

STARS nct08 用コマンド集

2011.1.25 版

STARS 経由で nct08（ツジ電子製 8CH カウンタ／タイマ NCT08-02、NCT08-01、NCT08-01B）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)f

※ □は半角スペースを意味しています

例) nct08□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) nct08>test□@GetValue□1000, 10, 0, 0, 0, 0, 0, 0, 10000000

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 nct08 の場合（Stars のノード名は nct08 I/O Client プログラム起動に指定します）

nct08

コントローラコマンド

カウンタのコントローラに対してメッセージを配信します

nct08.countername

カウンタコマンド

countername という名前のカウン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) nct08.counter01>test@Run Er: Busy.

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。

(nct08 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) nct08.counte01 GetValue
nct08>test@GetValue Er: nct08.counte01 is down.

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: Bad command or parameter

例) nct08.counter01 GetValue
nct08>test@GetValue Er: Bad command or parameter

コントローラコマンド	5
[メッセージ配信先名]	5
[コマンド]	5
hello	5
GetRomVersion	5
GetDeviceType	5
GetCounterList	6
GetCounterName 引数	6
GetCounterNumber 引数	6
SetStopMode	7
SetCountPreset	8
SetTimerPreset	8
GetStopMode	9
GetCountPreset	9
GetTimerPreset	9
CounterReset 引数なし	10
CounterReset 引数あり	10
GetValue 引数なし	11
GetValue 引数あり	11
IsOverflow 引数なし	12
IsOverflow 引数あり	12
CountStart	13
Stop	14
IsBusy	14
flushdata	14
flushdatatome	15
デバッグ用コマンド	17
devact NCT08-XX コマンド	17
devsend NCT08-XX コマンド	17
カウンタ/タイマコマンド	18
[メッセージ配信先名]	18
[コマンド]	18
hello	18
GetCounterNumber	18
CounterReset	18
GetValue	19
IsOverflow	19
コントローライベント	21
[イベント]	21

_ChangedIsBusy	21
カウンタ／タイマイベント	22
[イベント]	22
_ChangedValue	22
_ChangedIsOverflow	22

コントローラコマンド

[メッセージ配信先名]

nct08

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確認するコマンド。

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)

nct08□hello

(返信されてくる文字列)

nct08>test□@hello□nice□to□meet□you.

GetRomVersion

このコマンドを送信することで、プログラムから接続しているカウンタの機種名、バージョン情報を返します。

[例]

(送信側)

nct08□GetRomVersion

カウンタの機種・バージョン情報を問い合わせます。

(返信されてくる文字列)

nct08>test□@GetRomVersion□1.02□11-01-18□NCT08-02

機種名が NCT08-02 の場合、Ver1.02 の場合

GetDeviceType

このコマンドを送信することで、プログラムから接続しているカウンタの機種を返します。

[例]

(送信側)

nct08□GetDeviceType

カウンタの機種を問い合わせます。

(返信されてくる文字列)

nct08>test□@GetDeviceType□NCT08-02

機種名が NCT08-02 の場合

nct08>test□@GetDeviceType□NCT08-01

機種名が NCT08-01 の場合

nct08>test□@GetDeviceType□NCT08-01B

機種名が NCT08-01B の場合

GetCounterList

このコマンドを送信することで、全てのカウンタ／タイマの名前を” ”（スペース）区切りで返します。

[例]

（送信側）

nct08>GetCounterList

カウンタ／タイマ名を問い合わせます。

（返信されてくる文字列）

nct08>test @GetCounterList counter00 counter01 counter02 counter03 counter04 counter05
counter06 counter07 timer 正常にカウンタ／タイマ名のリストが返された場合

GetCounterName 引数

このコマンドを送信することで、引数のカウンタ／タイマに該当するカウンタ番号を返します。

[引数]

0 から 7 までの数字： カウンタ番号 CHx (x=0 to 7) を指定します。

8 の数字： タイマを指定します。

[リプライ・メッセージ]

@GetCounterName (カウンタ番号) (カウンタ名)

カウンタ番号 0 から 7 : CHx (x=0 to 7) に該当する
カウンタ名を返します。

@GetCounterName 8 (タイマ名)

番号 8 を指定するとタイマ名を返します。

@GetCounterName (引数) Er: Bad number.

