

STARS pm16c04 用コマンド集

2007.10.31 版

はじめに

ツジ電子製 PM16C04 ならびに PM16C04-X は 16 台のパルスモータを同時に 4 台まで駆動可能なパルスモータコントローラですが、「チャンネルにモータを選んで駆動する」といった動作が必要となるために若干扱いが面倒です。この STARS クライアント(I/O Client)を使うとチャンネルの選択は pm16c04 I/O Client 側で自動で行ってくれるので、ユーザは「どのモータを選んで」などという事を考える必要がなくなります。

PM16C04 STARS I/O Client(以下、単に PM16C04 とします)の作りはコントローラの先にモータが接続されているような形をイメージしているため、各モータへの命令は、送り先を"PM16C4.dth1"のように階層化したような形式にします。

```
+-----+   (コントローラ)   (モータ名)
| STARS | |-----PM16C04-----| th
| Server| |                               |
+-----+ |                               |
          +-----dth1
          |
          +-----all
          :
          : 略
```

メッセージフォーマット

STARS 経由でパルスモータ pm16c04 (ツジ電子製) を使用するには、下記フォーマットのメッセージを送信して行います。

メッセージ配信先名□コマンド□引数(必要な場合のみ)

※ □は半角スペースを意味しています

例) pm16c04.th□GetValue

メッセージを送った場合は必ず返事 (リプライメッセージを含んだ文字列) が返ってきます。

(メッセージ配信先)>(メッセージ送信元)□@コマンド□引数□値

例) pm16c04.th>test□@GetValue□10000

対応しているメッセージ配信先名は下記の通りです。

メッセージの配信先について

[メッセージ配信先名]

Stars のノード名が pm16c04 の場合（Stars のノード名は pm16c04 I/O Client プログラム起動に指定します）

pm16c04	コントローラコマンド パルスモータのコントローラに対してメッセージを配信します
pm16c04.motorname	モータコマンド motorname という名前のモータに対してメッセージを配信します

エラーメッセージについて

エラーが起こると返事（リプライメッセージ）として下記の形式の文字列が返ってきます。

(メッセージ配信先)>(メッセージ送信元)@コマンド引数Er: (エラー内容を表す文字列)

例) pm16c04.th>test@Preset1000000000Er:DataOutOfRange.

メッセージ配信先を誤って送った場合は下記のエラーを含んだ文字列が返ってきます。

(pm16c04 のノード名)>(メッセージ送信元)@コマンド引数Er: (誤って送ったメッセージ配信先名) isdown.

例) pm16c04.thetGetValue
pm16c04>test@GetValueEr:pm16c04.thetisdown.

用意されていないコマンドもしくは適切でない引数を含んだメッセージを送った場合は下記のエラーを含んだ文字列が返ってきます。

(メッセージ配信先)>(メッセージ送信元)@コマンド引数Er:Badcommandorparameter

例) pm16c04.thetGetValue
pm16c04>test@GetValueEr:Badcommandorparameter

はじめに	1
メッセージフォーマット	1
メッセージの配信先について	2
エラーメッセージについて	2
コントローラコマンド	7
[メッセージ配信先名]	7
pm16c04	7
[コマンド]	7
hello	7
help	7
GetMotorList	8
GetMotorName	8
GetRomVersion	9
◆速度および加減速レート表確認コマンド	9
GetSpeedList (PM16C-04 Only)	9
GetAccRateList	10
◆コントローラ Remote/Local 操作コマンド	10
Remote	10
Local	11
SetFunction	11
◆全モータ速度レベル一括選択コマンド	12
SpeedHigh	12
SpeedMiddle	12
SpeedLow	13
◆チャンネル関連コマンド	13
Select	13
SetHold	14
◆全モーター一括停止コマンド	14
Stop	14
StopEmergency	15
◆2 台同時駆動関連コマンド	15
Standby	15
SyncRun	15
◆イベントメッセージ強制発行コマンド	16
flushdata	16
flushdatatome	16
◆コントローラ現在状況確認コマンド	17
GetFunction	17
GetCtlIsBusy	17

GetChannelStatus (PM16C-04X Only)	18
GetFunctionStatus.....	19
GetStatus.....	20
◆コントローライベント	20
_ChangedCtlIsBusy イベント	20
_ChangedFunction イベント	21
◆ 2 補軸ドライブ機能関連コマンド (PM16C-04X Only)	21
DrawLine	21
DrawLineREL.....	22
DrawCircularCw.....	24
DrawCircularCcw	25
DrawCircularCwREL	26
DrawCircularCcwREL.....	27
DrawArc	29
DrawArcREL.....	30
DrawCircle.....	31
DrawCircleREL	33
モータコマンド	35
[メッセージ配信先名]	35
pm16c04.motorname	35
[コマンド]	35
hello	35
help	35
GetMotorNumber	36
◆モータ速度レベル選択コマンド (PM16C-04X Only)	36
SpeedHigh	36
SpeedMiddle.....	37
SpeedLow	37
GetSpeedSelected	37
◆モータ設定コマンド	38
SetHighSpeed	38
SetMiddleSpeed.....	39
SetLowSpeed	39
SetAccRate	40
SetDigitalCwLs.....	41
SetDigitalCcwLs.....	42
SetCancelBacklash	43
SetJogPulse	43
SetLimits.....	44
SetMotorSetup (PM16C-04X Only)	45

SetStopMode (PM16C-04X Only)	46
GetHighSpeed.....	46
GetMiddleSpeed	47
GetLowSpeed	47
GetAccRate.....	47
GetDigitalCwLs.....	48
GetDigitalCcwLs	48
GetCancelBacklash.....	49
GetJogPulse	49
GetLimits	49
GetMotorSetup (PM16C-04X Only)	50
GetStopMode (PM16C-04X Only)	51
◆HP 関連フラグ設定・読み出しコマンド (PM16C-04X Only)	51
SetHPMode.....	51
SetHPOffset	52
SetHomePosition	52
GetHPOffset	53
GetHomePosition.....	54
ScanHome.....	54
ReScanHome	55
◆タイミングアウト出力機能関連コマンド (PM16C-04X Only)	55
SetTimingOutMode	55
SetTimingOutStart	56
SetTimingOutEnd	56
SetTimingOutInterval	57
GetTimingOutMode.....	58
GetTimingOutStart	58
GetTimingOutEnd	58
GetTimingOutInterval.....	59
SetTimingOutReady	59
GetTimingOutReady.....	59
◆自動速度変更機能関連コマンド (PM16C-04X Only)	60
SetAutoChangeSpeed	60
GetAutoChangeSpeed.....	62
SetAutoChangeSpeedReady	64
GetAutoChangeSpeedReady	64
◆モータ座標のプリセットコマンド.....	65
Preset	65
◆モータ駆動・停止コマンド	65
JogCw.....	66

JogCcw	66
ScanCwConst.....	66
ScanCcwConst.....	67
ScanCw.....	67
ScanCcw	67
ScanCwHome	67
ScanCcwHome	68
SetValue	68
SetValueREL	69
Stop.....	70
StopEmergency.....	70
◆モータ駆動中コマンド	70
SetSpeedCurrent (PM16C-04X Only)	70
◆モータ現在状況確認コマンド	71
GetSelected.....	71
GetValue	71
IsBusy	72
◆モータイベント	72
_ChangedValue イベント	72
_ChangedIsBusy イベント	73
<補足>PM16C-04X について PM16C-04 からの変更点	74
PM16C-04X で使用できない PM16C04 のコマンド.....	74
GetSpeedList	74
PM16C-04 と PM16C-04X で仕様の異なるコマンド	74
GetHighSpeed/GetMiddleSpeed/GetLowSpeed/SetHighSpeed/SetMiddleSpeed/SetLowSpeed.....	74
GetAccRateList/GetAccRate/SetAccRate	74
GetLimits/SetLimits	74
GetValue/SetValue/SetValueREL/Preset/GetDigitalCwLs/SetDigitalCwLs/GetDigitalCcwLs/SetDigitalCcwLs ...	74
Select	74
<補足>PM16C-04X の新機能.....	75
モータ移動形式の追加 (モータコマンド)	75
モータ単位での速度レベルの選択 (モータコマンド)	75
原点検索機能コマンドの追加 (モータコマンド)	75
タイミングアウト出力機能 (モータコマンド)	75
モータ動作中の速度変更 (モータコマンド)	75
自動速度変更機能関連コマンド (モータコマンド)	75
2 補軸ドライブ機能関連 (コントローラコマンド)	75

コントローラコマンド[メッセージ配信先名]**pm16c04**

コントローラに対して命令（コマンド）を送信する場合の配信先名

[コマンド]**hello**

STARS の通信が行われているかをチェックするコマンド。

このコマンドを送信すると '@hello nice to meet you.' の文字列を返します。

[例]

(送信側)

pm16c04□hello

(返信されてくる文字列)

pm16c04>test□hello□@hello□nice□to□meet□you.

help

引数を指定しない場合はコントローラコマンドの一覧を返します。

引数にコマンド名を指定した場合はコマンドのヘルプを返します。

[例]

(送信側)

pm16c04□help

コントローラコマンドの一覧を表示します

(返信されてくる文字列)

pm16c04>test□@help□GetAccRate□GetAccRateCode□GetAccRateList□GetAutoChangeSpeed□
GetAutoChangeSpeedReady□GetCancelBacklash□GetDigitalCwLs□GetDigitalCwLs□GetHPMode□GetHP
Offset□GetHighSpeed□GetHomePosition□GetJogPulse□GetLimits□GetLowSpeed□GetMiddleSpeed□GetM
otorNumber□GetMotorSetup□GetSelected□GetSpeedSelected□GetStopMode□GetTimingOutEnd□GetTimin
gOutInterval□GetTimingOutMode□GetTimingOutReady□GetTimingOutStart□GetValue□IsBusy□JogCw□J
ogCw□Preset□ReScanHome□ScanCw□ScanCwConst□ScanCwHome□ScanCw□ScanCwConst□ScanCw
Home□ScanHome□SetAccRate□SetAutoChangeSpeed□SetAutoChangeSpeedReady□SetCancelBacklash□S
etDigitalCwLs□SetDigitalCwLs□SetHPMode□SetHPOffset□SetHighSpeed□SetHomePosition□SetJogPulse
□SetLimits□SetLowSpeed□SetMiddleSpeed□SetMotorSetup□SetSpeedCurrent□SetStopMode□SetTimingOu
tEnd□SetTimingOutInterval□SetTimingOutMode□SetTimingOutReady□SetTimingOutStart□SetValue□SetV
alueREL□SpeedHigh□SpeedLow□SpeedMiddle□Stop□StopEmergency□_ChangedCtlIsBusy□_ChangedFun
ction□_ChangedIsBusy□_ChangedValue□hello□help

(送信側)

pm16c04□help□hello

hello コマンドのヘルプを表示します

(返信されてくる文字列)

```
pm16c04>test @help hello Usage: hello ¥n The client returns "@hello nice to meet you."
¥n¥n
```

(送信側)

```
pm16c04 help helo                                引数として該当しないコマンドを指定した場合
pm16c04>test @help helo Er: Command "helo" not found.
```

GetMotorList

このコマンドを送信すると 16 個のモータ名称の一覧をスペース区切りで返します。

[リプライ・メッセージ]

```
@GetMotorList (モータ No.0 の名称) (モータ No.1 の名称) ... (モータ No.15 の名称)
```

[例]

(送信側)

```
pm16c04 GetMotorList
```

(返信されてくる文字列)

```
pm16c04>test @GetMotorList th dth1 d1 al1 Mt4 d2 al2 Mt7 Mt8 Mt9 Mta Mtb Mtc
Mtd Mte Mtf
```

左から順にモータ No.0 の名称'th'、No.1 の名称'dth1'...以降 No.15 の名称'Mtf'を返します

GetMotorName

モータ No.に対応するモータ名を返します。

[引数]

モータ No. 0 から 15 の数字文字列

[リプライ・メッセージ]

```
@GetMotorName (引数のモータ No.) (モータ名)
```

データ取得が正常におこなわれた場合

```
@GetMotorName Er: Bad command or parameter.
```

引数のモータ No.が指定されなくてエラーの場合

```
@GetMotorName (引数) Er: Bad motor number.
```

モータ No.が間違っていてエラーの場合

[例]

(送信側)

```
pm16c04 GetMotorName 0
```

モータ No.0 のモータ名を問い合わせます

(返信されてくる文字列)

pm16c04>test□@GetMotorName□0□th

モータ No.0 のモータ名'th'が返ってきます

GetRomVersion

ツジ製 PM16C-04 シリーズの場合は、このコマンドを送信するとコントローラのシステムプログラムのバージョンと日付を返します。

ツジ製 PM16C-04X シリーズの場合は、このコマンドを送信するとコントローラのシステムプログラムのバージョンと日付に加えて製品識別子'PM16C-04X'を返します。

[例]

(送信側)

pm16c04□GetRomVersion

(返信されてくる文字列)

pm16c4x>test□@GetRomVersion□1.13□07-09-13□PM16C-04X [PM16C-04X シリーズの場合]

pm16c04>test□@GetRomVersion□ 1.00□04-10-02 [PM16C-04 シリーズの場合]

◆速度および加減速レート表確認コマンド**GetSpeedList (PM16C-04 Only)**

このコマンドは PM16C-04 のみ有効です。PM16C-04X では無効です。

PM16C-04 では速度コード表と対応する値のみが速度として設定できます。

このコマンドを送信すると、PM16C-04 の速度コード表に基づく有効なモータの速度 (PPS) の一覧を返します。

(ちなみに PM16C-04X では速度コード表が撤廃され、1~5,000,000 までの速度値を直接設定できるようになりました)

[リプライ・メッセージ]

@GetSpeedList□ (有効なモータ速度 PPS 1) □ (有効なモータ速度 PPS 2) ...□ (有効なモータ速度 PPS N) [PM16C-04 シリーズの場合]

@GetSpeedList Ng: Set number from 1 to 5000000 in PM16C-04X.

