

SPACE サーバ(SCServer)使用説明書

ver.1.0 11

2010/03/18

目次

1.1. 起動に必要なファイル.....	3
----------------------	---

2. コマンドリファレンス 3

2.1. 規定動作(JCS にて起動).....	3
2.1.1. <i>dtm</i> (原点サーチ復帰動作) [put].....	3
2.1.2. <i>mount</i> (結晶マウント) [put].....	3
2.1.3. <i>unmount</i> (結晶アンマウント) [put].....	3
2.1.4. <i>move</i> (トレイ内・トレイ間移動) [put].....	3
2.1.5. <i>cleaning</i> (クリーニング動作) [put].....	4
2.1.6. <i>xafs</i> (XAFS 測定位置へ移動) [put].....	4
2.1.7. <i>trayin</i> (トレイセット位置へ移動) [put].....	4
2.1.8. <i>trayout</i> (トレイ回収位置へ移動) [put].....	4
2.1.9. <i>next</i> (Warning Stop 後の動作の継続) [put].....	4
2.1.10. <i>In2in</i> (予約コマンドでロボット先端の冷却) [put].....	4
2.1.11. <i>sc_evacuate/on</i> (退避位置へ移動) [put].....	4
2.1.12. <i>sc_evacuate/off</i> (マウント位置へ移動) [put].....	4
2.2. 任意動作 (JCS を使用しない)	4
2.2.1. <i>position</i> (軸を指定位置へ移動[put] / 軸の現在位置を得る[get])	4
2.2.2. <i>ccount</i> (<i>cleaning</i> 状況を得る。(<i>noclean.txt</i> の数字を得る) [get])	4
2.2.3. <i>preset</i> (軸のカウンタ値をセットする) [put].....	5
2.2.4. <i>speed</i> (動作速度の設定[put] / 現在の設定速度を得る[get])	5
2.2.5. <i>rise</i> (<i>Rise</i> の設定[put] / 現在の設定値を得る[get])	5
2.2.6. <i>fall</i> (<i>Fall</i> の設定[put] / 現在の設定値を得る[get])	5
2.2.7. <i>stop</i> (軸の停止) [put].....	5
2.2.8. <i>status</i> (軸のステータスを取る) [get].....	6
2.2.9. <i>trayid</i> (トレイ ID の設定[put] / トレイ ID の取得[get])	6
2.2.10. <i>traytype</i> (トレイの種類を取得) [get].....	6
2.2.11. <i>sc_cryo/open</i> (クライオシャッター開) [put].....	6
2.2.12. <i>sc_cryo/close</i> (クライオシャッター閉) [put].....	6
2.2.13. <i>sc_cryo/status</i> (クライオシャッターの状態取得) [get].....	6
2.2.14. <i>disconnect</i> (サーバからの切断) [put].....	6
2.2.15. <i>jcs</i> (任意の JCS を実行する) [put].....	6
2.2.16. <i>HALT</i> (<i>Accidental halt</i> の発生) [put].....	6
2.2.17. <i>recover</i> (<i>Accidental halt</i> の解除) [put].....	7
2.2.18. <i>QUIT</i> (サーバの終了) [put].....	7

3. JCS(JOB CONTROL SCRIPT)	7
3.1. JCS について	7
3.3. JCS の記述例	8
3.4. JCS で使用できる変数	9
3.4.1. <i>alias.txt</i>	9
3.5. JCS で使用できるコマンド	10
3.5.1. <i>IF · ELSE · ENDIF</i> (条件分岐)	10
3.5.2. <i>GOTO</i> (指定場所へ処理を移す)	12
3.5.3. <i>CALC</i> (演算)	13
3.5.4. <i>SAVE</i> (カレント値の保存)	14
3.5.5. <i>MES</i> (メッセージの保存)	14
3.5.6. <i>WARNING</i> (Warning stop)	14
3.5.7. <i>HALT</i> (Accidental halt)	15
4. 更新履歴	15

起動

1.1. 起動に必要なファイル

以下のファイルは、すべて、サーバ本体と同じディレクトリに置きます。

- SCServer.exe (サーバ本体)
- FbiDevio.dll (GPC-7212 用 DLL)
- FbiMtrnp.dll (GPC-7212 用 DLL)
- FbiPenc.dll (GPC-6204 用 DLL)
- FBIDIO.DLL (GPC-2000 用 DLL)
- tcl83.dll (Tcl 用 DLL)
- 環境によっては、その他 MFC 関係の DLL が必要になります。
- alias.txt (JCS 中で使用する変数の定義ファイル)
JCS 内では、\$Lh_speed というように使用します。
- traypos/XXX.txt (トレイ穴位置定義ファイル)
- JCS 類

