

STARS npm2c01 用コマンド集

2006.1.13 版

STARS 経由でパルスモータ npm2c01 (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) npm2c01.theta□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) npm2c01.theta>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。

[メッセージ配信先名]

Stars のノード名が npm2c01 の場合 (Stars のノード名は npm2c01 I/O Client プログラム起動に指定します)

npm2c01

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

npm2c01.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) npm2c01.theta>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。

(npm2c01 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) is down.

例) npm2c01.thetaGetValue
npm2c01>test@GetValueEr:npm2c01.thetaisdown.

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) npm2c01.thetaGetValue
npm2c01>test@GetValueEr:Badcommandorparameter

コントローラコマンド

[メッセージ配信先名]

npm2c01

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）

npm2c01□hello

（返信されてくる文字列）

npm2c01>test□@hello□nice□to□meet□you.

GetMotorList

このコマンドを送信すると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）

[例]

（送信側）

npm2c01□GetMotorList

（返信されてくる文字列）

npm2c01>test□@GetMotorList□theta□DTH

左から順にモータ No.0 の名称
'theta'、No.1 の名称'DTH'を返しま
す

[補足]

モータの名称の定義方法につきましては、npm2c01 取扱説明書を参照してください。

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 もしくは 1

[リプライ・メッセージ]

@GetMotorName□ (引数) □ (モータ名)

データ取得が正常にお
こなわれた場合

@GetMotorName□ (引数) □Er:□Bad□command□or□parameter

モータ No. が間違っ
ていてエラーの場合

[例]

(送信側)

npm2c01□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)

npm2c01>test□@GetMotorName□0□theta

モータ No.0 のモータ名 'theta' が返ってき
ます

[補足]

モータの名称の定義方法につきましては、npm2c01 取扱説明書を参照してください。

GetRomVersion

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。

[例]

(送信側)

npm2c01□GetRomVersion

(返信されてくる文字列)

npm2c01>test□@GetRomVersion□ 1.00□04-10-02 バージョンと日付が返ってきます

GetSpeedList

このコマンドを送信すると有効なモータの速度 (PPS) の一覧を返します。

[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)

[例]

(送信側)

npm2c01□GetSpeedList

(返信されてくる文字列)

npm2c01>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

[関連コマンド]

SetHighSpeed、SetMiddleSpeed、SetLowSpeed、GetHighSpeed、GetMiddleSpeed、GetLowSpeed

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率の一覧を返します。

[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)

npm2c01□GetAccRateList

(返信されてくる文字列)

npm2c01>test□@GetAccRateList 1000 800 600 500 400 300 200 150 125 100 75 50 30
 20 15 10 7.5 5 3

[関連コマンド]

SetAccRate、GetAccRate

GetSpeedSelect

このコマンドを送信するとモータ No.1、No2 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]

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

[例]

(送信側)

```
npm2c01□GetSpeedSelect
( 返信されてくる文字列 )
npm2c01>test□@GetSpeedSelect□M
```

[関連コマンド]

SpeedHigh、SpeedMiddle、SpeedLow

GetLSIV

このコマンドを送信することで、モータ No.1、No2 両方のリミットスイッチ入力ビット「Limit Switch Invert bit」の現在値を返します。

[戻り値]

ビット値 XX (X : 1 6 進数 1 桁目 : モータ No.1 用 2 桁目 : モータ No.0 用)

X のビット仕様

bit3	HP LS Invert Bit	(0: Normaly Open 1:Normaly Close)
bit2	NG LS Invert Bit	(0: Normaly Open 1:Normaly Close)
bit1	CCW LS Invert Bit	(0: Normaly Open 1:Normaly Close)
bit0	CW LS Invert Bit	(0: Normaly Open 1:Normaly Close)

[リブライ・メッセージ]

@GetLSIV□FF 正常に動作を終了した場合

[例]

