

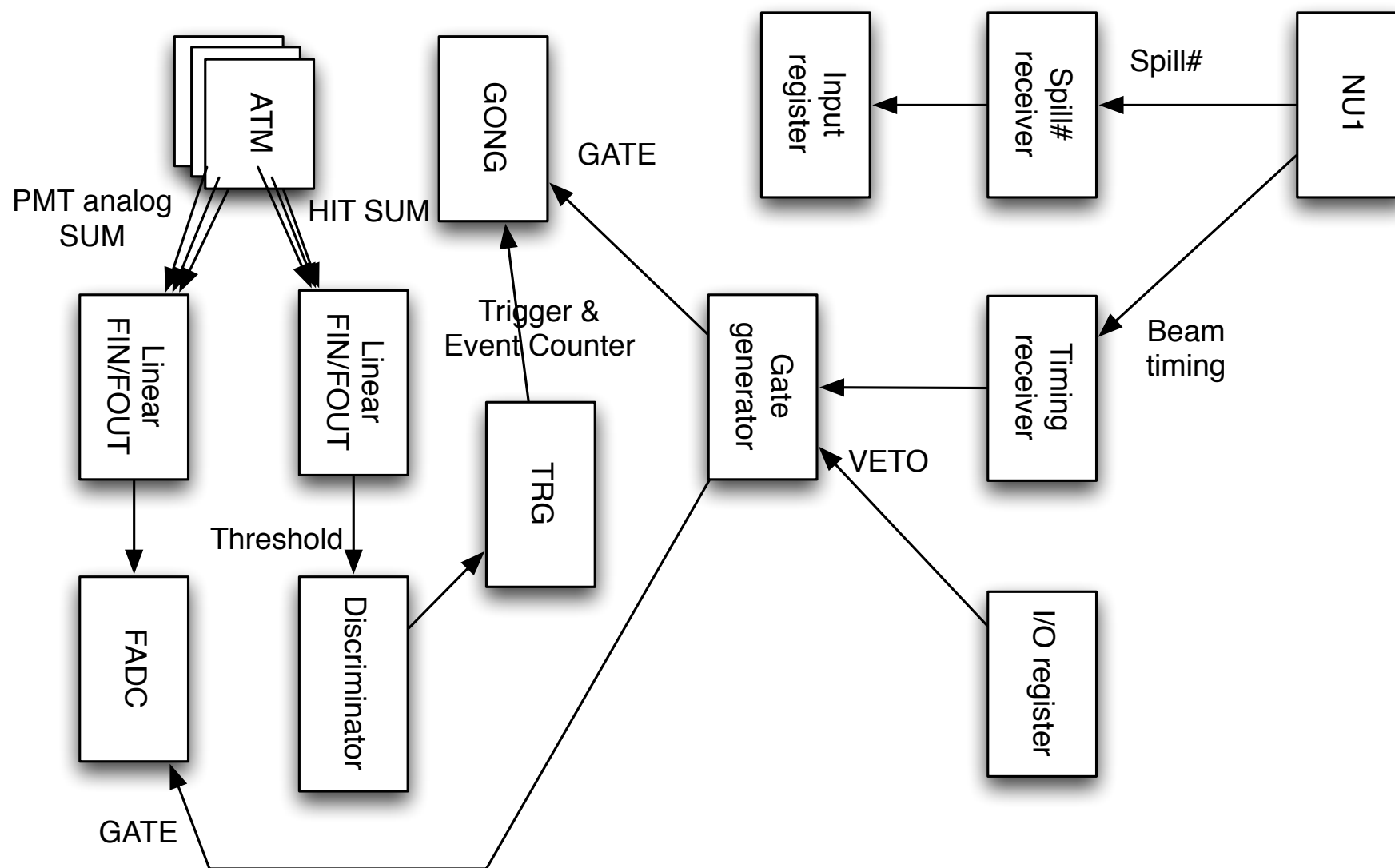
Mizuche DAQ

A.Murakami

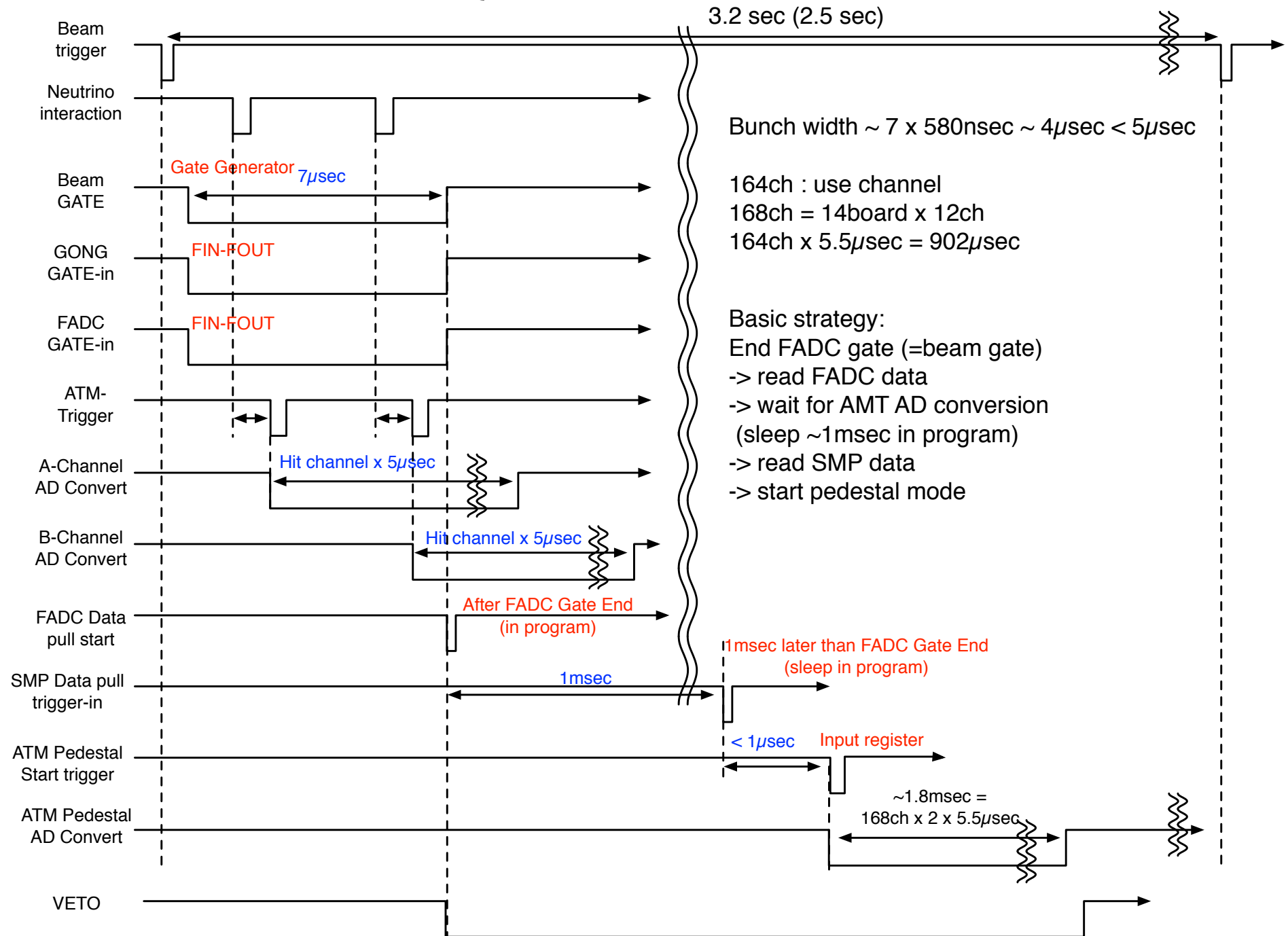
Contents

- DAQ system at Beam operation.
- DAQ system at Calibration.

