

STARS クライアント ソフトウェア

ツジ電子ステッピングモータコントローラ

NPM2C-01 制御プログラム

TSUJI DENSHI Stepping Motor Controller
Model NPM2C-01 Stars I/O Client

npm2c01 取扱説明書

第一版

平成17年12月16日発行

物質構造科学研究所

コラボラトリ開発プロジェクト

承認欄

	物構研	放射光	コラボP	コラボP	作成
					永谷

はじめに

本書は、STARSのネットワーク上で利用されるツジ電子製のステッピングモータコントローラ NPM2C-01のStars I/O Clientプログラム「npm2c01」の取り扱いについて記述したものです。

STARSについて（略称：“Simple Transmission and Retrieval System”）

ネットワーク上において、測定機器や制御機器を制御するI/O Client、GUI(Graphical User Interface) 操作や解析表示を行うApplication Client、これらClientを管理し実行を制御するServerと、処理の分散化を目的として開発されたシステムで、アプリケーション間のメッセージ配信を行う小規模システム向けの通信機構です。

STARS におけるアプリケーション間通信は、TCP/IP Socket を利用したテキストベースのコマンドの送受により行われるため、システムでの扱いが非常に容易となり、それと同時に、開発言語やOS 選択の幅が広がっています。さらに、コアの部分となるプログラムはPerl を使って開発されているので、STARS は様々なプラットフォーム上で動作可能です。

目次

1	はじめに.....	1
1.1	npm2c01 について.....	1
1.2	ステッピングモータコントローラと npm2c01 Stars I/O Client の通信について.....	1
1.3	npm2c01 Stars I/O Client におけるモータの指定方法について.....	1
1.4	npm2c01 の Stars のノード名について.....	1
2	プログラム構成.....	2
2.1	npm2c01 Stars I/O Client プログラム一覧.....	2
2.2	npm2c01 Stars I/O Client の Stars 認証 Key について.....	2
3	プログラムの初期設定.....	3
3.1	npm2c01 の設定ファイルについて.....	3
3.1.1	モータ名の設定.....	3
3.1.2	Stars サーバ IP アドレスの設定.....	3
3.1.3	NPM2C-01 の IP アドレスとポート番号の設定.....	3
4	プログラムの起動.....	5
4.1	npm2c01 のプログラム起動方法について.....	5
5	コマンド一覧.....	6
5.1	コマンド一覧.....	6
6	イベントメッセージについて.....	9
6.1	イベントメッセージについて.....	9
6.2	イベントメッセージの種類.....	9
6.3	イベントメッセージの受信例.....	9
7	2台同時運転について.....	10
7.1	2台同時運転の実行手順について.....	10
7.2	2台同時運転の指定が解除されるタイミングについて.....	10
7.3	2台同時運転の実行例.....	10
8	モータ駆動時に実行できないコマンドについて.....	11
	参考.....	13

1 はじめに

1.1 npm2c01 について

npm2c01 はツジ電子製のステッピングモータコントローラ NPM2C-01 の制御を目的に Perl 言語で作成された Stars I/O Client プログラムです。

NPM2C-01 取扱説明書「2CH パルスモータコントローラ NPM2C-01 (Ethernet/USB 通信ポート標準装備)取扱説明書(2386 改訂 1)」を参照して作成しています。(npm2c01 のプログラムと同一フォルダに「npm2c_01.pdf」という名前のファイルで置いてあります。)

また、ステッピングモータコントローラ本体のシステムプログラムのVersion“1.00 04-10-02”での動作を確認しております。

1.2 ステッピングモータコントローラと npm2c01 Stars I/O Client の通信について

ステッピングモータコントローラNPM2C-01はLANもしくはUSB通信により遠隔制御をおこなうことが可能です。

NPM2C-01にネットワークIPアドレスとポート番号(デフォルト7777)を設定してLANにNPM2C-01の機器をつなげると、npm2c01 Stars I/O Clientとの通信ができるようになります。