(送信側)

```
npm2c01□GetLSIV
```

(返信されてくる文字列)

```
npm2c01>test□@GetLSIV□FF
```

[関連コマンド]

SetLSIV、SetLSEN、GetLSEN、SetFLAG、GetFLAG、GetFunctionStatus
SetDigitalCwLs、SetDigitalCcwLs

GetLSEN

このコマンドを送信することで、モータ No.1、No2 両方のリミットスイッチ有効ビット「Limit Switch Enable bit」の現在値を返します。

[戻り値]

ビット値 XX (X : 1 6 進数 1 桁目 : モータ No.1 用 2 桁目 : モータ No.0 用)

X のビット仕様

bit3	HP LS Enable Bit	(1: Enable 0:Disable)
bit2	NG LS Enable Bit	(1: Enable 0:Disable)
bit1	CCW LS Enable Bit	(1: Enable 0:Disable)
bit0	CW LS Enable Bit	(1: Enable 0:Disable)

[リプライ・メッセージ]

@GetLSEN□FF 正常に動作を終了した場合

[例]

(送信側)

npm2c01□GetLSEN

(返信されてくる文字列)

npm2c01>test□@GetLSEN□FF

[関連コマンド]

SetLSEN、SetLSIV、GetLSIV、SetFLAG、GetFLAG、GetFunctionStatus
SetDigitalCwLs、SetDigitalCcwLs

SpeedHigh

このコマンドを送信することで、以降モータ No.1、No2 の速度は ' High ' の設定で動きます。
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。

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

[例]

(送信側)

npm2c01□SpeedHigh

(返信されてくる文字列)

npm2c01>test□@SpeedHigh□Ok:

[関連コマンド]

GetSpeedSelect、GetSpeedList、SetHighSpeed、GetHighSpeed

[注意]

電源を再投入すると、モータの速度は'Middle'で動くようリセットされます。

SpeedMiddle

このコマンドを送信することで、以降モータ No.1、No2 の速度は'Middle'で動きます。

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。

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

[例]

(送信側)

npm2c01□SpeedMiddle

(返信されてくる文字列)

npm2c01>test□@SpeedMiddle□Ok:

[関連コマンド]

GetSpeedSelect、GetSpeedList、SetMiddleSpeed、GetMiddleSpeed

[注意]

電源を再投入すると、モータの速度は'Middle'で動くようリセットされます。

SpeedLow

このコマンドを送信することで、以降モータ No.1、No2 の速度は'Low'で動きます。

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。

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

[例]

(送信側)

npm2c01□SpeedLow

(返信されてくる文字列)

npm2c01>test□@SpeedLow□Ok:

[関連コマンド]

GetSpeedSelect、GetSpeedList、SetLowSpeed、GetLowSpeed

[注意]

電源を再投入すると、モータの速度は'Middle'で動くようリセットされます。

SetLSIV

このコマンドを送信することで、モータ No.1、No2 両方のリミットスイッチ入力ビット「Limit Switch Invert bit」を設定します。

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。

[引数]

ビット値 XX (X: 1 6進数 1桁目: モータ No.1 用 2桁目: モータ No.0 用)

X のビット仕様

bit3	HP LS Invert Bit	(0: Normaly Open 1:Normaly Close)
bit2	NG LS Invert Bit	(0: Normaly Open 1:Normaly Close)
bit1	CCW LS Invert Bit	(0: Normaly Open 1:Normaly Close)
bit0	CW LS Invert Bit	(0: Normaly Open 1:Normaly Close)

[リプライ・メッセージ]

@SetLSIV□FF□Ok:	正常に動作を終了した場合
@SetLSIV□FF□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)

npm2c01□SetLSIV□FF

(返信されてくる文字列)

npm2c01>test□@SetLSIV□FF□Ok:

[関連コマンド]

GetLSIV、SetLSEN、GetLSEN、SetFLAG、GetFLAG、GetFunctionStatus
SetDigitalCwLs、SetDigitalCcwLs