2. コマンドリファレンス

2.1. 規定動作(JCS にて起動)

2.1.1. dtm (原点サーチ復帰動作) [put]

- I/put/sc_all/dtm (dtm_all.jcs)
- I/put/sc_lhead/dtm (dtm_lhead.jcs)
- I/put/sc_thead/dtm (dtm_thead.jcs)
- I/put/sc_tarm/dtm (dtm_tarm.jcs)
- I/put/sc_xtray/dtm (dtm_xtray.jcs)
- I/put/sc_ytray/dtm (dtm_ytray.jcs)
- I/put/sc_estage/dtm (dtm_estage.jcs)
- I/put/sc_hstage/dtm (dtm_hstage.jcs)
- I/put/sc_tstage/dtm (dtm_tstage.jcs)

2.1.2. mount (結晶マウント) [put]

- I/put/sc_all/mount_1_5 (mount.jcs)
トレイ 1 の 5 番目の結晶をマウントする

2.1.3. unmount (結晶アンマウント) [put]

- I/put/sc_all/unmount_1_5 (unmount.jcs)
ゴニオに載っている結晶をトレイ 1 の 5 番目にストックする

2.1.4. move (トレイ内・トレイ間移動) [put]

- I/put/sc_all/move_1_5_1_7 (move.jcs)
トレイ 1 の 5 番目の結晶をトレイ 1 の 7 番目に移動する

2.1.5. **cleaning** (クリーニング動作) [put]

- I/put/sc_all/cleaning (cleaning.jcs)

2.1.6. **xafs** (XAFS 測定位置へ移動) [put]

- I/put/sc_all/xafs (xafs.jcs)

2.1.7. **trayin** (トレイセット位置へ移動) [put]

- I/put/sc_all/trayin (trayin.jcs)

2.1.8. **trayout** (トレイ回収位置へ移動) [put]

- I/put/sc_all/trayout (trayout.jcs)

2.1.9. **next** (Warning Stop 後の動作の継続) [put]

- I/put/sc_all/next (next.jcs)
(warning.txt を削除します)

2.1.10. **ln2in** (予約コマンドでロボット先端の冷却) [put]

- I/put/sc_reserve/ln2in (ln2in.jcs)

2.1.11. **sc_evacuate/on** (退避位置へ移動) [put]

- I/put/sc_evacuate/on (evacuate_on.jcs)

2.1.12. **sc_evacuate/off** (マウント位置へ移動) [put]

- I/put/sc_evacuate/off (evacuate_off.jcs)

2.2. 任意動作 (JCS を使用しない)

2.2.1. **position** (軸を指定位置へ移動[put] / 軸の現在位置を得る[get])

- I/put/sc_lhead/position_1000 (lhead を 1000 の位置に移動)
 - I/get/sc_lhead/position (lhead の現在位置を得る)
Rsrv/put/sc_lhead/position_4500
 - I/get/sc_all/position (全ての軸の現在位置を得る)
Rsrv/put/sc_all/position_1000_1200_4500_120_300_1000
(前から、lhead, thead, tarm, xtray, ytray, estage の順)
- (注) put では、all は指定できません。

2.2.2. **ccount** (cleaning 状況を得る。(noclean.txt の数字を得る) [get])

- I/get/sc_all/dtm

Rsrv/put/sc_all/1times

- I/get/sc_all/cleaning

Rsrv/put/sc_all/1times

- I/get/sc_all/mount_X_X

Rsrv/put/sc_all/0times

- I/get/sc_all/unmount_X_X

Rsrv/put/sc_all/0times

2.2.3. **preset** (軸のカウンタ値をセットする) [put]

- I/put/sc_lhead/preset_1000 (lheadのカウンタ値を1000にする)

(注) allは指定できません。

2.2.4. **speed** (動作速度の設定[put] / 現在の設定速度を得る[get])

- I/put/sc_lhead/speed_8000 (lheadの動作速度を8000にする)

(注) allは指定できません。

- I/get/sc_lhead_speed (lheadの現在の設定速度を得る)

