

Stars I/O client for CryoCon 32B

ASAKURA, Hiroyuki

はじめに

CryoCon 32B の Stars I/O ドライバです。現時点(2013年11月29日)では, CryoCon で実行できるコマンドの内の一部のみが実装されています。

機器の接続

CryoCon 32B は RS-232C 経由で Nport に接続されています。Stars サーバとは Nport 経由で接続されています。

最小限の使い方

```
python cryocon.py -c BL5S1.cryocon.conf &
```

起動オプション

-c filename

ini 形式の設定ファイル **filename** を読み込む。

デフォルトの設定ファイル

Stars の接続に必要なパラメータは以下の様な設定ファイルに記述されています。BL5S1.cryocon.conf の例。

```
[starsnode]
NPORT=192.168.51.203
nport_port=4003

[starsnode]
NODENAME=cryocon
NODESERVER=localhost
NODEPORT=6057
```

また, NODENAME に対応した key ファイルが必要です。上記の設定ファイルの場合は, cryocon.key ファイルを同じディレクトリと takaserv-lib に保存して下さい。

コマンド一覧

一般コマンド

hello

Stars の通信が行われているかチェックするコマンドです。

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) cryocon hello

(返信されてくる文字列) cryocon>test @hello Nice to meet you!

help

引数を指定しない場合は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。

現在はうまく動作しません。

ユーティリティ系コマンド

Set 系のコマンドは、現在のところ正しく設定されたかどうかの validation は行わず、OK を返します。

GetID

CryoCon デバイスの ID を返します。

[例]

(送信側) cryocon GetID +

(返信されてくる文字列) cryocon>test @GetID Cryocon Model 32, Rev 6.20H

Reset

CryoCon デバイスを再起動します。Reset には15-18秒程度かかります。

現在は、同期的に動作してしまうので、**Reset** が完了してから返事が返ってきます。

[例]

(送信側) cryocon Reset

(返信されてくる文字列) cryocon>test @Reset OK

Clear

CryoCon の状態を初期化します。

[例]

(送信側) cryocon Clear

(返信されてくる文字列) cryocon>test @Clear OK

IsControlled

温度制御が行われているか確認します。制御状態に応じて、ON, OFF を返します。

[例]
(送信側) cryocon IsControlled
(返信されてくる文字列) cryocon>test @IsControlled ON

StartControl

温度制御を返します。

[例]
(送信側) cryocon StartControl
(返信されてくる文字列) cryocon>test @StartControl OK

Stop

温度制御を停止します。

[例]
(送信側) cryocon Stop
(返信されてくる文字列) cryocon>test @Stop OK

IsLocked

デバイス前面のボタンがロックされているか確認します。ロック状態に応じて、ON, OFF を返します。

[例]
(送信側) cryocon IsLocked
(返信されてくる文字列) cryocon>test @IsLocked ON

IsinLoop

(クローズド) ループ制御が行われているか確認します。ループ制御状態に応じて、ON, OFF を返します。

[例]
(送信側) cryocon IsinLoop
(返信されてくる文字列) cryocon>test @IsinLoop ON

SetTimeConstant time

測定温度の更新間隔を time (sec) に設定します。制御装置前面の値およびリモートで問い合わせた値の両方に有効です。

time に設定できる値は、0.5, 1, 2, 4, 8, 16, 32, 64 です。

[例]

(送信側) cryocon SetTimeConstant 8

(返信されてくる文字列) cryocon>test @SetTimeConstant 8 OK

GetTimeConstant

測定温度の更新間隔を取得します。

[例]

(送信側) cryocon GetTimeConstant

(返信されてくる文字列) cryocon>test @GetTimeConstant 8

GetSystemTemperature

デバイスの温度を取得します。

クライオ本体の温度ではないことに注意してください！

[例]

(送信側) cryocon GetSystemTemperature

(返信されてくる文字列) cryocon>test @GetSystemTemperature +34 C

GetHeatSinkTemperature

ヒートシンクの温度を取得します。 過熱を防ぐ意味合いがあります。 クライオ本体の温度ではないことに注意してください！

[例]

(送信側) cryocon GetHeatSinkTemperature

(返信されてくる文字列) cryocon>test @GetHeatSinkTemperature +30 C

DisplayHome

制御装置前面の表示をホーム画面に戻します。

[例]

(送信側) cryocon DisplayHome

(返信されてくる文字列) cryocon>test @DisplayHome OK

GetTemperature channel

channel の温度を取得します。

[例]

(送信側) cryocon GetTemperature A

(返信されてくる文字列) cryocon>test @GetTemperature A 224.524

ResetStatistics channel

channel の統計情報（連続動作時間等）を初期化します。

[例]

(送信側) cryocon ResetStatistics A

(返信されてくる文字列) cryocon>test @ResetStatistics A OK

GetElapsedTime channel

channel の経過時間を取得します。

[例]

(送信側) cryocon ResetStatistics A

(返信されてくる文字列) cryocon>test @ResetStatistics A OK

SetSource channel loopnumber

channel に対する Loop を *loopnumber* に設定します。 *loopnumber* は 1 か 2 をとります。

[例]

(送信側) cryocon SetSource CHA 1

(返信されてくる文字列) cryocon>test @SetSource CHA 1 OK

SetSetpoint temperature channel

channel に対する設定温度を *temperature* に設定します。

[例]