SetLSEN

このコマンドを送信することで、モータ No.1、No2 両方のリミットスイッチ有効ビット「Limit Switch Enable bit」を設定します。

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。

[戻り値]

ビット値 XX (X: 1 6 進数 1 桁目: モータ No.1 用 2 桁目: モータ No.0 用)

X のビット仕様

bit3	HP LS Enable Bit	(1: Enable 0:Disable)
bit2	NG LS Enable Bit	(1: Enable 0:Disable)
bit1	CCW LS Enable Bit	(1: Enable 0:Disable)
bit0	CW LS Enable Bit	(1: Enable 0:Disable)

[リプライ・メッセージ]

@SetLSEN□FF□Ok:	正常に動作を終了した場合
@SetLSEN□FF□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)

npm2c01□SetLSEN□FF

(返信されてくる文字列)

npm2c01>test□@SetLSEN□FF□Ok:

[関連コマンド]

GetLSEN、SetLSIV、GetLSIV、SetFLAG、GetFLAG、GetFunctionStatus
SetDigitalCwLs、SetDigitalCcwLs

Stop

このコマンドを送信すると全てのモータが減速停止します。

[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)

npm2c01□Stop

全てのモータが減速停止します

(返信されてくる文字列)

npm2c01>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。

[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)

npm2c01□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)

npm2c01>test□@StopEmergency□@Ok:

正常に動作を終了した場合

Standby

2 台同時駆動 Standby 状態にします。

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]

@Standby□Ok:

正常に動作を終了した場合

@Standby□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)

npm2c01□Standby

2 台同時駆動 Standby 状態にします。

(返信されてくる文字列)

npm2c01>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。

[リプライ・メッセージ]

@SyncRun□Ok:

正常に動作を終了した場合

[例]

(送信側)

npm2c01□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)

npm2c01>test□@SyncRun□@Ok:

正常に動作を終了した場合

場合

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、エンコーダ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□npm2c01.theta`モータ名 'theta' のイベントメッセージの配信を
依頼します`npm2c01□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)

`npm2c01□@flushdata□Ok:`

コマンドが正常送信されました

`npm2c01.theta>test□_ChangedIsBusy□0`モータ名 theta 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`npm2c01.theta>test□_ChangedValue□100`モータ名 theta 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、エンコーダ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 の送信元に直接返します。

[例]

(送信側)

`npm2c01□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)

`npm2c01□@flushdatatome□Ok:`

コマンドが正常送信されました

`npm2c01.theta>test□_ChangedIsBusy□0`モータ名 theta 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`npm2c01.theta>test□_ChangedValue□100`モータ名 theta 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`npm2c01.DTH>test□_ChangedIsBusy□0`モータ名 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`npm2c01.DTH>test□_ChangedValue□100`モータ名 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

モータコマンド

[メッセージ配信先名]

npm2c01.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）

npm2c01.theta□hello

（返信されてくる文字列）

npm2c01.theta>test□@hello□nice□to□meet□you.

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）

npm2c01.theta□GetMotorNumber

モータ theta のモータ No. 番号を取得します

（返信されてくる文字列）

npm2c01.theta>test□@GetMotorNumber□0 モータ theta のモータ No.0 が返ってきます

GetHighSpeed

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

[例]

（送信側）

npm2c01.theta□GetHighSpeed

モータ theta の速度 ' High ' の設定値を取得

（返信されてくる文字列）

npm2c01.theta>test□@GetHighSpeed□1000 正常に動作した場合

[関連コマンド]

SetHighSpeed、SpeedHigh、GetSpeedSelect

GetMiddleSpeed

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

[例]

(送信側)

npm2c01.theta□GetMiddleSpeed モータ theta の速度 ' Middle ' の設定値を取得

(返信されてくる文字列)

npm2c01.theta>test□@GetMiddleSpeed□500 正常に動作した場合

[関連コマンド]

