

APV8508
サンプルプログラム
取扱説明書
Python 版

第 1.0 版 2020 年 06 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.	キャリブレーション実行プログラム (calibration.py) の実行	7
3. 5.	ヒストグラム計測プログラム (apv8508_hist.py) の実行	8
3. 6.	リストデータ計測プログラム (apv8508_list.py) の実行	10
4.	トラブルシューティング	11
4. 1.	リスト計測プログラムにおいて保存ファイルがない	11

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 APV8508（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 IP 通信（UDP 及び TCP）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。

※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 06 月 03 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: Python2.7 及び 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 2.7 及び 3.6

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 apv8508_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/)

```
└─apv8508_python
  │ apv8508_hist.py    #ヒストグラム計測プログラム
  │ apv8508_list.py   #リストデータ計測プログラム
  │ calibration.py    #キャリブレーション実行プログラム
  │
  └─technoap
    └─device          #デバイス制御ライブラリ群
      │ └─apv8508    # →APV8508 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/apv8508_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 ~ ]$ cd /tmp/sample
[ tap@localhost sample ]$ unzip apv8508_python.zip
Archive:  apv8508_python.zip
  creating:  apv8508_python/
 inflating:  apv8508_python/apv8508_hist.py
 inflating:  apv8508_python/apv8508_list.py
 inflating:  apv8508_python/calibration.py
  creating:  apv8508_python/technoap/
  creating:  apv8508_python/technoap/device/
  creating:  apv8508_python/technoap/device/apv8508/
 inflating:  apv8508_python/technoap/device/apv8508/common.py
 inflating:  apv8508_python/technoap/device/apv8508/config.py
 inflating:  apv8508_python/technoap/device/apv8508/constant.py
 inflating:  apv8508_python/technoap/device/apv8508/fastdatatx.py
 inflating:  apv8508_python/technoap/device/apv8508/listdata.py
 inflating:  apv8508_python/technoap/device/apv8508/register.py
 inflating:  apv8508_python/technoap/device/apv8508/status.py
 inflating:  apv8508_python/technoap/device/apv8508/__init__.py
  creating:  apv8508_python/technoap/device/common/
 inflating:  apv8508_python/technoap/device/common/reginfo.py
 inflating:  apv8508_python/technoap/device/common/register.py
 inflating:  apv8508_python/technoap/device/common/registerRW.py
 inflating:  apv8508_python/technoap/device/common/__init__.py
 extracting: apv8508_python/technoap/device/__init__.py
  creating:  apv8508_python/technoap/num/
 inflating:  apv8508_python/technoap/num/bitconverter.py
 inflating:  apv8508_python/technoap/num/byteorder.py
 extracting: apv8508_python/technoap/num/__init__.py
  creating:  apv8508_python/technoap/sitcp/
 inflating:  apv8508_python/technoap/sitcp/constant.py
 inflating:  apv8508_python/technoap/sitcp/fasttxbase.py
 inflating:  apv8508_python/technoap/sitcp/rbcp.py
 inflating:  apv8508_python/technoap/sitcp/rbcpbase.py
 inflating:  apv8508_python/technoap/sitcp/rbcpformat.py
 inflating:  apv8508_python/technoap/sitcp/rbcppraw.py
 inflating:  apv8508_python/technoap/sitcp/__init__.py
  creating:  apv8508_python/technoap/time/
 inflating:  apv8508_python/technoap/time/stopwatch.py
 extracting: apv8508_python/technoap/time/__init__.py
 extracting: apv8508_python/technoap/__init__.py
[ tap@localhost sample ]$ cd apv8508_python/
[ tap@localhost apv8508_python ]$ pwd
/tmp/sample/apv8508_python
[ tap@localhost apv8508_python ]$
```

Callout boxes in the image:

- カレントディレクトリの移動 (points to `cd /tmp/sample`)
- ソースファイルの展開 (points to the `unzip` command)
- カレントディレクトリを表示 (points to `pwd`)
- カレントディレクトリの移動 (points to `cd apv8508_python/`)
- 現在のディレクトリを表示 (points to the final prompt)

図 1 プログラムソースの展開までの手順

3. 4. キャリブレーション実行プログラム (calibration.py) の実行

このプログラムは本装置の ADC に対してキャリブレーション実行を命令するプログラムです。本装置の電源投入後、1 度だけ実行します。

→実行コマンド: `python calibration.py`



```
192.168.10.62 - tap@localhost:/tmp/sample/apv8508_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 apv8508_python]$ python calibration.py
calibration start!!!
wating...
success!
[tap@localhost apv8508_python]$
```

図 2 キャリブレーションの実行

3. 5. ヒストグラム計測プログラム (apv8508_hist.py) の実行

このプログラム (apv8508_hist.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 (5 秒間計測) 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 apv8508_hist.py`

図 3 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/apv8508_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 apv8508_python]$ python apv8508_hist.py
##### 0.0 / 5.0 #####
throughput total counts = 0, 0, 3711, 0, 0, 0, 0, 0
throughput count rate = 0, 0, 1000016, 0, 0, 0, 0, 0
dead time counts = 0, 0, 61017, 0, 0, 0, 0, 0

##### 1.1 / 5.0 #####
throughput total counts = 0, 0, 1129044, 0, 0, 0, 0, 0
throughput count rate = 0, 0, 999418, 0, 0, 0, 0, 0
dead time counts = 0, 0, 98304610, 0, 0, 0, 0, 0

##### 2.2 / 5.0 #####
throughput total counts = 0, 0, 2234245, 0, 0, 0, 0, 0
throughput count rate = 0, 0, 1000016, 0, 0, 0, 0, 0
dead time counts = 0, 0, 195008503, 0, 0, 0, 0, 0