引数であるカウンタ／タイマの番号が間違っている場合

[例]

（送信側）

nct08>GetCounterName 1

名前 counter01 に該当する番号を問い合わせます。

（返信されてくる文字列）

nct08>test @ GetCounterName 1 counter01 カウンタ名が正常に返された場合

GetCounterNumber 引数

このコマンドを送信することで、引数のカウンタ／タイマ名に該当するカウンタ／タイマ番号を返します。

[引数]

カウンタもしくはタイマの名前

[リプライ・メッセージ]

@GetCounterNumber□ (カウンタ名) □ (カウンタ番号)

引数がカウンタ名の場合はカウンタ番号 0 から 7 までの数字: CHx (x=0 to 7) を返します。

@GetCounterNumber□ (タイマ名) □8

引数がタイマ名の場合は番号 8 の数字を返します。

@GetCounterNumber□ (引数) □Er:□Bad name.

引数に該当するカウンタ/タイマ名が間違っている場合

[例]

(送信側)

nct08□GetCounterNumber□counter01

名前 counter01 に該当する番号を問い合わせます。

(返信されてくる文字列)

nct08>test□@ GetCounterNumber□counter01□1 カウンタ番号が正常に返された場合

SetStopMode

このコマンドを送信することで、カウンタの停止方法を設定します。

このコマンドはカウンタが **Busy** 状態のときはエラーとなり実行されません。

[引数]

C	プリセットカウンタが設定値までカウントするとカウンタを自動停止します
T	タイマが設定値までカウントするとカウンタを自動停止します
N	マニュアルで停止コマンドを発行する、もしくは TTL の STOP 信号入力があるまで自動停止しません。

[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□Er:□Busy. カウンタが **Busy** 状態のため実行されなかった場合

[例]

(送信側)

nct08□SetStopMode□C

プリセットカウンタを使用した自動停止に設定する場合

(返信されてくる文字列)

nct08>test□@SetStopMode□C□Ok: 正常に設定された場合

[関連コマンド]

GetStopMode、SetTimerPreset、GetTimerPreset、SetCountPreset、GetCountPreset

SetCountPreset

このコマンドを送信することで、カウンタのプリセット値を設定します。

このコマンドはカウンタが **Busy** 状態のときはエラーとなり実行されません。

[引数]

NCT08-01、NCT08-01B 機器の場合 1 から 4294967925 までの数字

NCT08-02 機器の場合 1 から 281474976710655 までの数字

[リプライ・メッセージ]

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

@SetCountPreset□Er:□Busy. カウンタが **Busy** 状態のため実行されなかった場合

[例]

(送信側)

nct08□SetCountPreset□1000 プリセットカウンタの値を 1000cts に設定する場合

(返信されてくる文字列)

nct08>test□@SetCountPreset□1000□Ok:

[関連コマンド]

GetCountPreset、SetStopMode、GetStopMode

SetTimerPreset

このコマンドを送信することで、タイマのプリセット値を設定します。

このコマンドはカウンタが **Busy** 状態のときはエラーとなり実行されません。

[引数]

NCT08-01 機器の場合 1 から 4294967925 までの数字

NCT08-02、NCT08-01B 機器の場合 1 から 1099511627775 までの数字

(上記数字の単位はマイクロ秒)

[リプライ・メッセージ]

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

@SetTimerPreset□Er:□Busy. カウンタが **Busy** 状態のため実行されなかった場合

[例]

(送信側)

nct08□SetTimerPreset□1000000 タイマプリセットカウンタの値を 1 秒に設定する場合

(返信されてくる文字列)


```
nct08>test@SetTimerPreset1000000Ok:
```

[関連コマンド]

GetTimerPreset、SetStopMode、GetStopMode

GetStopMode

このコマンドを送信するとカウンタのカウンタの停止方法を返します。

[リプライ・メッセージのコマンド]

@GetStopModeC :	プリセットカウンタを使用しての自動停止の場合
@GetStopModeT :	タイマカウンタを使用しての自動停止場合
@GetStopModeN :	手動停止の場合

[例]

(送信側)

```
nct08GetStopMode
```

カウンタの計測モードを取得します

(返信されてくる文字列)

```
nct08>test@GetStopModeC
```

プリセットカウンタを使用しての自動停止が選択されている場合

[関連コマンド]

