

APU101 サンプルプログラム 取扱説明書 Linux, C 版

第 1.0 版 2017 年 07 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
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ダウンロード.....	5
3. 3.	プログラムのビルドと実行.....	6
4.	トラブルシューティング.....	8
4. 1.	入力カウントレートや出力カウントレートが0のまま変化しない.....	8

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(X)または APN101(X)（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、異臭などがあつ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つ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 C 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

2017年07月31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 release 6.7(Final)
- (2) コンパイラ: gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-16)

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_linux.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。

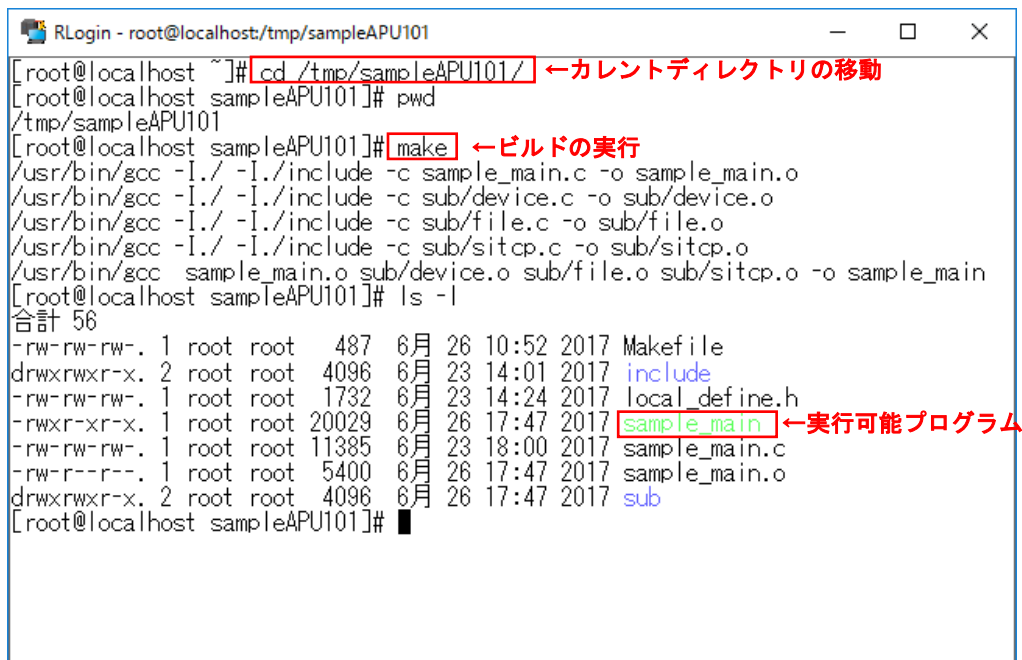
```
(解凍先ディレクトリ)
└─sampleAPU101
   └─include      # ヘッダファイル用ディレクトリ
   └─sub          # 共通関数ソース用ディレクトリ
```

説明の都合上、以降の説明では、解凍先ディレクトリとして /tmp にコピーした事を前提とします。実際のコピー先が異なる場合は、適宜そのディレクトリに読替えて下さい。

3. 3. プログラムのビルドと実行

- (1) サンプルプログラムのビルドはコマンドラインにて行います。
まず、ターミナルウィンドウを開き、下記のカレントディレクトリを移動するコマンドを実行して、目的のディレクトリに移動します。
 - カレントディレクトリを移動するコマンド： “cd /tmp/sampleAPU101”
- (2) 目的のディレクトリに移動したら、下記の通りにビルドを実行します。
 - ビルドの実行： “make”
- (3) ビルドが終了すると、同じディレクトリに、実行可能プログラムが作成されます。