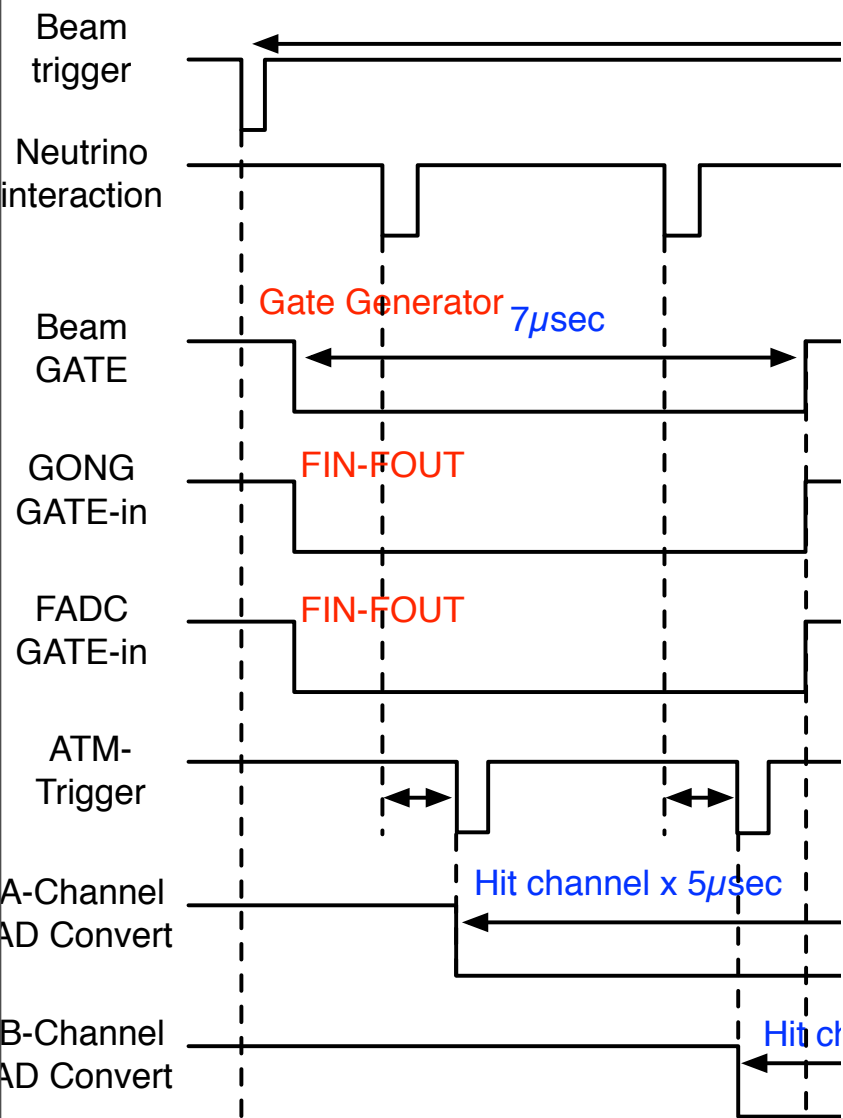
DAQ Overview (beam)



DAQ time chart



ATM beam data



- ATMのGate機能 : Gateが入っていないときの外部入力は無視→ $7\mu\text{sec}$ のBeam Gate.
- Neutrino反応によって、Thresholdを超えた信号が入力→HIT生成 (TDC Start)
- Thresholdに関しては後述.
- HITSUM($\sim 20\text{mV} \cdot 200\text{nsec} / \text{HIT}$) の和をFIN/FOUTで取り、ディスクリを通してxx HIT以上の場合にトリガー(AD convert start, TDC stop) を生成.
- xx HIT はMCでStudyして決定する.

- AD conversion には $\sim 5.5\mu\text{sec}$ / HIT
- ニュートリノ反応(CC反応)の場合、必ず2HIT以上 = $11\mu\text{sec}$ 以上かかる.
- どう頑張っても1スピン中に2回の反応までしかATMで測定できない。
- 3回以上の反応はPMTのアナログ信号の和をFADCで測定することで、統計的に処理.
- AD convert trigger はAD Clear (Start から $1.1\mu\text{sec}$) に間に合うように作成する.
- FIN/FOUT を駆使する方針.

