

STARS クライアント ソフトウェア

ツジ電子製 8CH カウンタ／タイマ

NCT08 制御プログラム

Tsuji Electronics Model NCT08-0x Series
8 Channel Counter/Timer Stars I/O Client

nct08 取扱説明書

第一版

平成 21年1月7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8CHカウンタ／タイマ「NCT08-0xシリーズ」のStars I/O Clientプログラム「nct08」の取扱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	nct08 について	1
1.2	NCT08-0x カウンタ本体と nct08 Stars I/O Client の通信について	1
1.3	nct08 Stars I/O Client におけるカウンタの指定方法について	1
1.4	nct08 の Stars のノード名について	1
2	プログラム構成	2
2.1	nct08 Stars I/O Client プログラム一覧	2
2.2	nct08 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	nct08 の設定ファイルについて	3
3.1.1	カウンタ名の設定	3
3.1.2	Stars サーバ IP アドレスの設定	3
3.1.3	NCT08-0x 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	nct08 のプログラム起動方法について	5
4.2	nct08 のプログラム起動オプション-flushdata 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	8
6.1	イベントメッセージについて	8
6.2	イベントメッセージの種類	8
6.3	イベントメッセージの受信例	9
7	カウンタ駆動時に実行できないコマンドについて	10
8	計測サンプル	11

1 はじめに

1.1 nct08 について

nct08 はツジ電子製の8CH カウンタ／タイマ NCT08-0x 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。

nct08 では、内部タイマを用いた計測、プリセットカウンタ(CH7 を使用)を用いた計測、ならびに外部からの TTL 信号を利用しての計測が可能です。

1.2 NCT08-0x カウンタ本体と nct08 Stars I/O Client の通信について

NCT08-0x本体はTCP/IPインターフェイスとUSBインターフェイスをサポートしています。

本プログラムはTCP/IPによる相互通信をおこなうことを想定して作られています。

NCT08-0x本体のIPアドレスの設定方法は、ツジ電子が提供するNCT08-0xの取扱説明書に書かれています。取扱説明書はツジ電子のHP (<http://www.tsuji-denshi.co.jp/>) からダウンロードができます。

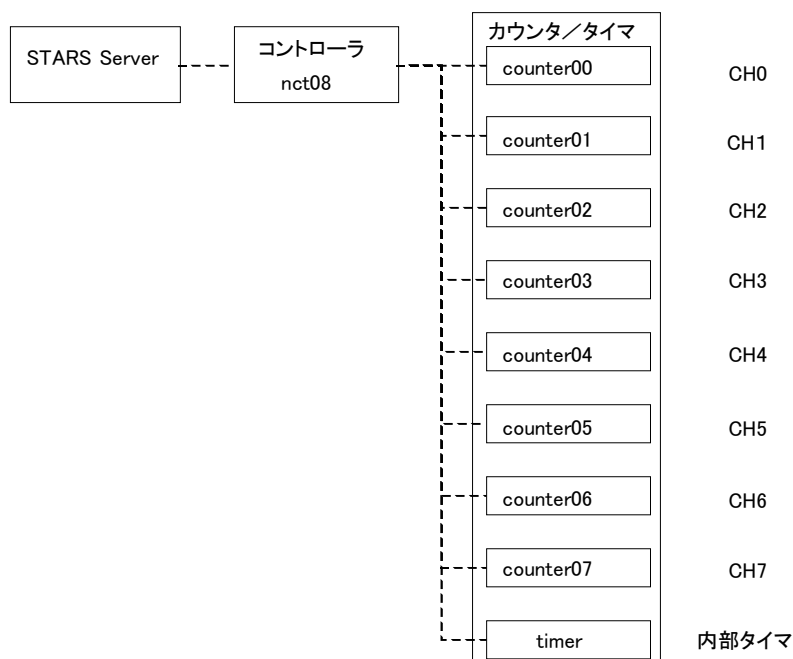
1.3 nct08 Stars I/O Client におけるカウンタの指定方法について

nct08では、コントローラに対してGetValueコマンドを送信することで、8CHのカウンタとタイマの全ての値を、(カンマ)区切りで同時に返します。また、nct08ではnct08.counter00のようにコントローラの先にカウンタ／タイマが接続されているような階層構造もサポートしていますので、カウンタへの命令をStarsのメッセージ配信先名“nct08.counter00”(「counter00」という名前のカウンタを制御対象とする場合)といった形式で指定することもできます。

[例] nct08の構成例

1.4 nct08 の Stars のノード名について

nct08のStarsのノード名は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 nct08 Stars I/O Client プログラム一覧

nct08 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「nct08」です。

<nct08 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
nct08	nct08 Stars I/O Client プログラム	perl
nct08.key	Stars ノード名「nct08」用の Stars 認証 Key	text
config.pl	nct08 Stars I/O Client プログラム設定ファイル	perl
nportserv.pm	Stars の NPORT 用ライブラリ	perl
stars.pm	Stars のライブラリ本体	perl