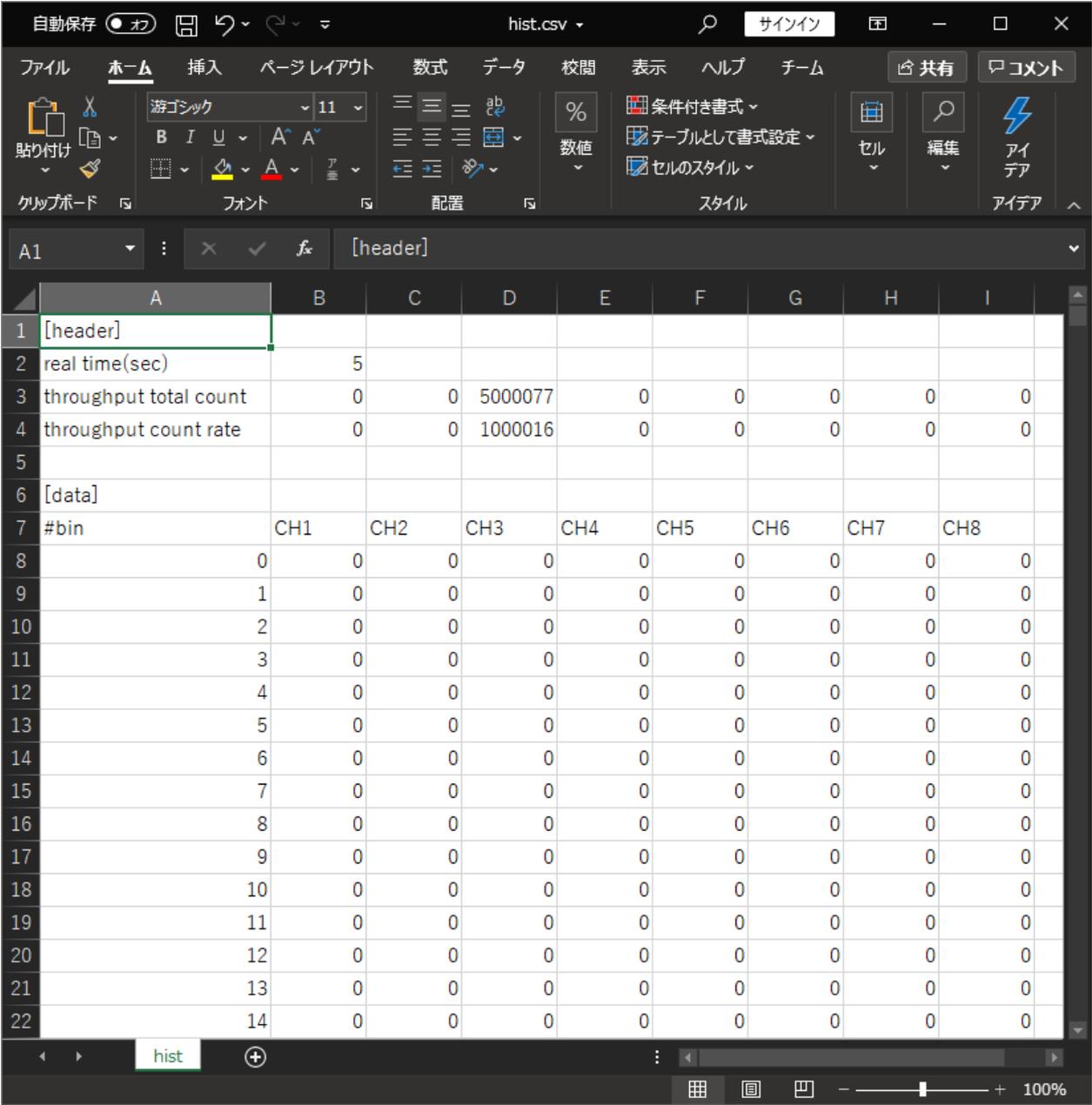
##### 3.3 / 5.0 #####
throughput total counts = 0, 0, 3342194, 0, 0, 0, 0, 0
throughput count rate = 0, 0, 1000015, 0, 0, 0, 0, 0
dead time counts = 0, 0, 291904575, 0, 0, 0, 0, 0

##### 4.5 / 5.0 #####
throughput total counts = 0, 0, 4463196, 0, 0, 0, 0, 0
throughput count rate = 0, 0, 1000016, 0, 0, 0, 0, 0
dead time counts = 0, 0, 389949993, 0, 0, 0, 0, 0

##### 5.6 / 5.0 #####
throughput total counts = 0, 0, 5000077, 0, 0, 0, 0, 0
throughput count rate = 0, 0, 1000015, 0, 0, 0, 0, 0
dead time counts = 1, 1, 490536296, 1, 1, 1, 1, 1
[tap@localhost apv8508_python]$ ls -la
合計 196
drwxrwxr-x 3 tap tap 106 6月 4 10:51 .
drwxrwxr-x 3 tap tap 54 6月 4 10:33 ..
-rw-rw-r-- 1 tap tap 8432 6月 3 17:41 apv8508_hist.py
-rw-rw-r-- 1 tap tap 8709 6月 3 16:44 apv8508_list.py
-rw-rw-r-- 1 tap tap 833 5月 29 14:25 calibration.py
-rw-rw-r-- 1 tap tap 171282 6月 4 10:51 hist.csv
drwxrwxr-x 6 tap tap 95 6月 4 10:38 technoap
[tap@localhost apv8508_python]$
```

図 3 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 (hist.csv) に保存します。保存先のディレクトリは apv8508_hist.py と同じ階層です。



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I
1	[header]								
2	real time(sec)	5							
3	throughput total count	0	0	5000077	0	0	0	0	0
4	throughput count rate	0	0	1000016	0	0	0	0	0
5									
6	[data]								
7	#bin	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8
8	0	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0
10	2	0	0	0	0	0	0	0	0
11	3	0	0	0	0	0	0	0	0
12	4	0	0	0	0	0	0	0	0
13	5	0	0	0	0	0	0	0	0
14	6	0	0	0	0	0	0	0	0
15	7	0	0	0	0	0	0	0	0
16	8	0	0	0	0	0	0	0	0
17	9	0	0	0	0	0	0	0	0
18	10	0	0	0	0	0	0	0	0
19	11	0	0	0	0	0	0	0	0
20	12	0	0	0	0	0	0	0	0
21	13	0	0	0	0	0	0	0	0
22	14	0	0	0	0	0	0	0	0

図 4 「apv8508_hist.py」 実行後に生成された CSV 形式のファイル

3. 6. リストデータ計測プログラム (apv8508_list.py) の実行

このプログラムはパラメータ設定→計測開始→リストデータ取込み+ステータス表示と計測開始から終了（5秒間計測）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 apv8508_list.py`