SetMiddleSpeed、SpeedMiddle、GetSpeedSelect

GetLowSpeed

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

[例]

(送信側)

npm2c01.theta□GetLowSpeed モータ theta の速度 ' Low ' の設定値を取得

(返信されてくる文字列)

npm2c01.theta>test□@GetLowSpeed□100 正常に動作した場合

[関連コマンド]

SetLowSpeed、SpeedLow、GetSpeedSelect

GetAccRate

このコマンドを送信することでモータの速度の加減速率の値 (mS/1000PPS) を返します。

[例]

(送信側)

npm2c01.theta□GetAccRate モータ theta の速度の加減速率の設定値を取得

(返信されてくる文字列)

npm2c01.theta>test□@GetAccRate□10 正常に動作した場合

[関連コマンド]

SetAccRate

GetDigitalCwLs

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

[例]

(送信側)

npm2c01.theta□GetDigitalCwLs

モータ theta 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)

npm2c01.theta>test□@GetCancelBacklash□40000

正常に動作した場合

[関連コマンド]

SetDigitalCwLs、GetLSEN、SetLSIV、GetLSIV、SetFLAG、GetFLAG、GetFunctionStatus

GetDigitalCwLs

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

[例]

(送信側)

npm2c01.theta□GetDigitalCwLs

モータ theta 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)

npm2c01.theta>test□@GetCancelBacklash□-40000

正常に動作した場合

[関連コマンド]

SetDigitalCwLs、GetLSEN、SetLSIV、GetLSIV、SetFLAG、GetFLAG、GetFunctionStatus

GetCancelBacklash

このコマンドを送信することで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。

[例]

(送信側)

npm2c01.theta□GetCancelBacklash

モータ theta 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)

npm2c01.theta>test□@GetCancelBacklash□100

正常に動作した場合

[関連コマンド]

SetCancelBacklash

GetFLAG

このコマンドを送信することで対応するモータの各種フラグビットの現在値を返します。

[戻り値]

各種フラグビット値 XX (X : 16 進数)

XX のビット仕様

bit4 DGLS Enable (CW/CCW LS 1:Enable 0:Disable)

bit0 LS Stop Mode (LS 検出時 0:Slow Stop 1:Emergency)

[リプライ・メッセージ]

@GetFLAG□10 正常に動作を終了した場合

[例]

(送信側)

npm2c01.theta□GetFLAG

(返信されてくる文字列)

npm2c01.theta>test□@GetFLAG□10

[関連コマンド]

SetFLAG、SetLSEN、GetLSEN、SetLSIV、GetLSIV、GetFunctionStatus

SetDigitalCwLs、SetDigitalCcwLs

GetLSIV

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入力ビット「Limit Switch Invert bit」の現在値を返します。

[戻り値]

ビット値 X (X : 16進数)

X のビット仕様

bit3	HP LS Invert Bit	(0: Normaly Open 1:Normaly Close)
bit2	NG LS Invert Bit	(0: Normaly Open 1:Normaly Close)
bit1	CCW LS Invert Bit	(0: Normaly Open 1:Normaly Close)
bit0	CW LS Invert Bit	(0: Normaly Open 1:Normaly Close)

[リプライ・メッセージ]

@GetLSIV□F 正常に動作を終了した場合

[例]

(送信側)

npm2c01.theta□GetLSIV

(返信されてくる文字列)

npm2c01.theta>test□@GetLSIV□F

[関連コマンド]

SetLSIV、SetLSEN、GetLSEN、SetFLAG、GetFLAG、GetFunctionStatus

SetDigitalCwLs、SetDigitalCcwLs

GetLSEN

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有効ビット「Limit Switch Enable bit」の現在値を返します。

[戻り値]

ビット値 X (X : 16進数)

