

APU101/APN101
サンプルプログラム
取扱説明書
Python 版

第 1.0 版 2020 年 04 月

株式会社 テクノエーピー

〒312-0012 茨城県ひたちなか市馬渡 2976-15

TEL : 029-350-8011

FAX : 029-352-9013

URL : <http://www.techno-ap.com>

e-mail : order@techno-ap.com

目次

1.	安全上の注意・免責事項.....	3
2.	概要.....	4
2. 1.	概要.....	4
2. 2.	改訂履歴.....	4
2. 3.	環境.....	4
2. 4.	その他.....	4
3.	取り扱い.....	5
3. 1.	ネットワークの設定確認.....	5
3. 2.	サンプルプログラムのダウンロード.....	5
3. 3.	プログラムソースの展開.....	6
3. 4.	ヒストグラム計測プログラム (apu101.py) の実行.....	7
3. 5.	高圧電源制御プログラム (apu101_hv.py) の実行.....	9
3. 6.	高圧電源パラメータ設定プログラム (apu101_hv2config.py) の実行.....	10
3. 7.	高圧電源ステータスマニタプログラム (apu101_hv2satus.py) の実行.....	11
3. 8.	高圧電源極性変更プログラム (apu101_hv2pol.py) の実行.....	12

1. 安全上の注意・免責事項

日頃、株式会社テクノエーピー（以下「弊社」）製品 APU101 または APN101（以下「本装置」）のご愛顧を頂き、誠にありがとうございます。本装置をご使用する前に、この「安全上の注意・免責事項」をお読みの上、内容を必ずお守りいただき、正しくご使用ください。

弊社製品のご使用によって発生した事故であっても、装置・検出器・接続機器・アプリケーションの異常、故障に対する損害、その他二次的な損害を含む全ての損害について、弊社は一切責任を負いません。

禁止事項

- 人命、事故に関わる特別な品質、信頼性が要求される用途にはご使用できません。
- 高温、高湿度、振動の多い場所などでのご使用はできません。
- 強い衝撃や振動を与えないでください。
- 分解、改造はしないでください。
- 水や結露などで濡らさないでください。濡れた手でのご操作もおやめください。
- 発熱、変形、変色、異臭などがあった場合は直ちにご使用を止めて弊社までご連絡ください。

注意事項

- 本装置の使用温度範囲は室温とし、結露無いようにご使用ください。
- 発煙や異常な発熱があった場合はすぐに電源を切ってください。
- 本装置は高精度な精密電子機器です。静電気にはご注意ください。
- 本装置は、ほこりの多い場所や高温・多湿の場所には保管しないでください。
- 携帯電話やトランシーバー等、強い電波を出す機器を近づけないでください。
- 電氣的ノイズの多い環境では誤作動のおそれがあります。
- 本装置の仕様や本書及び関連書類の内容は、予告無しに変更する場合があります。