<nct08 Stars I/O Client マニュアル類>

ファイル名	内容	ファイル種別
nct08General.pdf	nct08 Stars I/O Client の取扱説明書	pdf
nct08CommandReference.pdf	nct08 Stars I/O Client のコマンド集	pdf

2.2 nct08 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「nct08」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「nct08」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 nct08 の設定ファイルについて

nct08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 カウンタ名の設定

設定ファイル「config.pl」の記述方法を説明します。

[例] カウンタのCH0からCH7とタイマを「counter00」、「counter01」、・・・「counter07」、・・・「timer」という名前にする場合

< config.plの記述例 >

```
@::CounterName = qw(counter00 counter01 counter02 counter03 counter04 counter05 counter06 counter07 timer);
```

(カウンタ名の記述)

変数@::CounterName の qw 関数のパラメータとして、左から順に「CH0 のカウンタ名 □ CH1 のカウンタ名 □・・・□ CH7 □タイマ」といった風に、軸数分スペース1文字区切りでカウンタ名を記述していきます。カウン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 NCT08-0x の IP アドレスとポート番号の設定

NCT08-0x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]NCT08-0xと通信可能な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;
```

(NCT08-0x の IP アドレスの記述)

変数\$::NPORT_HOST に' NCT08-0x と通信可能な NPORT の IP アドレス'を記述します。'ホスト名'の指定も可能です。

(NCT08-0x 本体のポート番号の記述)

変数\$::NPORT_PORT に' NCT08-0x と通信可能な NPORT の Port 番号'を記述します。

4 プログラムの起動

4.1 nct08 のプログラム起動方法について

[例]nct08というStarsのノード名でStarsのI/O Client nct08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 nct08  
C:\nct08 >perl nct08 nct08
```

(1行目)

nct08 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 nct08 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。

-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:\nct08 >perl nct08 nct08 -d
```

4.2 nct08 のプログラム起動オプション-flushdata について

カウント動作計測中(Busy 状態)の時に、カウンタ/タイマの値読み出しコマンド(本プログラムの GetValue コマンド)を実行すると、NCT08-0x 本体の仕様により約 120ns の間全てのカウンタ/タイマのカウント動作が停止します。

そこで nct08 プログラムのデフォルト仕様は、カウントの精度を損なわないことを最優先とし、多くの STARS I/O Client プログラムで行なわれている Busy 状態時のカウンタ/タイマの値のインターバル読み出しおよび値変更時の _ChangedValue イベント送信は行いません。

カウント動作が停止してからカウンタ/タイマの値読み出しをおこない、そこで値が変更されていれば _ChangedValue イベントを送信します。

しかしながら上記の仕様を理解した上で、Busy 状態時のインターバル読み出しおよび値変更時の _ChangedValue イベント送信が行われることで、カウンタ/タイマの値変更の推移を監査できるというメリットを優先したい場合を考慮して、起動オプションとして-flushdata オプションをサポートしました。

【-flushdata オプションの指定方法】

-flushdata (カウンタ/タイマの読み出し間隔ミリ秒 100 以上 50 の倍数で指定してください)

```
C:\nct08 >perl nct08 nct08 -flushdata 1000
```

上記の例では、1秒のインターバルでカウンタ/タイマの現在値を読み、値変更時に_ChangeValue イベントを送信します。

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nct08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
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
	GetDeviceType	Stars が接続しているカウンタの機種名を返します
	GetCounterList	Stars で扱うカウンタ／タイマの名前リストを返します
	GetCounterName	番号を指定して、該当する Stars で扱うカウンタ／タイマの名前を返します
	GetCounterNumber	Stars で扱うカウンタ／タイマの名前を指定して、該当する番号を返します
	GetStopMode	カウント動作の停止方法を返します
	GetTimerPreset	内部タイマのプリセット値を返します
	GetCountPreset	カウンタのプリセット値を返します
設定	SetStopMode	カウンタ動作の停止方法を設定します
	SetTimerPreset	内部タイマを用いたカウント計測時に参照されるタイマプリセット値を設定します
	SetCountPreset	プリセットカウントを用いたカウント計測時に参照されるプリセット値を設定します
	CounterReset	全てのカウンタとタイマの値を0リセットする、もしくは番号を指定して該当するカウンタ／タイマの値を0リセットします
駆動	CountStart	カウンタの計測動作を開始します
	Stop	カウンタの計測動作を停止します
カウンタ値	GetValue	全てのカウンタとタイマの現在値を返す、もしくは番号を指定して該当するカウンタ／タイマの現在値を返します
	IsOverflow	全てのカウンタとタイマがオーバーフローしたか否かを返す、もしくは番号を指定して該当するカウンタ／タイマがオーバーフローしたか否かを返します
状態	IsBusy	カウント動作中か否かを返します
	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 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

カウンタコマンド一覧

	コマンド	機能
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
	GetCounterNumber	指定されたカウンタ／タイマに該当する番号を返します
設定	CounterReset	指定されたカウンタ／タイマの値を0リセットします
カウンタ値	GetValue	指定されたカウンタ／タイマの現在値を返します
	IsOverflow	指定されたカウンタ／タイマがオーバーフローしたか否かを返します

イベント一覧

	コマンド	機能
コントローラ	_ChangedIsBusy	カウンタが計測動作中か否かを返します
カウンタ	_ChangedValue	カウンタ／タイマの現在値を返します
／タイマ	_ChangedIsOverflow	カウンタ／タイマの値がオーバーフローしているか否かを返します

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
コントローラ	_ChangedIsBusy	カウンタが計測動作中か否かを返します。	コントローラコマンド「flushdata」もしくは「flushdatatome」実行時、もしくは駆動状態が切り替わったとき
カウンタ／タイマ	_ChangedValue	カウンタ／タイマの現在値を返します	コントローラコマンド「flushdata」もしくは「flushdatatome」実行時、 起動オプション-flushdata 使用時は値が変更されたとき 起動オプション-flushdata 未使用時は計測停止時に値が変更されたとき
	_ChangedIsOverflow	カウンタ／タイマの値がオーバーフローしているか否かを返します	コントローラコマンド「flushdata」もしくは「flushdatatome」実行時、もしくは値が変更されたとき

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 nct08
System>term1 @flgon Node nct08 has been registered.
System flgon nct08.counter01
System>term1 @flgon Node nct08.counter01 has been registered.
System flgon nct08.counter02
System>term1 @flgon Node nct08.counter02 has been registered.
System flgon nct08.timer
System>term1 @flgon Node nct08.timer has been registered.
nct08 flushdata
nct08 @flushdata Ok:
nct08>term1 _ChangedIsBusy 0
nct08.counter01>term1 _IsOverflow 0
nct08.counter02>term1 _IsOverflow 0
nct08.timer>term1 _IsOverflow 0
nct08.counter01>term1 _ChangedValue 1
nct08.counter02>term1 _ChangedValue 60
nct08.timer>term1 _ChangedValue 1000000
```