X のビット仕様

bit3	HP LS Enable Bit	(1: Enable 0:Disable)
bit2	NG LS Enable Bit	(1: Enable 0:Disable)
bit1	CCW LS Enable Bit	(1: Enable 0:Disable)
bit0	CW LS Enable Bit	(1: Enable 0:Disable)

[リプライ・メッセージ]

@GetLSEN□F

正常に動作を終了した場合

[例]

(送信側)

npm2c01.theta□GetLSEN

(返信されてくる文字列)

npm2c01.theta>test□@GetLSEN□F

[関連コマンド]

SetLSEN、SetLSIV、GetLSIV、GetFLAG、SetFLAG、GetFunctionStatus

SetDigitalCwLs、SetDigitalCcwLs

GetHold

このコマンドを送信することで対応するモータのホールドオンオフ信号の値を返します。

[リプライ・メッセージ]

@GetHold□0

モータがホールドオフ状態

@GetHold□1

モータがホールドオン状態

[例]

(送信側)

npm2c01.theta□GetHold

モータ theta の駆動状況を確認します

(返信されてくる文字列)

npm2c01.theta>test□@GetHold□1

正常に動作した場合

[関連コマンド]

SetHold

SetHighSpeed

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

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'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 npm2c01 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。

[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

[例]

(送信側)

npm2c01.theta□SetHighSpeed□10000

モータ theta 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)

npm2c01.theta>test□@SetHighSpeed□10000□Ok: 正常に動作を終了した場合

[関連コマンド]

GetSpeedList、GetHighSpeed、SpeedHigh、GetSpeedSelect

SetMiddleSpeed

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

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'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 npm2c01 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。

[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

[例]

(送信側)

npm2c01.theta□SetMiddleSpeed□500

モータ theta の速度 'Middle' の値 (PPS) を 500 で設定します

(返信されてくる文字列)

npm2c01.theta>test□@SetMiddleSpeed□500□Ok: 正常に動作を終了した場合

[関連コマンド]

GetSpeedList、GetMiddleSpeed、SpeedMiddle、GetSpeedSelect

SetLowSpeed

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

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 '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 npm2c01 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。

[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

[例]

(送信側)

npm2c01.theta□SetLowSpeed□100

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

(返信されてくる文字列)

npm2c01.theta>test□@SetLowSpeed□100□Ok: 正常に動作を終了した場合

[関連コマンド]

GetSpeedList、GetLowSpeed、SpeedLow、GetSpeedSelect

SetAccRate

このコマンドを送信することでモータの速度の加減速率の値 (mS/1000PPS) を設定します。

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率の値

正の数値文字列 (+ 符号なし)

設定可能な速度の加減速率は、GetAccRateList コマンドで確認できる npm2c01 の速度の加減速率データ表に定義されている値です。

引数の速度の加減速率に一致する値が速度の加減速率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率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率が、速度の加減速率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□1000□Ok:

正常に動作を終了した場合

@SetAccRate□1000□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)

npm2c01.theta□SetAccRate□10000

モータ theta の速度の加減速率を設定

(返信されてくる文字列)

npm2c01.theta>test□@SetAccRate□10000□Ok: 正常に動作した場合

[関連コマンド]

GetAccRate

SetDigitalCwLs

このコマンドを送信することでモータの 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 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□40000□Ok:

正常に動作を終了した場合

@SetDigitalCwLs□40000□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)

npm2c01.theta□SetDigitalCwLs□40000

モータ theta 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)

npm2c01.theta>test□@SetDigitalCwLs□40000 正常に動作した場合

[関連コマンド]

SetDigitalCwLs、GetLSEN、SetLSIV、GetLSIV、SetFLAG、GetFLAG、GetFunctionStatus

SetDigitalCwLs

このコマンドを送信することでモータの 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 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□-40000□Ok:

正常に動作を終了した場合

@SetDigitalCwLs□-40000□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)

npm2c01.theta□SetDigitalCwLs□-40000

モータ theta 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)

npm2c01.theta>test□@SetDigitalCwLs□-40000 正常に動作した場合

[関連コマンド]

SetDigitalCcwLs、GetLSEN、SetLSIV、GetLSIV、SetFLAG、GetFLAG、GetFunctionStatus

SetCancelBacklash

このコマンドを送信することで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

[リプライ・メッセージ]

@SetCancelBacklash□100□Ok:

正常に動作を終了した場合

@SetCancelBacklash□100□Er:□Busy.