2. 概要

2. 1. 概要

本装置は TCP/IP 通信を通して、ユーザー作成の Python プログラムにて計測制御を行うことが可能です。

本書は、本装置の Linux 環境向けサンプルプログラムの取り扱いについて説明するものです。

※コマンドの詳細については「[デジタルスペクトルメータ, APU101/APN101 コマンドマニュアル](#)」を参照ください。

※本書の記載内容は予告なく変更することがあります。

2. 2. 改訂履歴

2020年04月22日 第1.0版 初版

2. 3. 環境

以下の環境にて動作確認を行いました。

- (1) OS: CentOS 7.6 (64bit 版)
 Windows 10 (64bit 版)
- (2) Python ver: Python3.6

※本書は各 OS にて Python の環境が構築されている事を前提としています。

Python の環境構築手順については説明を割愛させて戴きます。Python の環境構築については、python のホームページを参照してください。

2. 4. その他

このサンプルプログラムは Windows, Linux (CentOS) で動作確認行いましたが、本書の説明は下記の環境である事を前提とさせて戴きます。

表 1 本書の取扱説明に於ける前提環境

#	項目	OS・バージョン等
1	OS	CentOS 7.6 64bit 版
2	Python	version 3.6.8

3. 取り扱い

3. 1. ネットワークの設定確認

本装置の出荷時 IP アドレスは、192.168.10.128 に設定されています。

プログラムから本装置にアクセスするためには、PC 側は 192.168.10.127 番以下の固定 IP アドレスを設定して下さい。

設定後、ターミナルウィンドウより「ping 192.168.10.128」を実行し、正常に通信ができることを確認しておいて下さい。

3. 2. サンプルプログラムのダウンロード

弊社サンプルプログラムのページより apu101_python.zip をダウンロードします。解凍後以下のディレクトリ構成でファイルが生成されます。

(解凍先ディレクトリ： /tmp/sample/)

```
└─apu101_python
  │ apu101.py                #ヒストグラム計測プログラム
  │ apu101_hv.py            #高圧電源制御プログラム
  │ apu101_hv2config.py     #高圧電源パラメータ設定プログラム
  │ apu101_hv2pol.py        #高圧電源極性変更プログラム
  │ apu101_hv2status.py     #高圧電源ステータスマニタプログラム
  │
  └─technoap
    └─device                 #デバイス制御ライブラリ群
      │ └─apu101             # →APU(N)101 用制御ライブラリ
      │ └─common             # →デバイス制御共通ライブラリ
      └─num                   #数値編集ライブラリ
    └─sitcp                   #SiTCP ライブラリ
    └─time                     #時間管理ライブラリ
```

説明の都合上、以降の説明では解凍先ディレクトリは“/tmp/sample”とします。また、サンプルプログラム (zip 形式) は“/tmp/sample”に配置した事を前提とします。実際のディレクトリ構成が異なる場合は、適宜そのディレクトリに読替えて下さい。

3. 3. プログラムソースの展開

サンプルプログラムの実行はCUI形式にて行いますので、ターミナルウィンドウを開きます。
まず、ターミナルウィンドウを開き、サンプルプログラムを配置したディレクトリに移動します。

→実行コマンド： `cd /tmp/sample`

その後、圧縮形式のサンプルプログラム（zip ファイル）を展開して、展開先のディレクトリに入ります。

図 1 はターミナルウィンドウに於ける「カレントディレクトリ移動～プログラムソースの展開」までのコマンド操作の例です。

```
192.168.10.62 - tap@localhost:/tmp/sample/apu101_python VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
[ tap@localhost ~ ] $ cd /tmp/sample
[ tap@localhost sample ] $ unzip apu101_python.zip
Archive:  apu101_python.zip
warning:  apu101_python.zip appears to use backslashes as path separators
inflating:  apu101_python/apu101.py
inflating:  apu101_python/apu101.pyproj
inflating:  apu101_python/apu101.sln
inflating:  apu101_python/apu101_hv.py
inflating:  apu101_python/apu101_hv2config.py
inflating:  apu101_python/apu101_hv2pol.py
inflating:  apu101_python/apu101_hv2status.py
inflating:  apu101_python/technoap/__init__.py
inflating:  apu101_python/technoap/device/__init__.py
inflating:  apu101_python/technoap/device/apu101/common.py
inflating:  apu101_python/technoap/device/apu101/config.py
inflating:  apu101_python/technoap/device/apu101/configHV.py
inflating:  apu101_python/technoap/device/apu101/constant.py
inflating:  apu101_python/technoap/device/apu101/fastdatatx.py
inflating:  apu101_python/technoap/device/apu101/register.py
inflating:  apu101_python/technoap/device/apu101/register.txt
inflating:  apu101_python/technoap/device/apu101/spectrum.py
inflating:  apu101_python/technoap/device/apu101/status.py
inflating:  apu101_python/technoap/device/apu101/statusHV.py
inflating:  apu101_python/technoap/device/apu101/__init__.py
inflating:  apu101_python/technoap/device/common/reginfo.py
inflating:  apu101_python/technoap/device/common/registerRW.py
inflating:  apu101_python/technoap/device/common/__init__.py
inflating:  apu101_python/technoap/num/bitconverter.py
inflating:  apu101_python/technoap/num/byteorder.py
inflating:  apu101_python/technoap/num/__init__.py
inflating:  apu101_python/technoap/sitcp/constant.py
inflating:  apu101_python/technoap/sitcp/fasttxbase.py
inflating:  apu101_python/technoap/sitcp/rbc.py
inflating:  apu101_python/technoap/sitcp/rbcbase.py
inflating:  apu101_python/technoap/sitcp/rbcformat.py
inflating:  apu101_python/technoap/sitcp/rbcpraw.py
inflating:  apu101_python/technoap/sitcp/__init__.py
inflating:  apu101_python/technoap/time/stopwatch.py
inflating:  apu101_python/technoap/time/__init__.py
[ tap@localhost sample ] $ cd apu101_python
[ tap@localhost apu101_python ] $ pwd
/tmp/sample/apu101_python
[ tap@localhost apu101_python ] $
```