ATM pedestal

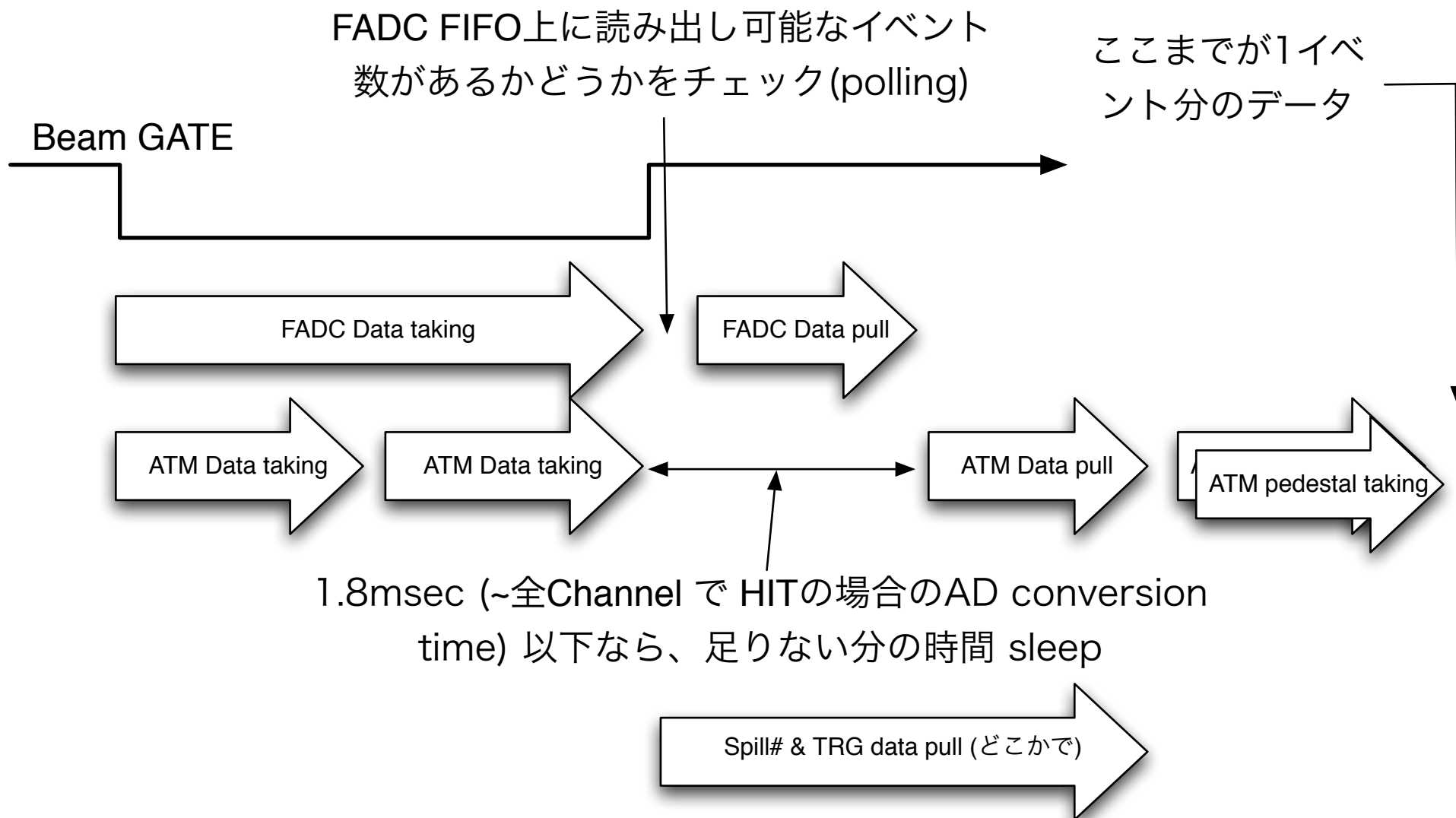
- ATM の pedestal mode を使用.
- 強制的に全channel のA/B系にHITを生成. (外部アナログ信号は入力されないものとする) → HIT Thresholdは最大値に.
- 外部からのトリガーと、プログラム上のスタート信号が必要.
- 外部トリガーは1回で良いっぽい? (この辺りまだ手探り) . スタート関数を読んだ分だけpedestal data が取れる.
- 各channelでA/B系のpedestalが必要→最低2回=1セット
- ビームトリガーの間(after beam data taking) に1セットの測定
→ 直前の beam data のペデスタルとして使用.

FADC

- 外部Gate(TTL signal)を入れている間だけData taking (sampling)を行う.
- Gate終了は、”読み出し可能なイベント数が少なくとも1つある”ことを引っかけ.
- それまで、pollingで register の値を見に行く.
- が、スマートじゃないのでもう少し練る.
- NIM module、I/O register を活用.

Data process

- 単純に1モジュールずつ処理する.