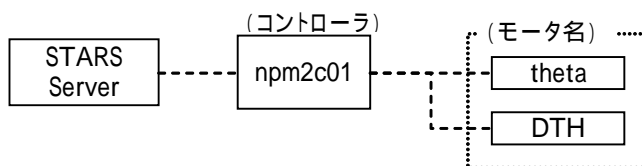
USB通信に関しては現時点でnpm2c01 Stars I/O Clientの動作検証をおこなっておりません。

USB通信ベースでnpm2c01 Stars I/O Clientを動かしたい場合は、別途ご相談ください。

1.3 npm2c01 Stars I/O Client におけるモータの指定方法について

npm2c01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ます。モータへの命令はStarsのメッセージ配信先名を"npm2c01.theta"('theta'という名前のモータを制御対象とする場合)といったようにStarsのメッセージの配信先名を階層化された形式で指定します。

[例] npm2c01の構成例



1.4 npm2c01 の Stars のノード名について

npm2c01のStarsのノード名は、npm2c01のI/O Client起動時の引数として指定します。

2 プログラム構成

2.1 npm2c01 Stars I/O Client プログラム一覧

npm2c01 Stars I/O Clientプログラムファイルは同一のフォルダで管理する必要があります。
デフォルトの格納フォルダ名は「npm2c01」です。

< npm2c01 Stars I/O Clientプログラムファイル >

ファイル名	内容	ファイル種別
npm2c01	Npm2c01 Stars I/O Client プログラム	perl
npm2c01.key	Stars ノード名「npm2c01」用の Stars 認証 Key	text
config.pl	Npm2c01 Stars I/O Client プログラム設定ファイル	perl
nportsvr.pl	Stars の NPORT 用ライブラリ	perl
stars.pm	Stars のライブラリ本体	perl

< npm2c01 Stars I/O Clientマニュアル類 >

ファイル名	内容	ファイル種別
npm2c01General.pdf	npm2c01 Stars I/O Client の取扱説明書	pdf
npm2c01CommandReference.pdf	npm2c01 Stars I/O Client のコマンド集	pdf
npm2c_01.pdf	ツジ電子ステッピングモータコントローラ NPM2C-01 取扱説明書 (npm2c01 Stars I/O Client 作成の参考元)	pdf

2.2 npm2c01 Stars I/O Client の Stars 認証 Key について

本プログラムには、Stars ノード名「npm2c01」用の Stars 認証 Key が既に入っておりますので、Stars のノード名を「npm2c01」とする場合には、準備された Stars 認証 Key を使ったプログラムの実行が可能です。

ちなみに、準備された Stars 認証 Key でなく、ユニークな Stars 認証 Key を新たに作成して使用することも可能です。Stars 認証 Key の作成方法に関して知りたい場合は別途お問い合わせください。

3 プログラムの初期設定

3.1 npm2c01 の設定ファイルについて

npm2c01のプログラムは設定ファイル「config.pl」を参照して動作します。
設定ファイルconfig.plの中身はperl言語の文法に則って記述する必要があります。

3.1.1 モータ名の設定

設定ファイル「config.pl」の記述方法を説明します。

[例] モータの1軸目を「theta」、2軸目を「DTH」という名前にする場合

< config.plの記述例 >

```
@::MotorName = qw(theta DTH);          # Channel A:theta B:DTH
```

(モータ名の記述)

変数@::MotorName の qw 関数のパラメータとして、左から順に「Channel A のモータ名 ChannelB のモータ名」といった風に、軸数分スペース1文字区切りでモータ名を記述していきます。モータ名にスペースを含めてはいけません。

3.1.2 Stars サーバ IP アドレスの設定

StarsサーバのIPアドレスを、設定ファイルconfig.plに記述します。

[例]StarsサーバのIPアドレスを192.168.11.100にする場合

< config.plの記述例 >

```
$::Server = 192.168.11.100;
```

(サーバ名の記述)

変数\$::Server に Stars サーバの IP アドレスを記述します。ホスト名による指定も可能です。

3.1.3 NPM2C-01 の IP アドレスとポート番号の設定

NPM2C-01本体のIPアドレスとポート番号を、設定ファイルconfig.plに記述します。

[例]NPM2C-01本体のIPアドレスが192.168.11.121、ポート番号7777の場合

< config.plの記述例 >

```
$::NPORT_HOST = '192.168.11.121';  
$::NPORT_PORT = 7777;
```