[PM16C-04X シリーズの場合]

[例 : PM16C-04 シリーズの場合]

(送信側)

pm16c04□GetSpeedList

(返信されてくる文字列)

```
pm16c04>test□@GetSpeedList 5 10 25 50 75 100 150 200 250 300 350 400 450 500 550
600 650 700 750 800 900 1000 1100 1200 1300 1400 1500 1600 1700 1800 1900 2000
2100 2200 2300 2400 2500 2600 2700 2800 2900 3000 3100 3200 3300 3400 3500 3600
```

3700 3800 3900 4000 4100 4200 4300 4400 4500 4600 4700 4800 4900 5000 5100 5200
 5300 5400 5500 5600 5700 5800 5900 6000 6100 6200 6300 6400 6500 6600 6700 6800
 6900 7000 7100 7200 7300 7400 7500 7600 7700 7800 7900 8000 8200 8400 8600 8800
 9000 9200 9400 9600 9800 10000 10200 10400 10600 10800 11010 11210 11410 11600
 11800 11990 12200 12400 12600 12790 12990 13200 13400 13620 13810 14000 14200 14
 400 14620 14830 15010 15200 15390 15580 15770 15970 16180 16400 16610 16830 17060
 17240 17420 17600 17800 17990 18180 18380 18660 18940 19230 19530 19840 20160
 20500 20830 21190 21550 21930 22320 22730 23150 23590 24040 24510 25000 25510 26040
 26600 27170 27620 28090 28570 29070 29590 30120 30680 31250 31850 32470 33110 33780
 34480 35210 35970 36500 37040 37600 38170 38760 39370 40000

GetAccRateList

このコマンドを送信すると有効なモータの速度の加減速レートの一覧を返します。

[リプライ・メッセージ]

@GetAccRateList□ (有効なモータ速度の加減速率 1) □ (有効なモータ速度の加減速率 2)
 ...□ (有効なモータ速度の加減速率 N)

[例]

(送信側)

pm16c04□GetAccRateList

(返信されてくる文字列)

[PM16C-04X シリーズの場合]

pm16c04>test□@GetAccRateList□1000□910□820□750□680□620□560□510□470□430□390□360□
 330□300□270□240□220□200□180□160□150□130□120□110□100□91□82□75□68□62□56□51□47
 □43□39□36□33□30□27□24□22□20□18□16□15□13□12□11□10□9.1□8.2□7.5□6.8□6.2□5.6□5.1□4.
 7□4.3□3.9□3.6□3.3□3.0□2.7□2.4□2.2□2.0□1.8□1.6□1.5□1.3□1.2□1.1□1.0□0.91□0.82□0.75□0.68□0.62
 □0.56□0.51□0.47□0.43□0.39□0.36□0.33□0.30□0.27□0.24□0.22□0.20□0.18□0.16□0.15□0.13□0.12□
 0.11□0.10□0.091□0.082□0.075□0.068□0.062□0.056□0.051□0.047□0.043□0.039□0.036□0.033□0.0
 3□0.027□0.024□0.022□0.020□0.018□0.016

[PM16C-04 シリーズの場合]

pm16c04>test□@GetAccRateList 1000 800 600 500 400 300 200 150 125 100 75 50 30
 20 15 10 7.5 5 3

◆コントローラ Remote/Local 操作コマンド

Remote

このコマンドを送信することで Remote 状態に設定します。

[リプライ・メッセージ]

@Remote□Ok:

正常に動作を終了した場合

@Remote□Er:□Could□not□changed□function. なんらかの理由で Remote に変更できなかった場合

[例]

(送信側)

pm16c04□Remote	Remote 状態に設定します
pm16c04□GetFunction	Remote/Local の状態を確認します

(返信されてくる文字列)

pm16c04>test□@Remote□Ok:	
pm16c04>test□@GetFunction□1	Remote 状態です

Local

このコマンドを送信することで Local 状態に設定します。

[リプライ・メッセージ]

@Local□Ok:	正常に動作を終了した場合
@Local□Er:□Could□not□changed□function.	なんらかの理由で Local に変更できなかった場合

[例]

(送信側)

pm16c04□Local	Local 状態に設定します
pm16c04□GetFunction	Remote/Local の状態を確認します

(返信されてくる文字列)

pm16c04>test□@Remote□Ok:	
pm16c04>test□@GetFunction□0	Local 状態です

SetFunction

このコマンドを送信することで Remote もしくは Local 状態に設定します。

それぞれ Remote コマンドもしくは Local コマンドと同一の機能です。

[引数]

0	Local
1	Remote

[リプライ・メッセージ]

@SetFunction□ (引数) □Ok:	正常に動作を終了した場合
@SetFunction□Er:□Could□not□changed□function.	なんらかの理由で変更できなかった場合

[例]

(送信側)

pm16c04□SetFunction□1	Remote 状態に設定します
-----------------------	-----------------

(返信されてくる文字列)

```
pm16c04>test□@SetFunction□1□Ok:
```

```
pm16c04>test□@GetFunction□1 Remote 状態です
```

◆全モータ速度レベル一括選択コマンド

以下のコマンドを使って全てのモータの速度レベルを一括で設定します。

PM16C-04X シリーズからは、モータ個別に速度レベルを設定できるようになりました。

PM16C-04 シリーズでは、モータ個別の速度レベルの設定はサポートされていません。

SpeedHigh

このコマンドを送信することで、以降全てのモータの速度が' High' の設定で動きます。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータが 1 つでも Busy 状態のときはエラーとなり実行されません。

[リプライ・メッセージ]

```
@SpeedHigh□Ok:
```

正常に動作を終了した場合

```
@SpeedHigh□Er:□Busy.
```

モータが 1 つでも Busy 状態のため実行されなかった場合

```
@SpeedHigh□Er:□Offline.
```

Local 状態で実行した場合

[例]

(送信側)

```
pm16c04□SpeedHigh
```

(返信されてくる文字列)

```
pm16c04>test□@SpeedHigh□Ok:
```

SpeedMiddle

このコマンドを送信することで、以降全てのモータの速度が'Middle'で動きます。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータが 1 つでも Busy 状態のときはエラーとなり実行されません。

[リプライ・メッセージ]

```
@SpeedMiddle□Ok:
```

正常に動作を終了した場合

```
@SpeedMiddle□Er:□Busy.
```

モータが 1 つでも Busy 状態のため実行されなかった場合

```
@SpeedMiddle□Er:□Offline.
```

Local 状態で実行した場合

[例]

(送信側)

```
pm16c04□SpeedMiddle
```

(返信されてくる文字列)

pm16c04>test□@SpeedMiddle□Ok:

SpeedLow

このコマンドを送信することで、以降全てのモータの速度が'Low'で動きます。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータが 1 つでも Busy 状態のときはエラーとなり実行されません。

[リプライ・メッセージ]

@SpeedLow□Ok:

正常に動作を終了した場合

@SpeedLow□Er:□Busy.

モータが 1 つでも Busy 状態のため実行されなかった場合

@SpeedLow□Er:□Offline.

Local 状態で実行した場合

[例]

(送信側)

pm16c04□SpeedLow

(返信されてくる文字列)

pm16c04>test□@SpeedLow□Ok:

◆チャンネル関連コマンド

Select

このコマンドを送信すると手動でチャンネルにモータを割り当てることができます。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータもしくはチャンネルが Busy 状態のときはエラーとなり実行されません。

[引数]

- | | |
|------------|------------------|
| 1. チャンネル | A、B、C、D のいずれかの文字 |
| 2. モータ No. | 0 から 15 の数字文字列 |

[リプライ・メッセージ]

@Select□ (引数のチャンネル) □ (引数のモータ No.) □Ok: 正常に動作を終了した場合

@Select□ (引数のチャンネル) □ (引数のモータ No.) □Er:□Busy.

モータもしくはチャンネル A が Busy 状態のため実行されなかった場合

@ Select□ (引数のチャンネル) □ (引数のモータ No.) □Er:□Offline.

Local 状態で実行した場合

[例]

(送信側)

正常に動作を終了した場合

(返信されてくる文字列)

pm16c04>test□@Stop□@Ok:

正常に動作を終了した場合

StopEmergency

このコマンドを送信すると全てのモータが緊急停止します。

このコマンドはコントローラが Local 状態の場合操作は無視されます。

[リプライ・メッセージ]

@StopEmergency□Ok:

正常に動作を終了した場合

[例]

(送信側)

pm16c04□StopEmergency

全てのモータを緊急停止します

(返信されてくる文字列)

pm16c04>test□@StopEmergency□@Ok:

正常に動作を終了した場合

◆ 2 台同時駆動関連コマンド**Standby**

2 台同時駆動 Standby 状態にします。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

[リプライ・メッセージ]

@Standby□Ok:

正常に動作を終了した場合

@Standby□Er:□Offline.

Local 状態で実行した場合

[例]

(送信側)

pm16c04□Standby

2 台同時駆動 Standby 状態にします。

(返信されてくる文字列)

pm16c04>test□@Standby□@Ok:

正常に動作を終了した場合

SyncRun

このコマンドを送信すると 2 台同時駆動の動作を開始します。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

[リプライ・メッセージ]

@SyncRun□Ok:

正常に動作を終了した場合

@SyncRun□Er:□Offline.

Local 状態で実行した場合

[例]

(送信側)

pm16c04□SyncRun

2 台同時駆動の動作を開始します

(返信されてくる文字列)

pm16c04>test□@SyncRun□@Ok:

正常に動作を終了した場合

◆イベントメッセージ強制発行コマンド

flushdata

このコマンドを送信するとコントローラおよびモータの全てのステータス情報をイベントメッセージとして Stars の TAK サーバ'System'に返します。

ステータス情報をイベントメッセージとして受け取るには、この当コマンドを発行する前に Stars の TAK サーバ'System'に対してイベントメッセージ配信依頼のコマンドを送信しておく必要があります。

[リプライ・メッセージ]

@flushdata□Ok:

コマンドが正常に送信された場合

[例]

(送信側)

System□flgon□pm16c04

コントローラのイベントメッセージの配信を
依頼します

System□flgon□pm16c04.th

モータ名'th'のイベントメッセージの配信を
依頼します

pm16c04□flushdata

イベントメッセージ配信の実行を依頼します

(返信されてくる文字列)

pm16c04□@flushdata□Ok:

コマンドが正常送信されました

pm16c04>test□_ChangedCtlIsBusy□0

モータ全てが Busy か否かがイベントメッ
セージの値として返ってきます

pm16c04>test□_ChangedFunction□1

コントローラが Remote か Local かがイベ
ントメッセージの値として返ってきます

pm16c04.th>test□_ChangedIsBusy□0

モータ名 th の Busy 状態がイベントメッ
セージの値として返ってきます

pm16c04.th>test□_ChangedValue□100

モータ名 th の現在値がイベントメッセー
ジの値として返ってきます

flushdatatome

このコマンドを送信するとコントローラおよびモータの全てのステータス情報をイベントメッセージとして当コマンドの Stars 送信元に直接返します。

[例]

(送信側)

pm16c04□flushdatatome

イベントメッセージ配信の実行を依頼します

(返信されてくる文字列)

pm16c04□@flushdatatome□Ok:

コマンドが正常送信されました

pm16c04>test□_ChangedCtlIsBusy□0

モータ全てが **Busy** か否かがイベントメッセージの値として返ってきます

pm16c04>test□_ChangedFunction□1

コントローラが **Remote** か **Local** かがイベントメッセージの値として返ってきます

pm16c04.th>test□_ChangedIsBusy□0

モータ 0 の th の **Busy** 状態がイベントメッセージの値として返ってきます

pm16c04.th>test□_ChangedValue□100

モータ 0 の th の現在値がイベントメッセージの値として返ってきます

pm16c04.DTH>test□_ChangedIsBusy□0

モータ 1 の DTH の **Busy** 状態がイベントメッセージの値として返ってきます

pm16c04.DTH>test□_ChangedValue□100

モータ 1 の DTH の現在値がイベントメッセージの値として返ってきます

...

以下、モータ No. 2 から 1 5 までの **_ChangedIsBusy** イベントと **_ChangedValue** イベントが返されます。（例では省略しています）

◆コントローラ現在状況確認コマンド

GetFunction

このコマンドを送信することで **Remote/Local** の状態を取得します。

[リプライ・メッセージ]

@GetFunction□0

Local

@GetFunction□1

Remote

[例]

(送信側)

pm16c04□GetFunction

Remote/Local の状態を確認します

(返信されてくる文字列)

pm16c04>test□@GetFunction□0

Local 状態の場合

pm16c04>test□@GetFunction□1

Remote 状態の場合

GetCtlIsBusy

このコマンドを送信することで 4 チャンネル全てが使用中か否かのデータを取得します。

[リプライ・メッセージ]

@GetCtIsBusy□1

4チャンネル全てが使用中である場合

@Get CtlIsBusy□0

空きチャンネルがある場合

[例]

(送信側)

pm16c04□GetCtIsBusy

4チャンネル全てが使用中か否か 状況を確認 します

(返信されてくる文字列)

pm16c04>test□@GetCtIsBusy□0

空きチャンネルがある場合

pm16c04>test□@GetCtIsBusy□1

4チャンネル全てが使用中の場合

GetChannelStatus (PM16C-04X Only)

このコマンドは PM16C-04X のみ有効です。PM16C-04X では無効です。

このコマンドを送信することで全てのチャンネル A～D に関する詳細な情報を返します。

[戻り値]

下記のフォーマット※の文字列を返します。

(R|L)abcd/PNNS/VVVV/HHJJKKLL/+-uu. . /+-vv. . /+-ww. . /+-xx. . .

※ 文字列のフォーマット

R|L R もしくは L (R: Remote L: Local)

abcd 左から A～D それぞれのチャンネルに割り当てられているモータ (値: 0～F)

PNNS 左から A～D それぞれのチャンネルの駆動状態を表す

(P: cw 方向に駆動中 N: ccw 方向に駆動中 S: 停止中)

VVVV 左から A～D それぞれのチャンネルの LimitSwitch 状態を表す 1 6 進数

bit3 Hold off (0: Hold on 1: Hold off)

bit2 HP LS (0: Non Active 1: Active)

bit1 CCW LS (0: Non Active 1: Active)

bit0 CW LS (0: Non Active 1: Active)

HH,JJ,KK,LL 左から A～D それぞれのチャンネルの内部ステータスを表す 2 桁の 1 6 進数

bit7 ESEND (1: 急停止コマンドあり)

bit6 SSEND (1: 減速停止コマンドあり)

bit5 LSEND (1: リミット停止受付)

bit4 COMERR (1: エラー)

bit3 ACCN (1: 減速中)

bit2 ACCP (1: 加速中)

bit1 DRIVE (1: パルス出力中)

bit0 BUSY (1: データ処理中もしくはパルス出力中)

uu..., vv..., ww..., xx... A～D それぞれのチャンネルに割り当てられているモータの
現在位置

[リプライ・メッセージ]

@GetFunctionStatus□ (各 LS／ Remote/Local の状態を表す値)

正常に動作を終了した場合

[例]

(送信側)

pm16c04□GetFunctionStatus

最新ステータス値を取得します

(返信されてくる文字列)

pm16c04>test□@GetFunctionStatus□32631

正常に動作を終了した場合

GetFunctionStatus

このコマンドを送信することで全てのチャンネル A～D の各 LS、ならびに Remote/Local の状態を返します。

[戻り値]

各 LS／ Remote/Local の状態※を表す 1 0 進数の数字文字列を返します。

※ 戻り値の構成

bit15	未使用	
bit14	チャンネル B の HP LS	(1: Non Active 0:Active)
bit13	チャンネル B の CCW LS	(1: Non Active 0:Active)
bit12	チャンネル B の CW LS	(1: Non Active 0:Active)
bit11	Remote/Local ステータス	(0: Local 1:Remote)
bit10	チャンネル A の HP LS	(1: Non Active 0:Active)
bit9	チャンネル A の CCW LS	(1: Non Active 0:Active)
bit8	チャンネル A の CW LS	(1: Non Active 0:Active)
bit7	未使用	
bit6	チャンネル D の HP LS	(1: Non Active 0:Active)
bit5	チャンネル D の CCW LS	(1: Non Active 0:Active)
bit4	チャンネル D の CW LS	(1: Non Active 0:Active)
bit3	未使用	
bit2	チャンネル C の HP LS	(1: Non Active 0:Active)
bit1	チャンネル C の CCW LS	(1: Non Active 0:Active)
bit0	チャンネル C の CW LS	(1: Non Active 0:Active)

[リプライ・メッセージ]

@GetFunctionStatus□ (各 LS／ Remote/Local の状態を表す値)

正常に動作を終了した場合

[例]

(送信側)

pm16c04□GetFunctionStatus

最新ステータス値を取得します

(返信されてくる文字列)

pm16c04>test□@GetFunctionStatus□32631

正常に動作を終了した場合

GetStatus

このコマンドを送信することで該当するチャンネルの最新の内部ステータスを返します。

[引数]

チャンネル A、B、C、D のいずれかの文字

[戻り値]

内部ステータス※を表す 10 進数

※ 戻り値の構成

bit7	ESEND	(1: 急停止コマンドあり)
bit6	SSEND	(1: 減速停止コマンドあり)
bit5	LSEND	(1: リミット停止受付)
bit4	COMERR	(1: エラー)
bit3	ACCN	(1: 減速中)
bit2	ACCP	(1: 加速中)
bit1	DRIVE	(1: パルス出力中)
bit0	BUSY	(1: データ処理中もしくはパルス出力中)

[リプライ・メッセージ]

@GetStatus□ (引数) □ (内部ステータスを表す値)

[例]

(送信側)

pm16c04□GetStatus□A

チャンネル A の最新の内部 CPU
ステータス値を取得します

(返信されてくる文字列)

pm16c04 >test□@GetStatus□A□0

正常に動作を終了した場合

◆コントローライベント**_ChangedCtlIsBusy イベント**

このイベントは 4 チャンネル全てが使用中か否かが切り替わった場合、もしくは "flushdata" コマンド、"flushdatatome" コマンドを発行した場合に Stars Server から送られます。

[イベント・メッセージ]

_ChangedCtlIsBusy□1

4 チャンネル全てが使用中である場合

_ChangedCtlIsBusy□0

[例]

(送信側)

System flgon pm16c04

コントローラのイベント監視を開始します

pm16c04 flushdata

イベントメッセージの強制送信を指示します

(返信されてくる文字列)

System>test□@flgon Node pm16c04 has been registered.

