



KEITHLEY Model2701 デジタルマルチメータ 制御プログラム

KEITHLEY Model2701DMM Data Acquisition System I/O Client

for Python

m2701drv.py 取扱説明書

rev1.0/2023.03.23 ; 初版

本書について

本書は、STARSのネットワーク上で利用されるケースレー製デジタルマルチメータ Model2701DMM用制御プログラム「m2701drv.py」の取り扱いについて記述したものです。

m2701drv.pyは、STARSサーバの下でコントローラを制御するひとつのSTARS I/O Client(以下、m2701と呼ぶ)を構成するものです。

■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	4
1.1 m2701drv.py について	4
1.2 Model2701DMM と m2701drv の通信について	4
1.3 m2701drv の Stars のノード名について	5
1.4 メッセージフォーマット	5
1.5 STARS チャンネルについて	5
2 プログラム構成	6
2.1 m2701drv プログラム一覧	6
3 プログラムの初期設定	7
3.1 m2701drv の設定ファイル（コンフィグファイル）について	7
4 プログラムの起動	8
5 イベントメッセージ	10
5.1 イベントメッセージについて	10

1 はじめに

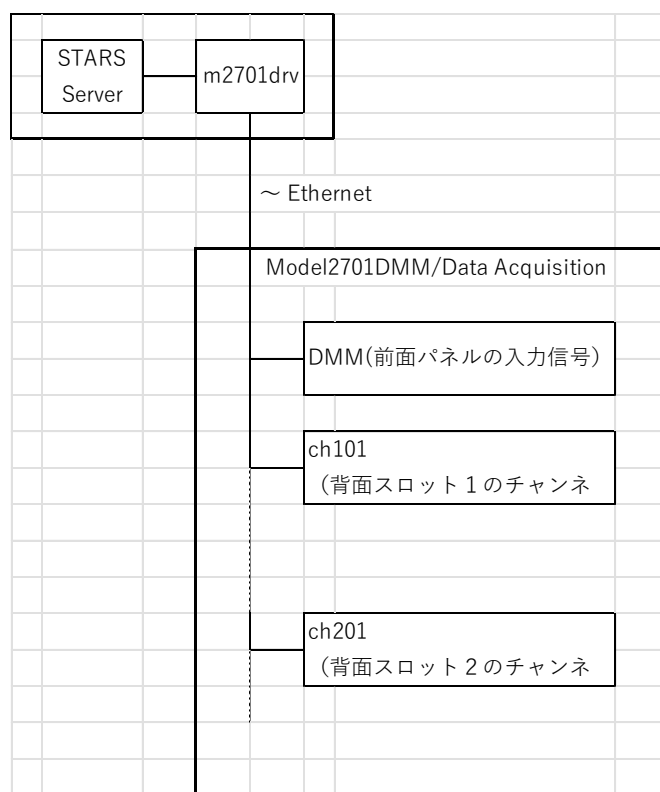
1.1 m2701drv.py について

m2701drv.py はPython言語で作成されたm2701drv用プログラムです。「Model 2701 Ethernet-Based DMM/Data Acquisition User's Manual」(※)に基づいて作成されています。

※ m2701drv.py のプログラムと同一のフォルダに「2701-900-01G_Feb_2016.pdf」として格納してあります。

1.2 Model2701DMM と m2701drv の通信について

Model2701DMMは、外部I/FとしてDIGITAL-I/O、RS232CおよびLAN(Ethernet TCP/IP)が搭載されています。STARSはTCP/IPベースのネットワークであり、接続にはLANを用います。



※ 背面スイッチングモジュールはオプションなので指定できない場合もあります。

1.3 m2701drv の Stars のノード名について

m2701drvのStarsのノード名は、m2701のI/O Client起動時の引数として指定します。
ノード名は下記のものを用いてください。

・ Model2701DMM : m2701drv

1.4 メッセージフォーマット

STARS 経由で m2701drv を使用するには、下記フォーマットのメッセージを通信して行います。

m2701drv□**STARS 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) m2701drv Run (CH101),1
 m2701drv 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。

1.5 STARS チャンネルについて

STARS チャンネルは、model2701 に実装された測定用の物理チャンネルに対応付けて用いられる操作便宜上の仮想チャンネルです。

STARS チャンネルは次のように記載されます。

フロントパネル入力 ; DMM

リヤスロット実装カードからの入力 ; ch01、ch02、… （番号は任意）

物理チャンネルは次のように記載されます。

フロントパネル入力 ; CHFRONT / DMM

リヤスロット実装カードからの入力 ; CHyxx

y … 背面スロットの番号、 xx…実装されたカードの入力チャンネル番号

STARS チャンネルと物理チャンネルとの対応付けは、起動時のオプション「--rearchannelnamelist」で行われます。また、後述の SetChannelNameList コマンドで再設定することが可能です。

後述のコマンドにおけるチャンネルの指定は、STARS コマンド、物理コマンドの双方を用いることが可能です。記述の仕方については、それぞれのコマンドの説明を参照してください。

尚、DMM はスキャン動作時のチャンネル指定としても用いられる場合があります。

2 プログラム構成

2.1 m2701drv プログラム一覧

m2701drvの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「m2701drv.channel.py」です。

