2,\$3 8              \$1 = WBD(\$1) 9              \$2 = WBD(\$2) 10             \$3 = WBK(\$3) 11         }座面補正加速度 </pre>

## 7.3. 快適性・座面補正角加速度

ProcFile 名	SS0703.prc
機能	座面部 3 軸角加速度データに iso2631-1 補正フィルタ処理を行います。
呼び出し文	<b>call proc 座面補正モーメント \$1,\$2,\$3</b>
引数	\$1:[X 軸(ロールレート)座面モーメント]<in/out><数値属性参照チャネル> \$2:[Y 軸(ピッチレート)座面モーメント]<in/out><数値属性参照チャネル> \$3:[Z 軸(ヨーレート)座面モーメント]<in/out><数値属性参照チャネル>
備考	<ul style="list-style-type: none"> <li>■ 座面部は、座骨結節の直下</li> <li>■ チャネルは、数値属性参照チャネルで与えます。</li> <li>■ X 軸(ロール)We、Y 軸(ピッチ)We、Z 軸(ヨー)We のウエイティングフィルタを掛けます。</li> </ul>
記述例	\$1 = #7 \$2 = #8 \$3 = #9 call proc 座面補正モーメント \$1,\$2,\$3
Script	<pre> 1      /*----- 2      \$1:[X 軸(ロール)座面モーメント]&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 3      \$2:[Y 軸(ピッチ)座面モーメント]&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 4      \$3:[Z 軸(ヨー)座面モーメント]&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 5      -----*/ 6      proc 座面補正モーメント{ 7          def inherit_ch \$1,\$2,\$3 8              \$1 = WBE(\$1) 9              \$2 = WBE(\$2) 10             \$3 = WBE(\$3) 11         }座面補正モーメント </pre>

## 7.4. 快適性-足支持補正加速度

ProcFile 名	SS0704.prc
機能	足支持部 3 軸加速度データに ISO2631-1 補正フィルタ処理を行います。
呼び出し文	<b>call proc 足支持補正加速度 \$1,\$2,\$3</b>
引数	\$1:[X 軸足支持加速度]<in/out><数値属性参照チャネル> \$2:[Y 軸足支持加速度]<in/out><数値属性参照チャネル> \$3:[Z 軸足支持加速度]<in/out><数値属性参照チャネル>
備考	<ul style="list-style-type: none"> <li>■ 足支持面部は足が置かれる機会の最も多い支持面</li> <li>■ チャネルは、数値属性参照チャネルで与えます。</li> <li>■ X 軸(前後)Wk、Y 軸(左右)Wk、Z 軸(上下)Wk のウエイティングフィルタを掛けます。</li> </ul>
記述例	<pre>\$1 = #10 \$2 = #11 \$3 = #12 call proc 足支持補正加速度 \$1,\$2,\$3</pre>
Script	<pre>1 /*----- 2 \$1:[X 軸足支持加速度]&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 3 \$2:[Y 軸足支持加速度]&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 4 \$3:[Z 軸足支持加速度]&lt;in/out&gt;&lt;数値属性参照チャネル&gt; 5 -----*/ 6 proc 足支持補正加速度[ 7     def inherit_ch \$1,\$2,\$3 8     \$1 = WBK(\$1) 9     \$2 = WBK(\$2) 10    \$3 = WBK(\$3) 11    ]足支持補正加速度</pre>

## 7.5. 快適性・背もたれ補正加速度実効値

ProcFile 名	SS0705.prc
機能	背もたれ部補正フィルタ処理済みのデータから指定した時定数で移動実効値演算を行い、同時に合成軸を求めます。
呼び出し文	<b>call proc 背もたれ補正加速度実効値 \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8</b>
引数	<p>\$1:【背もたれ X 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$2:【背もたれ Y 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$3:【背もたれ Z 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$4:【時定数】&lt;in&gt;&lt;数値属性参照チャネル&gt;                単位は秒(sec)            \$5:【背もたれ X 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt;            \$6:【背もたれ Y 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt;            \$7:【背もたれ Z 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt;            \$8【背もたれ合成補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt;</p>
備考	■ 合成軸演算補正係数は、x 軸 0.8、Y 軸 0.5、Z 軸 0.4
記述例	<pre>\$1 = #1 \$2 = #2 \$3 = #3 call proc 背もたれ補正加速度 \$1,\$2,\$3 \$4 "時定数:sec" = 1 call proc 背もたれ補正加速度実効値 \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8</pre>
Script	<pre>1 /*-----*/ 2 \$1:【背もたれ X 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt; 3 \$2:【背もたれ Y 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt; 4 \$3:【背もたれ Z 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt; 5 \$4:【時定数】&lt;in&gt;&lt;数値属性参照チャネル&gt; 6 単位は秒(sec) 7 \$5:【背もたれ X 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt; 8 \$6:【背もたれ Y 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt; 9 \$7:【背もたれ Z 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt; 10 \$8【背もたれ合成補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt; 11 /*-----*/ 12 proc 背もたれ補正加速度実効値[ 13     def inherit_ch \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8 14     \$5 = RRV(\$4,\$1) 15     \$6 = RRV(\$4,\$2) 16     \$7 = RRV(\$4,\$3) 17     \$8 = SQR((0.8*\$5)^2+(0.5*\$6)^2+(0.4*\$7)^2) 18 ]背もたれ補正加速度実効値</pre>