pm16c04>test□_ChangedCtlIsBusy□0

pm16c04>test□_ChangedFunction□1

pm16c04>test□@flushdata□Ok:

_ChangedFunction イベント

このイベントは Remote もしくは Local 状態が切り替わった場合、もしくは "flushdata" コマンド、"flushdatatome" コマンドを発行した場合に Stars Server から送られます。

[イベント・メッセージ]

_ChangedFunction□1

Remote 状態である場合

_ChangedFunction□0

Local 状態である場合

[例]

(送信側)

System flgon pm16c04

コントローラのイベント監視を開始します

pm16c04 flushdata

イベントメッセージの強制送信を指示します

(返信されてくる文字列)

System>test□@flgon Node pm16c04 has been registered.

pm16c04>test□_ChangedCtlIsBusy□0

pm16c04>test□_ChangedFunction□1

pm16c04>test□@flushdata□Ok:

◆ 2 補軸ドライブ機能関連コマンド (PM16C-04X Only)

2 補軸ドライブ機能は PM16C-04X からの新機能です。実行にあたっては、ツジ電子（株）が提供する取扱説明書の該当箇所をよく読んでからおこなってください。

DrawLine

このコマンドを送信することで、2つの軸を X、Y 座標とした平面上の直線を描くように、2つのモータを現在座標から目標座標（絶対座標）まで動かします。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータもしくは選択チャンネルが Busy 状態のときはエラーとなり実行されません。

[引数]

- | | |
|----------------------|---|
| 1. チャンネル選択 (省略可) | 0 もしくは AB ChannelA と B を使います
1 もしくは CD ChannelC と D を使います
省略された場合は、AB もしくは CD いずれかを
自動選択します。 |
| 2. X 軸のモータ No. | 0～15 の数字文字列 |
| 3. Y 軸のモータ No. | 0～15 の数字文字列 |
| 4. X 軸のモータの移動目標の絶対座標 | -999999999 から 999999999 までの数値文字
列 (+符号の指定は不可) |
| 5. Y 軸のモータの移動目標の絶対座標 | -999999999 から 999999999 までの数値文字
列 (+符号の指定は不可) |

[リプライ・メッセージ]

- | | |
|---|--|
| @DrawLine□0□0□1□1000□1000□Ok: | 正常に動作した場合 (チャンネル選択
あり、ここでは AB を選択) |
| @DrawLine□0□1□1000□1000□Ok: | 正常に動作した場合 (チャンネル選択省略
、AB もしくは CD が自動選択される) |
| @DrawLine□0□0□1□1000□1000□Er:□Busy | チャンネル選択時、選択チャンネルが別の
モータに使用されている、もしくは
X 軸もしくは Y 軸のモータが使用されて
いる |
| @DrawLine□0□1□1000□1000□Er:□Busy | チャンネル選択省略時、AB、CD のペアで
の空きチャンネルがない、もしくは
X 軸もしくは Y 軸のモータが使用されて
いる |
| @DrawLine□ (正しい引数) □Er:□Offline. | Local 状態で実行した場合 |
| @DrawLine□17□1□1000□1000□Er: Bad motor number. | X 軸もしくは Y 軸のモータ No.が範囲外 |
| @DrawLine□0□1□1000□999999999□Er:□Data□out□of□range. | X 軸もしくは Y 軸のモータ移動目標座標
が範囲外 |

[例]

(送信側)

pm16c04□DrawLine□0□1□1000□1000

(返信されてくる文字列)

pm16c04>test□@DrawLine□0□1□1000□1000

DrawLineREL

このコマンドを送信することで、2つの軸を X、Y 座標とした平面上の直線を描くように、2つのモータを現在座標から目標座標 (現在座標からの相対座標) まで動かします。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータもしくは選択チャンネルが Busy 状態のときはエラーとなり実行されません。

[引数]

- | | |
|----------------------|---|
| 1. チャンネル選択 (省略可) | 0 もしくは AB ChannelA と B を使います
1 もしくは CD ChannelC と D を使います
省略された場合は、AB もしくは CD いずれかを自動選択します。 |
| 2. X 軸のモータ No. | 0～15 の数字文字列 |
| 3. Y 軸のモータ No. | 0～15 の数字文字列 |
| 4. X 軸のモータの移動目標の相対座標 | --99999999 から 99999999 までの数値文字列 (+符号の指定は不可) |
| 5. Y 軸のモータの移動目標の相対座標 | -99999999 から 99999999 までの数値文字列 (+符号の指定は不可) |

[リプライ・メッセージ]

- | | |
|--|---|
| @DrawLineREL□0□0□1□1000□1000□Ok: | 正常に動作した場合 (チャンネル選択あり、ここでは AB を選択) |
| @DrawLineREL□0□1□1000□1000□Ok: | 正常に動作した場合 (チャンネル選択省略、AB もしくは CD が自動選択される) |
| @DrawLineREL□0□0□1□1000□1000□Er:□Busy | チャンネル選択時、選択チャンネルが別のモータに使用されている、もしくは X 軸もしくは Y 軸のモータが使用されている |
| @DrawLineREL□0□1□1000□1000□Er:□Busy | チャンネル選択省略時、AB、CD のペアでの空きチャンネルがない、もしくは X 軸もしくは Y 軸のモータが使用されている |
| @DrawLineREL□ (正しい引数) □Er:□Offline. | Local 状態で実行した場合 |
| @DrawLineREL□17□1□1000□1000□Er: Bad motor number. | X 軸もしくは Y 軸のモータ No.が範囲外 |
| @DrawLineREL□0□1□1000□999999999□Er:□Data□out□of□range. | X 軸もしくは Y 軸のモータ移動目標座標が範囲外 |

[例]

(送信側)

```
pm16c04□DrawLineREL□0□1□1000□1000
```

(返信されてくる文字列)

```
pm16c04>test□@DrawLineREL□0□1□1000□1000
```

DrawCircularCw

このコマンドを送信することで、2つの軸を X、Y 座標とした平面上の円弧を描くように、2つのモータを中心座標を元に現在座標から目標座標（絶対座標）まで CW 方向に向かって動かします。このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。またこのコマンドはモータもしくは選択チャンネルが Busy 状態のときはエラーとなり実行されません。

[引数]

- | | |
|----------------------|---|
| 1. チャンネル選択（省略可） | 0 もしくは AB ChannelA と B を使います
1 もしくは CD ChannelC と D を使います
省略された場合は、AB もしくは CD いずれかを自動選択します。 |
| 2. X 軸のモータ No. | 0～15 の数字文字列 |
| 3. Y 軸のモータ No. | 0～15 の数字文字列 |
| 4. X 軸のモータの移動目標の絶対座標 | -999999999 から 999999999 までの数値文字列（+符号の指定は不可） |
| 5. Y 軸のモータの移動目標の絶対座標 | -999999999 から 999999999 までの数値文字列（+符号の指定は不可） |
| 6. X 軸のモータの円弧の中心絶対座標 | -999999999 から 999999999 までの数値文字列（+符号の指定は不可） |
| 7. Y 軸のモータの円弧の中心絶対座標 | -999999999 から 999999999 までの数値文字列（+符号の指定は不可） |

[リプライ・メッセージ]

- | | |
|--|---|
| @DrawCircularCw□0□0□1□1000□1000□0□0□Ok: | 正常に動作した場合（チャンネル選択あり、ここでは AB を選択） |
| @DrawCircularCw□0□1□1000□1000□0□0□Ok: | 正常に動作した場合（チャンネル選択省略、AB もしくは CD が自動選択される） |
| @DrawCircularCw□0□0□1□1000□1000□0□0□Er:□Busy | チャンネル選択時、選択チャンネルが別のモータに使用されている、もしくは X 軸もしくは Y 軸のモータが使用されている |
| @DrawCircularCw□0□1□1000□1000□0□0□Er:□Busy | チャンネル選択省略時、AB、CD のペアでの空きチャンネルがない、もしくは X 軸もしくは Y 軸のモータが使用されている |
| @DrawCircularCw□（正しい引数）□Er:□Offline. | Local 状態で実行した場合 |
| @DrawCircularCw□17□1□1000□1000□0□0□Er: Bad motor number. | |

X 軸もしくは Y 軸のモータ No.が範囲外

@DrawCircularCw001099999999999910000000Er:Dataoutofrange.

X 軸もしくは Y 軸のモータ移動目標座標
が範囲外、

もしくは X 軸もしくは Y 軸のモータ円弧
中心座標が範囲外

[例]

(送信側)

pm16c04@DrawCircularCw0010100010000000

(返信されてくる文字列)

pm16c04>test@DrawCircularCw0010100010000000Ok:

DrawCircularCcw

このコマンドを送信することで、2つの軸を X、Y 座標とした平面上の円弧を描くように、2つのモータを中心座標を元に現在座標から目標座標（絶対座標）まで CW 方向に向かって動かします。このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。またこのコマンドはモータもしくは選択チャンネルが Busy 状態のときはエラーとなり実行されません。

[引数]

- | | |
|----------------------|---|
| 1. チャンネル選択 (省略可) | 0 もしくは AB ChannelA と B を使います
1 もしくは CD ChannelC と D を使います
省略された場合は、AB もしくは CD いずれかを自動選択します。 |
| 2. X 軸のモータ No. | 0～15 の数字文字列 |
| 3. Y 軸のモータ No. | 0～15 の数字文字列 |
| 4. X 軸のモータの移動目標の絶対座標 | -999999999 から 999999999 までの数値文字列 (+符号の指定は不可) |
| 5. Y 軸のモータの移動目標の絶対座標 | -999999999 から 999999999 までの数値文字列 (+符号の指定は不可) |
| 6. X 軸のモータの円弧の中心絶対座標 | -999999999 から 999999999 までの数値文字列 (+符号の指定は不可) |
| 7. Y 軸のモータの円弧の中心絶対座標 | -999999999 から 999999999 までの数値文字列 (+符号の指定は不可) |

[リプライ・メッセージ]

@DrawCircularCcw000010100010000000Ok:	正常に動作した場合 (チャンネル選択あり、ここでは AB を選択)
@DrawCircularCcw0010100010000000Ok:	正常に動作した場合 (チャンネル選択省略、AB もしくは CD が自動選択される)

@DrawCircularCcw□0□0□1□1000□1000□0□0□Er:□Busy	チャンネル選択時、選択チャンネルが別のモータに使用されている、もしくは X 軸もしくは Y 軸のモータが使用されている
@DrawCircularCcw□0□1□1000□1000□0□0□Er:□Busy	チャンネル選択省略時、AB、CD のペアでの空きチャンネルがない、もしくは X 軸もしくは Y 軸のモータが使用されている
@DrawCircularCcw□ (正しい引数) □Er:□Offline.	Local 状態で実行した場合
@DrawCircularCcw□17□1□1000□1000□0□0□Er: Bad motor number.	X 軸もしくは Y 軸のモータ No.が範囲外
@DrawCircularCcw□0□1□999999999□1000□0□0□Er:□Data□out□of□range.	X 軸もしくは Y 軸のモータ移動目標座標が範囲外、 もしくは X 軸もしくは Y 軸のモータ円弧中心座標が範囲外

[例]

(送信側)

pm16c04□DrawCircularCcw□0□1□1000□1000□0□0

(返信されてくる文字列)

pm16c04>test□@DrawCircularCcw□0□1□1000□1000□0□0□Ok:

DrawCircularCwREL

このコマンドを送信することで、2つの軸を X、Y 座標とした平面上の円弧を描くように、2つのモータを中心座標を元に現在座標から目標座標（現在座標からの相対座標）までCW 方向に向かって動かします。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータもしくは選択チャンネルが Busy 状態のときはエラーとなり実行されません。

[引数]

- | | |
|----------------------|---|
| 1. チャンネル選択 (省略可) | 0 もしくは AB ChannelA と B を使います
1 もしくは CD ChannelC と D を使います
省略された場合は、AB もしくは CD いずれかを自動選択します。 |
| 2. X 軸のモータ No. | 0～15 の数字文字列 |
| 3. Y 軸のモータ No. | 0～15 の数字文字列 |
| 4. X 軸のモータの移動目標の相対座標 | -999999999 から 999999999 までの数値文字列 (+符号の指定は不可) |

- 5. Y 軸のモータの移動目標の相対座標 -999999999 から 999999999 までの数値文字列 (+符号の指定は不可)
- 6. X 軸のモータの円弧の中心相対座標 -999999999 から 999999999 までの数値文字列 (+符号の指定は不可)
- 7. Y 軸のモータの円弧の中心相対座標 -999999999 から 999999999 までの数値文字列 (+符号の指定は不可)

[リプライ・メッセージ]

- @DrawCircularCwREL□0□0□1□1000□1000□0□0□Ok: 正常に動作した場合 (チャンネル選択あり、ここでは AB を選択)
- @DrawCircularCwREL□0□1□1000□1000□0□0□Ok: 正常に動作した場合 (チャンネル選択省略、AB もしくは CD が自動選択される)
- @DrawCircularCwREL□0□0□1□1000□1000□0□0□Er:□Busy チャンネル選択時、選択チャンネルが別のモータに使用されている、もしくは X 軸もしくは Y 軸のモータが使用されている
- @DrawCircularCwREL□0□1□1000□1000□0□0□Er:□Busy チャンネル選択省略時、AB、CD のペアでの空きチャンネルがない、もしくは X 軸もしくは Y 軸のモータが使用されている
- @DrawCircularCwREL□ (正しい引数) □Er:□Offline. Local 状態で実行した場合
- @DrawCircularCwREL□17□1□1000□1000□0□0□Er: Bad motor number.
X 軸もしくは Y 軸のモータ No.が範囲外
- @DrawCircularCwREL□0□1□999999999□1000□0□0□Er:□Data□out□of□range.
X 軸もしくは Y 軸のモータ移動目標座標が範囲外、
もしくは X 軸もしくは Y 軸のモータ円弧中心座標が範囲外

[例]

(送信側)

pm16c04□DrawCircularCwREL□0□1□1000□1000□0□0

(返信されてくる文字列)

pm16c04>test□@DrawCircularCwREL□0□1□1000□1000□0□0□Ok:

DrawCircularCwREL

このコマンドを送信することで、2つの軸を X、Y 座標とした平面上の円弧を描くように、2つのモータを中心座標を元に現在座標から目標座標 (現在座標からの相対座標) まで CCW 方向に向かって動かします。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータもしくは選択チャンネルが Busy 状態のときはエラーとなり実行されません。

[引数]

- | | |
|----------------------|---|
| 1. チャンネル選択 (省略可) | 0 もしくは AB ChannelA と B を使います
1 もしくは CD ChannelC と D を使います
省略された場合は、AB もしくは CD いずれかを自動選択します。 |
| 2. X 軸のモータ No. | 0～15 の数字文字列 |
| 3. Y 軸のモータ No. | 0～15 の数字文字列 |
| 4. X 軸のモータの移動目標の相対座標 | -999999999 から 999999999 までの数値文字列 (+符号の指定は不可) |
| 5. Y 軸のモータの移動目標の相対座標 | -999999999 から 999999999 までの数値文字列 (+符号の指定は不可) |
| 6. X 軸のモータの円弧の中心相対座標 | -999999999 から 999999999 までの数値文字列 (+符号の指定は不可) |
| 7. Y 軸のモータの円弧の中心相対座標 | -999999999 から 999999999 までの数値文字列 (+符号の指定は不可) |

[リプライ・メッセージ]

- | | |
|---|--|
| @DrawCircularCcwREL□0□0□1□1000□1000□0□0□Ok: | 正常に動作した場合 (チャンネル選択あり、ここでは AB を選択) |
| @DrawCircularCcwREL□0□1□1000□1000□0□0□Ok: | 正常に動作した場合 (チャンネル選択省略、AB もしくは CD が自動選択される) |
| @DrawCircularCcwREL□0□0□1□1000□1000□0□0□Er:□Busy | チャンネル選択時、選択チャンネルが別のモータに使用されている、もしくは
X 軸もしくは Y 軸のモータが使用されている |
| @DrawCircularCcwREL□0□1□1000□1000□0□0□Er:□Busy | チャンネル選択省略時、AB、CD のペアでの空きチャンネルがない、もしくは X 軸もしくは Y 軸のモータが使用されている |
| @DrawCircularCcwREL□ (正しい引数) □Er:□Offline. | Local 状態で実行した場合 |
| @DrawCircularCcwREL□17□1□1000□1000□0□0□Er: Bad motor number. | X 軸もしくは Y 軸のモータ No.が範囲外 |
| @DrawCircularCcwREL□0□1□999999999□1000□0□0□Er:□Data□out□of□range. | X 軸もしくは Y 軸のモータ移動目標座標が範囲外、
もしくは X 軸もしくは Y 軸のモータ円弧 |