Data process

- データはとりあえず生データをそのまま記録
(1イベントずつ詰めていく)

Raw data structure of one event

- spill#
- event Count of TRG (and time stamp ?)
- FADC data (data size, sampling)
- signal ATM data
- pedestal ATM data

ATM data

- data size
- event count of GONG (16bit)
- event count (8bit), board#, ch#, A/B
- TDC, QDC

- オンライン解析(デコード・オンラインモニター)は
メインのDAQとは別プロセスでやるようにしたい。

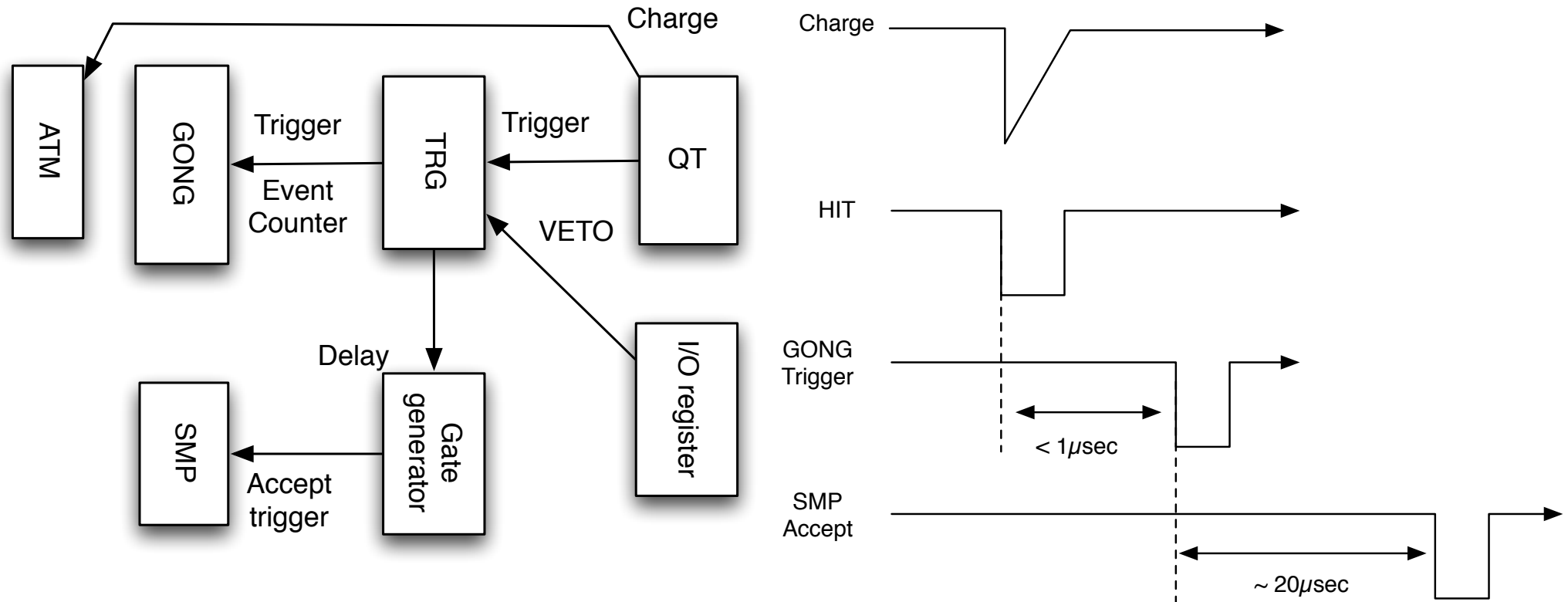
ToDo ...

- FIN/FOUTを用いたトリガー生成は未トライ.
- Input register の実装.
- とりあえず一連の動作をするプログラムはある（メイン関数にベタ書き）.
- 要長時間試運転、要改良.
- オンラインモニター、ランコントロール用の画面はまだ。

Calibration

- Threshold DAC の確認
- Pedestal QDC分布
- QDCリニアリティ

DAQ Overview (Ich用)