SetStopMode、SetTimerPreset、GetTimerPreset、SetCountPreset、GetCountPreset

GetCountPreset

このコマンドを送信するとカウンタのプリセット値を返します

[リプライ・メッセージのコマンド]

@GetCountPreset (値)	カウンタのプリセット値が返された場合
---------------------	--------------------

[例]

(送信側)

```
nct08GetCountPreset
```

カウンタのプリセット値を取得します

(返信されてくる文字列)

```
nct08>test@GetCountPreset1000
```

カウンタのプリセット値 1000 が返された場合

[関連コマンド]

SetCountPreset、SetStopMode、GetStopMode

GetTimerPreset

このコマンドを送信するとタイマプリセット値を返します

[リプライ・メッセージのコマンド]

@GetTimerPreset□ (値)

タイマプリセット値が返された場合

[例]

(送信側)

nct08□GetTimerPreset

タイマプリセット値を取得します

(返信されてくる文字列)

nct08>test□@GetTimerPreset□1000

タイマプリセット値 1000 が返された場合

[関連コマンド]

SetTimerPreset、SetStopMode、GetStopMode

CounterReset 引数なし

このコマンドを送信することで、全てのカウンタ／タイマの値を 0 リセットします。

[リプライ・メッセージ]

@CounterReset□Ok:

正常に動作を終了した場合

@CounterReset□Er:□Busy.

カウンタが **Busy** 状態のため実行されなかった場合

[例]

(送信側)

nct08□CounterReset

全てのカウンタとタイマの値を 0 リセットします

(返信されてくる文字列)

nct08>test□@CounterReset□Ok:

正常に動作を終了した場合

CounterReset 引数あり

このコマンドを送信することで、指定されたカウンタ／タイマの値を 0 リセットします。

[引数]

0 から 7 までの数字 : CHx (x=0 to 7) を指定します。

8 の数字 : タイマを指定します。

[リプライ・メッセージ]

@CounterReset□ (引数) □Ok:

正常に動作を終了した場合

@CounterReset□ (引数) □Er:□Busy.

カウンタが **Busy** 状態のため実行されなかった場合

[例]

(送信側)

nct08□CounterReset□1

CH1 のカウンタを 0 リセットします

(返信されてくる文字列)

nct08>test□@CounterReset□1□Ok:

正常に動作を終了した場合

GetValue **引数なし**

このコマンドを送信することで、全てのカウンタ／タイマの値を返します。

[リプライ・メッセージのコマンド]

@GetValue□<データ>

データが複数ある場合は、(カンマ) 区切りで返します

[例]

nct08□GetValue

カウンタ値を取得します

(返信されてくる文字列)

nct08>test□@GetValue□1000, 10, 0, 0, 0, 0, 0, 0, 10000000

カウンタ値が正常に返された場合

[使用上の注意]

カウント計測中 (Busy 状態) の時にこのコマンドを実行すると、NCT08-XX 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。

GetValue **引数あり**

このコマンドを送信することで、指定されたカウンタ／タイマの値を返します。

[引数]

0 から 7 までの数字 : CHx (x=0 to 7) を指定します。

8 の数字 : タイマを指定します。

[リプライ・メッセージのコマンド]

@GetValue□ (引数) □<データ>

データが複数ある場合は、(カンマ) 区切りで返します

[例]

nct08□GetValue□1

CH1 のカウンタの値を取得します

(返信されてくる文字列)

nct08>test□@GetValue□1□0

カウンタ値が正常に返された場合

[使用上の注意]

カウント計測中 (Busy 状態) の時にこのコマンドを実行すると、NCT08-XX 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。

IsOverflow 引数なし

このコマンドを送信することで、全てのカウンタ／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。

(NCT08-01 機器の場合、カウンタ CH1 から CH7 は 4294967925 カウントを超えるとオーバーフローします。

タイマカウンタは 4294967925 マイクロ秒（1 時間 1 1 分 3 5 秒弱）を超えるとオーバーフローします）

(NCT08-02 機器の場合、カウンタ CH1 から CH7 は 281474976710655 カウントを超えるとオーバーフローします。

タイマカウンタは 1099511627775 マイクロ秒（1 2 日 1 7 時間 2 5 分 1 2 秒弱）を超えるとオーバーフローします）

(NCT08-01B 機器の場合、カウンタ CH1 から CH7 は 4294967925 カウントを超えるとオーバーフローします。

タイマカウンタは 1099511627775 マイクロ秒（1 2 日 1 7 時間 2 5 分 1 2 秒弱）を超えるとオーバーフローします）

[オーバーフローの値]