[例]

(送信側)

pm16c04□DrawCircularCcwREL□0□1□1000□1000□0□0

(返信されてくる文字列)

pm16c04>test□@DrawCircularCcwREL□0□1□1000□1000□0□0□Ok:

DrawArc

このコマンドを送信することで、2つの軸を X、Y 座標とした平面上の円弧を描くように、2つのモータを通過座標を元に現在座標から目標座標（絶対座標）まで動かします。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータもしくは選択チャンネルが Busy 状態のときはエラーとなり実行されません。

[引数]

- | | |
|----------------------|---|
| 1. チャンネル選択 (省略可) | 0 もしくは AB ChannelA と B を使います
1 もしくは CD ChannelC と D を使います
省略された場合は、AB もしくは CD いずれかを自動選択します。 |
| 2. X 軸のモータ No. | 0～15 の数字文字列 |
| 3. Y 軸のモータ No. | 0～15 の数字文字列 |
| 4. X 軸のモータの移動目標の絶対座標 | -999999999 から 999999999 までの数値文字列 (+符号の指定は不可) |
| 5. Y 軸のモータの移動目標の絶対座標 | -999999999 から 999999999 までの数値文字列 (+符号の指定は不可) |
| 6. X 軸のモータの通過絶対座標 | -999999999 から 999999999 までの数値文字列 (+符号の指定は不可) |
| 7. Y 軸のモータの通過絶対座標 | -999999999 から 999999999 までの数値文字列 (+符号の指定は不可) |

[リプライ・メッセージ]

@DrawArc□0□0□1□1000□1000□0□0□Ok:

正常に動作した場合 (チャンネル選択あり、ここでは AB を選択)

@DrawArc□0□1□1000□1000□0□0□Ok:

正常に動作した場合 (チャンネル選択省略、AB もしくは CD が自動選択される)

@DrawArc□0□0□1□1000□1000□0□0□Er:□Busy

チャンネル選択時、選択チャンネルが別のモータに使用されている、もしくは
X 軸もしくは Y 軸のモータが使

用されている

@DrawArc□0□1□1000□1000□0□0□Er:□Busy

チャンネル選択省略時、AB、CD

のペアでの空きチャンネルがない、もしくは X 軸もしくは Y 軸

のモータが使用されている

@DrawArc□ (正しい引数) □Er:□Offline.

Local 状態で実行した場合

@DrawArc□17□1□1000□1000□0□0□Er: Bad motor number.

X 軸もしくは Y 軸のモータ No.が範囲外

@DrawArc□0□1□999999999□1000□0□0□Er:□Data□out□of□range.

X 軸もしくは Y 軸のモータ移動目標座標が範囲外、

もしくは X 軸もしくは Y 軸のモータ通過座標が範囲外

[例]

(送信側)

pm16c04□DrawArc□0□1□1000□1000□0□0

(返信されてくる文字列)

pm16c04>test□@DrawArc□0□1□1000□1000□0□0□Ok:

DrawArcREL

このコマンドを送信することで、2つの軸を X、Y 座標とした平面上の円弧を描くように、2つのモータを通過座標を元に現在座標から目標座標（現在座標からの相対座標）まで動かします。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータもしくは選択チャンネルが Busy 状態のときはエラーとなり実行されません。

[引数]

- | | |
|----------------------|---|
| 1. チャンネル選択 (省略可) | 0 もしくは AB ChannelA と B を使います
1 もしくは CD ChannelC と D を使います
省略された場合は、AB もしくは CD いずれかを自動選択します。 |
| 2. X 軸のモータ No. | 0～15 の数字文字列 |
| 3. Y 軸のモータ No. | 0～15 の数字文字列 |
| 4. X 軸のモータの移動目標の相対座標 | -999999999 から 999999999 までの数値文字列 (+符号の指定は不可) |
| 5. Y 軸のモータの移動目標の相対座標 | -999999999 から 999999999 までの数値文字列 (+符号の指定は不可) |
| 6. X 軸のモータの通過相対座標 | -999999999 から 999999999 までの数値文字列 (+符号の指定は不可) |
| 7. Y 軸のモータの通過相対座標 | -999999999 から 999999999 までの数値文字列 |

列（+符号の指定は不可）

[リプライ・メッセージ]

@DrawArcREL□0□0□1□2000□0□1000□-1000□Ok:	正常に動作した場合（チャンネル選択あり、ここでは AB を選択）
@DrawArcREL□0□1□2000□0□1000□-1000□Ok:	正常に動作した場合（チャンネル選択省略、AB もしくは CD が自動選択される）
@DrawArcREL□0□0□1□2000□0□1000□-1000□Er:□Busy	チャンネル選択時、選択チャンネルが別のモータに使用されている、もしくは X 軸もしくは Y 軸のモータが使用されている
@DrawArcREL□0□1□2000□0□1000□-1000□Er:□Busy	チャンネル選択省略時、AB、CD のペアでの空きチャンネルがない、もしくは X 軸もしくは Y 軸のモータが使用されている
@DrawArcREL□（正しい引数）□Er:□Offline.	Local 状態で実行した場合
@DrawArcREL□17□1□2000□0□1000□-1000□Er: Bad motor number.	X 軸もしくは Y 軸のモータ No.が範囲外
@DrawArcREL□0□1□999999999□0□1000□-1000□Er:□Data□out□of□range.	X 軸もしくは Y 軸のモータ移動目標座標が範囲外、 もしくは X 軸もしくは Y 軸のモータ通過座標が範囲外

[例]

(送信側)

pm16c04□DrawArcREL□0□1□2000□0□1000□-1000

(返信されてくる文字列)

pm16c04>test□@DrawArcREL□0□1□2000□0□1000□-1000□Ok:

DrawCircle

このコマンドを送信することで、2つの軸を X、Y 座標とした平面上の円を描くように、2つのモータを現在座標から2つの通過座標（絶対座標）を通して動かします。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータもしくは選択チャンネルが Busy 状態のときはエラーとなり実行されません。

[引数]

1. チャンネル選択（省略可）	0 もしくは AB	ChannelA と B を使います
	1 もしくは CD	ChannelC と D を使います

省略された場合は、AB もしくは CD いずれかを自動選択します。

- | | |
|---------------------|--|
| 2. X 軸のモータ No. | 0～15 の数字文字列 |
| 3. Y 軸のモータ No. | 0～15 の数字文字列 |
| 4. X 軸のモータの通過絶対座標 1 | -99999999 から 99999999 までの数値文字列（+ 符号の指定は不可） |
| 5. Y 軸のモータの通過絶対座標 1 | -99999999 から 99999999 までの数値文字列（+ 符号の指定は不可） |
| 6. X 軸のモータの通過絶対座標 2 | -99999999 から 99999999 までの数値文字列（+ 符号の指定は不可） |
| 7. Y 軸のモータの通過絶対座標 2 | -99999999 から 99999999 までの数値文字列（+ 符号の指定は不可） |

[リプライ・メッセージ]

- | | |
|---|--|
| @DrawCircle□0□0□1□1000□1000□0□-1000□Ok: | 正常に動作した場合（チャンネル選択あり、ここでは AB を選択） |
| @DrawCircle□0□1□1000□1000□0□-1000□Ok: | 正常に動作した場合（チャンネル選択省略、AB もしくは CD が自動選択される） |
| @DrawCircle□0□0□1□1000□1000□0□-1000□Er:□Busy | チャンネル選択時、選択チャンネルが別のモータに使用されている、もしくは
X 軸もしくは Y 軸のモータが使用されている |
| @DrawCircle□0□1□1000□1000□0□-1000□Er:□Busy | チャンネル選択省略時、AB、CD のペアでの空きチャンネルがない、もしくは X 軸もしくは Y 軸のモータが使用されている |
| @DrawCircle□（正しい引数）□Er:□Offline. | Local 状態で実行した場合 |
| @DrawCircle□17□1□1000□1000□0□-1000□Er: Bad motor number. | X 軸もしくは Y 軸のモータ No.が範囲外 |
| @DrawCircle□0□1□999999999□1000□0□0□Er:□Data□out□of□range. | X 軸もしくは Y 軸のモータ通過座標 1 が範囲外、
もしくは X 軸もしくは Y 軸のモータ通過座標 2 が範囲外 |

[例]

（送信側）

pm16c04□DrawCircle□0□1□1000□1000□0□-1000

（返信されてくる文字列）

pm16c04>test@DrawCircle001000000-1000Ok:

DrawCircleREL

このコマンドを送信することで、2つの軸を X、Y 座標とした平面上の円を描くように、2つのモータを現在座標から2つの通過座標（現在座標からの相対座標）を通して動かします。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータもしくは選択チャンネルが Busy 状態のときはエラーとなり実行されません。

[引数]

- | | |
|---------------------|---|
| 1. チャンネル選択（省略可） | 0 もしくは AB ChannelA と B を使います
1 もしくは CD ChannelC と D を使います
省略された場合は、AB もしくは CD いずれかを自動選択します。 |
| 2. X 軸のモータ No. | 0～15 の数字文字列 |
| 3. Y 軸のモータ No. | 0～15 の数字文字列 |
| 4. X 軸のモータの通過相対座標 1 | -99999999 から 99999999 までの数値文字列（+ 符号の指定は不可） |
| 5. Y 軸のモータの通過相対座標 1 | -99999999 から 99999999 までの数値文字列（+ 符号の指定は不可） |
| 6. X 軸のモータの通過相対座標 2 | -99999999 から 99999999 までの数値文字列（+ 符号の指定は不可） |
| 7. Y 軸のモータの通過相対座標 2 | -99999999 から 99999999 までの数値文字列（+ 符号の指定は不可） |

[リプライ・メッセージ]

- | | |
|--------------------------------------|--|
| @DrawCircleREL0001000000-1000Ok: | 正常に動作した場合（チャンネル選択あり、ここでは AB を選択） |
| @DrawCircleREL001000000-1000Ok: | 正常に動作した場合（チャンネル選択省略、AB もしくは CD が自動選択される） |
| @DrawCircleREL0001000000-1000Er:Busy | チャンネル選択時、選択チャンネルが別のモータに使用されている、もしくは
X 軸もしくは Y 軸のモータが使用されている |
| @DrawCircleREL001000000-1000Er:Busy | チャンネル選択省略時、AB、CD のペアでの空きチャンネルがない、もしくは X 軸もしくは Y 軸のモータが使用されている |

@DrawCircleREL (正しい引数) Er:Offline. Local 状態で実行した場合

@DrawCircleREL 17 1 2000 0 1000 -1000 Er: Bad motor number.

X 軸もしくは Y 軸のモータ No.が範囲外

@DrawCircleREL 0 1 999999999 0 1000 -1000 Er:Data out of range.

X 軸もしくは Y 軸のモータ通過座標 1 が
範囲外、

もしくは X 軸もしくは Y 軸のモータ通過
座標 2 が範囲外

[例]

(送信側)

pm16c04 DrawCircleREL 0 1 2000 0 1000 -1000

(返信されてくる文字列)

pm16c04>test @DrawCircleREL 0 1 2000 0 1000 -1000 Ok:

モータコマンド

[メッセージ配信先名]

pm16c04.motorname

motorname のモータに対して命令（コマンド）を送信する場合の配信先名

[コマンド]

hello

STARS の通信が行われているかをチェックするコマンド。

このコマンドを送信すると '@hello nice to meet you.' の文字列を返します。

[例]

(送信側)

```
pm16c04.th□hello
```

(返信されてくる文字列)

```
pm16c04.th>test□@hello□@hello□nice□to□meet□you.
```

help

引数を指定しない場合はモータコマンドの一覧を返します。

引数にコマンド名を指定した場合はコマンドのヘルプを返します。

[例]

(送信側)

```
pm16c04.th□help
```

コントローラコマンドの一覧を表示します

(返信されてくる文字列)

```
pm16c04.th>test□@help□GetAccRate□GetAccRateCode□GetAccRateList□GetAutoChangeSpeed
□GetAutoChangeSpeedReady□GetCancelBacklash□GetDigitalCwLs□GetDigitalCwLs□GetHPM
ode□GetHPOffset□GetHighSpeed□GetHomePosition□GetJogPulse□GetLimits□GetLowSpeed□G
etMiddleSpeed□GetMotorNumber□GetMotorSetup□GetSelected□GetSpeedSelected□GetStopMod
e□GetTimingOutEnd□GetTimingOutInterval□GetTimingOutMode□GetTimingOutReady□GetTim
ingOutStart□GetValue□IsBusy□JogCcw□JogCw□Preset□ReScanHome□ScanCcw□ScanCcwCons
t□ScanCcwHome□ScanCw□ScanCwConst□ScanCwHome□ScanHome□SetAccRate□SetAutoCha
ngeSpeed□SetAutoChangeSpeedReady□SetCancelBacklash□SetDigitalCwLs□SetDigitalCwLs□
SetHPMode□SetHPOffset□SetHighSpeed□SetHomePosition□SetJogPulse□SetLimits□SetLowSpe
ed□SetMiddleSpeed□SetMotorSetup□SetSpeedCurrent□SetStopMode□SetTimingOutEnd□SetTim
ingOutInterval□SetTimingOutMode□SetTimingOutReady□SetTimingOutStart□SetValue□SetValu
eREL□SpeedHigh□SpeedLow□SpeedMiddle□Stop□StopEmergency□_ChangedCtlIsBusy□_Chan
gedFunction□_ChangedIsBusy□_ChangedValue□hello□help
```

(送信側)

pm16c04.th□help□hello

hello コマンドのヘルプを表示します

(返信されてくる文字列)

pm16c04.th>test□@help□hello□Usage:□hello¥n□The□client□returns□"@hello□nice□to□meet□you.
u."¥n¥n

(送信側)

pm16c04.th help helo

引数として該当しないコマンドを指定した場合

pm16c04.th>test @help helo Er: Command "helo" not found.

GetMotorNumber

このコマンドを送信することでモータ名に対応するモータ No.を取得します。

[例]

(送信側)

pm16c04.th□GetMotorNumber

モータ th のモータ No.番号を取得します

(返信されてくる文字列)

pm16c04.th>test□@GetMotorNumber□0

モータ th のモータ No.0 が返ってきます

◆モータ速度レベル選択コマンド (PM16C-04X Only)

PM16C-04X では、モータ毎に速度レベル”High”、 ”Middle”、 ”Low”を設定できるようになりました。

SpeedHigh

このコマンドを送信することで、以降対応するモータの速度は’ High’ の設定で動きます。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが Busy 状態のときはエラーとなり実行されません。

[リプライ・メッセージ]

@SpeedHigh□Ok:

正常に動作を終了した場合

@SpeedHigh□Er:□Busy.

モータが Busy 状態のため実行されなかった場合

@SpeedHigh□Er:□Offline.

Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□SpeedHigh

(返信されてくる文字列)

pm16c04.th>test□@SpeedHigh□Ok:

SpeedMiddle

このコマンドを送信することで、以降対応するモータの速度は' Middle' の設定で動きます。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが Busy 状態のときはエラーとなり実行されません。

[リプライ・メッセージ]

@SpeedMiddle□Ok:	正常に動作を終了した場合
@SpeedMiddle□Er:□Busy.	モータが Busy 状態のため実行されなかった場合
@SpeedMiddle□Er:□Offline.	Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□SpeedMiddle

(返信されてくる文字列)

pm16c04.th>test□@SpeedMiddle□Ok:

SpeedLow

このコマンドを送信することで、以降対応するモータの速度は' Low' の設定で動きます。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが Busy 状態のときはエラーとなり実行されません。

[リプライ・メッセージ]

@SpeedLow□Ok:	正常に動作を終了した場合
@SpeedLow□Er:□Busy.	モータが Busy 状態のため実行されなかった場合
@SpeedLow□Er:□Offline.	Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□SpeedLow

(返信されてくる文字列)

pm16c04.th>test□@SpeedLow□Ok:

GetSpeedSelected

このコマンドを送信すると対応するモータの速度がどの設定 (Low/Middle/High)で動くのかを返します

[リプライ・メッセージのコマンド]

@GetSpeedSelected□H	:	選択されている速度が' High'の場合
---------------------	---	----------------------

@GetSpeedSelected□M : 選択されている速度が'Middle'の場合
 @GetSpeedSelected□L : 選択されている速度が'Low'の場合

[例]

(送信側)

pm16c04□GetSpeedSelected

(返信されてくる文字列)

pm16c04>test□@GetSpeedSelected□M

◆モータ設定コマンド

以下のコマンドは Stars の GUI プログラム 'pm16config'から参照・更新できます。

SetHighSpeed

このコマンドを送信することで対応するモータの速度'High'の値 (PPS) を設定します。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが Busy 状態のときはエラーとなり実行されません。

[引数]

[PM16C-04 シリーズの場合]

モータの速度'High'の値 (PPS) ※ 正の数値文字列 (+符号なし)

※設定可能な速度 (PPS) は、GetSpeedList コマンドで確認できる pm16c04 の速度データ表に定義されている値です。

引数の速度 (PPS) に一致する速度が速度データ表にない場合は、コード表に定義されている速度の中から引数の速度 (PPS) 未満の近似値の速度 (PPS) を使用します。

引数の速度 (PPS) が、速度データ表に定義されている最小速度より小さい場合は最小速度を使用します。

[PM16C-04X シリーズの場合]

1 から 5000000 までの数字文字列 (PPS) (+符号の指定は不可)

[リプライ・メッセージ]

@SetHighSpeed□ (引数) □Ok:

正常に動作を終了した場合

@SetHighSpeed□ (引数) □Er:□Busy.

モータが Busy 状態のため実行されなかった場合

@SetHighSpeed□ (引数) □Er:□Offline.

Local 状態で実行した場合

@SetHighSpeed□10000000□Er:□Data□out□of□range.

引数が有効範囲を超えた場合

[例]

(送信側)

pm16c04.th□SetHighSpeed□10000

モータ th の速度'High'の値 (PPS) を
10000 で設定します

(返信されてくる文字列)

pm16c04.th>test□@SetHighSpeed□10000□Ok: 正常に動作を終了した場合

SetMiddleSpeed

このコマンドを送信することで対応するモータの速度' Middle' の値 (PPS) を設定します。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが Busy 状態のときはエラーとなり実行されません。

[引数]

[PM16C-04 シリーズの場合]

モータの速度'Middle'の値 (PPS) ※ 正の数値文字列 (+符号なし)

※設定可能な速度 (PPS) は、GetSpeedList コマンドで確認できる pm16c04 の速度データ表に定義されている値です。

引数の速度 (PPS) に一致する速度が速度データ表にない場合は、コード表に定義されている速度の中から引数の速度 (PPS) 未満の近似値の速度 (PPS) を使用します。

引数の速度 (PPS) が、速度データ表に定義されている最小速度より小さい場合は最小速度を使用します。

[PM16C-04X シリーズの場合]

1 から 5000000 までの数字文字列 (PPS) (+符号の指定は不可)

[リプライ・メッセージ]

@SetMiddleSpeed□ (引数) □Ok:

正常に動作を終了した場合

@SetMiddleSpeed□ (引数) □Er:□Busy.

モータが Busy 状態のため実行されなかった場合

@SetMiddleSpeed□ (引数) □Er:□Offline.

Local 状態で実行した場合

@SetMiddleSpeed□10000000□Er:□Data□out□of□range.

引数が有効範囲を超えた場合

[例]

(送信側)

pm16c04.th□SetMiddleSpeed□10000

モータ th の速度'Middle'の値 (PPS) を
10000 で設定します

(返信されてくる文字列)

pm16c04.th>test□@SetMiddleSpeed□10000□Ok: 正常に動作を終了した場合

SetLowSpeed

このコマンドを送信することで対応するモータの速度' Low' の値 (PPS) を設定します。

このコマンドはコントローラが **Local** 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが **Busy** 状態のときはエラーとなり実行されません。

[引数]

[PM16C-04 シリーズの場合]

モータの速度'Low'の値 (PPS) ※ 正の数値文字列 (+符号なし)

※設定可能な速度 (PPS) は、GetSpeedList コマンドで確認できる pm16c04 の速度データ表に定義されている値です。

引数の速度 (PPS) に一致する速度が速度データ表にない場合は、コード表に定義されている速度の中から引数の速度 (PPS) 未満の近似値の速度 (PPS) を使用します。

引数の速度 (PPS) が、速度データ表に定義されている最小速度より小さい場合は最小速度を使用します。

[PM16C-04X シリーズの場合]

1 から 5000000 までの数字文字列 (PPS) (+符号の指定は不可)

[リプライ・メッセージ]

@SetLowSpeed□ (引数) □Ok:

正常に動作を終了した場合

@SetLowSpeed□ (引数) □Er:□Busy.

モータが **Busy** 状態のため実行されなかった場合

@SetLowSpeed□ (引数) □Er:□Offline.

Local 状態で実行した場合

@SetLowSpeed□0□Er:□Data□out□of□range.

引数が有効範囲を超えた場合

[例]

(送信側)

pm16c04.th□SetLowSpeed□10000

モータ th の速度'Low'の値 (PPS) を
10000 で設定します

(返信されてくる文字列)

pm16c04.th>test□@SetLowSpeed□10000□Ok: 正常に動作を終了した場合

SetAccRate

このコマンドを送信することで対応するモータの速度の加減速レートの値 (ms/1000PPS) を設定します。

このコマンドはコントローラが **Local** 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが **Busy** 状態のときはエラーとなり実行されません。

[引数]

モータの速度の加減速レートの値※

正の数値文字列 (+符号なし)

※設定可能な速度の加減速レートは、GetAccRateList コマンドで確認できる速度の加減速レートデータ表に定義されている値です。

PM16C-04 シリーズと PM16C-04X シリーズで設定可能な加減速レートデータ表の値は異なりますのでご注意ください。

引数の速度の加減速レートに一致する値が速度の加減速レートデータ表にない場合は、コード表に定義されている速度の加減速レートの中から引数の値未満の近似値を使用します。

引数の速度の加減速レートが、速度の加減速レートデータ表に定義されている最小値より小さい場合は最小値を使用します。

[リプライ・メッセージ]

@SetAccRate□ (引数) □Ok:	正常に動作を終了した場合
@SetAccRate□ (引数) □Er:□Busy.	モータが Busy 状態のため実行されなかった場合
@SetAccRate□ (引数) □Er:□Offline.	Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□SetAccRate□1000

モータ th の速度の加減速レートを設定

(返信されてくる文字列)

pm16c04.th>test□@SetAccRate□1000□Ok:

正常に動作した場合

SetDigitalCwLs

このコマンドを送信することで対応するモータの CW ソフトウェアリミットスイッチの値を設定します。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが Busy 状態のときはエラーとなり実行されません。

[引数]

[PM16C-04X シリーズの場合]

モータの CW ソフトウェアリミットスイッチの値 (パルス)

-2147483647 から 2147483647 までの数値文字列 (+符号の指定は不可)

[PM16C-04 シリーズの場合]

モータの CW ソフトウェアリミットスイッチの値 (パルス)

-8388607 から ~8388607 までの数値文字列 (+符号の指定は不可)

[リプライ・メッセージ]

@SetDigitalCwLs□ (引数) □Ok:	正常に動作を終了した場合
@SetDigitalCwLs□ (引数) □Er:□Busy.	モータが Busy 状態のため実行されなかった場合

た場合

@SetDigitalCwLs□ (引数) □Er:□Offline. Local 状態で実行した場合

@SetDigitalCwLs□1000000000□Er:□Data□out□of□range. 引数が有効範囲を超えた場合

[例]

(送信側)

pm16c04.th□SetDigitalCwLs□40000

モータ th の CW ソフトウェアリミットスイッチの値を設定

(返信されてくる文字列)

pm16c04.th>test□@SetDigitalCwLs□40000 正常に動作した場合

SetDigitalCwLs

このコマンドを送信することで対応するモータの CCW ソフトウェアリミットスイッチの値を設定します。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが Busy 状態のときはエラーとなり実行されません。

[引数]

[PM16C-04X シリーズの場合]

モータの CCW ソフトウェアリミットスイッチの値 (パルス)

-2147483647 から 2147483647 までの数値文字列 (+符号の指定は不可)

[PM16C-04 シリーズの場合]

モータの CCW ソフトウェアリミットスイッチの値 (パルス)

-8388607 から ~8388607 までの数値文字列 (+符号の指定は不可)

[リプライ・メッセージ]

@SetDigitalCwLs□ (引数) □Ok: 正常に動作を終了した場合

@SetDigitalCwLs□ (引数) □Er:□Busy. モータが Busy 状態のため実行されなかった場合

@SetDigitalCwLs□ (引数) □Er:□Offline. Local 状態で実行した場合

@SetDigitalCwLs□-1000000000□Er:□Data□out□of□range. 引数が有効範囲を超えた場合

[例]

(送信側)

pm16c04.th□SetDigitalCwLs□-40000

モータ th の CCW ソフトウェアリミットスイッチの値を設定

(返信されてくる文字列)

pm16c04.th>test□@SetDigitalCwLs□-40000 正常に動作した場合

SetCancelBacklash

このコマンドを送信することで対応するモータのバックラッシュ補正ステップ数の値を設定します。

このコマンドはコントローラが **Local** 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが **Busy** 状態のときはエラーとなり実行されません。

[引数]

バックラッシュ補正ステップ数（パルス） -9999～9999 の数字文字列

[リプライ・メッセージ]

@SetCancelBacklash□（引数）□Ok:	正常に動作を終了した場合
@SetCancelBacklash□（引数）□Er:□Busy.	モータが Busy 状態のため実行されなかった場合
@SetCancelBacklash□（引数）□Er:□Offline.	Local 状態で実行した場合
@SetCancelBacklash□-10000□Er: □Bad□parameter.	引数が有効範囲を超えた場合

[例]

（送信側）

```
pm16c04.th□SetCancelBacklash□100
```

モータ **th** のバックラッシュ補正ステップ数の値を設定

（返信されてくる文字列）

```
pm16c04.th>test□@SetCancelBacklash□100      正常に動作した場合
```

SetJogPulse

このコマンドを送信することで、ローカルモードで **Jog** 操作をする際の、モータの 1 **Jog** 単位の値を設定します。

このコマンドはコントローラが **Local** 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが **Busy** 状態のときはエラーとなり実行されません。

[引数]

JOG 単位の値（パルス） 1～9999 の数字文字列

[リプライ・メッセージ]

@SetJogPulse□（引数）□Ok:	正常に動作を終了した場合
@SetJogPulse□（引数）□Er:□Busy.	モータが Busy 状態のため実行されなかった場合
@SetJogPulse□（引数）□Er:□Offline.	Local 状態で実行した場合
@SetJogPulse□10000□Er:□Jog□pulse□out□of□range.(1□to□9999)	

引数が有効範囲を超えた場合

[例]

(送信側)

pm16c04.th□SetJogPulse□10

モータ th の 1 Jog 単位の値を設定

(返信されてくる文字列)

pm16c04.th>test□@SetJogPulse□10□Ok:

正常に動作した場合

SetLimits

このコマンドを送信することで、対応するモータのリミットスイッチ関連フラグの値を設定します。

このコマンドはコントローラが **Local** 状態の場合はエラーとなり実行されません。

またこのコマンドはモータが **Busy** 状態のときはエラーとなり実行されません。

[引数]

[PM16C-04X シリーズの場合]

下記のフォーマット※の 8 桁の数字文字列

※ 数字文字列のフォーマット ABCDEFGH

A	Digital LS Enable Bit	(1: enable 0: disable)
B	HP LS Enable Bit	(1: enable 0: disable)
C	CCW LS Enable Bit	(1: enable 0: disable)
D	CW LS Enable Bit	(1: enable 0: disable)
E	未使用	(0)
F	HP LS Invert Bit	(1:Normaly Close 0: Normaly Open)
G	CCW LS Invert Bit	(1:Normaly Close 0: Normaly Open)
H	CW LS Invert Bit	(1:Normaly Close 0: Normaly Open)

[PM16C-04 シリーズの場合]

下記 Bit※を 1 0 進数で表現した数字文字列

※ Bit 構成

Bit7	Motor Off	(1: enable 0: disable)
Bit6	Hold Status	(1: On 0: Off)
Bit5	Digital LS Enable Bit	(1: enable 0: disable)
Bit4	CCW LS Enable Bit	(1: enable 0: disable)
Bit3	CW LS Enable Bit	(1: enable 0: disable)
Bit2	HP A/B	ツジ電子取説記載確認できず
Bit1	CCW LS A/B	ツジ電子取説記載確認できず
Bit0	CW LS A/B	ツジ電子取説記載確認できず

[リプライ・メッセージ]

[PM16C-04X シリーズの場合]

@SetLimits□ (引数) □Ok:	正常に動作を終了した場合
@SetLimits□ (引数) □Er:□Busy.	モータが Busy 状態のため実行されなかった場合
@SetLimits□ (引数) □Er:□Offline.	Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□SetLimits□11110000

(返信されてくる文字列)

pm16c04.th>test□@SetLimits□11110000□Ok:

SetMotorSetup (PM16C-04X Only)

このコマンドを送信することで、対応するモータの基本特性の値を設定します。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータが Busy 状態のときはエラーとなり実行されません。

[引数]

下記のフォーマット※の 4 桁の数字文字列

※ 数字文字列のフォーマット ABCD

A	モータの有効無効 (1: drive enable 0: disable)
B	Hold 特性 (1: hold on 0: hold off)
C	モータの移動形式 (2: S 字 1: 台形 0: 定速)
D	外部設置のモータドライバの信号入力 (1: P-D 方式 0: P-P 方式)

[リプライ・メッセージ]

@SetMotorSetup□ (引数) □Ok:	正常に動作を終了した場合
@SetMotorSetup□ (引数) □Er:□Busy.	モータが Busy 状態のため実行されなかった場合
@SetMotorSetup□ (引数) □Er:□Offline.	Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□SetMotorSetup□1010

(返信されてくる文字列)

pm16c04.th>test□SetMotorSetup□1010□Ok:

SetStopMode (PM16C-04X Only)

このコマンドを送信することで、対応するモータの停止方法の値を設定します。

このコマンドはコントローラが **Local** 状態の場合はエラーとなり実行されません。

またこのコマンドはモータが **Busy** 状態のときはエラーとなり実行されません。

[引数]

下記のフォーマット※の 2 桁の数字文字列

※ 数字文字列のフォーマット AB

A	CW/CCW リミットスイッチ	(1: 緊急停止 0: 減速停止)
B	パネルの STOP スイッチ	(1: 緊急停止 0: 減速停止)

[リプライ・メッセージ]

@SetStopMode□ (引数) □Ok:	正常に動作を終了した場合
@SetStopMode□ (引数) □Er:□Busy.	モータが Busy 状態のため実行されなかった場合
@SetStopMode□ (引数) □Er:□Offline.	Local 状態で実行した場合

[例]

(送信側)

```
pm16c04.th□SetStopMode□00
```

(返信されてくる文字列)

```
pm16c04.th>test□@SetStopMode□00□Ok:
```

GetHighSpeed

このコマンドを送信することで対応するモータの速度 'High' の設定値 (PPS) を返します。

[戻り値]

[PM16C-04X シリーズの場合]

1 から 5000000 までの数字文字列 (PPS)

[PM16C-04 シリーズの場合]

正の数値文字列 (+ 符号なし)

GetSpeedList コマンドで確認できる pm16c04 の速度データ表に定義されている値のいずれかが返ってきます。

[例]

(送信側)

```
pm16c04.th□GetHighSpeed
```

モータ th の速度 'High' の設定値を取得

(返信されてくる文字列)

```
pm16c04.th>test□@GetHighSpeed□1000
```

正常に動作した場合

GetMiddleSpeed

このコマンドを送信することで対応するモータの速度' Middle' の設定値 (PPS) を返します。

[戻り値]