(NPM2C-01 本体の IP アドレスの記述)

変数\$::NPORT_HOST に ' NPM2C-01 本体の IP アドレス' を記述します。'ホスト名'の指定も可能です。

(NPM2C-01 本体のポート番号の記述)

変数\$::NPORT_PORT に ' NPM2C-01 本体の Port 番号 'を記述します。

4 プログラムの起動

4.1 npm2c01 のプログラム起動方法について

[例]npm2c01というStarsのノード名でStarsのI/O Client npm2c01を起動する場合

コマンドプロンプトから起動します。

```
C:¥>cd npm2c01  
C:¥npm2c01>perl npm2c01 npm2c01
```

(1行目)

npm2c01 プログラムファイル格納フォルダをカレントディレクトリとします。

(2行目)

Perl コマンドの1番目の引数は Stars I/O Client npm2c01 の Perl プログラムのファイル名です。

Perl コマンドの2番目の引数は Stars のノード名です。ノード名はユーザが自由に決めることができますが、その場合は Stars 認証 Key(ノード名.key)の準備を忘れずおこなってください。

ちなみに、-d オプションを付けることによってデバッグメッセージをスクリーン出力することができます。

```
C:¥npm2c01>perl npm2c01 npm2c01 -d
```


5 コマンド一覧

5.1 コマンド一覧

使用できるコマンドは以下の通りです。各コマンドの使用方法は別冊のnpm2c01のコマンド解説書をご確認ください。

コントローラコマンド一覧

	コマンド	機能	pm16c04 との互換性
情報	hello	STARS の通信が行われているかをチェックするコマンド 単に@hello nice to meet you.の文字列を返します	
	GetMotorList	モータ名のリストを返します	
	GetMotorName	モータ番号(0, 1)に対応するモータ名を返します	
	GetRomVersion	PMC のファームウェアのバージョン情報を返します	
	GetSpeedList	モータ速度のコード表のリストを返します	
	GetAccRateList	モータ速度の加減速率のコード表のリストを返します	
	GetSpeedSelect	現在選択されている速度の種類(Low/Middle/High)を返します	-
	GetLSIV	Channel A・B 両方の「Limit Switch Invert bit」の値を返します。	-
	GetLSEN	Channel A・B 両方の「Limit Switch Enable bit」の値を返します。	-
設定	SpeedHigh	モータは速度'High'の設定で動きます	1
	SpeedMiddle	モータは速度'Middle'の設定で動きます	1
	SpeedLow	モータは速度'Low'の設定で動きます	1
	SetLSIV	Channel A・B 両方の「Limit Switch Invert bit」を設定します 設定値:XX(16進数2桁 1桁目:bit 4-7, 2桁目:bit 0-3) {bit 7: Bch HP LS, 6: Bch NG LS, 5: Bch CCW LS, 4: Bch CW LS, bit 3: Ach HP LS, 2: Ach NG LS, 1: Ach CCW LS, 0: Ach CW LS} 全てのリミットスイッチを NC(Normaly Close)とするときは'FF'、NO(Normaly Open)とするときは'00'となります	-
	SetLSEN	Channel A・B 両方の「Limit Switch Enable bit」を設定します 設定値:XX(16進数2桁 1桁目:bit 4-7, 2桁目:bit 0-3) {bit 7: Bch HP LS, 6: Bch NG LS, 5: Bch CCW LS, 4: Bch CW LS, bit 3: Ach HP LS, 2: Ach NG LS, 1: Ach CCW LS, 0: Ach CW LS} 全てのリミットスイッチを有効とするときは'FF'、無効とするときは'00'となります	-
駆動	Stop	全てのモータを台形駆動の設定に沿って減速停止させます	
	StopEmergency	全てのモータを台形駆動の設定に関係なく瞬時に停止させます	
	Standby	同時駆動のために全チャンネルを Standby 状態にします	
	SyncRun	同時駆動を開始します	
状態	flushdata	コントローラ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	2