カレントディレクトリの移動

ソースファイルの展開

カレントディレクトリの移動

カレントディレクトリを表示

現在のディレクトリを表示

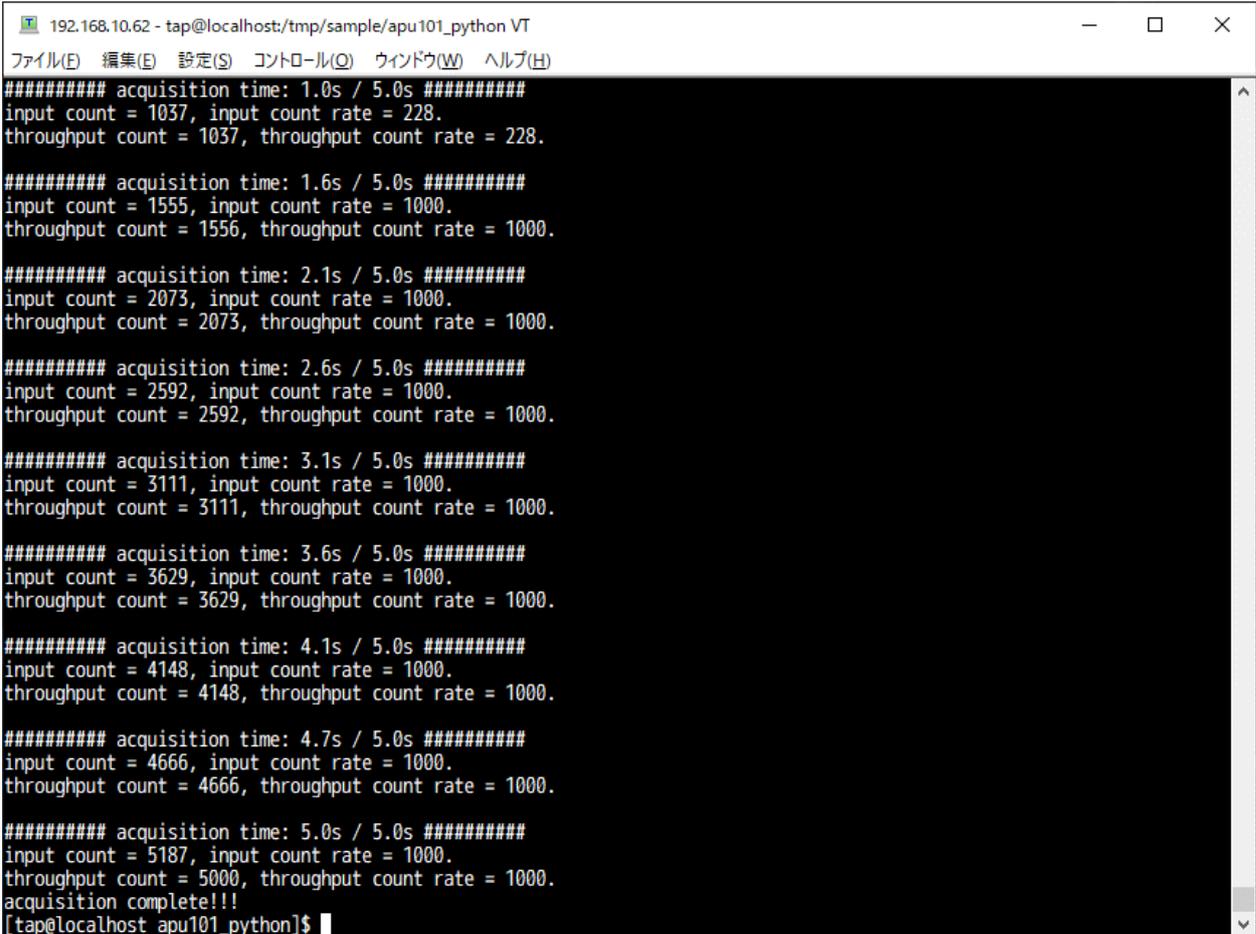
図 1 プログラムソースの展開までの手順

3. 4. ヒストグラム計測プログラム (apu101.py) の実行

このプログラム (apu101.py) はパラメータ設定→計測開始→スペクトル取込み→ステータス表示と計測開始から終了まで一連の流れを行うプログラムです。
プログラムの実行は下記の通りにコマンドを実行します。

- 実行コマンド: `python apu101.py`

図 2 はヒストグラム計測プログラムの実行例です。



```
192.168.10.62 - tap@localhost:/tmp/sample/apu101_python VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
##### acquisition time: 1.0s / 5.0s #####
input count = 1037, input count rate = 228.
throughput count = 1037, throughput count rate = 228.

##### acquisition time: 1.6s / 5.0s #####
input count = 1555, input count rate = 1000.
throughput count = 1556, throughput count rate = 1000.

##### acquisition time: 2.1s / 5.0s #####
input count = 2073, input count rate = 1000.
throughput count = 2073, throughput count rate = 1000.