## 7.6. 快速性・座面補正加速度実効値

ProcFile 名	SS0706.prc
機能	座面部補正フィルタ処理済みのデータから指定した時定数で移動実効値演算を行い、同時に合成軸を求めます。
呼び出し文	<b>call proc 座面補正加速度実効値 \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8</b>
引数	<p>\$1:【座面 X 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$2:【座面 Y 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$3:【座面 Z 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$4:【時定数】&lt;in&gt;&lt;数値属性参照チャネル&gt;                単位は秒(sec)            \$5:【座面 X 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt;            \$6:【座面 Y 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt;            \$7:【座面 Z 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt;            \$8:【座面合成補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt;</p>
備考	■ 合成軸演算補正係数は、x 軸 1、Y 軸 1、Z 軸 1
記述例	<pre>\$1 = #4 \$2 = #5 \$3 = #6 call proc 座面補正加速度 \$1,\$2,\$3 \$4 "時定数:sec" = 1 call proc 座面補正加速度実効値 \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8</pre>
Script	<pre>1      /*----- 2      \$1:【座面 X 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt; 3      \$2:【座面 Y 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt; 4      \$3:【座面 Z 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt; 5      \$4:【時定数】&lt;in&gt;&lt;数値属性参照チャネル&gt; 6      単位は秒(sec) 7      \$5:【座面 X 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt; 8      \$6:【座面 Y 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt; 9      \$7:【座面 Z 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt; 10     \$8【座面合成補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt; 11     -----*/ 12     proc 座面補正加速度実効値[ 13         def inherit_ch \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8 14         \$5 = RRV(\$4,\$1) 15         \$6 = RRV(\$4,\$2) 16         \$7 = RRV(\$4,\$3) 17         \$8 = SQR((0.8*\$5)^2+(0.5*\$6)^2+(0.4*\$7)^2) 18     ]座面補正加速度実効値</pre>

## 7.7. 快速性・座面モーメント補正加速度実効値

ProcFile 名	SS0707.pro
機能	座面部角加速度補正フィルタ処理済みのデータから指定した時定数で移動実効値演算を行い、同時に合成軸を求めます。
呼び出し文	<b>call proc 座面モーメント補正加速度実効値 \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8</b>
引数	<p>\$1:【座面モーメント X 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$2:【座面モーメント Y 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$3:【座面モーメント Z 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt;            \$4:【時定数】&lt;in&gt;&lt;数値属性参照チャネル&gt;                単位は秒(sec)            \$5:【座面モーメント X 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt;            \$6:【座面モーメント Y 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt;            \$7:【座面モーメント Z 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt;            \$8:【座面モーメント合成補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt;</p>
備考	■ 合成軸演算補正係数は、x 軸 0.63、Y 軸 0.4、Z 軸 0.2
記述例	<pre>\$1 = #7 \$2 = #8 \$3 = #9 call proc 座面補正モーメント \$1,\$2,\$3 \$4 "時定数:sec" = 1 call proc 座面モーメント補正加速度実効値 \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8</pre>
Script	<pre>1 /*----- 2     \$1:【座面モーメント X 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt; 3     \$2:【座面モーメント Y 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt; 4     \$3:【座面モーメント Z 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt; 5     \$4:【時定数】&lt;in&gt;&lt;数値属性参照チャネル&gt; 6         単位は秒(sec) 7     \$5:【座面モーメント X 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt; 8     \$6:【座面モーメント Y 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt; 9     \$7:【座面モーメント Z 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt; 10    \$8:【座面モーメント合成補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt; 11    -----*/ 12    proc 座面モーメント補正加速度実効値( 13        def inherit_ch \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8 14            \$5 = RRV(\$4,\$1) 15            \$6 = RRV(\$4,\$2) 16            \$7 = RRV(\$4,\$3) 17            \$8 = SQR((0.25*\$5)^2+(0.25*\$6)^2+(0.4*\$7)^2) 18    )座面モーメント補正加速度実効値</pre>