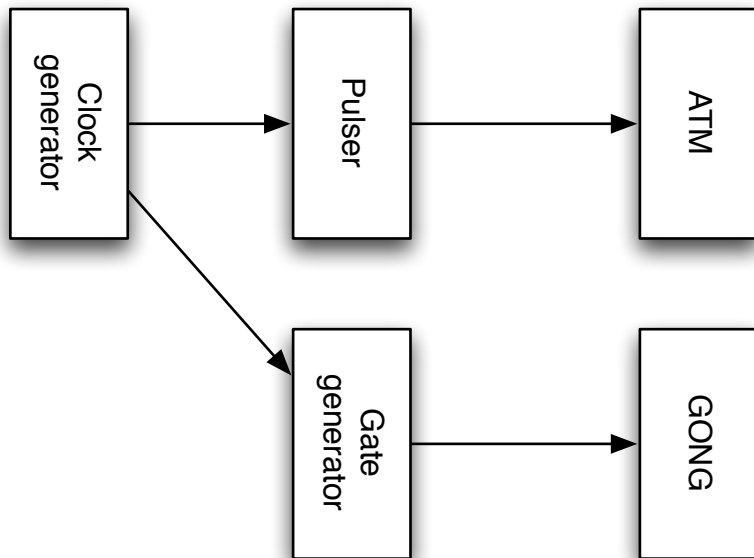
- Event counter によるイベント同期
- ATM → SMP のデータ転送は外部トリガー(Accept trigger)で開始
- AD conversion が十分に終了するタイミングで開始

Data taking

- SMP → PC のデータ転送はSMPからの割り込みを使用.
- SMPに一定数(20イベント)以上のイベントが貯まったら開始.
- Pedestal測定, Charge測定 はそれぞれランを分ける.
 - Charge : $10\text{pC} \sim 100\text{pC} = 200\text{p.e.}$ (gain: $3\text{e}6$) の間で 10pC 刻み.
 - Pedestal : 1回だけ → 各Charge測定の前に行うべきか ?
- Trigger rate : 100Hz (temporary)

ATM Threshold

- ATMのHITのThresholdはプログラム上で設定.
- 0 ~ -1240 mV : 入力信号を100倍した値と比べる.
- Thresholdの設定として、これで正しいかを確認
- パルサーとオシロを使用.



Pulser → ATM には LEMO-ECL(フラットケーブルコネクタ)の変換ケーブルを使用.

ATM Threshold

- Input pulse : -10mV の矩形波 (ch1:yellow)



FWHM:**20nsec**, raise:5nsec

Threshold DAC : -400mV~-1000mV

→ HIT (10trigger/10trigger)

Threshold : -1100mV

→ No HIT

入力信号 × 100倍 と比較.



FWHM:**10nsec**, raise:5nsec

Threshold DAC : -400mV~-500mV

→ HIT (10trigger/10trigger)

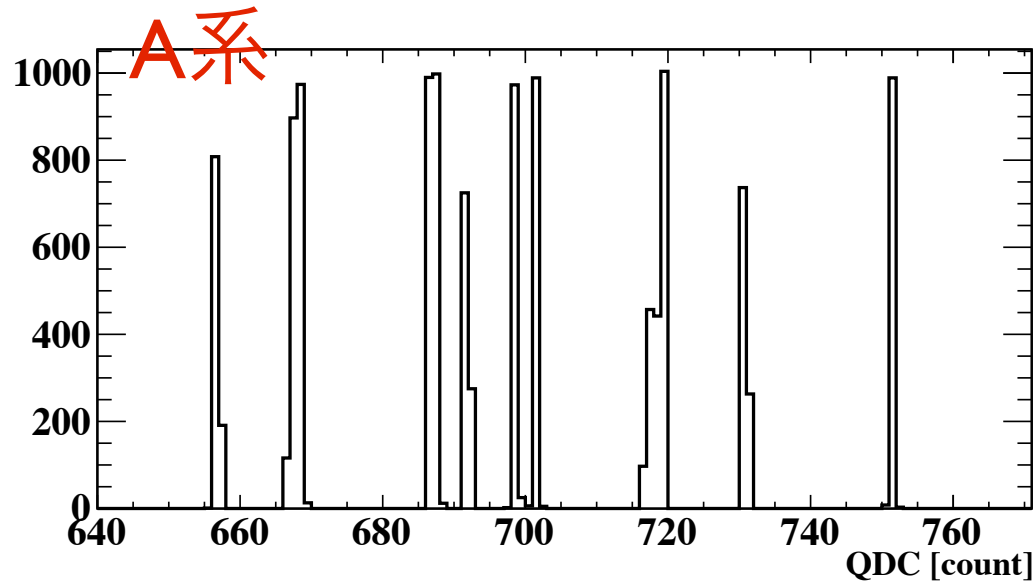
Threshold : -800mV

→ Sometimes (3trigger/10trigger)