		サーバ'System'に返します	
	flushdatatome	コントローラ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	2

1 電源再投入時'Middle'にリセットされます

2 返されるイベントの種類は異なります

モータコマンド一覧

	コマンド	機能	pm16c04 との互換性
情報	hello	STARS の通信が行われているかをチェックするコマンド 単に@hello nice to meet you.の文字列を返します	
	GetMotorNumber	モータ名に対応するモータ番号を返します	
	GetHighSpeed	モータの速度'High'の値(PPS)を返します	
	GetMiddleSpeed	モータの速度'Middle'の値(PPS)を返します	
	GetLowSpeed	モータの速度'Low'の値(PPS)を返します	
	GetAccRate	モータの速度の加減速率の値(mS/1000PPS)を返します	
	GetDigitalCwLs	モータの CW ソフトウェアリミットスイッチの値(パルス)を返します	
	GetDigitalCcwLs	モータの CCW ソフトウェアリミットスイッチの値(パルス)を返します	
	GetCancelBacklash	モータのバックラッシュ補正ステップ数の値(パルス)を返します	
	GetFLAG	モータに対応するチャンネルの各種フラグ値を返します	-
	GetLSIV	モータに対応するチャンネルの「Limit Switch Invert bit」の値を返します。	-
	GetLSEN	モータに対応するチャンネルの「Limit Switch Enable bit」の値を返します。	-
	GetHold	モータに対応するチャンネルのホールドオフ信号の値を返します。	-
設定	SetHighSpeed	モータの速度'High'の値(PPS)を設定します	
	SetMiddleSpeed	モータの速度'Middle'の値(PPS)を設定します	
	SetLowSpeed	モータの速度'Low'の値(PPS)を設定します	
	SetAccRate	モータの速度の加減速率の値(mS/1000PPS)を設定します	
	SetDigitalCwLs	モータの CW ソフトウェアリミットスイッチの値(パルス)を設定します	
	SetDigitalCcwLs	モータの CCW ソフトウェアリミットスイッチの値(パルス)を設定します	
	SetCancelBacklash	モータのバックラッシュ補正ステップ数の値(パルス)を設定します	
	SetFLAG	モータに対応するチャンネルの各種フラグを設定します 設定値:XX(16進数2桁 1桁目:bit 4-7, 2桁目:bit 0-3) {bit 5: backlash mode, 4: DGLS enable, 0: LS Stop mode}	-
	SetLSIV	モータに対応するチャンネルの「Limit Switch Invert bit」を設定します 設定値:X(16進数1桁 bit 0-3) {bit 3: Ach HP LS, 2: Ach NG LS, 1: Ach CCW LS, 0: Ach CW LS} 全てのリミットスイッチを NC(Normaly Close)とするときは'F'、NO(Normaly Open)とするときは'0'となります	-

	SetLSEN	モータに対応するチャンネルの「Limit Switch Enable bit」を設定します 設定値:X(16進数1桁 bit 0-3) {bit 3: HP LS, 2: NG LS, 1: CCW LS, 0: CW LS} 全てのリミットスイッチを有効とするときは'F'、無効とするときは'0'となります	-
	SetHold	モータに対応するチャンネルのホールドオフ信号(=0)もしくはホールドオフ解除信号(=1)を送ります	
	Preset	モータの現在値をプリセットします。	
駆動	JogCw	CW の方向にモータを1Jog 単位(=1パルス/Jog)動かします	
	JogCcw	CCW の方向にモータを1Jog 単位(=1パルス/Jog)動かします	
	ScanCwConst	CW の方向に一定速走行でモータを連続して動かします	
	ScanCcwConst	CCW の方向に一定速走行でモータを連続して動かします	
	ScanCw	CW の方向にモータを連続して動かします	
	ScanCcw	CCW の方向にモータを連続して動かします	
	ScanCwHome	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	
	ScanCcwHome	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	
	SetValue	絶対位置管理によりモータを目的位置まで動かします。	
	SetValueREL	相対位置管理によりモータを目的位置まで動かします。	
	Stop	モータを台形駆動の設定に沿って減速停止させます	
	StopEmergency	モータを台形駆動の設定に関係なく瞬時に停止させます	
座標	GetValue	モータの現在値を返します	
状態	IsBusy	モータが駆動中か否かを返します	
	GetFunctionStatus	モータに対応するチャンネルの CW LS/CCW LS/HP LS/Hold Off 状態を返します	× ₃
	GetStatus	モータに対応するチャンネルの内部 CPU ステータス(Busy Bit etc.)を返します	× ₃