```
192.168.10.62 - tap@localhost:/tmp/sample/apv8508_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 apv8508_python]$ python apv8508_list.py
##### 1.0 / 5.0 #####
throughput total counts = 0, 0, 1037514, 0, 0, 0, 0, 0
throughput count rate = 0, 0, 990900, 0, 0, 0, 0, 0
dead time counts = 0, 0, 90543541, 0, 0, 0, 0, 0

##### 2.1 / 5.0 #####
throughput total counts = 0, 0, 2061163, 0, 0, 0, 0, 0
throughput count rate = 0, 0, 1000015, 0, 0, 0, 0, 0
dead time counts = 0, 0, 180089905, 0, 0, 0, 0, 0

##### 3.1 / 5.0 #####
throughput total counts = 0, 0, 3085255, 0, 0, 0, 0, 0
throughput count rate = 0, 0, 1000016, 0, 0, 0, 0, 0
dead time counts = 0, 0, 269675578, 0, 0, 0, 0, 0

##### 4.1 / 5.0 #####
throughput total counts = 0, 0, 4109089, 0, 0, 0, 0, 0
throughput count rate = 0, 0, 1000015, 0, 0, 0, 0, 0
dead time counts = 0, 0, 359215083, 0, 0, 0, 0, 0

##### 5.1 / 5.0 #####
throughput total counts = 0, 0, 5000077, 0, 0, 0, 0, 0
throughput count rate = 0, 0, 1000016, 0, 0, 0, 0, 0
dead time counts = 1, 1, 449499440, 1, 1, 1, 1, 1
[tap@localhost apv8508_python]$ ls -la
合計 78156
drwxrwxr-x 3 tap tap 106 6月 4 10:53 .
drwxrwxr-x 3 tap tap 54 6月 4 10:33 ..
-rw-rw-r-- 1 tap tap 8432 6月 3 17:41 apv8508_hist.py
-rw-rw-r-- 1 tap tap 8709 6月 3 16:44 apv8508_list.py
-rw-rw-r-- 1 tap tap 833 5月 29 14:25 calibration.py
-rw-rw-r-- 1 tap tap 80001232 6月 4 10:53 list.bin
drwxrwxr-x 6 tap tap 95 6月 4 10:38 technoap
[tap@localhost apv8508_python]$
```

図5 リストデータ計測プログラムの実行

計測により収集したデータはバイナリ形式のファイル (list.bin) に保存します。保存先のディレクトリは apv8508_list.py と同じ階層です。

4. トラブルシューティング

4. 1. リスト計測プログラムにおいて保存ファイルがない

リスト計測プログラムにて、計測を実行しても計測データを保存したファイル（以下、保存ファイル）がない場合があります。その場合、本装置が入力信号を検出できていない事が考えられます。本装置は信号を検出できないと計測自体を行わず、保存ファイルが生成する要因がない為です。この場合は入力信号の確認を行い、本製品の仕様を満たした信号を入力してください。

以上