##### acquisition time: 2.6s / 5.0s #####
input count = 2592, input count rate = 1000.
throughput count = 2592, throughput count rate = 1000.

##### acquisition time: 3.1s / 5.0s #####
input count = 3111, input count rate = 1000.
throughput count = 3111, throughput count rate = 1000.

##### acquisition time: 3.6s / 5.0s #####
input count = 3629, input count rate = 1000.
throughput count = 3629, throughput count rate = 1000.

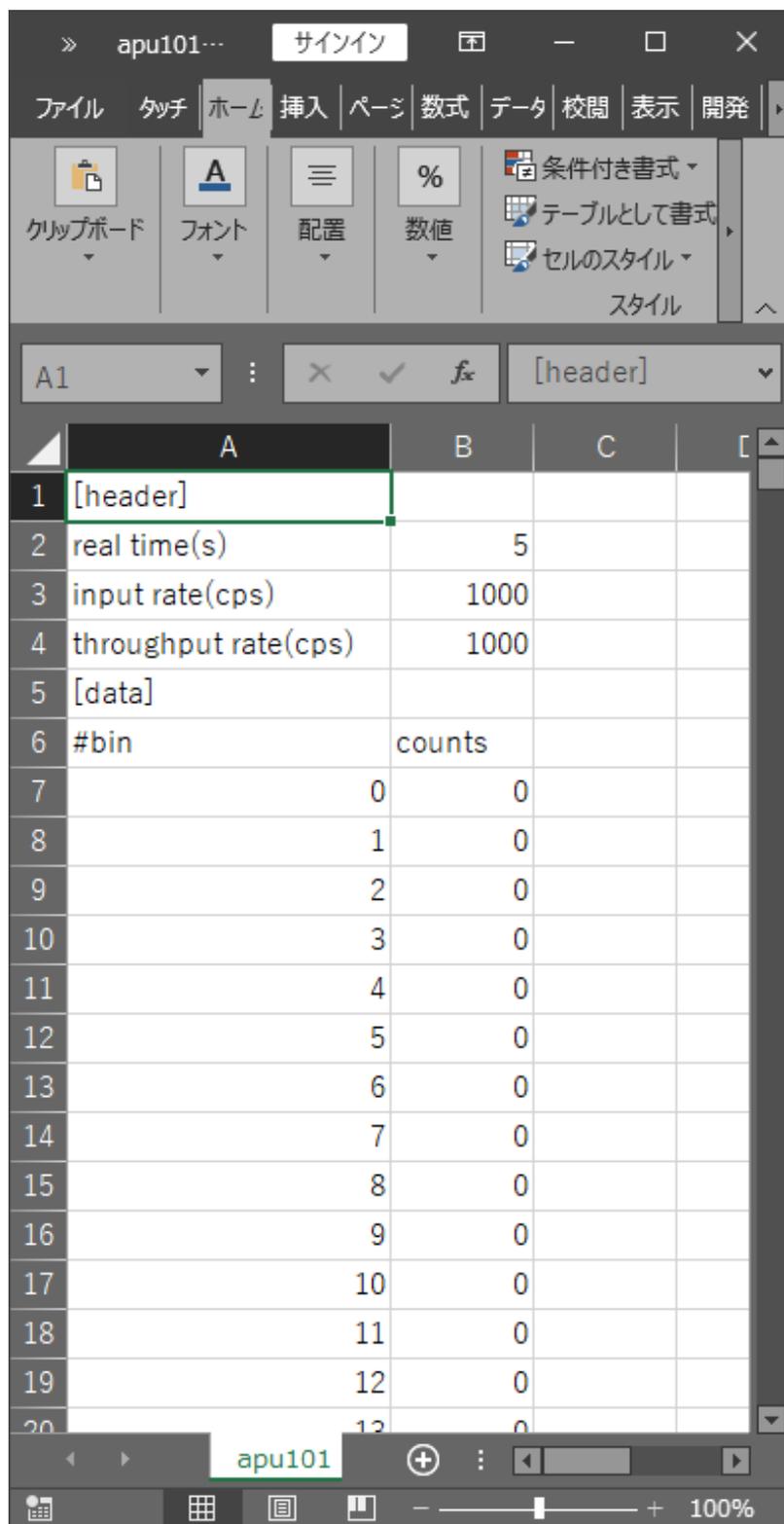
##### acquisition time: 4.1s / 5.0s #####
input count = 4148, input count rate = 1000.
throughput count = 4148, throughput count rate = 1000.

##### acquisition time: 4.7s / 5.0s #####
input count = 4666, input count rate = 1000.
throughput count = 4666, throughput count rate = 1000.

##### acquisition time: 5.0s / 5.0s #####
input count = 5187, input count rate = 1000.
throughput count = 5000, throughput count rate = 1000.
acquisition complete!!!
[tap@localhost apu101_python]$
```