3 返される BIT の仕様が異なります

イベント一覧

	コマンド	機能	pm16c04 との互換性
モータ	_ChangedIsBusy	モータが駆動中か否かを返します	
	_ChangedValue	モータの現在の位置情報を返します	
	_ChangedFunctionStatus	モータの現在の LS、Hold Off 状態を返します	-

6 イベントメッセージについて

6.1 イベントメッセージについて

_で始まるイベントメッセージは、Starsサーバ「System」に対して送信されます。

Starsのクライアントは「System flgon」コマンドを実行しておくことで、イベントメッセージの受信が可能となります。

6.2 イベントメッセージの種類

	イベント	返す値	System への送信タイミング
モータ	_ChangedIsBusy	1: 駆動中 0: 非駆動状態	コントローラコマンド「flushdata」実行時、 もしくは駆動状態が切り替わったとき
	_ChangedValue	モータの現在の位置情報(パルス)	コントローラコマンド「flushdata」実行時、もしくは値が変わったとき
	_ChangedFunctionStatus	モータの現在の LS、Hold Off ステータス情報を返します	コントローラコマンド「flushdata」実行時、もしくは値が変わったとき

6.3 イベントメッセージの受信例

[Starsコマンドターミナル上での実行例]

```
System flgon npm2c01.theta
System>term1 @flgon Node npm2c01.theta has been registered.
System flgon npm2c01.DTH
System>term1 @flgon Node npm2c01.DTH has been registered.
npm2c01 flushdata
npm2c01 @flushdata Ok:
npm2c01.theta>term1 _ChangedIsBusy 0
npm2c01.theta>term1 _ChangedValue 100
npm2c01.theta>term1 _ChangedFunctionStatus 0
npm2c01.DTH>term1 _ChangedIsBusy 0
npm2c01.DTH>term1 _ChangedValue 1
npm2c01.DTH>term1 _ChangedFunctionStatus 0
```

7 2台同時運転について

7.1 2台同時運転の実行手順について

1. コントローラコマンド「StandBy」を実行します。駆動が一時停止状態となります。
2. 1台のモータに対して移動命令コマンド「SetValue」を実行します。
モータへの移動命令が本体機器のメモリに記憶されます。この時点でモータは停止しています。
3. もう1台のモータに対して移動命令コマンド「SetValue」を実行します。
モータへの移動命令が本体機器のメモリに記憶されます。この時点でモータは停止しています。
4. コントローラコマンド「SyncRun」を実行します。2台のモータは同時に動き始めます。

7.2 2台同時運転の指定が解除されるタイミングについて

以下のタイミングで2台同時運転の指定は解除されます。

1. コントローラコマンド「SyncRun」コマンドを実行した場合。
2台同時運転が開始された時点で2台同時運転の指定は解除されます。

7.3 2台同時運転の実行例

[Starsコマンドターミナル上での実行例]

```
npm2c01 Standby
npm2c01>term1 @Standby Ok:
npm2c01.theta SetValue 110
npm2c01.theta>term1 @SetValue 110 :Ok
npm2c01.DTH SetValue 2
npm2c01.DTH>term1 @SetValue 2 Ok:
npm2c01 SyncRun
npm2c01>term1 @SyncRun Ok:
```

