

Electronics Memo

January 7, 2020

1 測定準備のためのコマンド

- 接続している device の確認

```
$ nmcli device
```

```
$ nmcli con up enp0s31f6
```

```
$ setnew
$ setord
```

- Pyrame のストップ・スタート

```
$ stoppy[systemctl stop pyrame]
$ startpy[systemctl start pyrame]
```

- thresholdDAC の値の設定
以下のコマンドの***に threshold DAC の値を代入する。

```
$ change_trigth.sh /opt/calicoes/config/wagasci_config_dif1_chip1.txt ***
```

- preampDAC の値の設定

```
$ wgChangeConfig -c -r -f /opt/calicoes/config/wagasci_config_dif1_chip1.txt
-m 3 -b 36 -v 53
```

- InputDAC の値の設定

```
$ wgChangeConfig -c -r -f /opt/calicoes/config/wagasci_config_dif1_chip1.txt
-m 2 -b 36 -v 41
```

引数の意味は以下の表の通り。

Table 1: wgChangeConfig の引数

引数名	意味	備考
m		
b	チャンネル指定	36 なら全チャンネルを、0-35 ならその ch を意味する。
v	DAC の値	

なお、txt ファイルの dif、chip の後の数字でそれぞれ DIF と Chip の番号を指定できる。

2 実際の測定の流れ

1. GDCC/CCC の立ち上げ
LV_GDCC/CCC を on (1.2V ぐらいから少しずつ上がり 1.8~1.9V になる)
2. Low voltage (LV) の立ち上げ
LV_IF を on (0.54V)
3. xml ファイルの読み込み・Configure
ASU が 1 枚の場合と 3 枚の場合はそれぞれ、

```
load\_config\_file.sh /opt/calicoes/config/wagasci\_config\_1asu.xml [load]
load\_config\_file.sh /opt/calicoes/config/wagasci\_config\_3asu.xml [load3]
```

さらに、初期化、configure を行う。

```
$ init[initialize.sh]
$ conf[configure.sh]
```

この時点で LV_IF の電圧が少し上昇する。[Ex.) 0.54 A → 0.58-0.61 A]

これなんだろう？

```
spill_setinfo.sh p
spill_setinfo.sh a
spill_getinfo.sh
```

4. HV の立ち上げ
HV を入れる、3.5 V から出力し、徐々に (1 V ずつ) 上げていく。0 V から始めないのは、微小なバイアス電圧をかけた段階で反対方向に電流が流れるのを防ぐため？。
5. spill の開始
internal spill を利用する場合は以下のコマンドを利用する。

```
$ spill_internal.sh
```

この時、GDCC の前面の Light が点滅する。なお、internal spill で送られる信号は周期 260 msec、パルス幅 5 msec の方形波である。

外部トリガーを使って測定を行う場合は、

```
$ spill_external.sh
```

を用いる。その際、写真の場所に信号を入れる。

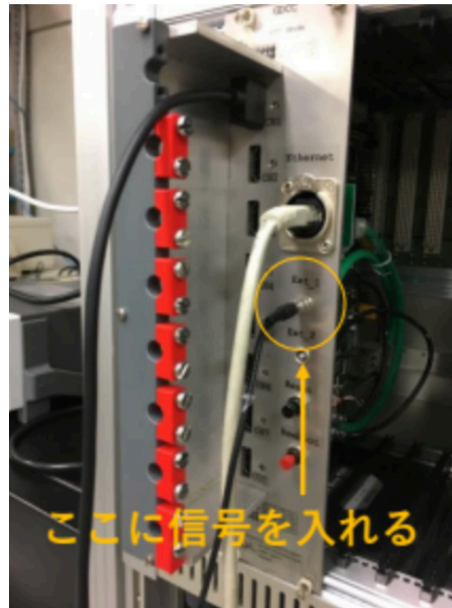


Figure 1: 外部トリガーの入力端子

spill の設定を変更したいときは、/SiTCP/script の中の spill_setinfo.sh というコマンドを用いる。
 ※このコマンドがないときは Analysis/bin/wgChangeConfig を使用。

```
$ wgChangeConfig -f CONFIG_FILE_NAME.txt
```

で同じように確認できるはず。これを使用すれば config をこのコマンドのみで変更できる。

6. 測定の開始

以下のコマンドで測定のスタート、ストップができる。

```
$ start_run.sh ***** remove
```

```
$ stop_run.sh *****
```

7. 終了方法

測定前と逆の手順を行う。

- HV を消す
- LV_IF を消す
- LV_GDCC/CCC を消す

8. データを見る

raw データをデコードする必要がある。

```
Decoder -f /home/data/prototech/*****/*****.raw
```

これにより、/home/data/decode/ 以下に root ファイルが生成される (旧: /home/data/rootfile/)。
 root ファイルができる場所を指定する (DIR_NAME) には、デコードする際に

```
Decoder -f /home/data/prototech/*****/*****.raw -o /PATH/TO/DIR_NAME
```