Rsrv/put/sc_lhead/speed_8000

- I/get/sc_all/speed (全軸の設定速度を得る)

Rsrv/put/sc_all/speed_8000_10000_10000_6000_6000_4000

(前から、lhead, thead, tarm, xtray, ytray, estageの順)

2.2.5. **rise** (Riseの設定[put] / 現在の設定値を得る[get])

- I/put/sc_lhead/speed_200 (lheadのriseの設定値を200にする)

- I/get/sc_lhead/rise (lheadのriseの設定値を得る)

Rsrv/put/sc_lhead/rise_100

- I/get/sc_all/rise (全軸のriseの設定値を得る)

Rsrv/put/sc_all/rise_100_100_200_100_100_200

(前から、lhead, thead, tarm, xtray, ytray, estageの順)

2.2.6. **fall** (Fallの設定[put] / 現在の設定値を得る[get])

- I/put/sc_lhead/speed_200 (lheadのfallの設定値を200にする)

- I/get/sc_lhead/fall (lheadのfallの設定値を得る)

Rsrv/put/sc_lhead/fall_100

- I/get/sc_all/fall (全軸のfallの設定値を得る)

Rsrv/put/sc_all/fall_100_100_200_100_100_200

(前から、lhead, thead, tarm, xtray, ytray, estageの順)

2.2.7. **stop** (軸の停止) [put]

- I/put/sc_lhead/stop (lhead軸の停止)

- I/put/sc_all/stop (全軸の停止)

2.2.8. **status** (軸のステータスを取る) [get]

- I/get/sc_lhead/status (lhead 軸のステータスを取る)

Rsrv/put/sc_lhead/status_0_1_0

(前から、EL+, ORG, EL-信号)

- I/get/sc_all/status (全軸のステータスを取る)

Rsrv/put/sc_all/status_0_-1_0_0_-1_0_0_-1_0_0_-1_0_0_-1_0_0_-1_0

(前から、lhead, thead, tarm, xtray, ytray, estage の EL+, ORG, EL-信号の順)

2.2.9. **trayid** (トレイ ID の設定[put] / トレイ ID の取得[get])

- I/put/sc_all/trayid_1_XXXYYYZZZ

トレイ 1 の ID を XXXYYYZZZ に設定する。

- I/get/sc_all/trayid_2

トレイ 2 の ID を取得する。

2.2.10. **traytype** (トレイの種類を取得) [get]

- I/get/sc_all/traytype_1

トレイ 1 の種類を取得する。

2.2.11. **sc_cryo/open** (クライオシャッター開) [put]

- I/put/sc_cryo/open

2.2.12. **sc_cryo/close** (クライオシャッター閉) [put]

- I/put/sc_cryo/close

2.2.13. **sc_cryo/status** (クライオシャッターの状態取得) [get]

- I/get/sc_cryo/status

2.2.14. **disconnect** (サーバからの切断) [put]

- I/put/sc_all/disconnect

2.2.15. **jcs** (任意の JCS を実行する) [put]

- I/put/sc_all/jcs_XXX.jcs

XXX.jcs を実行します

2.2.16. **HALT** (Accidental halt の発生) [put]

- I/put/sc_all/HALT

動作中の軸を則停止し、解除命令があるまでコマンドの実行、受付を禁止します。

(I/put/sc_all/accident を返します)

accident.txt を作成し、中に発生日時を記録します。

l/put/sc_all/recover コマンドのみで解除することができます。サーバを再起動しても、解除できません。

2.2.17. **recover** (Accidental halt の解除) [put]

- l/put/sc_all/recover
Accidental halt を解除します
(accident.txt を削除します)

2.2.18. **QUIT** (サーバの終了) [put]

- l/put/sc_all/QUIT
サーバを終了します。動作中の軸は則停止します。
(accident.txt を作成し、中に発生日時を記録します)

3. JCS(Job Control Script)

3.1. JCS について

JCS は動作を 1 動作 1 行で、シーケンシャルに記述するファイルです。

記述方法は、/(スラッシュ)で区切り、

軸/機能/動作設定/移動速度/移動パルス数・移動方向/相対・絶対位置/単・複数の軸動作
の順で記述します。

#で始まる行はコメント行として読み飛ばします。

その他、:で始まるコマンドを受け付けます。(JCS で使用できるコマンド章を参照)