[PM16C-04X シリーズの場合]

1 から 5000000 までの数字文字列 (PPS)

[PM16C-04 シリーズの場合]

正の数値文字列 (+符号なし)

GetSpeedList コマンドで確認できる pm16c04 の速度データ表に定義されている値のいずれかが返ってきます。

[例]

(送信側)

pm16c04.th□GetMiddleSpeed モータ th の速度' Middle' の設定値を取得

(返信されてくる文字列)

pm16c04.th>test□@GetMiddleSpeed□500 正常に動作した場合

GetLowSpeed

このコマンドを送信することで対応するモータの速度' Low' の設定値 (PPS) を返します。

[戻り値]

[PM16C-04X シリーズの場合]

1 から 5000000 までの数字文字列 (PPS)

[PM16C-04 シリーズの場合]

正の数値文字列 (+符号なし)

GetSpeedList コマンドで確認できる pm16c04 の速度データ表に定義されている値のいずれかが返ってきます。

[例]

(送信側)

pm16c04.th□GetLowSpeed モータ th の速度' Low' の設定値を取得

(返信されてくる文字列)

pm16c04.th>test□@GetLowSpeed□100 正常に動作した場合

GetAccRate

このコマンドを送信することで対応するモータの速度の加減速レートの値 (mS/1000PPS) を返します。

[戻り値]

モータの速度の加減速レートの値※

※加減速レートとして取りうる値は、GetAccRateList コマンドで確認できる速度の加減速レートデータ表に定義されている値です。

[例]

(送信側)

pm16c04.th□GetAccRate

モータ th の速度の加減速レートの設定値を取得

(返信されてくる文字列)

pm16c04.th>test□@GetAccRate□300 正常に動作した場合

GetDigitalCwLs

このコマンドを送信することで対応するモータの CW ソフトウェアリミットスイッチの値を返します。

[戻り値]

[PM16C-04X シリーズの場合]

モータの CW ソフトウェアリミットスイッチの値 (パルス)

-2147483647 から 2147483647 までの数値文字列

[PM16C-04 シリーズの場合]

モータの CW ソフトウェアリミットスイッチの値 (パルス)

-8388607 から～8388607 までの数値文字列

[例]

(送信側)

pm16c04.th□GetDigitalCwLs

モータ th の CW ソフトウェアリミットスイッチの値を取得

(返信されてくる文字列)

pm16c04.th>test□@GetCancelBacklash□40000 正常に動作した場合

GetDigitalCcwLs

このコマンドを送信することで対応するモータの CCW ソフトウェアリミットスイッチの値を返します。

[戻り値]

[PM16C-04X シリーズの場合]

モータの CCW ソフトウェアリミットスイッチの値 (パルス)

-2147483647 から 2147483647 までの数値文字列

[PM16C-04 シリーズの場合]

モータの CCW ソフトウェアリミットスイッチの値 (パルス)

-8388607 から～8388607 までの数値文字列

[例]

(送信側)

pm16c04.th□GetDigitalCcwLs

モータ th の CCW ソフトウェアリミットスイッチの値を取得

(返信されてくる文字列)

pm16c04.th>test□@GetCancelBacklash□-40000

正常に動作した場合

GetCancelBacklash

このコマンドを送信することで対応するモータのバックラッシュ補正ステップ数の値を返します。

[戻り値]

バックラッシュ補正ステップ数 (パルス) -9999～9999 の数字文字列

[例]

(送信側)

pm16c04.th□GetCancelBacklash

モータ th のバックラッシュ補正ステップ数の値を取得

(返信されてくる文字列)

pm16c04.th>test□@GetCancelBacklash□100

正常に動作した場合

GetJogPulse

このコマンドを送信することで、ローカルモードで Jog 操作をする際の、モータの 1 Jog 単位の値を返します。

[戻り値]

JOG 単位の値 (パルス) 1～9999 の数字文字列

[例]

(送信側)

pm16c04.th□GetJogPulse

モータ th の 1 Jog 単位の値を設定

(返信されてくる文字列)

pm16c04.th>test□@GetJogPulse□10

正常に動作した場合

GetLimits

このコマンドを送信することで対応するモータのリミットスイッチ関連フラグの値を返します。

[戻り値]

[PM16C-04X シリーズの場合]

下記のフォーマット※の 8 桁の数字文字列を返します。

※ 数字文字列のフォーマット ABCDEFGH

A	Digital LS Enable Bit	(1: enable 0: disable)
B	HP LS Enable Bit	(1: enable 0: disable)
C	CCW LS Enable Bit	(1: enable 0: disable)
D	CW LS Enable Bit	(1: enable 0: disable)
E	未使用	(0)
F	HP LS Invert Bit	(1:Normaly Close 0: Normaly Open)
G	CCW LS Invert Bit	(1:Normaly Close 0: Normaly Open)
H	CW LS Invert Bit	(1:Normaly Close 0: Normaly Open)

[PM16C-04 シリーズの場合]

下記 Bit※を 1 0 進数で表現した数字文字列

※ Bit 構成

Bit7	Motor Off	(1: enable 0: disable)
Bit6	Hold Status	(1: On 0: Off)
Bit5	Digital LS Enable Bit	(1: enable 0: disable)
Bit4	CCW LS Enable Bit	(1: enable 0: disable)
Bit3	CW LS Enable Bit	(1: enable 0: disable)
Bit2	HP A/B	ツジ電子取説記載確認できず
Bit1	CCW LS A/B	ツジ電子取説記載確認できず
Bit0	CW LS A/B	ツジ電子取説記載確認できず

[例]

(送信側)

```
pm16c04.th□GetLimits
```

(返信されてくる文字列)

```
pm16c04.th>test□@GetLimits□11110000
```

GetMotorSetup (PM16C-04X Only)

このコマンドを送信することで対応するモータの基本特性の値を返します。

[戻り値]

下記のフォーマット※の 4 桁の数字文字列を返します。

※ 数字文字列のフォーマット ABCD

A	モータの有効無効	(1: drive enable 0: disable)
B	Hold 特性	(1: hold on 0: hold off)
C	モータの移動形式	(2: S 字 1:台形 0:定速)

D 外部設置のモータドライバの信号入力 (1: P-D 方式 0: P-P 方式)

[例]

(送信側)

```
pm16c04.th□GetMotorSetup
```

(返信されてくる文字列)

```
pm16c04.th>test□@GetMotorSetup□1010
```

GetStopMode (PM16C-04X Only)

このコマンドを送信することで対応するモータの停止方法の値を返します。

[戻り値]

下記のフォーマット※の 2 桁の数字文字列を返します。

※ 数字文字列のフォーマット AB

A CW/CCW リミットスイッチ (1: 緊急停止 0: 減速停止)

B パネルの STOP スイッチ (1: 緊急停止 0: 減速停止)

[例]

(送信側)

```
pm16c04.th□GetStopMode
```

(返信されてくる文字列)

```
pm16c04.th>test□@GetStopMode□00
```

◆HP 関連フラグ設定・読み出しコマンド (PM16C-04X Only)

PM16C-04X では、HP 検索機能が強化され新たなコマンドが追加されました。実行にあたっては、ツジ電子 (株) が提供する取扱説明書の該当箇所をよく読んでからおこなってください。

SetHPMode

このコマンドを送信することで対応するモータの原点位置設定方法や検出状況の強制書き換えをおこないます。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータが Busy 状態のときはエラーとなり実行されません。

[引数]

下記のフォーマット※の 4 桁の数字文字列

※ 数字文字列のフォーマット ABCD

A 未使用 (0)

B 原点検出状況 (1: 検出あり 0: 検出なし)

C	原点検出方向	(1: CCW 0: CW)
D	自動原点検出開始方向	(1: CCW 0: CW)

[リプライ・メッセージ]

@SetHPMode□ (引数) □Ok:	正常に動作した場合
@SetHPMode□ (引数) □Er:□Busy.	モータが Busy 状態のため実行されなかった場合
@SetHPMode□ (引数) □Er:□Offline.	Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□SetHPMode□0001

(返信されてくる文字列)

pm16c04.th>test□@SetHPMode□0001□Ok:

SetHPOffset

このコマンドを送信することでモータの原点オフセットの値を設定します。
このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。
またこのコマンドはモータが Busy 状態のときはエラーとなり実行されません。

[引数]

原点オフセット値 (パルス) 0～9999 の数字文字列

[リプライ・メッセージ]

@SetHPOffset□ (引数) □Ok:	正常に動作した場合
@SetHPOffset□10000□Er:□Data□out□of□range.	引数が有効範囲を超えた場合
@SetHPOffset□ (引数) □Er:□Busy.	モータが Busy 状態のため実行されなかった場合
@SetHPOffset□ (引数) □Er:□Offline.	Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□SetHPOffset□100□Ok: モータ th の原点オフセットの値を設定します

(返信されてくる文字列)

pm16c04.th>test□@SetHPOffset□100□Ok: 正常に動作した場合

SetHomePosition

このコマンドを送信することで対応するモータの原点位置データを書き換えます。
このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータが **Busy** 状態のときはエラーとなり実行されません。

[引数]

原点位置データの値 (パルス)

-2147483647 から ~2147483647 までの数値文字列 (+符号の指定は不可)

[リプライ・メッセージ]

@SetHomePosition□ (引数) □Ok: 正常に動作した場合

@SetHomePosition□10000000000□Er:□Data□out□of□range. 引数が有効範囲を超えた場合

@SetHomePosition□ (引数) □Er:□Busy. モータが **Busy** 状態のため実行されなかった場合

@SetHomePosition□ (引数) □Er:□Offline. Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□SetHomePosition□0 モータ th の原点位置データの値を書き換えます

(返信されてくる文字列)

pm16c04.th>test□@SetHomePosition□0□Ok: 正常に動作した場合

GetHPMode

このコマンドを送信することで対応するモータの原点位置設定方法や検出状況の値を返します。

[戻り値]

下記のフォーマット※の 4 桁の数字文字列

※ 数字文字列のフォーマット ABCD

A	未使用	(0)
B	原点検出状況	(1: 検出あり 0: 検出なし)
C	原点検出方向	(1: CCW 0: CW)
D	自動原点検出開始方向	(1: CCW 0: CW)

[例]

(送信側)

pm16c04.th□GetHPMode

(返信されてくる文字列)

pm16c04.th>test□@GetHPMode□0001

GetHPOffset

このコマンドを送信することで対応するモータの原点オフセットの値を取得します。

[戻り値]

原点オフセット値（パルス） 0～9999 の数字文字列

[例]

（送信側）

pm16c04.th□GetHPOffset モータ th の原点オフセットの値を取得します

（返信されてくる文字列）

pm16c04.th>test□@GetHPOffset□100 モータ th の原点オフセットは 100 です

GetHomePosition

このコマンドを送信することで対応するモータの原点位置データを取得します。

[戻り値]

原点位置データがある場合は、原点位置データの値（パルス）

-2147483647 から～2147483647 までの数値文字列

原点位置データがない場合は、文字列 “Ng: □NO□H.P.” を返します。

[リプライ・メッセージ]

@GetHomePosition□（戻り値）

正常に動作を終了した場合

@GetHomePosition□Ng: □NO□H.P.

原点位置データが設定されていない場合

[例]

（送信側）

pm16c04.th□GetHomePosition モータ th の原点位置データの値を取得します

（返信されてくる文字列）

pm16c04.th>test□@GetHomePosition□100 モータ th の原点位置データは 100 です

pm16c04.th>test□@GetHomePosition□NO□H.P. モータ th の原点位置データは未設定です

ScanHome

このコマンドを送信することでモータの自動原点検出を開始します

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが Busy 状態のときはエラーとなり実行されません。

[リプライ・メッセージ]

@ScanHome□Ok:

正常に動作を終了した場合

@ScanHome□Er:□Busy.

モータが Busy 状態のため実行さ

れなかった場合

@ScanHome□Er:□Offline.

Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□ScanHome

モータ th の自動原点検出を開始します

(返信されてくる文字列)

pm16c04.th>test□@ScanHome□Ok:

正常に動作を開始した場合

ReScanHome

あらかじめモータの原点が検出されている時、このコマンドを送信することで、原点位置近傍に移動しそこから低速で原点を検出します

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが Busy 状態のときはエラーとなり実行されません。

[リプライ・メッセージ]

@ReScanHome□Ok:

正常に動作を終了した場合

@ReScanHome□Er:□Busy.

モータが Busy 状態のため実行されなかった場合

@ReScanHome□Er:□Offline.

Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□ReScanHome

モータ th の原点の再検出を開始します

(返信されてくる文字列)

pm16c04.th>test□@ReScanHome□Ok:

正常に動作を開始した場合

◆タイミングアウト出力機能関連コマンド (PM16C-04X Only)

タイミングアウト出力機能は PM16C-04X からの新機能です。実行にあたっては、ツジ電子 (株) が提供する取扱説明書の該当箇所をよく読んでからおこなってください。

SetTimingOutMode

このコマンドを送信することで対応するモータのタイミングアウトモードの設定を設定します。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータが Busy 状態のときはエラーとなり実行されません。

[引数]

- 0 タイミングパルス出力をおこないません
- 1 開始点から終了点の間、TTL のゲートパルスを出力
- 2 開始点から終了点の間、設定インターバル毎に、200ns の TTL パルスを出力

- 3 開始点から終了点の間、設定インターバル毎に、 $10\mu\text{s}$ の TTL パルスを出力
- 4 開始点から終了点の間、設定インターバル毎に、 $100\mu\text{s}$ の TTL パルスを出力
- 5 開始点から終了点の間、設定インターバル毎に、 1ms の TTL パルスを出力

[リプライ・メッセージ]

@SetTimingOutMode□ (引数) □Ok:	正常に動作した場合
@SetTimingOutMode□ (引数) □Er:□Busy.	モータが Busy 状態のため実行されなかった場合
@SetTimingOutMode□ (引数) □Er:□Offline.	Local 状態で実行した場合
@SetTimingOutMode□7□Er:□Data□invalid.	引数が範囲外の場合

[例]

(送信側)

pm16c04.th□SetTimingOutMode□1

(返信されてくる文字列)

pm16c04.th>test□@SetTimingOutMode□1□Ok:

SetTimingOutStart

このコマンドを送信することで対応するモータのタイミングアウトの開始点を設定します。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータが Busy 状態のときはエラーとなり実行されません。

[引数]

-2147483647 から 2147483647 までの数値文字列（+符号の指定は不可）

[リプライ・メッセージ]

@SetTimingOutStart□ (引数) □Ok:	正常に動作した場合
@SetTimingOutStart□ (引数) □Er:□Busy.	モータが Busy 状態のため実行されなかった場合
@SetTimingOutStart□ (引数) □Er:□Offline.	Local 状態で実行した場合
@SetTimingOutStart□9999999999□Er:□Data□out□of□range.	引数が範囲外の場合

[例]

(送信側)

pm16c04.th□SetTimingOutStart□-10000

(返信されてくる文字列)

pm16c04.th>test□@SetTimingOutStart□-10000□Ok:

SetTimingOutEnd

このコマンドを送信することで対応するモータのタイミングアウトの終了点を設定します。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータが Busy 状態のときはエラーとなり実行されません。

[引数]

-2147483647 から 2147483647 までの数値文字列（+符号の指定は不可）

[リプライ・メッセージ]

@SetTimingOutEnd□（引数）□Ok:	正常に動作した場合
@SetTimingOutEnd□（引数）□Er:□Busy.	モータが Busy 状態のため実行されなかった場合
@SetTimingOutEnd□（引数）□Er:□Offline.	Local 状態で実行した場合
@SetTimingOutEnd□9999999999□Er:□Data□out□of□range.	引数が範囲外の場合

[例]

（送信側）

```
pm16c04.th□SetTimingOutEnd□10000
```

（返信されてくる文字列）

```
pm16c04.th>test□@SetTimingOutEnd□10000□Ok:
```

SetTimingOutInterval

このコマンドを送信することで対応するモータのタイミングアウトのインターバルを設定します。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータが Busy 状態のときはエラーとなり実行されません。

[引数]

0 から 2147483647 までの数値文字列（+符号の指定は不可）

[リプライ・メッセージ]