とする。

running 中のデータを見るには、ls を用いる。

3 環境変数の設定

3.1 環境変数の設定

WAGASCI 関連の環境変数は setting_wagasci.sh に記述されている。

Table 2: setting_wagasci.sh

ディレクトリパス	意味
WAGASCI_RAWDATADIR	raw data が出力されるディレクトリ
WAGASCI_DECODEDIR	Decode した _tree.root の出力先のデフォルト
WAGASCI_HISTDIR	wgMakeHist でヒストグラムを生成した _hist.root の出力先のデフォルト
WAGASCI_RECONDIR	wgRecon で飛跡再構成した _recon.root の出力先のデフォルト
WAGASCI_XMLDATADIR	wgAnaHist で _hist.root を解析した xml file の出力先のデフォルト
WAGASCI_IMGDATADIR	諸々で図を生成した場合の出力先のデフォルト
WAGASCI_LOGDIR	コマンド実行時のログの出力先のデフォルト
WAGASCI_MAINDIR	Analysis ディレクトリへのパス
WAGASCI_RUNCOMMANDDIR	RunCommand ディレクトリへのパス
WAGASCI_CALICOE	/opt/calicoe ディレクトリへのパス
WAGASCI_CALIBDATADIR	Calibration 結果をまとめているディレクトリへのパス
WAGASCI_BSDDIR	beam summary data へのパス
WAGASCI_DQDIR	wgDQCheck で出力するデータクオリティ用のファイルへのパス
WAGASCI_DQHISTORYDIR	wgDQHistory で出力するデータクオリティの結果を纏めたファイルの出力先

3.2 新しくディレクトリを作る場合

1. ディレクトリの作成
もし対応するディレクトリが存在しない場合, mkdir でディレクトリを作って
2. setting_wagasci.sh の編集
そのディレクトリへの絶対パスを setting_wagasci.sh に書き込み、

```
$ source setting\_wagasci.sh
```

で設定を読み込む。

3. bash.rc の編集
環境設定はターミナル終了時に消えてしまうので、ターミナル起動時に自動で読み込みを行うために
/.bashrc に

```
source /PATH/T0/setting\_wagasci.sh
```

と書き込む。

4 CCC との接続

- enp0s31f6 の IP アドレスを適切なもの (192.168.10.2 / 255.255.255.0 など) に設定する (やりづらければ、nmtui を使用する。)
- nmcli device を実行したら接続されているデバイスが表示される
- nmcli con up enp0s31f6 で接続が行われる
- systemctl restart NetworkManager で接続の反映

5 txt ファイルの確認方法

```
# cd Analysis
# make clean
# make
# ./bin/wgChangeConfig -f wagasci_config_dif1_chip1.txt

/usr/local/src/wagasci_software/Analysis/bin/wgChangeConfig
-f /opt/calicoes/config/wagasci_config_dif1_chip1.txt
```

※ bin の中に入っている実行ファイルは

```
# ./bin/wgChangeConfig -h
```

のようにしてそれぞれヘルプが見れます。

6 よくあるエラー

- 急に configure.sh がうまくいかなくなる。
→ LV を付け直し、まず GDCC/CCC の LV をつけ 1.2A ぐらいから上昇し、1.9V ぐらいまで上昇したことを確認してから、IF の LV をつける。(この作業は正確に)
- IF の LV の値がおかしい
→ フラットケーブル (ASU - IF) の接続が正しくない
→ シルクに合わせる + 全体的に中心になるようにセット
参考 ASU 一つの場合
 - 正常なとき
5 V をかけたときの電流値: 0.54 A → (config が通る) 0.58-0.61 A (CV に明かりが点灯)
 - 正常ではないとき
5 V をかけたときの電流値: 2.2 A とか ?? (CC に明かりが点灯)

7 SPIROC について

7.1 PADAC について

1つはプリアンプの値が低すぎると安定してトリガーをかけるのが難しくなることに由来してもう1つはプリアンプの値が低い場合、ゲインの線形性があまり良くなるということです。

後者はどの程度か見積もることが出来てませんが、DAC 値が 50 程度なら問題なさそうです。(30,40 になるとどうなるかはわかりませんが。)

ここはどう tuning するかによって変わってくると思うので、うまく最適化してください。

For the Gain of each channel, the (voltage) gain is given by $15\text{pF}/C_{\text{feedback}}$ (for the HG) C_{feedback} is $(63 - 6 - \text{bit_Gain_Code}) \times 25 \text{ fF}$.

The C_{feedback} can be tuned by 25 fF steps. However, there is a parasitic capacitance on the feedback, so the theoretical gain with 25 fF should be $15 \text{ p}/25 \text{ f} = 600$ but real gain is lower.

→理論的には 600 ずつ変えられるが...

7.2 4-bit threshold adjustment について

個々のチャンネルに対してスレショルドの値を微調整しています。これを使わない場合、chip 内のスレショルドは一律の値になります。

single MPPC card を使用するとどうなるかわかりませんが、僕の知る限りでは個々のチャンネルの微調整をしなくとも一律の p.e. にトリガーをかけることが出来ています。single MPPC の方が array 型のものより MPPC の個体差が小さいので多分使わずともうまく調整できるのではないかと。

7.3 Slow shaper fast shaper

Basically, the fast shaper is used to generate the trigger and measure the time.

The slow shaper is intended to measure the charge (thus it is more filtered in order to increase the S/N ratio). It is not intended to reduce pileup !

7.4 thresholdDAC

Q

Using 10-bit threshold DAC, The threshold voltage (V_{th}) is set as follows.

$$V_{\text{th}} = V_1 + (V_2 - V_1) \times \text{DAC}/1024$$

What is the value of V_1 and V_2 ?

I don't know the relationship between a threshold voltage and the value of 10-bit DAC.

A

Concerning the point 1, I join you a linearity measurement of the Threshold DAC :

$$V_{\text{out}} = A \times \text{DAC_Code} + B$$

You can see that the slope is $A = 2.22 \text{ mV/DAC unit}$ with an offset of about $B = 830 \text{ mV}$.