※下図はターミナルウィンドウにおける「カレントディレクトリ移動～ビルド」までのコマンドラインの入出力の例です。



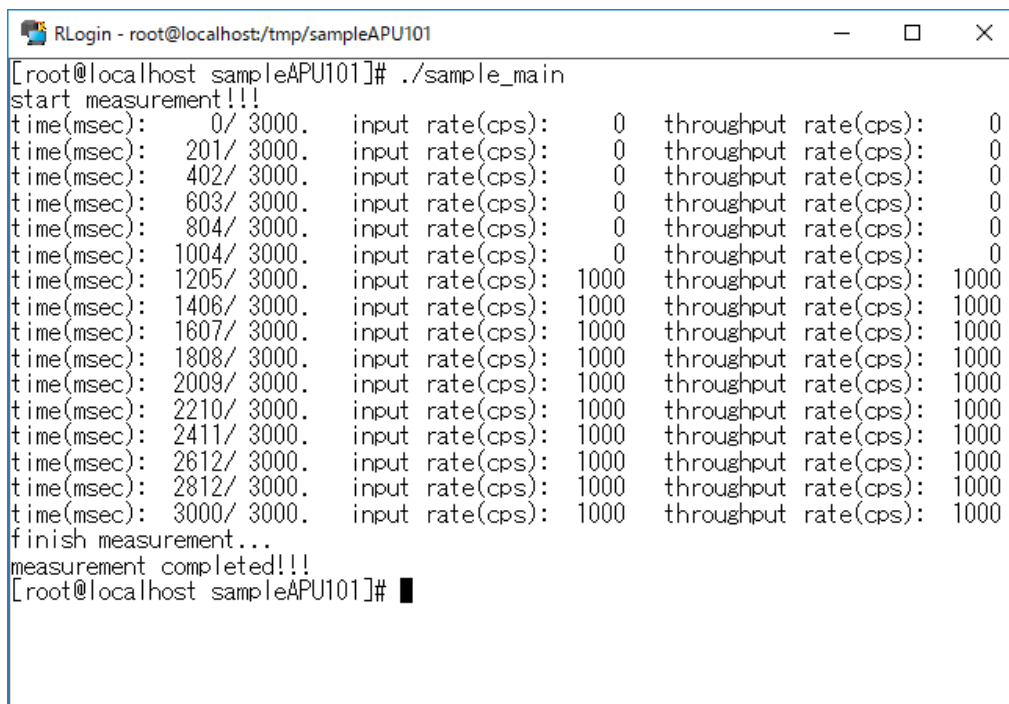
```
RLogin - root@localhost/tmp/sampleAPU101
[root@localhost ~]# cd /tmp/sampleAPU101/ ←カレントディレクトリの移動
[root@localhost sampleAPU101]# pwd
/tmp/sampleAPU101
[root@localhost sampleAPU101]# make ←ビルドの実行
/usr/bin/gcc -I./ -I./include -c sample_main.c -o sample_main.o
/usr/bin/gcc -I./ -I./include -c sub/device.c -o sub/device.o
/usr/bin/gcc -I./ -I./include -c sub/file.c -o sub/file.o
/usr/bin/gcc -I./ -I./include -c sub/sitcp.c -o sub/sitcp.o
/usr/bin/gcc sample_main.o sub/device.o sub/file.o sub/sitcp.o -o sample_main
[root@localhost sampleAPU101]# ls -l
合計 56
-rw-rw-rw-. 1 root root 487 6月 26 10:52 2017 Makefile
drwxrwxr-x. 2 root root 4096 6月 23 14:01 2017 include
-rw-rw-rw-. 1 root root 1732 6月 23 14:24 2017 local_define.h
-rwxr-xr-x. 1 root root 20029 6月 26 17:47 2017 sample_main ←実行可能プログラム
-rw-rw-rw-. 1 root root 11385 6月 23 18:00 2017 sample_main.c
-rw-r--r--. 1 root root 5400 6月 26 17:47 2017 sample_main.o
drwxrwxr-x. 2 root root 4096 6月 26 17:47 2017 sub
[root@localhost sampleAPU101]# █
```

図 1 ビルドの手順(コマンドラインの入出力の例)

(4) 検出器のプリアンプ出力信号をリニアアンプ等にて波形整形したガウス波形信号または模擬信号を入力し、実行可能プログラムのプロセスを起動します。起動方法は下記の通りです。

・実行可能プログラムの起動： “./sample_main”

実行後、Real time(秒)、Input rate(cps)、throughput rate(cps)を3秒間、下図のように更新します。



```
RLogin - root@localhost:/tmp/sampleAPU101
[root@localhost sampleAPU101]# ./sample_main
start measurement!!!
time(msec): 0/ 3000.   input rate(cps): 0   throughput rate(cps): 0
time(msec): 201/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 402/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 603/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 804/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 1004/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 1205/ 3000. input rate(cps): 1000  throughput rate(cps): 1000
time(msec): 1406/ 3000. input rate(cps): 1000  throughput rate(cps): 1000
time(msec): 1607/ 3000. input rate(cps): 1000  throughput rate(cps): 1000
time(msec): 1808/ 3000. input rate(cps): 1000  throughput rate(cps): 1000
time(msec): 2009/ 3000. input rate(cps): 1000  throughput rate(cps): 1000
time(msec): 2210/ 3000. input rate(cps): 1000  throughput rate(cps): 1000
time(msec): 2411/ 3000. input rate(cps): 1000  throughput rate(cps): 1000
time(msec): 2612/ 3000. input rate(cps): 1000  throughput rate(cps): 1000
time(msec): 2812/ 3000. input rate(cps): 1000  throughput rate(cps): 1000
time(msec): 3000/ 3000. input rate(cps): 1000  throughput rate(cps): 1000
finish measurement...
measurement completed!!!
[root@localhost sampleAPU101]#
```

図 2 プログラム実行(コマンドラインの入出力の例)

3秒経過後、計測したデータをファイル hist_data.csv に保存します。その保存先ディレクトリは sample_main と同じディレクトリとなっています。