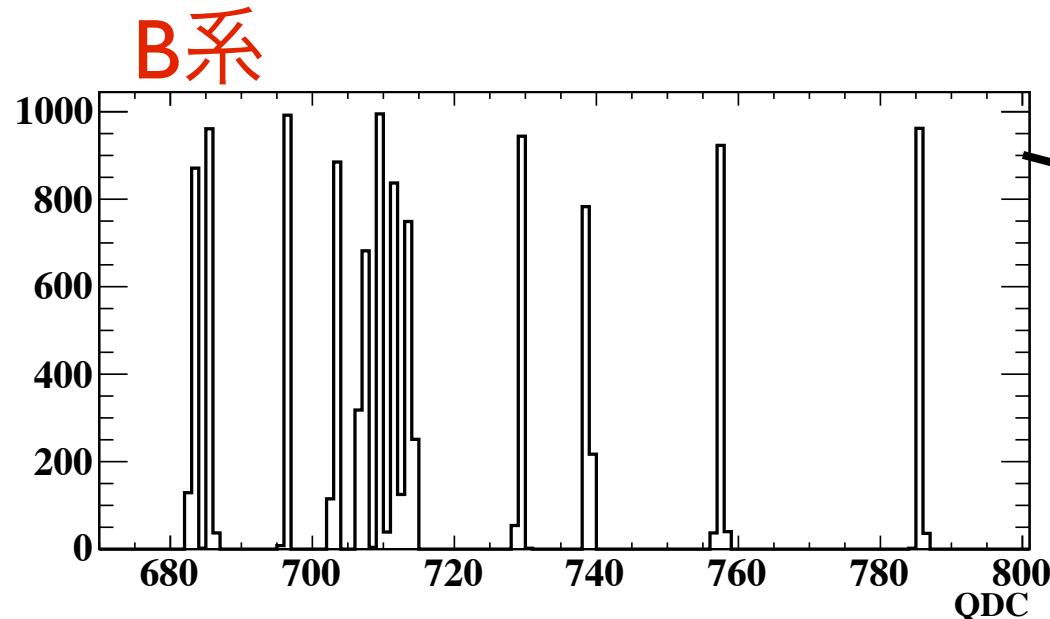
パルス幅>10nsecが必要.

Pedestal 測定

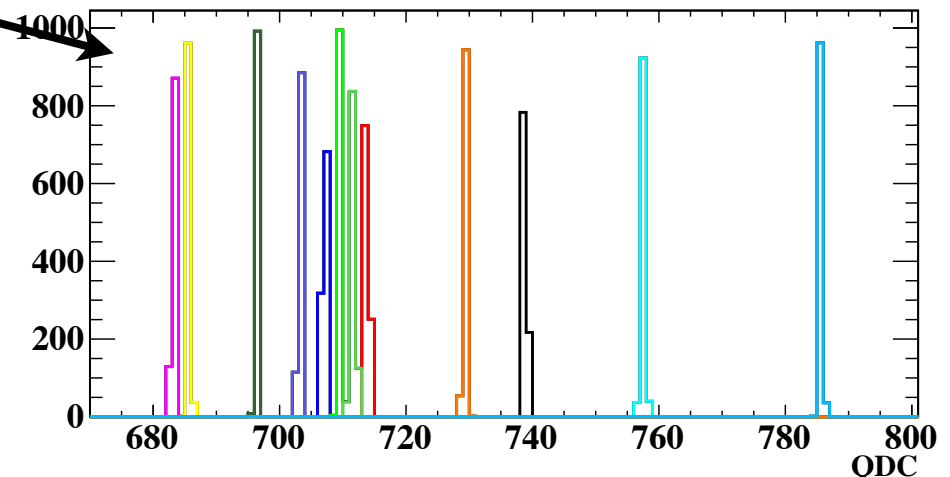
- Board#1 についてのみ. 各channel 1000 events



1 board, 12 channel の Pedestal
各channelで pedestalの平均値は違うが、各チャンネル安定している
→長時間測定でどうなるか(ToDo)