(送信側) cryocon SetSetpoint 300 1

(返信されてくる文字列) cryocon>test @SetSetpoint 300 1 OK

SetControlType type loopnumber

測定開始基準を *type* に設定します。

loopnumber は 1 か 2 をとります。 デフォルト値は1です。

type には Off, PID, Man, Table, RampP のいずれかを選択することができます。

それぞれの挙動はマニュアルを参照してください。

通常は Table を選択します。 **Table** はあらかじめ設定された **PID** パラメータのリストです

[例]

(送信側) cryocon SetControlType Table 1

(返信されてくる文字列) cryocon>test @SetControlType Table 1 OK

GetControlType loopnumber

測定開始基準を *source* に設定します。

loopnumber は 1 か 2 をとります。 デフォルト値は 1 です。

type は Off, PID, Man, Table, RampP のいずれかです。

[例]

(送信側) cryocon GetControlType 1

(返信されてくる文字列) cryocon>test @GetControlType 1 TABLE

SetOutputRange range loopnumber

測定開始基準を *range* に設定します。

loopnumber は 1 か 2 をとります。デフォルト値は1です。

range には HI, MID, LOW のいずれかを選択することができます。

それぞれの挙動はマニュアルを参照してください。

正しく設定されたかの確認はせずに単に OK を返します。

[例]

(送信側) cryocon SetOutputRange HI 1

(返信されてくる文字列) cryocon>test @SetOutputRange HI 1 OK

GetOutputRange loopnumber

loopnumber は 1 か 2 をとります。デフォルト値は 1 です。

range として HI, MID, LOW のいずれかを返します。

[例]

(送信側) cryocon GetOutputRange 1

(返信されてくる文字列) cryocon>test @GetOutputRange 1 HI

IsLoopRamp loopnumber

loopnumber は 1 か 2 をとります。デフォルト値は 1 です。

loopnumber が temperature ramp 中であるかどうかを ON, OFF で返します。

[例]

(送信側) cryocon IsLoopRamp 1

(返信されてくる文字列) cryocon>test @IsLoopRamp 1 ON

SetLoopRamp rate loopnumber

loopnumber は 1 か 2 をとります。デフォルト値は1です。

rate に毎分あたりの温度変化量を設定することができます。

正しく設定されたかの確認はせずに単に OK を返します。

[例]

(送信側) cryocon SetLoopRamp 0.01 1

(返信されてくる文字列) cryocon>test @SetLoopRamp 0.01 1 OK

GetLoopRamp loopnumber

loopnumber は 1 か 2 をとります。デフォルト値は 1 です。

毎分あたりの温度変化量設定値を返します。

[例]

(送信側) cryocon GetLoopRamp 1

(返信されてくる文字列) cryocon>test @GetLoopRamp 1 0.010000

SetPGain gain loopnumber

loopnumber は 1 か 2 をとります。デフォルト値は1です。

gain に Proportional gain を設定します。

正しく設定されたかの確認はせずに単に OK を返します。

[例]

(送信側) cryocon SetPGain 20 1

(返信されてくる文字列) cryocon>test @SetPGain 20 1 OK

GetPGain loopnumber

loopnumber は 1 か 2 をとります。デフォルト値は 1 です。

Proportional Gain を返します。

[例]

(送信側) cryocon GetPGain 1

(返信されてくる文字列) cryocon>test @GetPGain 1 20.000000

SetIGain gain loopnumber

loopnumber は 1 か 2 をとります。デフォルト値は1です。

gain に Integral gain を設定します。

正しく設定されたかの確認はせずに単に OK を返します。

[例]

(送信側) cryocon SetIGain 12 1

(返信されてくる文字列) cryocon>test @SetIGain 12 1 OK

GetIGain loopnumber

loopnumber は 1 か 2 をとります。デフォルト値は 1 です。

Integral gain を返します。

[例]

(送信側) cryocon GetIGain 1

(返信されてくる文字列) cryocon>test @GetIGain 1 12.000000

SetDGain gain loopnumber

loopnumber は 1 か 2 をとります。デフォルト値は1です。

gain に Differentiator gain を設定します。

正しく設定されたかの確認はせずに単に OK を返します。

[例]

(送信側) cryocon SetDGain 12 1

(返信されてくる文字列) cryocon>test @SetDGain 12 1 OK

GetDGain loopnumber

loopnumber は 1 か 2 をとります。デフォルト値は 1 です。

Differentiator gain を返します。

[例]

(送信側) cryocon GetDGain 1

(返信されてくる文字列) cryocon>test @GetDGain 1 2.8000000

GetCurrentPower loopnumber

loopnumber は 1 か 2 をとります。デフォルト値は 1 です。

現在の電流値を返します。

[例]

(送信側) cryocon GetCurrentPower 1

(返信されてくる文字列) cryocon>test @GetCurrentPower 1 0.0000000

IsBusy

Busy 状態を返します。

但し、非同期的に動作しているので、Busy だった場合は、Busy 状態が解けてから 0 を返します。

[例]

(送信側) cryocon IsBusy

(返信されてくる文字列) cryocon>test @IsBusy 0

例

以下では、Loop 1 に温度計が、Input A にヒーターが接続されている通常の状態を仮定しています。

cryocon は初期状態で、PID パラメータが目標温度ごとに表になっている TABLE モードで起動するので、以下の様に単に目標温度を設定して、制御を開始するだけで、一定温度に保たれる。

最も簡単な使い方

```
# cryocon をリセットする（朝に1回だけで十分）
cryocon Reset

# cryocon の状態をクリアする（必須ではない）
cryocon Clear

# 目標温度(K)を設定する
cryocon SetSetpoint 200

# He 冷却装置を動かし始める

# 温度制御をスタートする
cryocon StartControl

# ..

# 温度制御をストップする
cryocon Stop
```