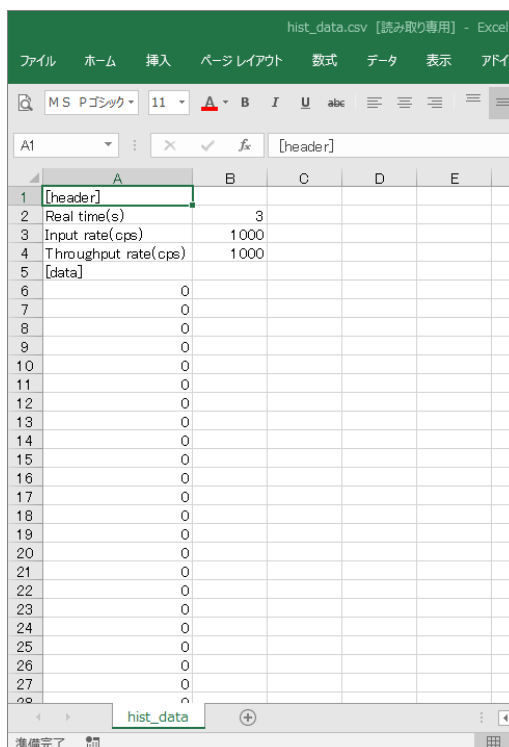


図 3 「sample_main」実行後生成された csv 形式の hist データファイル

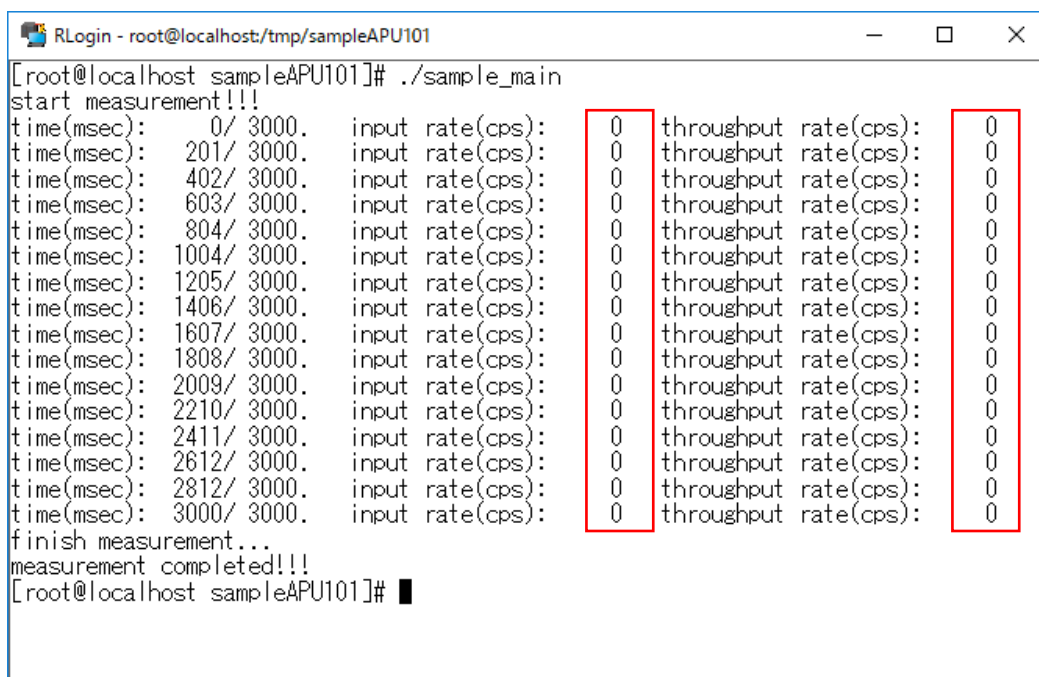
4. トラブルシューティング

4. 1. 入力カウントレートや出力カウントレートが0のまま変化しない

【現象】

プログラム起動後、表示される入力カウントレート (input rate) や出力カウントレート (throughput rate) の値が0のまま変化しない。

また、データファイルの[data]欄の値も全てオール0となってしまう。



```
RLogin - root@localhost:/tmp/sampleAPU101
[root@localhost sampleAPU101]# ./sample_main
start measurement!!!
time(msec): 0/ 3000.   input rate(cps): 0   throughput rate(cps): 0
time(msec): 201/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 402/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 603/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 804/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 1004/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 1205/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 1406/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 1607/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 1808/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 2009/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 2210/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 2411/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 2612/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 2812/ 3000. input rate(cps): 0   throughput rate(cps): 0
time(msec): 3000/ 3000. input rate(cps): 0   throughput rate(cps): 0
finish measurement...
measurement completed!!!
[root@localhost sampleAPU101]#
```

【対策】

sample.cにあるrun_measurement関数について、msleep(100)の値を200に変更してみてください。

```
static int run_measurement(DEVICE_COM_INFO *pComInfo, unsigned qword ms_count)
{
    int ret;
    unsigned qword now_count;
    unsigned int now_msec;
    unsigned short status;
    unsigned int input_cps;
    unsigned int throughput_cps;

    // clear data
    device_write_3step(pComInfo, DEVICE_COMMAND_CLR);
    // wait for completion of clear processing
    msleep(100);

    // start measurement
    device_start_measurement(pComInfo);

    while(1) {
```

改善が見られない場合は、装置の主電源をOFFにして、数秒後に再度ONにしてください。

以上