8 モータ駆動時に実行できないコマンドについて

モータ駆動時に、以下のコマンドを実行するとエラーが返ってきます。

コマンド		機能
コントローラ	SpeedHigh	モータは速度 'High' の設定で動きます
	SpeedMiddle	モータは速度 'Middle' の設定で動きます
	SpeedLow	モータは速度 'Low' の設定で動きます
	SetLSIV	Channel A・B 両方の「Limit Switch Invert bit」を設定します 設定値:XX(16 進数2桁 1桁目:bit 4-7, 2桁目:bit 0-3) {bit 7: Bch HP LS, 6: Bch NG LS, 5: Bch CCW LS, 4: Bch CW LS, bit 3: Ach HP LS, 2: Ach NG LS, 1: Ach CCW LS, 0: Ach CW LS} 全てのリミットスイッチを NC(Normaly Close)とするときは'FF'、NO(Normaly Open)とするときは'00'となります
	SetLSEN	Channel A・B 両方の「Limit Switch Enable bit」を設定します 設定値:XX(16 進数2桁 1桁目:bit 4-7, 2桁目:bit 0-3) {bit 7: Bch HP LS, 6: Bch NG LS, 5: Bch CCW LS, 4: Bch CW LS, bit 3: Ach HP LS, 2: Ach NG LS, 1: Ach CCW LS, 0: Ach CW LS} 全てのリミットスイッチを有効とするときは'FF'、無効とするときは'00'となります
モータ	SetHighSpeed	モータの速度 'High' の値(速度表のコード値)を設定します
	SetMiddleSpeed	モータの速度 'Middle' の値(速度表のコード値)を設定します
	SetLowSpeed	モータの速度 'Low' の値(速度表のコード値)を設定します
	SetAccRate	モータの速度の加減速率の値(速度の加減速率コード値)を設定します
	SetDigitalCwLs	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
	SetDigitalCcwLs	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
	SetCancelBacklash	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
	SetFLAG	モータに対応するチャンネルの各種フラグを設定します 設定値:XX(16 進数2桁 1桁目:bit 4-7, 2桁目:bit 0-3) {bit 5: backlash mode, 4: DGLS enable, 0: LS Stop mode}
	SetLSIV	モータに対応するチャンネルの「Limit Switch Invert bit」を設定します 設定値:X(16 進数1桁 bit 0-3) {bit 3: Ach HP LS, 2: Ach NG LS, 1: Ach CCW LS, 0: Ach CW LS}

			全てのリミットスイッチを NC(Normaly Close)とするときは'F'、NO (Normaly Open)とするときは'0'となります
		SetLSEN	モータに対応するチャンネルの「Limit Switch Enable bit」を設定します 設定値:X(16進数1桁 bit 0-3) {bit 3: HP LS, 2: NG LS, 1: CCW LS, 0: CW LS} 全てのリミットスイッチを有効とするときは'F'、無効とするときは'0'となります
		SetHold	モータに対応するチャンネルのホールドオフ信号(=0)もしくはホールドオフ解除信号(=1)を送ります
		Preset	モータの現在値をプリセットします。
駆動		JogCw	CW の方向にモータを1Jog 単位(=1パルス/Jog)動かします
		JogCcw	CCW の方向にモータを1Jog 単位で(=1パルス/Jog)動かします
		ScanCwConst	CW の方向に一定速走行でモータを連続して動かします
		ScanCcwConst	CCW の方向に一定速走行でモータを連続して動かします
		ScanCw	CW の方向にモータを連続して動かします
		ScanCcw	CCW の方向にモータを連続して動かします
		ScanCwHome	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
		ScanCcwHome	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
		SetValue	絶対位置管理によりモータを目的位置まで動かします。
		SetValueREL	相対位置管理によりモータを目的位置まで動かします。

参考

PM16C04 にあって NPM2C02 にないコマンド一覧

	コマンド/イベント	理由
モータ	GetLimits	NPMC02 に対応する命令が存在しないため
	SetLimits	NPMC02 に対応する命令が存在しないため
	GetJogPulse	NPMC02 が 1Jog 辺りのパルス数が 1 固定のため
	SetJogPulse	NPMC02 が 1Jog 辺りのパルス数が 1 固定のため
	_ChangedFunction	NPMC02 に Remote/Local 切り替えがないため
	_ChangedCtlIsBusy	NPMC02 がモータとチャンネルが 1 対 1 対応のため