モータが Busy 状態のため実行されなかつ

た場合

[例]

(送信側)

npm2c01.theta□SetCancelBacklash□100

モータ theta 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)

npm2c01.theta>test□@SetCancelBacklash□100 正常に動作した場合

[関連コマンド]

GetCancelBacklash

SetFLAG

このコマンドを送信することで、対応するモータの各種フラグビット値を設定します。
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]

各種フラグビット値 XX (X : 1 6 進数)

XX のビット仕様

bit4 DGLS Enable (CW/CCW LS 1:Enable 0:Disable)

bit0 LS Stop Mode (LS 検出時 0:Slow Stop 1:Emergency)

[リプライ・メッセージ]

@SetFLAG□10□Ok:

正常に動作を終了した場合

@SetFLAG□10□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)

npm2c01.theta□SetFLAG□10

(返信されてくる文字列)

npm2c01.theta>test□@SetFLAG□10□Ok:

[関連コマンド]

GetFLAG、SetLSEN、GetLSEN、SetLSIV、GetLSIV、GetFunctionStatus

SetDigitalCwLs、SetDigitalCcwLs

SetLSIV

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入力ビット「Limit Switch Invert bit」を設定します。

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。

[引数]

ビット値 X (X: 16進数)

X のビット仕様

bit3	HP LS Invert Bit	(0: Normaly Open 1:Normaly Close)
bit2	NG LS Invert Bit	(0: Normaly Open 1:Normaly Close)
bit1	CCW LS Invert Bit	(0: Normaly Open 1:Normaly Close)
bit0	CW LS Invert Bit	(0: Normaly Open 1:Normaly Close)

[リプライ・メッセージ]

@SetLSIV□F□Ok: 正常に動作を終了した場合

@SetLSIV□F□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)

npm2c01.theta□SetLSIV□F

(返信されてくる文字列)

npm2c01.theta>test□@SetLSIV□F□Ok:

[関連コマンド]

GetLSIV、SetLSEN、GetLSEN、SetFLAG、GetFLAG、GetFunctionStatus

SetDigitalCwLs、SetDigitalCcwLs

SetLSEN

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有効ビット「Limit Switch Enable bit」を設定します。

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。

[引数]

ビット値 X (X: 16進数)

ビット仕様

bit3	HP LS Enable Bit	(1: Enable 0:Disable)
bit2	NG LS Enable Bit	(1: Enable 0:Disable)
bit1	CCW LS Enable Bit	(1: Enable 0:Disable)
bit0	CW LS Enable Bit	(1: Enable 0:Disable)

[リプライ・メッセージ]

@SetLSEN□F□Ok:	正常に動作を終了した場合
@SetLSEN□F□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)

npm2c01.theta□SetLSEN□F

(返信されてくる文字列)

npm2c01.theta>test□@SetLSEN□F□Ok:

[関連コマンド]

GetLSEN、SetLSIV、GetLSIV、SetFLAG、GetFLAG、GetFunctionStatus
SetDigitalCwLs、SetDigitalCcwLs

SetHold

このコマンドを送信することで対応するモータのホールドオンオフ信号の値を設定します。

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	ホールドオフ
1	ホールドオン

[リプライ・メッセージ]

@SetHold□0□Ok:	正常に動作した場合
@SetHold□0□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)

npm2c01.theta□SetHold□1

モータ theta をホールドオンに設定します

(返信されてくる文字列)

npm2c01.theta >test□@SetHold□1□Ok:

正常に動作した場合

[関連コマンド]

GetHold

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

[例]

(送信側)

npm2c01.theta□Preset□10000

モータ theta の現在値を 10000 に設定する

(返信されてくる文字列)

npm2c01.theta >test□@Preset□10000□Ok:

正常に動作を終了した場合

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 Jog 単位 (= 1 パルス/Jog) 動かします。

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□Ok:

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

[例]

(送信側)

npm2c01.theta□JogCw

モータ theta を CW の方向に 1Jog 移動する

(返信されてくる文字列)

npm2c01.theta>test□@JogCw□Ok:

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 Jog 単位 (= 1 パルス/Jog) 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□Ok:

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

[例]

(送信側)

npm2c01.theta□ScanCwConst

モータ theta 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)

npm2c01.theta>test□@ScanCwConst□Ok:

正常に動作を終了した場合

ScanCcwConst

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 ScanCcwConst、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□Ok:

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

[例]

(送信側)

npm2c01.theta□ScanCw

モータ theta 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)

npm2c01.theta>test□@ScanCw□Ok:

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□Ok:

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

[例]

(送信側)

npm2c01.theta□ScanCwHome

モータ theta 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)

npm2c01.theta>test□@ScanCwHome□Ok:

正常に動作を終了した場合

ScanCcwHome

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]

モータの移動目標位置の絶対値

-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

[例]

(送信側)

npm2c01.theta□SetValue□10000

モータ theta を絶対値 10000 まで移動する

(返信されてくる文字列)

npm2c01.theta>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]

モータの移動目標位置の現在位置からの相対値

-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

[例]

(送信側)

npm2c01.theta□SetValueREL□100

モータ theta 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)

npm2c01.theta>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モータが減速停止します。

[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)

npm2c01.theta□Stop

モータ theta を減速停止します

(返信されてくる文字列)

npm2c01.theta>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モータが緊急停止します。

[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)

npm2c01.theta□StopEmergency

モータ theta を緊急停止します

(返信されてくる文字列)

npm2c01.theta>test□@StopEmergency□@Ok:

正常に動作を終了した場合

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。

[例]

(送信側)

npm2c01.theta□GetValue

モータ theta の現在位置の値を取得します

(返信されてくる文字列)

npm2c01.theta>test□@GetValue□10000 モータ theta 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]

@Busy□0

モータが停止状態

@Busy□1

モータが駆動中の状態

[例]

(送信側)

npm2c01.theta□IsBusy

モータ theta の駆動状況を確認します

(返信されてくる文字列)

npm2c01.theta >test□@IsBusy□0

モータ theta が停止状態の場合

npm2c01.theta >test□@IsBusy□1

モータ theta が駆動中の状態の場合

GetFunctionStatus

このコマンドを送信することでモータの最新の LS、Hold Off ステータスを返します。

[戻り値]

LS、Hold Off ステータスビット X (X : 16 進数)

X のビット仕様

bit3 Hold Off (0: Hold Off 1:Hold On)

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

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

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

[リプライ・メッセージ]

@GetFunctionStatus□0

正常に動作を終了した場合

[例]

(送信側)

npm2c01.theta□GetFunctionStatus

モータ theta の最新ステータス値を取得します

(返信されてくる文字列)

npm2c01.theta >test□@GetFunctionStatus□0

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モータの最新の内部 CPU ステータスを返します。

[戻り値]

内部 CPU ステータスビット XX (X : 16 進数)

仕様については、ツジ電子が提供している取扱説明書に記述されているコマンド「S2x1」

を参照してください。

[リプライ・メッセージ]

@GetStatus□00

[例]

(送信側)

npm2c01.theta□GetStatus

モータ theta の最新ステータス値を取得します

(返信されてくる文字列)

npm2c01.theta>test□@GetStatus□00

正常に動作を終了した場合