3.2. JCS リファレンス

・軸:

lhead
thead
tarm
xtray
ytray
estage
hstage
tstage
counter ¹
gauge ¹
spring ²
cryo ³

・機能:

MOTION (動作) ^{1, 2, 3}は、指定不可能

WRITECOUNTER (カウンタ値設定)
CLEARCOUNTER (カウンタ値ゼロクリア)
OFFSET (gauge のオフセットリセット) ²のみ指定可能
OPEN (クライオシャッター開) ³のみ指定可能
CLOSE (クライオシャッター閉) ³のみ指定可能

・動作設定 : (MtrnpSetMotion のヘルプも参考にしてください。 については、動作未確認)

JOG (連続動作)
ORG (原点復帰動作)
PTP (PTP 動作)
TIMER (タイマ動作)
SINGLE_STEP (シングルステップ)
ORG_SEARCH (原点サーチ動作)
ORG_EXIT (原点抜け出し動作)
ORG_ZERO (0点復帰動作)

・移動速度 :

0.073 ~ 1000000[Hz]

・移動パルス数、移動方向 :

JOG, ORG, SINGLE_STEP, ORG_EXIT:	MTR_CW (正の値): +方向
	MTR_CCW (負の値): -方向
PTP:	-134217728 ~ +134217727
TIMER:	1 ~ +134217727
ORG_SEARCH:	1 ~ +134217727
	-1 ~ -134217727
ORG_ZERO:	未使用

次の ABS/REL にしたがって、パルス数を記述します。

・相対・絶対 :

ABS(絶対位置)
REL(相対位置)

・単・複数の軸動作 :

SINGLE or S (単一軸動作)
MULTI or M (MULTI から次の間に書かれた複数の軸を同時駆動)
END or E (MULTI と対)

3.3. JCS の記述例

comment

```

lhead/MOTION/PTP/10000/-4000/REL/SINGLE
tarm/MOTION/PTP/10000/12000/ABS/SINGLE
lhead/MOTION/PTP/3333/2000/REL/MULTI
thead/MOTION/PTP/10000/7110/REL/END
# thead dtm
thead/MOTION/PTP/10000/-500/REL/SINGLE
thead/MOTION/ORG/500/1/REL/SINGLE
thead/CLEARCOUNTER

```

3.4. JCS で使用できる変数

3.4.1. alias.txt

JCS の中で頻繁に使う値は、変数値として置くことができます。変数値は alias.txt に記述しておきます。また、alias.txt 中では、tary .txt 中の値を反映する事や、計算式も使用できます。

[用例]

```

Demo 1
T_Lh 11300
Pickup_T_Lh1 ($T_Lh)-1340+($Demo)*100
    Pickup_T_Lh1 に 10060 が代入されます。

```

```

Ytray_Ln2 $Tray1(23,y)-6000
    tray1*.txt の 23 番穴の y 値から、6000 を引いた値が代入されます。

```

(注意)

alias.txt 中で定義した変数(T_Lh)等を使うには、必ず、(\$T_Lh)のように、\$を付け、()で囲んでください。

[関数および演算子について]

この演算ルーチンは、内部的にTclインタプリタを利用していますので、以下に示すような、Tcl の expr コマンド中で有効な関数および演算子が使用することができます。

数学関数：

acos	cos	hypot	sinh
asin()	cosh()	log()	sqrt()
atan()	exp()	log10()	tan()
atan2()	floor()	pow()	tanh()
ceil()	fmod()	sin()	

絶対値：abs(arg)

倍精度値：double(arg)

整数値 : int(arg)
乱数値 : rand()
整数への丸め値 : round(arg)
乱数の初期値 : srand(arg)

演算子 :

- + ~ ! (負符号、正符号、補数、否定)
* / % (乗算、除算、剰余)
+ - (加算、減算)
<< >> (左シフト、右シフト)
< > <= >= (左不等、右不等、等価左不等、等価右不等)
== != (等価、非等価)
& (ビット積(AND))
^ (ビット差(XOR))
| (ビット和(OR))
&& (積結合)
|| (和結合)
x?y:z (条件)

alias.txt 中の以下の変数値は特別な意味を持ちます。(削除しないでください)

・ SpringLimitMAX, SpringLimitMIN

ロボット動作中に gauge 値が、SpringLimitMAX 以上、または SpringLimitMIN 以下になると、ロボットは Accidental halt を発生して、緊急停止します。