- 0 : オーバフローしていない状態
- 1 : オーバフローしている状態

[リプライ・メッセージのコマンド]

@IsOverflow□<データ>

データが複数ある場合は、(カンマ) 区切りで返します

[例]

nct08□IsOverflow

カウンタ値を取得します

(返信されてくる文字列)

nct08>test□@IsOverflow□0, 0, 0, 0, 0, 0, 0, 1

カウンタ値が正常に返された場合

[使用上の注意]

一度オーバーフロー状態（値が 0 から 1）になると、オーバーフローしたカウンタが 0 リセットされるまでオーバーフローの状態（値が 1 のまま）は変わりません。

IsOverflow 引数あり

このコマンドを送信することで、指定されたカウンタ／タイマがオーバーフローしているか否かを返します。

(NCT08-01 機器の場合、カウンタ CH1 から CH7 は 4294967925 カウントを超えるとオーバーフローします。

タイマカウンタは 4294967925 マイクロ秒（1 時間 1 1 分 3 5 秒弱）を超えるとオーバーフロー

タイマカウンタは 1099511627775 マイクロ秒（12日17時間25分12秒弱）を超えるとオーバーフローします）

タイマカウンタは 1099511627775 マイクロ秒（12日17時間25分12秒弱）を超えるとオーバーフローします）

0 から 7 までの数字： CHx (x=0 to 7) を指定します。
8 の数字： タイマを指定します。

@IsOverflow (引数) 0	カウンタ／タイマがオーバーフローしていない状態
@IsOverflow (引数) 1	カウンタ／タイマがオーバーフローしている状態

nct08□IsOverflow□1 (返信されてくる文字列)	CH1 のカウンタの値を取得します
nct08>test□@IsOverflow□1□0	オーバフローか否か値が正常に返された 場合

一度オーバーフロー状態（値が 0 から 1）になると、オーバーフローしたカウンタが 0 リセットされるまでオーバーフローの状態（値が 1 のまま）は変わりません。

このコマンドを送信するとカウンタの計測動作を開始します。

このコマンドはカウンタが **Busy** 状態のときはエラーとなり実行されません。

@CountStartOk:	正常に動作を終了した場合
@CountStartEr: Busy.	カウンタが Busy 状態のため実行されなかった場合

nct08□CountStart	カウンタ計測の動作を開始します
------------------	-----------------

(返信されてくる文字列)

nct08>test□@CountStart□Ok:

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カウンタの計測動作を停止します。

[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)

nct08□Stop

カウンタの計測動作を停止します

(返信されてくる文字列)

nct08>test□@Stop□Ok:

正常に動作を終了した場合

IsBusy

このコマンドを送信することでカウンタ計測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]

@IsBusy□0

カウンタ計測が停止状態

@IsBusy□1

カウンタ計測が駆動中の状態

[例]

(送信側)

nct08□IsBusy

カウンタ計測の駆動状況を確認します

(返信されてくる文字列)

nct08>test□@IsBusy□0

カウンタ計測が停止状態の場合

nct08>test□@IsBusy□1

カウンタ計測が駆動中の状態の場合

flushdata

このコマンドを送信するとコントローラおよびカウン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□nct08

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□nct08.counter01

カウンタ名 'counter01' のイベントメッセージの配信
を依頼します

nct08□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)

nct08□@flushdata□Ok:

コマンドが正常送信されました

nct08>test□_ChangedIsBusy□0

コントローラの Busy 状態がイベントメッ
セージの値として返ってきます

nct08.counter01>test□_ChangedIsOverflow□0

カウンタ名 counter01 がオーバーフローし
ているか否かイベントメッセージの値と
して返ってきます

nct08.counter01>test□_ChangedValue□100

カウンタ名 counter01 の現在値がイベント
メッセージの値として個別に返ってきま
す

flushdatatome

このコマンドを送信するとコントローラおよびカウン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 の送信元に直接返します。

[例]

(送信側)

nct08□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)

nct08□@flushdatatome□Ok:

コマンドが正常送信されました

nct08>test□_ChangedIsBusy□0

コントローラの Busy 状態がイベントメッ
セージの値として返ってきます

nct08.counter01>test□_ChangedIsOverflow□0

カウンタ名 counter01 がオーバーフローし
ているか否かイベントメッセージの値と
して返ってきます

nct08.counter02>test□_ChangedIsOverflow□0

以下残り全てのカウンタ／タイマのオー
バーフローか否かが返ってきます。

...

nct08.counter07>test□_ChangedIsOverflow□0

nct08.timer>test□_ChangedIsOverflow□0

nct08.counter01>test□_ChangedValue□1000

カウンタ名 counter01 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

以下残り全てのカウンタ／タイマの現在
値が返ってきます。

```
nct08.counter02>test□_ChangedValue□0
```

```
...
```

```
nct08.counter07>test□_ChangedValue□0
```

```
nct08.timer>test□_ChangedValue□1000000
```


デバッグ用コマンド**devact NCT08-XX コマンド**

引数の NCT08-XX コマンドを直接 NCT08-XX に送信し応答を待つ値を返します。

NCT08-XX のコマンドのうち応答を返すコマンド（?を含むコマンド）を引数として指定して下さい。

デバッグ時に使用してください。有効でないコマンドを指定した場合の動作は保証されませんのでご了承ください。

[例]

(送信側)

nct08□devact□MOD?

MOD?コマンドを実行します。

(返信されてくる文字列)

nct08□@devact□MOD?□R_SN_T_O

コマンドが正常送信されました

devsend NCT08-XX コマンド

引数の NCT08-XX コマンドを直接 NCT08-XX に送信し応答を待つ値を返します。

NCT08-XX のコマンドのうち応答を返さないコマンド（?を含まないコマンド）を引数として指定して下さい。

デバッグ時に使用してください。有効でないコマンドを指定した場合の動作は保証されませんのでご了承ください。

[例]

(送信側)

nct08□devsend□STRT

STRT コマンドを実行します。

(返信されてくる文字列)

nct08□@devsend□STRT□Ok:

コマンドが正常送信されました

カウンタ／タイマコマンド

[メッセージ配信先名]

nct08.counter *name*

countername のカウンタ／タイマに対して命令（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)

nct08.counter01□hello

(返信されてくる文字列)

nct08.counter01>test□@hello□nice□to□meet□you.

GetCounterNumber

このコマンドを送信することで、指定されたカウンタ／タイマに該当するカウンタ／タイマ番号を返します。

[リプライ・メッセージ]

@GetCounterNumber□ (カウンタ番号)

カウンタの場合はカウンタ番号 0 から 7 までの数字：CHx (x=0 to 7) を返します。

@GetCounterNumber□8

タイマの場合は番号 8 の数字を返します。

[例]

(送信側)

nct08.counter01□GetCounterNumber□

名前 counter01 に該当する番号を問い合わせます。

(返信されてくる文字列)

nct08.counter01>test□@ GetCounterNumber□1

カウンタ番号が正常に返された場合

CounterReset

このコマンドを送信することで、指定されたカウンタ／タイマの値を 0 リセットします。

[リプライ・メッセージ]

@CounterReset□Ok:

正常に動作を終了した場合

@CounterReset□Er:□Busy.

カウンタが Busy 状態のため実行されなかった場合

[例]

(送信側)

nct08.counter01□CounterReset

カウンタ名 counter01 のカウンタの値を 0 リセット
します

(返信されてくる文字列)

nct08.counter01>test□@CounterReset□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、指定されたカウンタ／タイマの値を返します。

[リプライ・メッセージのコマンド]

@GetValue□<データ>

正常にデータが返された場合

[例]

nct08.counter01□GetValue

カウンタ名 counter01 のカウンタ値を取得
します

(返信されてくる文字列)

nct08.counter01>test□@GetValue□1000

カウンタ値が正常に返された場合

[使用上の注意]

カウント計測中 (Busy 状態) の時にこのコマンドを実行すると、NCT08-XX 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。

IsOverflow

このコマンドを送信することで、指定されたカウンタ／タイマがオーバーフローしているか否かを返します。

(NCT08-01 機器の場合、カウンタ CH1 から CH7 は 4294967925 カウントを超えるとオーバーフローします。

タイマカウンタは 4294967925 マイクロ秒 (1 時間 1 1 分 3 5 秒弱) を超えるとオーバーフローします) 4.9710276909722222222222222222222222