@SetTimingOutInterval□（引数）□Ok:	正常に動作した場合
@SetTimingOutInterval□（引数）□Er:□Busy.	モータが Busy 状態のため実行されなかった場合
@SetTimingOutInterval□（引数）□Er:□Offline.	Local 状態で実行した場合
@SetTimingOutInterval□-1000□Er:□Data□out□of□range.	引数が範囲外の場合

[例]

（送信側）

```
pm16c04.th□SetTimingOutInterval□1000
```

（返信されてくる文字列）

```
pm16c04.th>test□@SetTimingOutInterval□1000□Ok:
```

GetTimingOutMode

このコマンドを送信することで対応するモータのタイミングアウトモードの値を返します。

[戻り値]

- 0 タイミングパルス出力をおこないません
- 1 開始点から終了点の間、TTL のゲートパルスを出力
- 2 開始点から終了点の間、設定インターバル毎に、200ns の TTL パルスを出力
- 3 開始点から終了点の間、設定インターバル毎に、10 μ s の TTL パルスを出力
- 4 開始点から終了点の間、設定インターバル毎に、100 μ s の TTL パルスを出力
- 5 開始点から終了点の間、設定インターバル毎に、1ms の TTL パルスを出力

[例]

(送信側)

```
pm16c04.th□GetTimingOutMode
```

(返信されてくる文字列)

```
pm16c04.th>test□@GetTimingOutMode□0
```

GetTimingOutStart

このコマンドを送信することで対応するモータのタイミングアウトの開始点の値を返します。

[戻り値]

-2147483647 から 2147483647 までの数値文字列

[例]

(送信側)

```
pm16c04.th□GetTimingOutStart
```

(返信されてくる文字列)

```
pm16c04.th>test□@GetTimingOutStart□-10000
```

GetTimingOutEnd

このコマンドを送信することで対応するモータのタイミングアウトの終了点の値を返します。

[戻り値]

-2147483647 から 2147483647 までの数値文字列

[例]

(送信側)

```
pm16c04.th□GetTimingOutEnd
```

(返信されてくる文字列)

```
pm16c04.th>test□@GetTimingOutEnd□10000
```

GetTimingOutInterval

このコマンドを送信することで 対応するモータのタイミングアウトのインターバルの値を返します。

[戻り値]

0 から 2147483647 までの数値文字列

[例]

(送信側)

```
pm16c04.th>GetTimingOutInterval
```

(返信されてくる文字列)

```
pm16c04.th>test>@GetTimingOutInterval>1000
```

SetTimingOutReady

このコマンドを送信することで対応するモータのタイミングアウトの ready を設定します。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータが Busy 状態のときはエラーとなり実行されません。

[引数]

0 タイミングアウトの ready 状態を解除します

1 タイミングアウトの ready 状態を設定します

[リプライ・メッセージ]

```
@SetTimingOutReady>1>Ok:
```

正常に動作した場合

```
@SetTimingOutReady>1>Er:>Busy.
```

モータが Busy 状態のため実行されなかった場合

[例]

(送信側)

```
pm16c04.th>SetTimingOutReady>1
```

```
pm16c04.th>GetTimingOutReady
```

ready 状態の確認をおこないます

(返信されてくる文字列)

```
pm16c04.th>test>@SetTimingOutReady>1>Ok:
```

```
pm16c04.th>test>@GetTimingOutReady>1
```

ready 状態が設定されていました

GetTimingOutReady

このコマンドを送信することで対応するモータのタイミングアウトの ready の値を返します。

[戻り値]

- 0 タイミングアウトの ready 状態は解除されています
- 1 タイミングアウトの ready 状態は設定されています

[例]

(送信側)

```
pm16c04.th□GetTimingOutReady
```

(返信されてくる文字列)

```
pm16c04.th>test□@GetTimingOutReady□1
```

◆自動速度変更機能関連コマンド (PM16C-04X Only)

自動速度変更機能機能は PM16C-04X からの新機能です。実行にあたっては、ツジ電子（株）が提供する取扱説明書の該当箇所をよく読んでからおこなってください。

SetAutoChangeSpeed

このコマンドを送信することで対応するモータの自動速度変更テーブルを設定します。

当コマンドの値は Stars の GUI プログラム ‘pm16cconfigXACS’ を使って更新できます。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはモータが Busy 状態のときはエラーとなり実行されません。

[引数]

1. 行番号 0～127 までの数字文字列
2. 速度条件コード
 - ADD 開始位置から指定された相対座標（引数 3.）に達したら
 - TIM 前行の条件が実行されてから指定秒数（引数 3.）が経過したら
 - ACC 一定の速度（引数 3.）に達したら（但し加速中の場合）
 - DCC 一定の速度（引数 3.）に達したら（但し減速中の場合）
 - END 条件設定終了（引数 3. 以降の引数指定はありません）
3. 速度条件コード値
 - ・速度条件コードが ADD の場合座標値指定、-2147483647 から 2147483647（パルス）までの数値文字列（+符号の指定は不可）
 - ・速度条件コードが TIM の場合秒数指定、0 から 65535 までの数値文字列（ms）（+符号の指定は不可）
 - ・速度条件コードが ACC もしくは DCC の場合速度 pps 指定、1 から 5000000 (pps) までの数値文字列（+符号の指定は不可）
4. 動作種別
 - SPD 指定速度（引数 5.）への変更を開始します。
 - RTE 指定加減速レート（引数 5.）への変更を開始します。
 - SLW 減速停止を開始します（引数 5. 以降の引数指定はありません）
 - FST 緊急停止を開始します（引数 5. 以降の引数指定はありません）

NOP コメント行にします。当行の設定は無視されます。（引数 5. 以降の引数指定はありません）

5. 動作種別値

- ・動作種別が SPD の場合速度 pps 指定、1 から 5000000 (pps) までの数値文字列（+符号の指定は不可）
 - ・動作種別が RTE の場合加減速レートコード指定、0 から 115 (RateCode) までの数値文字列（+符号の指定は不可）
- 但し、GetAccRate コマンドの値に依存して動作可能となる加減速レートコードが絞り込まれます。詳細はツジ電子（株）の取扱説明書をお読みください。

[リプライ・メッセージ]

@SetAutoChangeSpeed□0□ADD□10000□SPD□2000□Ok:	正常に動作した場合(ADDxSPD)
@SetAutoChangeSpeed□0□ADD□10000□RTE□10□Ok:	正常に動作した場合(ADDxRTE)
@SetAutoChangeSpeed□0□ADD□10000□SLW□Ok:	正常に動作した場合(ADDxSLW)
@SetAutoChangeSpeed□0□ADD□10000□FST□Ok:	正常に動作した場合(ADDxFST)
@SetAutoChangeSpeed□0□ADD□10000□NOP□Ok:	正常に動作した場合(ADDxNOP)
@SetAutoChangeSpeed□0□TIM□1000□SPD□2000□Ok:	正常に動作した場合(TIMxSPD)
@SetAutoChangeSpeed□0□TIM□1000□RTE□10□Ok:	正常に動作した場合(TIMxRTE)
@SetAutoChangeSpeed□0□TIM□1000□SLW□Ok:	正常に動作した場合(TIMxSLW)
@SetAutoChangeSpeed□0□TIM□1000□FST□Ok:	正常に動作した場合(TIMxFST)
@SetAutoChangeSpeed□0□TIM□1000□NOP□Ok:	正常に動作した場合(TIMxNOP)
@SetAutoChangeSpeed□0□ACC□1000□SPD□2000□Ok:	正常に動作した場合(ACCxSPD)
@SetAutoChangeSpeed□0□ACC□1000□RTE□16□Ok:	正常に動作した場合(ACCxRTE)
@SetAutoChangeSpeed□0□ACC□1000□SLW□Ok:	正常に動作した場合(ACCxSLW)
@SetAutoChangeSpeed□0□ACC□1000□FST□Ok:	正常に動作した場合(ACCxFST)
@SetAutoChangeSpeed□0□ACC□1000□NOP□Ok:	正常に動作した場合(ACCxNOP)
@SetAutoChangeSpeed□0□DCC□300□SPD□10□Ok:	正常に動作した場合(DCCxSPD)
@SetAutoChangeSpeed□0□DCC□300□RTE□0□Ok:	正常に動作した場合(DCCxRTE)
@SetAutoChangeSpeed□0□DCC□300□SLW□Ok:	正常に動作した場合(DCCxSLW)
@SetAutoChangeSpeed□0□DCC□300□FST□Ok:	正常に動作した場合(DCCxFST)
@SetAutoChangeSpeed□0□DCC□300□NOP□Ok:	正常に動作した場合(DCCxNOP)
@SetAutoChangeSpeed□1□END□Ok:	正常に動作した場合(END)
@SetAutoChangeSpeed□ (正しい引数) □Er:□Busy.	モータが Busy 状態のため実行されなかった場合
@SetAutoChangeSpeed□ (正しい引数) □Er:□Offline.	Local 状態で実行した場合
@SetAutoChangeSpeed□128□ADD□10000□SLW□Er:□Bad□data□number.	行番号が範囲外の場合
@SetAutoChangeSpeed□0□ADT□10000□SLW□Er:□Bad□point.	速度条件コードが範囲外の場合
@SetAutoChangeSpeed□0□TIM□-1□SLW□Er:□Data□out□of□range.	

速度条件コード値が範囲外の場合

@SetAutoChangeSpeed00TIM010000RTF00Er:Badfunc.

速度種別が範囲外の場合

@SetAutoChangeSpeed00TIM010000RTE150Er:Dataoutofrange.

動作種別値が範囲外の場合

[例]

(送信側)

pm16c04.th>SetAutoChangeSpeed00ADD010000SPD02000

自動速度条件 0 : 開始座標から+10000 で速度を 2000pps にあげる。

pm16c04.th>SetAutoChangeSpeed01ADD020000SPD010000

自動速度条件 1 : 開始座標から+20000 で速度を 10000pps にあげる。

pm16c04.th>SetAutoChangeSpeed02ADD0120000SLW

自動速度条件 2 : 開始座標から+120000 で減速停止する。

pm16c04.th>SetAutoChangeSpeed03END

自動速度条件 3 : END なので自動速度変更の条件設定はここで終了。

(返信されてくる文字列)

pm16c04.th>test>@SetAutoChangeSpeed00ADD010000SPD02000Ok:

pm16c04.th>test>@SetAutoChangeSpeed00ADD020000SPD010000Ok:

pm16c04.th>test>@SetAutoChangeSpeed02ADD0120000SLWOk:

pm16c04.th>test>@SetAutoChangeSpeed03ENDOk:

GetAutoChangeSpeed

このコマンドを送信することで対応するモータの自動速度変更テーブルの値を返します。

当コマンドの値は Stars の GUI プログラム 'pm16cconfigXACS' を使って参照できます。

[引数]

行番号 0～127 までの数字文字列

[戻り値]

1. 速度条件コード

ADD 開始位置から指定された相対座標 (引数 2.) に達したら

TIM 前行の条件が実行されてから指定秒数 (引数 2.) が経過したら

ACC 一定の速度 (引数 2.) に達したら (但し加速中の場合)

DCC 一定の速度 (引数 2.) に達したら (但し減速中の場合)

END 条件設定終了 (引数 3. 以降はありません)

2. 速度条件コード値

- 速度条件コードが ADD の場合座標値、-2147483647 から 2147483647 (パルス) までの数値文字列

- 速度条件コードが TIM の場合秒数、0 から 65535 (ms) までの数値文字列

- ・ 速度条件コードが ACC もしくは DCC の場合速度 pps、1 から 5000000 (pps)
- ・ までの数値文字列

3. 動作種別

SPD	指定速度 (引数 4.) への変更を開始します。
RTE	指定加減速レート (引数 4.) への変更を開始します。
SLW	減速停止を開始します (引数 4. 以降はありません)
FST	緊急停止を開始します (引数 4. 以降はありません)
NOP	コメント行にします。当行の設定は無視されます。 (引数 4. 以降はありません)

4. 動作種別値

- ・ 動作種別が SPD の場合速度 pps、1 から 5000000 (pps) までの数値文字列
- ・ 動作種別が RTE の場合加減速レートコード、0 から 115 (RateCode) までの数値文字列

[リプライ・メッセージ]

@GetAutoChangeSpeed□0□ADD□10000□SPD□2000	正常に動作した場合(ADDxSPD)
@GetAutoChangeSpeed□0□ADD□10000□RTE□10	正常に動作した場合(ADDxRTE)
@GetAutoChangeSpeed□0□ADD□10000□SLW	正常に動作した場合(ADDxSLW)
@GetAutoChangeSpeed□0□ADD□10000□FST	正常に動作した場合(ADDxFST)
@GetAutoChangeSpeed□0□ADD□10000□NOP	正常に動作した場合(ADDxNOP)
@GetAutoChangeSpeed□0□TIM□1000□SPD□2000	正常に動作した場合(TIMxSPD)
@GetAutoChangeSpeed□0□TIM□1000□RTE□10	正常に動作した場合(TIMxRTE)
@GetAutoChangeSpeed□0□TIM□1000□SLW	正常に動作した場合(TIMxSLW)
@GetAutoChangeSpeed□0□TIM□1000□FST	正常に動作した場合(TIMxFST)
@GetAutoChangeSpeed□0□TIM□1000□NOP	正常に動作した場合(TIMxNOP)
@GetAutoChangeSpeed□0□ACC□1000□SPD□2000	正常に動作した場合(ACCxSPD)
@GetAutoChangeSpeed□0□ACC□1000□RTE□16	正常に動作した場合(ACCxRTE)
@GetAutoChangeSpeed□0□ACC□1000□SLW	正常に動作した場合(ACCxSLW)
@GetAutoChangeSpeed□0□ACC□1000□FST	正常に動作した場合(ACCxFST)
@GetAutoChangeSpeed□0□ACC□1000□NOP	正常に動作した場合(ACCxNOP)
@GetAutoChangeSpeed□0□DCC□300□SPD□10	正常に動作した場合(DCCxSPD)
@GetAutoChangeSpeed□0□DCC□300□RTE□0	正常に動作した場合(DCCxRTE)
@GetAutoChangeSpeed□0□DCC□300□SLW	正常に動作した場合(DCCxSLW)
@GetAutoChangeSpeed□0□DCC□300□FST	正常に動作した場合(DCCxFST)
@GetAutoChangeSpeed□0□DCC□300□NOP	正常に動作した場合(DCCxNOP)
@GetAutoChangeSpeed□1□END	正常に動作した場合(END)

[例]

(送信側)

pm16c04.th□GetAutoChangeSpeed□0

```
pm16c04.th□GetAutoChangeSpeed□1
pm16c04.th□GetAutoChangeSpeed□2
pm16c04.th□GetAutoChangeSpeed□3
```

(返信されてくる文字列)

```
pm16c04.th>test□@SetAutoChangeSpeed□0□ADD□10000□SPD□2000
pm16c04.th>test□@SetAutoChangeSpeed□0□ADD□20000□SPD□10000
pm16c04.th>test□@SetAutoChangeSpeed□2□ADD□120000□SLW
pm16c04.th>test□@ SetAutoChangeSpeed□3□END
```

SetAutoChangeSpeedReady

このコマンドを送信することで対応するモータの自動速度変更機能の **ready** を設定します。
このコマンドはコントローラが **Local** 状態の場合はエラーとなり実行されません。
またこのコマンドはモータが **Busy** 状態のときはエラーとなり実行されません。

[引数]

- 0 自動速度変更機能の **ready** 状態を解除します
- 1 自動速度変更機能の **ready** 状態を設定します

[リプライ・メッセージ]

@SetAutoChangeSpeedReady□1□Ok:	正常に動作した場合
@SetAutoChangeSpeedReady□1□Er:□Busy.	モータが Busy 状態のため実行されなかった場合
@SetAutoChangeSpeedReady□1□Er:□Offline.	Local 状態で実行した場合

[例]

(送信側)

```
pm16c04.th□SetAutoChangeSpeedReady□1
pm16c04.th□GetAutoChangeSpeedReady
```

ready 状態の確認をおこないます

(返信されてくる文字列)

```
pm16c04.th>test□@SetAutoChangeSpeedReady□1□Ok:
pm16c04.th>test□@GetAutoChangeSpeedReady□1
```

ready 状態が設定されていました

GetAutoChangeSpeedReady

このコマンドを送信することで対応するモータの自動速度変更機能の **ready** の値を返します。

[戻り値]

- 0 自動速度変更機能の **ready** 状態は解除されています
- 1 自動速度変更機能の **ready** 状態は設定されています

[例]

(送信側)

pm16c04.th□GetTimingOutReady

(返信されてくる文字列)

pm16c04.th>test□@GetTimingOutReady□1

◆モータ座標のプリセットコマンド

Preset

このコマンドを送信することでモータの現在位置を指定値で書き換えます。

このコマンドはコントローラが **Local** 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが **Busy** 状態のときはエラーとなり実行されません。

[引数]

[PM16C-04X シリーズの場合]

モータ位置の絶対値

-2147483647 から 2147483647 までの数値文字列（+符号の指定は不可）

[PM16C-04 シリーズの場合]

モータ位置の絶対値

-8388607 から ~8388607 までの数値文字列（+符号の指定は不可）

[リプライ・メッセージ]

@Preset□ (引数) □Ok:

正常に動作を終了した場合

@Preset□ (引数) □Er:□Data□Out□Of□Range.