3.5. JCS で使用できるコマンド

JCS では、以下のコマンドを使用することができます。コマンド行は、:で始まるように記述します。

3.5.1. IF・ELSE・ENDIF (条件分岐)

条件により、JCS の処理の分岐を行ないたい場合に使用します。ELSE は省略することができます。

[書式]

:IF/番号/条件式

番号は、0~99 までの任意の整数値を入れます。番号により区別しています。条件式は、例えば、(Ihead>5000)を条件にしたい場合は、Ihead/GT/5000 のように書きます。条件式左項には、以下に示す軸名、各軸のリミットスイッチ名その他、alias.txt に記述した変数が使用できます。中項には、比較演算子を記入してください。右項には、alias.txt に記述した変数、数値が使用できます。

(比較演算子) EQ (=), NE (!=), LE (<=), LT (<), GE (>=), GT (>)

(軸名) Ihead, thead, tarm, xtray, ytray, estage, counter, gauge, spring

(各軸のリミットスイッチ名)

軸名	+EL	ORG	-EL
Ihead	Ihead_ELp	Ihead_ORG	Ihead_ELm
tarm	tarm_ELp	tarm_ORG	tarm_ELm
xtray	xtray_ELp	xtray_ORG	xtray_ELm
ytray	ytray_ELp	ytray_ORG	ytray_ELm

(リミットスイッチステータス)

- ・入光時 --- 0
- ・遮光時 --- 1

[用例 1]

```

:IF/1/Ihead/GT/0                                条件判断
Ihead/MOTION/PTP/5000/3000/REL/SINGLE          真なら
:ELSE/1
Ihead/MOTION/PTP/5000/-3000/REL/SINGLE         偽なら
:ENDIF/1

```

[用例 2]

```

Ihead/MOTION/PTP/4000/-30000/REL/SINGLE
:IF/0/Ihead_ELm/NE/1                            Ihead 軸-EL が遮光されていなければ
:MES/Ihead_ELm was not detect.
:HALT
:ENDIF/0

```

[用例 3] (多重構造)

```

:IF/1/Ihead/GT/0                                条件判断 1
Ihead/MOTION/PTP/5000/3000/REL/SINGLE
:IF/2/tarm/GT/12500                             条件判断 2
tarm/MOTION/PTP/5000/-5000/REL/SINGLE
:ELSE/2
tarm/MOTION/PTP/5000/5000/REL/SINGLE
:ENDIF/2
:ELSE/1
Ihead/MOTION/PTP/5000/-3000/REL/SINGLE
:ENDIF/1

```

次項の、GOTO と組み合わせできます。

3.5.2. GOTO (指定場所へ処理を移す)

GOTO 行は、JCS の処理をラベルのついた行に移したい場合に使用します。ラベルは、GOTO の前後を問いません。(GOTO よりも前に処理を移す場合は、無限ループにならないように、注意してください)

[書式]

:GOTO/ラベル

:ラベル

ラベルは任意の文字列(100 文字未満)です。

[用例 1]

```
Ihead/MOTION/PTP/5000/3000/REL/SINGLE
:GOTO/XXXXX                                XXXXX ラベル行にジャンプ
Ihead/MOTION/PTP/5000/-3000/REL/SINGLE      この行は実行されない
:XXXXX
tarm/MOTION/PTP/5000/12500/ABS/SINGLE
```

[用例 2] (IF との組み合わせにより、複雑な分岐が可能になります)

```
:IF/1/Ihead/GT/0
Ihead/MOTION/PTP/5000/3000/REL/SINGLE
:IF/2/gauge/GT/2000
:GOTO/SKIP1
:ELSE/2
:GOTO/SKIP2
:ENDIF/2
:ELSE/1
Ihead/MOTION/PTP/5000/-3000/REL/SINGLE
:ENDIF/1
:GOTO/END
##### Comment Line
:SKIP1
Ihead/MOTION/PTP/500/-100/REL/SINGLE
:GOTO/END
##### Comment Line
:SKIP2
Ihead/MOTION/PTP/500/100/REL/SINGLE
:GOTO/END
##### Comment Line
:END
```

3.5.3. CALC (演算)

CALC 行は、JCS の中で計算を行ないたいときに使用します。CALC 行で計算した結果は、\$_で参照することができます。CALC 行の中では、軸のカレントの値および、alias.txt で記述された変数値を使用できます。