## 7.8. 快適性-足支持補正加速度実効値

ProcFile 名	SS0708.prc
機能	足支持部補正フィルタ処理済みのデータから指定した時定数で移動実効値演算を行い、同時に合成軸を求めます。
呼び出し文	<b>call proc 足支持補正加速度実効値 \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8</b>
引数	\$1:【足支持 X 軸補正加速度】<in><数値属性参照チャネル> \$2:【足支持 Y 軸補正加速度】<in><数値属性参照チャネル> \$3:【足支持 Z 軸補正加速度】<in><数値属性参照チャネル> \$4:【時定数】<in><数値属性参照チャネル> 単位は秒(sec) \$5:【足支持 X 軸補正加速度実効値】<out><数値属性参照チャネル> \$6:【足支持 Y 軸補正加速度実効値】<out><数値属性参照チャネル> \$7:【足支持 Z 軸補正加速度実効値】<out><数値属性参照チャネル> \$8:【足支持合成補正加速度実効値】<out><数値属性参照チャネル>
備考	■ 合成軸演算補正係数は、x 軸 0.25、Y 軸 0.25、Z 軸 0.4
記述例	<pre> \$1 = #10 \$2 = #11 \$3 = #12 call proc 足支持補正加速度 \$1,\$2,\$3 \$4 "時定数:sec" = 1 call proc 足支持補正加速度実効値 \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8           </pre>
Script	<pre> 1      /*----- 2      \$1:【足支持 X 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt; 3      \$2:【足支持 Y 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt; 4      \$3:【足支持 Z 軸補正加速度】&lt;in&gt;&lt;数値属性参照チャネル&gt; 5      \$4:【時定数】&lt;in&gt;&lt;数値属性参照チャネル&gt; 6      単位は秒(sec) 7      \$5:【足支持 X 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt; 8      \$6:【足支持 Y 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt; 9      \$7:【足支持 Z 軸補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt; 10     \$8:【足支持合成補正加速度実効値】&lt;out&gt;&lt;数値属性参照チャネル&gt; 11     -----*/ 12     proc 足支持補正加速度実効値[ 13         def inherit_ch \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8 14         \$5 = RRV(\$4,\$1) 15         \$6 = RRV(\$4,\$2) 16         \$7 = RRV(\$4,\$3) 17         \$8 = SQR((0.25*\$5)^2+(0.25*\$6)^2+(0.4*\$7)^2) 18     ]足支持補正加速度実効値           </pre>

## 7.9. 快速性・全体合成補正加速度実効値

ProcFile 名	SS0709.prc
機能	測定部位毎に求めた合成補正加速度実効値から全体の合成補正加速度実効値を求めます。
呼び出し文	<b>call proc 全体合成補正加速度実効値 \$1,\$2,\$3,\$4,\$5</b>
引数	\$1:【背もたれ合成補正加速度実効値】<in><数値属性参照チャネル> \$2:【座面合成補正加速度実効値】<in><数値属性参照チャネル> \$3:【座面合成補正モーメント加速度実効値】<in><数値属性参照チャネル> \$4:【足支持合成補正加速度実効値】<in><数値属性参照チャネル> \$5:【全体合成補正加速度実効値】<out><数値属性参照チャネル>
備考	■ 計測していない測定部位の合成補正加速度実効値は0として与えます。
記述例	\$1 = #1 \$2 = #2 \$3 = #3 \$4 = #4 \$5 = #5 \$6 = #6 \$7 = #7 \$8 = #8 \$9 = #9 \$10 = #10 \$11 = #11 \$12 = #12 \$20 “時定数:sec” = 1 call proc 背もたれ補正加速度 \$1,\$2,\$3 call proc 座面補正加速度 \$4,\$5,\$6 call proc 座面補正モーメント \$7,\$8,\$9 call proc 背もたれ補正加速度実効値 \$1,\$2,\$3,\$20,\$21,\$22,\$23,\$41 call proc 座面補正加速度実効値 \$4,\$5,\$6,\$20,\$24,\$25,\$26,\$42 call proc 座面モーメント補正加速度実効値 \$7,\$8,\$9,\$20,\$27,\$28,\$29,\$43 call proc 足支持補正加速度 \$10,\$11,\$12 call proc 足支持補正加速度実効値 \$10,\$11,\$2,\$20,\$30,\$31,\$32,\$44 call proc 全体合成補正加速度実効値 \$41,\$42,\$43,\$44,\$45
Script	1 /*-----*/ 2 \$1:【背もたれ合成補正加速度実効値】<in><数値属性参照チャネル> 3 \$2:【座面合成補正加速度実効値】<in><数値属性参照チャネル> 4 \$3:【座面合成補正モーメント加速度実効値】<in><数値属性参照チャネル> 5 \$4:【足支持合成補正加速度実効値】<in><数値属性参照チャネル> 6 \$5:【全体合成補正加速度実効値】<out><数値属性参照チャネル> 7 -----*/ 8 proc 全体合成補正加速度実効値{ 9 def inherit_ch \$1,\$2,\$3,\$4,\$5 10 \$5 = SQR(\$1^2+\$2^2+\$3^2+\$4^2) 11 }全体合成補正加速度実効値



**株式会社 デイシー**

〒205-0002 東京都羽村市栄町3-3-6

電話: 042-570-7121

メール: [info@deicy.co.jp](mailto:info@deicy.co.jp)

© Copyright 2006, 2007, 2008, 2009, 2010 DEICY Corporation

**株式会社 デイシーインスツルメンツ**

〒205-0002 東京都羽村市栄町3-3-6

電話: 042-570-7085

メール: [info@deicy.co.jp](mailto:info@deicy.co.jp)