(NCT08-02 機器の場合、カウンタ CH1 から CH7 は 281474976710655 カウントを超えるとオーバーフローします。

タイマカウンタは 1099511627775 マイクロ秒 (1 2 日 1 7 時間 2 5 分 1 2 秒弱) を超えるとオーバーフローします)

(NCT08-01B 機器の場合、カウンタ CH1 から CH7 は 4294967925 カウントを超えるとオーバーフローします。

タイマカウンタは 1099511627775 マイクロ秒（1 2 日 1 7 時間 2 5 分 1 2 秒弱）を超えるとオーバーフローします）

[リプライ・メッセージのコマンド]

@IsOverflow□（引数）□0

カウンタ／タイマがオーバーフローしていない状態

@IsOverflow□（引数）□1

カウンタ／タイマがオーバーフローしている状態

[例]

nct08.counter01□IsOverflow

カウンタ名 counter01 のカウンタがオーバーフローしているか否かを取得します

（返信されてくる文字列）

nct08.counter01>test□@IsOverflow□0

正常に値が返された場合

[使用上の注意]

一度オーバーフロー状態（値が 0 から 1）になると、オーバーフローしたカウンタが 0 リセットされるまでオーバーフローの状態（値が 1 のまま）は変わりません。

コントローライベント

[イベント]

_ChangedIsBusy

カウンタ／タイマが **Busy**（動作中か否か）を返します。
状態が変化した場合もしくはコントローラコマンド「flushdata」もしくは「flushdatatome」実行時に、
Stars のイベントメッセージとして返されます。

[イベントメッセージ]

_ChangedIsBusy □1	カウンタ／タイマが Busy 状態（計測中）の場合
_ChangedIsBusy □0	カウンタ／タイマが Busy 状態でない（停止中）の場合

[関連コマンド]

flushdata、flushdatatome

カウンタ／タイマイイベント

[イベント]

_ChangedValue

カウンタ／タイマの値を返します。

コンローラコマンド「flushdata」もしくは「flushdatatome」実行時に、Stars のイベントメッセージとして返されます。

プログラム起動時に-flushdata オプションを使用すると、カウント計測中（Busy 状態）インターバルでカウンタ／タイマ値を読み、値が変化した場合に Stars のイベントメッセージとして返します。

プログラム起動時に-flushdata オプションを使用しないと、カウント計測中（Busy 状態）はカウンタ／タイマ値を読みません。カウント計測停止（Busy 状態が解除）したタイミングで値を読みにいき、値が変化していた場合に Stars のイベントメッセージとして返します。

[イベントメッセージ]

_ChangedValue□<データ>

[関連コマンド]

flushdata、flushdatatome

_ChangedIsOverflow

カウンタ／タイマの値がオーバーフローしたか否かを返します。

(NCT08-01 機器の場合、カウンタ CH1 から CH7 は 4294967925 カウントを超えるとオーバーフローします。

タイマカウンタは 4294967925 マイクロ秒（1 時間 1 1 分 3 5 秒弱）を超えるとオーバーフローします） 4.9710276909722222222222222222222222

(NCT08-02 機器の場合、カウンタ CH1 から CH7 は 281474976710655 カウントを超えるとオーバーフローします。

タイマカウンタは 1099511627775 マイクロ秒（1 2 日 1 7 時間 2 5 分 1 2 秒弱）を超えるとオーバーフローします）

(NCT08-01B 機器の場合、カウンタ CH1 から CH7 は 4294967925 カウントを超えるとオーバーフローします。

タイマカウンタは 1099511627775 マイクロ秒（1 2 日 1 7 時間 2 5 分 1 2 秒弱）を超えるとオーバーフローします）

値が変化した場合もしくはコンローラコマンド「flushdata」もしくは「flushdatatome」実行時に、Stars のイベントメッセージとして返されます。

[イベントメッセージ]

`_ChangedIsOverflow`□1

カウンタがオーバーフロー状態の場合

`_ChangedIsOverflow`□0

カウンタがオーバーフロー状態でないの場合

[関連コマンド]

`flushdata`、`flushdatatome`