[書式]

:CALC/計算式

[用例 1]

:CALC/(2+3)*2

\$_に 10 が代入されます。

[用例 2] (alias.txt の変数値の代入)

:CALC/((Lh_speed)+(Th_speed))*2

((Lh_speed)+(Th_speed))*2 の結果が、\$_に代入されます。

[用例 3] (軸のカレント値の代入)

:CALC/((Ihead)+(spring))*2

((Ihead)+(spring))の結果が、\$_に代入されます。

(使用可能な軸名: Ihead, thead. tarm, xtray, ytray, counter, gauge, spring)

[計算結果の使用方法]

:CALC/(\$Pickup_T_Lh1)+200

thead/MOTION/PTP/1000/\$_/ABS/SINGLE

(注意)

CALC 行の中で、alias.txt で定義した変数(Lh_speed)等を使うには、必ず、(Lh_speed)のように、()で囲んでください。

\$Lh_speed+\$Th_speed

は、

(\$Lh_speed)+(\$Th_speed)

のように書いてください。

軸のカレント値も同様に、()で囲んでください。

()例えば、(0.9*gauge)を計算する場合、gauge を()で囲まないと、gauge=-1000 のとき、(0.9*-1000)となってしまう、思い通りの計算がなされない可能性があります。

[関数および演算子について]

この演算ルーチンは、内部的にTcl インタプリタを利用していますので、以下に示すような、Tcl の expr コマンド中で有効な関数および演算子を使用することができます。

数学関数：

acos	cos	hypot	sinh
asin()	cosh()	log()	sqrt()
atan()	exp()	log10()	tan()
atan2()	floor()	pow()	tanh()
ceil()	fmod()	sin()	

絶対値：abs(arg)

倍精度値：double(arg)

整数値：int(arg)

乱数値：rand()

整数への丸め値：round(arg)

乱数の初期値：srand(arg)

演算子：

- + ~ !	(負符号、正符号、補数、否定)
* / %	(乗算、除算、剰余)
+ -	(加算、減算)
<< >>	(左シフト、右シフト)
< > <= >=	(左不等、右不等、等価左不等、等価右不等)
== !=	(等価、非等価)
&	(ビット積(AND))
^	(ビット差(XOR))
	(ビット和(OR))
&&	(積結合)
	(和結合)
x?y:z	(条件)

3.5.4. SAVE (カレント値の保存)

SAVE が呼び出されると、その時点での counter, gauge, spring の値が savefile_*****.txt に追記されます。ロボットの動作確認、JCS のデバッグ等に使用できます。

3.5.5. MES (メッセージの保存)

MES が呼び出されると、その行の " / " 以降の文字が savefile_*****.txt に追記されます。

3.5.6. WARNING (Warning stop)

WARNING が呼ばれると、ロボットの動作を停止し、コマンドの実行、受付を禁止します。

(/put/sc_all/warning を返します)

その時点の日時が warning.txt に記述されます。/put/sc_all/next コマンドを受けると、動作を継続します。また、サーバを再起動すると解除されます。

3.5.7. HALT (Accidental halt)

HALT が呼ばれると、解除命令があるまで動作を停止し、コマンドの実行、受付を禁止します。
(I/put/sc_all/accident を返します)

その時点の日時が warning.txt に記述されます。I/put/sc_all/recover コマンドのみで解除することができます。サーバを再起動しても、解除できません。

4. 更新履歴

Ver.1.0		2003年6月19日	作成
Ver.1.0	4	2005年1月7日	alias.txt 内で計算可能に変更
Ver.1.0	5	2005年2月28日	tray*.txt の値を、alias.txt に反映できるように変更
Ver.1.0	6	2007年7月9日	MES コマンド追加
Ver.1.0	7	2008年4月11日	各軸のリミットスイッチステータスを判断できるように変更
Ver.1.0	8	2008年9月29日	Estage 軸追加
Ver.1.0	9	2008年11月21日	sc_reserve、sc_evacuate 機能追加
Ver.1.0	10	2009年5月19日	IF 文、条件式左項に alias.txt 変数を使用できるように変更
Ver.1.0	11	2010年3月18日	Hstage,Tstage 軸追加、PCI-2756AL(クライオシャッター)追加