1 行目の System flgon コマンドで Stars ノード名 nct08 (コントローラ)をみるよう設定します。

3 行目の System flgon コマンドで Stars ノード名 nct08.counter01 (CH1)をみるよう設定します。

5 行目の System flgon コマンドで Stars ノード名 nct08.counter02 (CH2)をみるよう設定します。

7行目の System flgon コマンドで Stars ノード名 nct08.timer (内部タイマ)をみるよう設定します。

9行目の nct08 flushdata コマンドでコントローラ／カウンタ／タイマの現在の状態の取得を要求します。

11行目で NCT08-01 が動作停止状態であることがわかります。

12、13、14行目で nct08 の CH1 と CH2 と内部タイマがオーバーフローしていないことがわかります。11

5、16、17行目で nct08 の CH1 と CH2 と内部タイマそれぞれの現在値が返ってきています。

7 カウンタ駆動時に実行できないコマンドについて

カウンタ駆動時に、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	SetStopMode	カウンタ動作の停止方法を設定します
	SetTimerPreset	内部タイマを用いたカウント計測時に参照されるタイマプリセット値を設定します
	SetCountPreset	プリセットカウントを用いたカウント計測時に参照されるプリセット値を設定します

8 計測サンプル

[Starsコマンドターミナル上での実行例]

内部タイマ1秒でCH0への入力を計測します。

```
(コントローラ、CH0、内部タイマについて System flgon をおこないます)

nct08 GetCounterList ... カウンタ／タイマ名を確認します

nct08>term1 @GetCounterList C00 C01 C02 C03 C04 C05 C06 C07 CTM

System flgon nct08

System>term1 @flgon Node nct08 has been registered.

System flgon nct08.C00

System>term1 @flgon Node nct08.C00 has been registered.

System flgon nct08.CTM

System>term1 @flgon Node nct08.CTM has been registered.


(計測停止中の状態を確認して、1秒間のタイマ計測を選択します)

nct08 IsBusy

nct08>term1 @IsBusy 0

nct08 SetStopMode T

nct08>term1 @SetStopMode T Ok:

nct08 SetTimerPreset 1000000

nct08>term1 @SetTimerPreset 1000000 Ok:


(全てのカウンタ／タイマを0リセットして計測を開始します)

nct08 CounterReset

nct08>term1 @CounterReset Ok:

nct08 CountStart

nct08>term1 _ChangedIsBusy 1 . ... イベント送信:計測が開始されました

nct08>term1 @CountStart Ok:


nct08>term1 _ChangedIsBusy 0 . ... イベント送信:計測が終わりました

nct08.C00>term1 _ChangedValue 1000011 ... イベント送信:CH0の現在値

nct08.CTM>term1 _ChangedValue 1000000 ... イベント送信:内部タイマの現在値


(計測停止してからカウンタ／タイマの現在値を読みます)

nct08 IsBusy

nct08>term1 @IsBusy 0

nct08.C00 GetValue

nct08.C00>term1 @GetValue 1000011

nct08.CTM GetValue

nct08.CTM>term1 @GetValue 1000000
```