ファイル名	内容	ファイル 種別
m2701drv.py	Model2701DMM I/O Client プログラム	python
m2701drv.key	Stars ノード名「m2701drv」用の Stars 認証 Key	text
config.cfg	m2701drv Stars I/O Client プログラム設定ファイル	python
nportserv.py	m2701drv 用ライブラリ	python
singlestars.py	m2701drv 用ライブラリ	python
libstreg.py	m2701drv 用ライブラリ	python
stars.py	STARS サーバ本体プログラム	python
pystarsutil/pystarsutilargparser.py	m2701drv 用ライブラリ	python
pystarsutil/pystarsutilconfig.py	m2701drv 用ライブラリ	python
configfiles/xxxx.txt	Model2701DMM 設定ファイル	text

※ keyファイルはカーネル（サーバー）のライブラリに登録（保存）されている必要があります。

3 プログラムの初期設定

3.1 m2701drv の設定ファイル（コンフィグファイル）について

m2701drvのプログラムは、起動時に設定ファイル「config.cfg」を参照することで制御に必要な環境を整えます。config.cfgに記載の必要な項目は次の通りです。

- ・ STARSサーバのIPアドレスまたはホスト名
- ・ デバイスのIPアドレス
- ・ デバイスのポート番号

※ STARSサーバのポート番号は6057の固定となっています。

記載例を示します。

```
m2701drv.channel > ⚙ config.cfg
1  [DEFAULT]
2
3  [m2701drv]
4  StarsServerHost=localhost
5  DeviceHost=172.16.0.53
6  DevicePort=1394
7
```

尚、STARSサーバのIPアドレス、STARSサーバのポート番号、デバイスのIPアドレスおよびポート番号は、m2701drv.py起動時のオプションにより設定することもできます。また、設定ファイルを起動時のオプションで指定することができます。

4 プログラムの起動

4.1 m2701drv.py の起動方法について

コマンドプロンプトを起動し、m2701drv.pyが格納されているホルダをカレントディレクトリにして、次のように入力します。m2701drv.pyが起動することでm2701drvが稼働します。

```
> python m2701drv.py --nodename m2701drv ...
... ; オプション
```

起動時のオプションは次の通りです。

--nodename xxx	； ノード名指定
-d、--debug	； デバッグモードを有効にする
--config xxx	； コンフィグファイルの指定 (※2)
--serverhost	； STARSサーバ側IPアドレス、ホスト名の指定
--serverport	； STARSサーバ側ポート番号の指定
--devicehost	； コントローラ側IPアドレスの指定
--deviceport	； コントローラ側ポート番号の指定
--logenable	； ログ取得を有効にする
--logdir	； ログファイルを格納するディレクトリの指定
--rearchannelnamelist	； Model7700等スイッチングモジュール装着時のチャンネル定義 (※3)
--defaultinputselection	； 測定対象となる信号の入力先の指定 (※4)
	0：背面実装のスイッチングモジュール
	1：前面パネル

(入力例)

```
C:¥>cd m2701drv.py
C:¥m2701drv.py> python m2701drv.py --nodename m2701drv --rearchannelnamelist
"ch01:101,ch02:102,ch03:103,ch04:104,ch05:121,ch06:122,ch07:123,ch08:124"
--defaultinputselection 0
```

※1. Model2701DMMの制御は、サーバーにログインしたユーザがサーバーを介してSTARSコマンドをm2701drvに送信しm2701drvがこれを実行することで行われます。

- ※2. コンフィグファイルは、デフォルトの名称が“config.cfg”です。オプションで指定する場合は異なるファイル名を用い、m2701drv.pyが格納されているホルダに格納してください。
- ※3. Model 7700の場合、電圧、抵抗、周波数、周期、および温度測定用として使用できる20 系統の入力チャンネル（1～20）と、電流測定用に2チャンネル(21、22)が利用可能です。そしてModel 2701はModel 7700を装着できるスロットが2個あります。入力先は番号で、スロット番号とチャンネル番号を組み合わせ、スロット1のチャンネル1なら101,スロット2のチャンネル10なら210となります。
- 入力先の番号にチャンネル名を関連付けたものがチャンネル入力識別子で、使用する入力チャンネル識別子を,(カンマ)区切りでつなげた文字列を当オプションパラメータとして指定します。
- 《入力チャンネル識別子》 チャンネル名+‘:’(コロン)+スロットチャンネル番号
オプションパラメータに、“ch1:101,ch2:102,ch3:103”（”を取り除いた文字列）と指定した場合は、チャンネル名ch1に対し入力先番号101、ch2には102、ch3には103が対応します
- ※4. スイッチングモジュールはオプション的位置付けであるため、デフォルトは「1」の設定となっています。
- ※5. STARS コマンドは STARS コマンドリファレンスマニュアル「STARS to Model2701 用コマンド集」を参照してください。

5 イベントメッセージ

5.1 イベントメッセージについて

イベントメッセージは、コントローラの状態をサーバーに通知するためのものです。コントローラの状態が変わったときや、所定のコマンド（flushdata等）が実行されたときにサーバーに送信されます。

ユーザーがサーバーに対してイベントメッセージ発生の配信を登録（flgon）することで、サーバーから発生したイベントメッセージを通知してもらうことができます。登録は配信先（ノード名）を用いて行います。

（登録例）

・コントローラを登録する場合

```
> System flgon m2701drv
```

登録の解除は次のように行います。

```
> System flgoff m2701drv
```

5.2 イベントメッセージの種類

イベントメッセージは、登録する対象（コントローラ、モータ、エンコーダ）によって異なり、次のものがあります。

イベントメッセージ	対象
_Connected	コントローラ
_Disconnected	コントローラ
_ChangesIsBusy 0 1	コントローラ

それぞれの意味は次の通りです。

_Connected	；コントローラとの接続が確立
_Disconnected	；コントローラとの接続が解除
_ChangedIsBusy 0 1	；0:停止中 1:稼動中