図 2 ヒストグラム計測プログラムの実行

計測終了後、計測したデータを CSV 形式のファイル (apu101.csv) に保存します。保存先のディレクトリは apu101.py と同じ階層です。



The screenshot shows a spreadsheet application window titled 'apu101... サインイン'. The spreadsheet contains the following data:

	A	B	C
1	[header]		
2	real time(s)	5	
3	input rate(cps)	1000	
4	throughput rate(cps)	1000	
5	[data]		
6	#bin	counts	
7	0	0	
8	1	0	
9	2	0	
10	3	0	
11	4	0	
12	5	0	
13	6	0	
14	7	0	
15	8	0	
16	9	0	
17	10	0	
18	11	0	
19	12	0	
20	13	0	

図3 「apu101.py」実行後に生成された CSV 形式のファイル

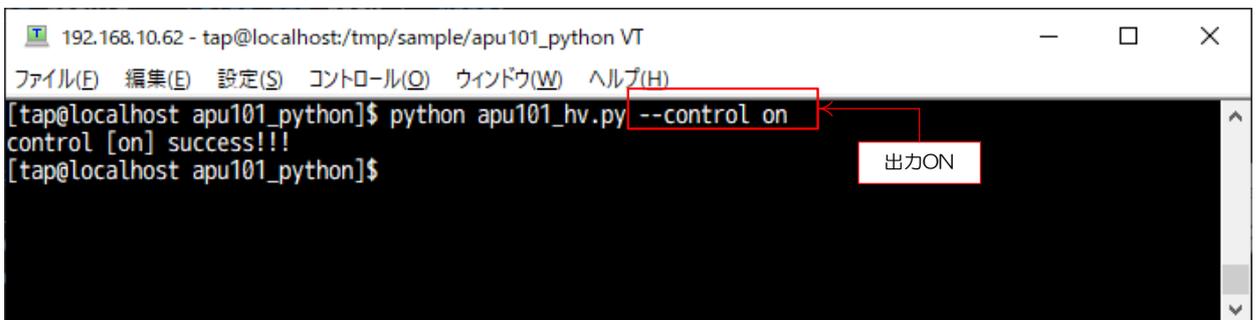
3. 5. 高圧電源制御プログラム (apu101_hv.py) の実行

このプログラム (apu101_hv.py) は高圧電源モジュールの出力を制御 (ON/OFF) するプログラムです。コマンド実行時は引数として出力制御 (ON/OFF, 必須) を渡します。コマンド実行の形式は以下の通りです。