モータのプリセット位置（絶対値）が有効範囲を超えたためエラーの場合

@Preset□ (引数) □Er:□Busy.

モータが **Busy** 状態のため実行されなかった場合

@Preset□ (引数) □Er:□Offline.

Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□Preset□10000

モータ th の現在値を 10000 に設定する

(返信されてくる文字列)

pm16c04.th>test□@Preset□10000□Ok:

正常に動作を終了した場合

◆モータ駆動・停止コマンド

以下、モータを駆動するコマンドです。

JogCw

このコマンドを送信することで CW の方向にモータを 1 パルス動かします。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが Busy 状態のときはエラーとなり実行されません。

[リプライ・メッセージ]

@JogCw□1

正常に動作を終了した場合

@JogCw□Er:□Busy.

モータが Busy 状態のため実行されなかった場合

@JogCw□Er:□Offline.

Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□JogCw

モータ th を CW の方向に 1Jog 移動する

(返信されてくる文字列)

pm16c04.th>test□@JogCw□1

正常に動作を終了した場合

JogCcw

このコマンドを送信することで CCW の方向にモータを 1 パルス動かします。

使用方法およびリプライ・メッセージはコマンド JogCw の説明を JogCcw、CW を CCW に置き換えてお読みください。

ScanCwConst

このコマンドを送信することで CW の方向に一定速走行でモータを連続して動かします。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが Busy 状態のときはエラーとなり実行されません。

[リプライ・メッセージ]

@ScanCwConst□1

正常に動作を終了した場合

@ScanCwConst□Er:□Busy.

モータが Busy 状態のため実行されなかった場合

@ScanCwConst□Er:□Offline.

Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□ScanCwConst

モータ th を CW の方向に一定速走行で連続して動かします

(返信されてくる文字列)

pm16c04.th>test□@ScanCwConst□1

正常に動作を終了した場合

ScanCwConst

このコマンドを送信することで CCW の方向に一定速走行でモータを連続して動かします。

使用方法およびリプライ・メッセージはコマンド ScanCwConst の説明を ScanCwConst、CW を CCW に置き換えてお読みください。

ScanCw

このコマンドを送信することで CW の方向にモータを連続して動かします。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが Busy 状態のときはエラーとなり実行されません。

[リプライ・メッセージ]

@ScanCw□1

正常に動作を終了した場合

@ScanCw□Er:□Busy.

モータが Busy 状態のため実行されなかった場合

@ScanCw□Er:□Offline.

Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□ScanCw

モータ th を CW の方向に連続して動かします

(返信されてくる文字列)

pm16c04.th>test□@ScanCw□1

正常に動作を終了した場合

ScanCcw

このコマンドを送信することで CW の方向にモータを連続して動かします。

使用方法およびリプライ・メッセージはコマンド ScanCw の説明を ScanCcw、CW を CCW に置き換えてお読みください。

ScanCwHome

このコマンドを送信することで CW の方向に HP（原点）を検出するまでモータを連続して動かします

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが Busy 状態のときはエラーとなり実行されません。

[リプライ・メッセージ]

@ScanCwHome□1

正常に動作を終了した場合

@ScanCwHome□Er:□Busy.

モータが Busy 状態のため実行されなかった場合

@ScanCwHome□Er:□Offline.

Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□ScanCwHome

モータ th を CW の方向に HP 検出まで連続して動かします

(返信されてくる文字列)

pm16c04.th>test□@ScanCwHome□1

正常に動作を終了した場合

ScanCwHome

このコマンドを送信することで CCW の方向に HP (原点) を検出するまでモータを連続して動かします

使用方法およびリプライ・メッセージはコマンド ScanCwHome の説明を ScanCcwHome、CW を CCW に置き換えてお読みください。

SetValue

このコマンドを送信することでモータの位置を絶対値で指定された値まで移動します。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが Busy 状態のときはエラーとなり実行されません。

また、コントローラコマンド「Standby」を実行して 2 台同時駆動 Standby 状態に入っている場合、コントローラコマンド「SyncRun」を実行するまで移動動作はおこなわれません。

[引数]

[PM16C-04X シリーズの場合]

モータ位置の絶対値

-2147483647 から 2147483647 までの数値文字列 (+符号の指定は不可)

[PM16C-04 シリーズの場合]

モータ位置の絶対値

-8388607 から ~8388607 までの数値文字列 (+符号の指定は不可)

[リプライ・メッセージ]

@SetValue□ (引数) □Ok:

正常に動作を終了した場合

@SetValue□ (引数) □Er:□Data□Out□Of□Range.

モータの移動目標位置 (絶対値) が有効範囲を超えたためエラーの場合

@SetValue□ (引数) □Er:□Busy.

モータが Busy 状態のため実行されなかった場合

@SetValue□ (引数) □Er:□Offline.

Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□SetValue□10000

モータ th を絶対値 10000 まで移動する

(返信されてくる文字列)

pm16c04.th>test□@SetValue□10000□Ok:

正常に動作を終了した場合

SetValueREL

このコマンドを送信することでモータの位置を現在位置からの相対値で指定された値まで移動します。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

またこのコマンドはメッセージ配信先として指定したモータが Busy 状態のときはエラーとなり実行されません。

また、コントローラコマンド「Standby」を実行して 2 台同時駆動 Standby 状態に入っている場合、コントローラコマンド「SyncRun」を実行するまで移動動作はおこなわれません。

[引数]

[PM16C-04X シリーズの場合]

モータ位置の絶対値

-2147483647 から 2147483647 までの数値文字列 (+ 符号の指定は不可)

[PM16C-04 シリーズの場合]

モータ位置の絶対値

-8388607 から ~8388607 までの数値文字列 (+ 符号の指定は不可)

[リプライ・メッセージ]

@SetValueREL□ (引数) □Ok:

正常に動作を終了した場合

@SetValueREL□ (引数) □Er:□Data□Out□Of□Range.

引数のモータの移動目標位置 (現在位置からの相対値) が有効範囲を超えたためエラーの場合

@SetValueREL□ (引数) □Er:□Busy.

モータが Busy 状態のため実行されなかった場合

@SetValueREL□ (引数) □Er:□Offline.

Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□SetValueREL□100

モータ th を現在位置から+の方向に 100 移動する

(返信されてくる文字列)

pm16c04.th>test□@SetValueREL□10000

正常に動作を終了した場合

Stop

このコマンドを送信すると対応するモータが減速停止します。

このコマンドはコントローラが Local 状態の場合操作は無視されます。

[リプライ・メッセージ]

@Stop□Ok: 正常に動作を終了した場合

[例]

(送信側)

pm16c04.th□Stop モータ th を減速停止します

(返信されてくる文字列)

pm16c04.th>test□@Stop□@Ok: 正常に動作を終了した場合

StopEmergency

このコマンドを送信すると対応するモータが緊急停止します。

このコマンドはコントローラが Local 状態の場合操作は無視されます。

[リプライ・メッセージ]

@StopEmergency□Ok: 正常に動作を終了した場合

[例]

(送信側)

pm16c04.th□StopEmergency モータ th を緊急停止します

(返信されてくる文字列)

pm16c04.th>test□@StopEmergency□@Ok: 正常に動作を終了した場合

◆モータ駆動中コマンド**SetSpeedCurrent (PM16C-04X Only)**

このコマンドを送信することで駆動中のモータの速度を変更します。

このコマンドはコントローラが Local 状態の場合はエラーとなり実行されません。

(モータが停止している場合は、エラーとならず設定が無視されます)

[引数]

モータの速度の値 (PPS) ※ 正の数値文字列 (+ 符号なし)

[リプライ・メッセージ]

@SetSpeedCurrent□ (引数) □Ok: 正常に動作を終了した場合

@SetSpeedCurrent□ (引数) □Er:□Offline. Local 状態で実行した場合

[例]

(送信側)

pm16c04.th□SetSpeedCurrent□10000

モータ th の速度値 (PPS) を
10000 で設定します

(返信されてくる文字列)

pm16c04.th>test□@SetSpeedCurrent□10000□Ok: 正常に動作を終了した場合

◆モータ現在状況確認コマンド

以下、モータの現在の状況を確認するコマンドです。

GetSelected

このコマンドを送信することで対応するモータのチャンネル割当状況がわかります。

[戻り値]

チャンネルが選択されている場合

A、B、C、D のいずれかの文字

チャンネルが選択されていない場合

N

[リプライ・メッセージ]

@GetSelected□A

A チャンネルに割り当てられている場合

@GetSelected□B

B チャンネルに割り当てられている場合

@GetSelected□C

C チャンネルに割り当てられている場合

@GetSelected□D

D チャンネルに割り当てられている場合

@GetSelected□N

A～D チャンネルいずれにも割り当てられていない場合

[例]

(送信側)

pm16c04.th□GetSelected

モータ th のチャンネル割当状況
を取得します

(返信されてくる文字列)

pm16c04.th>test□@GetSelected□A

A チャンネルに割り当てられて
います

GetValue

このコマンドを送信することでモータの現在位置を絶対値で取得します。

[戻り値]

[PM16C-04X シリーズの場合]

モータ位置の絶対値

-2147483647 から 2147483647 までの数値文字列

[PM16C-04 シリーズの場合]

モータ位置の絶対値

-8388607 から～8388607 までの数値文字列

[例]

(送信側)

pm16c04.th>GetValue

モータ th の現在位置の値を取得します

(返信されてくる文字列)

pm16c04.th>test@GetValue10000

モータ th の現在位置の値 10000 が返ってきます

IsBusy

このコマンドを送信することでモータが駆動しているか否かのデータを取得します。

[リプライ・メッセージ]

@Busy0

モータが停止状態

@Busy1

モータが駆動中の状態

[例]

(送信側)

pm16c04.th>IsBusy

モータ th の駆動状況を確認します

(返信されてくる文字列)

pm16c04.th>test@IsBusy0

モータ th が停止状態の場合

pm16c04.th>test@IsBusy1

モータ th が駆動中の状態の場合

◆モータイベント

_ChangedValue イベント

この イベントは対応 するモータの現在位置が変化した場合、もしくは ”flushdata” コマンド、”flushdatatome”コマンドを発行した場合に Stars Server から送られます。

[イベント・メッセージ]

_ChangedValue (モータの現在位置)

[例]

(送信側)

System flgon pm16c04.th

モータ th のイベント監視を開始します

pm16c04 flushdata

イベントメッセージの強制送信を指示します

(返信されてくる文字列)

System>test@flgon Node pm16c04.th has been registered.

pm16c04.th>test_ChangedValue10000

pm16c04.th>test_ChangedIsBusy0

pm16c04>test@flushdataOk:

_ChangedIsBusy イベント

このイベントは対応するモータの駆動状態が切り替わった場合、もしくは ”flushdata” コマンド、”flushdatatome” コマンドを発行した場合に Stars Server から送られます。

[イベント・メッセージ]

_ChangedIsBusy□1	モータが駆動中の状態
_ChangedIsBusy□0	モータが停止状態

[例]

(送信側)

System flgon pm16c04.th	モータ th のイベント監視を開始します
pm16c04 flushdata	イベントメッセージの強制送信を指示します

(返信されてくる文字列)

```
System>test□@flgon Node pm16c04.th has been registered.
pm16c04.th>test□_ChangedValue□10000
pm16c04.th>test□_ChangedIsBusy□0
pm16c04>test□@flushdata□Ok:
```

<補足>PM16C-04X について PM16C-04 からの変更点**PM16C-04X で使用できない PM16C04 のコマンド****GetSpeedList**

PM16C-04 では速度の値はコード表に基づいて設定されていましたが、PM16C-04X から速度を 1～5,000,000 までの値を直接設定できるようになったので、速度のコード表を参照する必要がなくなりました。

PM16C-04 と PM16C-04X で仕様の異なるコマンド**GetHighSpeed/GetMiddleSpeed/GetLowSpeed/SetHighSpeed/SetMiddleSpeed/SetLowSpeed**

PM16C-04 では速度はコード表に基づいた値を設定しますが、PM16C-04X では 1～5,000,000 までの値を設定するようになりました。

GetAccRateList/GetAccRate/SetAccRate

PM16C-04、PM16C-04X とともに加減速レートはコード表に基づいた値を設定しますが、設定する加減速レートのコード表が変わりました。

GetLimits/SetLimits

PM16C-04 と PM16C-04X で設定する値の仕様が変わりました。

GetValue/SetValue/SetValueREL/Preset/GetDigitalCwLs/SetDigitalCwLs/GetDigitalCcwLs/SetDigitalCcwLs

PM16C-04X で設定できる値の範囲が -2147483647 から 2147483647 まで拡張されました。

Select

PM16C-04 ではモータのチャンネル選択をおこなう際に既に別チャンネルでモータが選択されていると選択できませんでしたが、PM16C-04X では選択可能となり前に選択されていたチャンネルには別のモータが自動選択されます。

＜補足＞PM16C-04X の新機能

モータ移動形式の追加（モータコマンド）

PM16C-04X では、モータ移動形式として台形駆動／S字駆動／定速設定が選択できるようになりました。
モータの SetMotorSetup/GetMotorSetup コマンドをご参照ください。

モータ単位での速度レベルの選択（モータコマンド）

PM16C-04X では各モータ毎に動作速度レベル("Low","Middle","High")を設定できるようになりました。
モータコマンドの SpeedHigh/SpeedMiddle/SpeedLow/GetSpeedSelected. コマンドをご参照ください。

原点検索機能コマンドの追加（モータコマンド）

PM16C-04X では、原点検索機能が強化され新たなコマンドが追加されました。
モータコマンドの SetHPMode/SetHPOffset/SetHomePosition/GetHPMode/GetHPOffset/GetHomePosition
/ScanHome/ReScanHome をご参照ください。

タイミングアウト出力機能（モータコマンド）

PM16C-04X では、モータ動作中の TTL 出力をおこなえる機能が追加されました。
モータコマンドの SetTimingOutMode/SetTimingOutStart/SetTimingOutEnd/SetTimingOutInterval
/ SetTimingOutReady/GetTimingOutMode/GetTimingOutStart/GetTimingOutEnd /GetTimingOutInterval
/GetTimingOutReady をご参照ください。

モータ動作中の速度変更（モータコマンド）

PM16C-04X ではモータが動作中に速度を変更できるコマンドが追加されました。
モータコマンドの SetSpeedCurrent コマンドをご参照ください。

自動速度変更機能関連コマンド（モータコマンド）

PM16C-04X では、モータが動作してからの移動座標・経過時間・速度に応じて、自動的にモータの速度／加速度レート変更もしくは停止をおこなえる機能が追加されました。
モータコマンドの SetAutoChangeSpeed/SetAutoChangeSpeedReady/GetAutoChangeSpeed/
GetAutoChangeSpeedReady をご参照ください。

2 補軸ドライブ機能関連（コントローラコマンド）

PM16C-04X では、直線、円、円弧の軌跡を描くようにモータを動作させる機能が追加されました。
コントローラコマンドの DrawLine/DrawLineREL/DrawCircularCw/ DrawCircularCcw/DrawCircularCwREL
/DrawCircularCcwREL/DrawArc/DrawArcREL/DrawCircle/DrawCircleREL/をご参照ください。