- 実行コマンド : `python apu101_hv2pol.py --control status`
status : on =出力 ON
 off =出力 OFF

※尚、通常はこのプログラムを実行する前に高圧電源モジュールのパラメータを設定する必要があります。

図 4 は高圧電源の出力を ON に実行した場合の実行例です。



```
192.168.10.62 - tap@localhost:/tmp/sample/apu101_python VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
[tap@localhost apu101_python]$ python apu101_hv.py --control on
control [on] success!!!
[tap@localhost apu101_python]$
```

図 4 高圧電源制御 (→ON) プログラムの実行

3. 6. 高圧電源パラメータ設定プログラム (apu101_hv2config.py) の実行

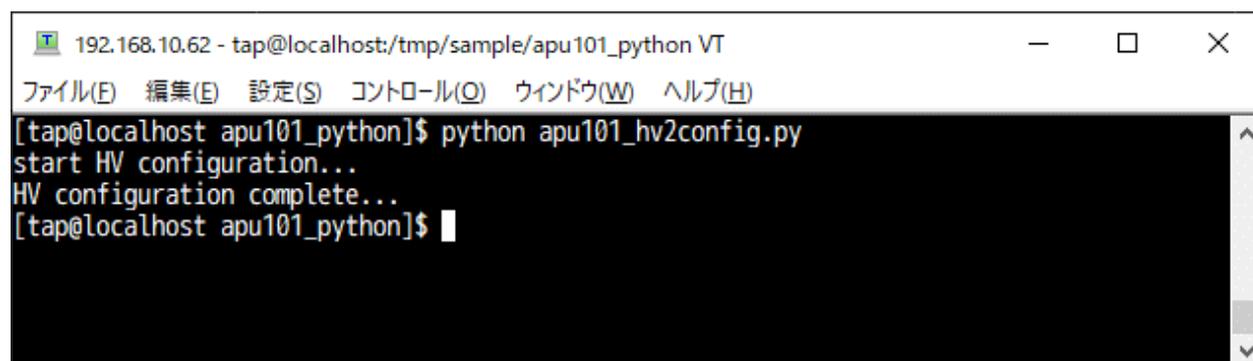
このプログラム (apu101_hv2config.py) は高圧電源モジュールに対するパラメータを設定するプログラムです。パラメータとしては、出力電圧、掃引時間、バイアスシャットダウンの設定等があります。

尚、設定するパラメータの一部には、**同じ型式の製品でもデバイス固有のもの**があります。該当パラメータ値は製品番号をご確認の上、弊社までお問合せをお願いします。従いまして、サンプルプログラムは**弊社よりご連絡しましたパラメータ値に修正する必要があります**があります。

高圧電源モジュールのパラメータの設定は下記の通りにコマンドを実行します。

- 実行コマンド： `python apu101_hv2config.py`

図 5 は高圧電源モジュールのパラメータを設定するプログラムの実行例です。



```
192.168.10.62 - tap@localhost:/tmp/sample/apu101_python VT
ファイル(E) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
[tap@localhost apu101_python]$ python apu101_hv2config.py
start HV configuration...
HV configuration complete...
[tap@localhost apu101_python]$
```

図 5 パラメータを設定するプログラムの実行

※高圧電源パラメータ設定プログラムは**一部のパラメータ値をデバイス固有の値に修正してある事を前提**とします。

また、該当するパラメータについては「デジタルスペクトルメータ, APU101/APN101 コマンドマニュアル」の「3. 8. HV (高圧電源コマンド) 説明」の項を参照してください。

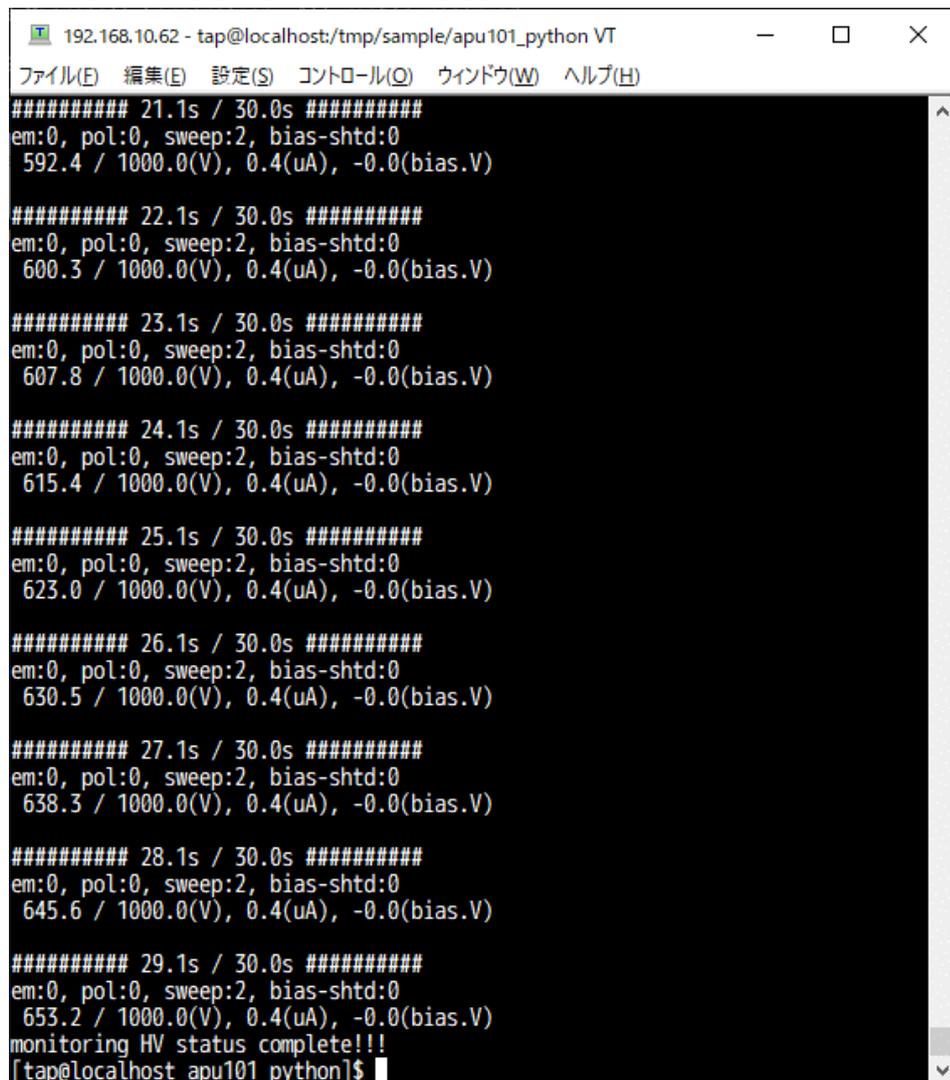
3. 7. 高圧電源ステータスマニタプログラム (apu101_hv2satus.py) の実行

このプログラム (apu101_hv2status.py) は高圧電源の状態を表示するプログラムです。所定の時間、一定周期で高圧電源の状態を表示します。コマンドの実行の形式は以下の通りです。

- 実行コマンド: `python apu101_hv2status.py --time sec1 --interval sec2`
 - sec1: モニタリング時間 (任意, 小数での指定)
→ デフォルト値 = 30.0 秒
 - sec2: モニタリング周期 (任意, 小数での指定)
→ デフォルト値 = 1.0 秒

図 6 は高圧電源ステータスプログラムを実行した例です。

→ パラメータ無しのため、モニタリング時間と周期はデフォルト値になります。



```
192.168.10.62 - tap@localhost:/tmp/sample/apu101_python VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
##### 21.1s / 30.0s #####
em:0, pol:0, sweep:2, bias-shtd:0
592.4 / 1000.0(V), 0.4(uA), -0.0(bias.V)

##### 22.1s / 30.0s #####
em:0, pol:0, sweep:2, bias-shtd:0
600.3 / 1000.0(V), 0.4(uA), -0.0(bias.V)

##### 23.1s / 30.0s #####
em:0, pol:0, sweep:2, bias-shtd:0
607.8 / 1000.0(V), 0.4(uA), -0.0(bias.V)

##### 24.1s / 30.0s #####
em:0, pol:0, sweep:2, bias-shtd:0
615.4 / 1000.0(V), 0.4(uA), -0.0(bias.V)

##### 25.1s / 30.0s #####
em:0, pol:0, sweep:2, bias-shtd:0
623.0 / 1000.0(V), 0.4(uA), -0.0(bias.V)

##### 26.1s / 30.0s #####
em:0, pol:0, sweep:2, bias-shtd:0
630.5 / 1000.0(V), 0.4(uA), -0.0(bias.V)

##### 27.1s / 30.0s #####
em:0, pol:0, sweep:2, bias-shtd:0
638.3 / 1000.0(V), 0.4(uA), -0.0(bias.V)

##### 28.1s / 30.0s #####
em:0, pol:0, sweep:2, bias-shtd:0
645.6 / 1000.0(V), 0.4(uA), -0.0(bias.V)

##### 29.1s / 30.0s #####
em:0, pol:0, sweep:2, bias-shtd:0
653.2 / 1000.0(V), 0.4(uA), -0.0(bias.V)
monitoring HV status complete!!!
[tap@localhost apu101_python]$
```

図 6 高圧電源ステータスマニタコマンドの実行

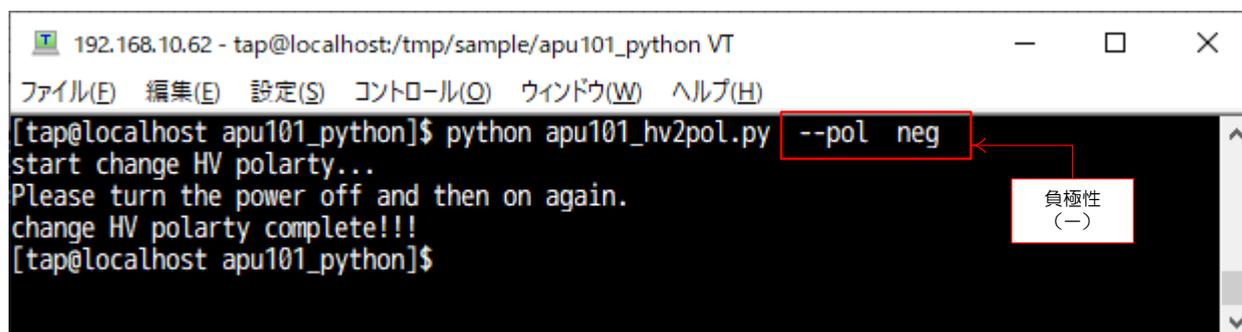
3. 8. 高圧電源極性変更プログラム (apu101_hv2pol.py) の実行

このプログラム (apu101_hv2pol.py) は高圧電源モジュールの極性を変更します。コマンド実行時は引数として極性 (必須) を渡します。コマンド実行の形式は以下の通りです。

- 実行コマンド: `python apu101_hv2pol.py --pol polarity`
polarity : pos= 正極性 (+)
 neg= 負極性 (-)

尚、コマンドを実行しても極性はすぐには反映されません。一度電源を入れ直す必要があります。コマンド実行後は電源を入れ直してください。

図 7 は負極性に変更する場合のコマンド実行の例です。



```
192.168.10.62 - tap@localhost:/tmp/sample/apu101_python VT
ファイル(E) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
[tap@localhost apu101_python]$ python apu101_hv2pol.py --pol neg
start change HV polarity...
Please turn the power off and then on again.
change HV polarity complete!!!
[tap@localhost apu101_python]$
```

The image shows a terminal window with a black background and white text. The command `python apu101_hv2pol.py --pol neg` is entered and executed. The output indicates that the polarity has been changed to negative. A red box highlights the `--pol neg` part of the command, and a red arrow points from this box to a white box containing the text "負極性 (-)".

図 7 負極性を変更するコマンドの実行

以上