Channel毎に色分け

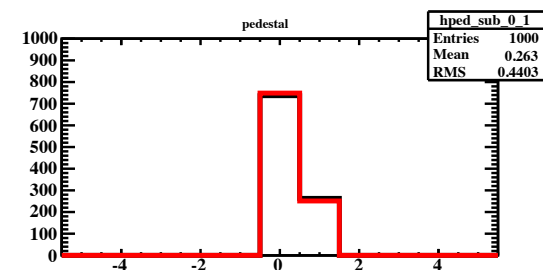
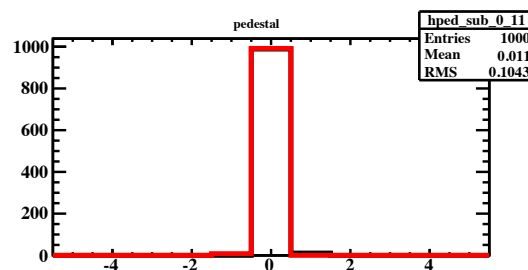
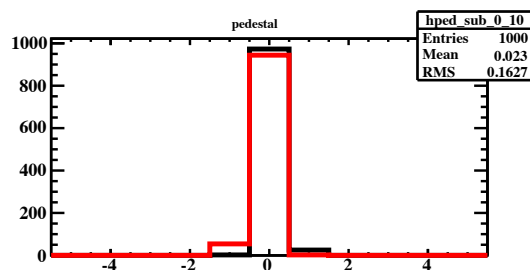
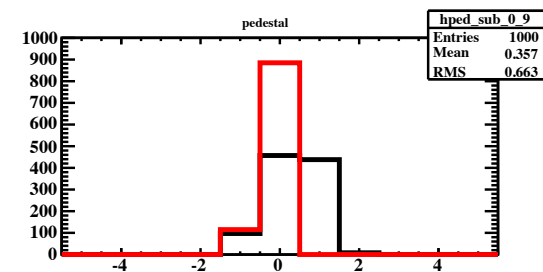
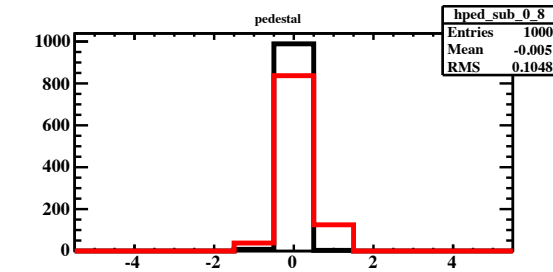
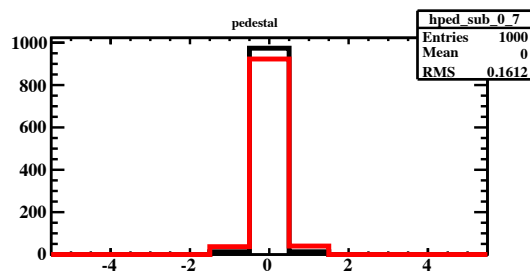
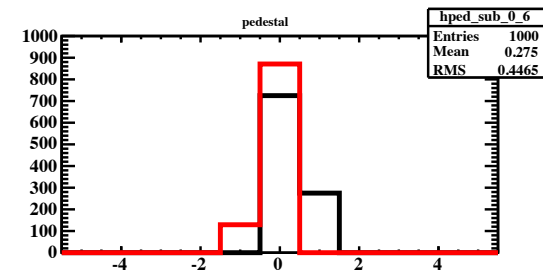
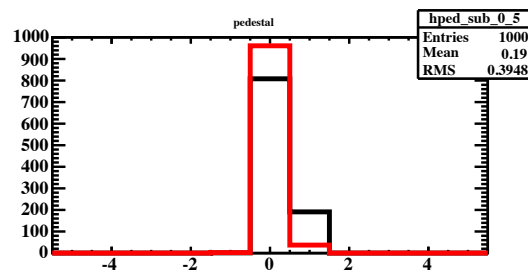
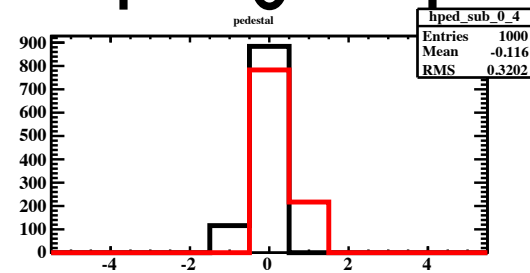
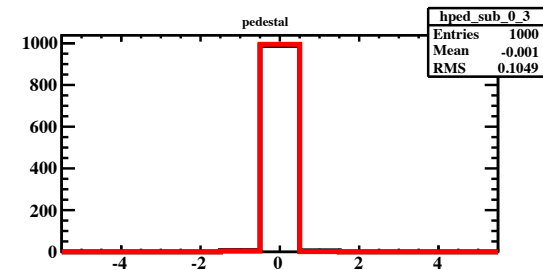
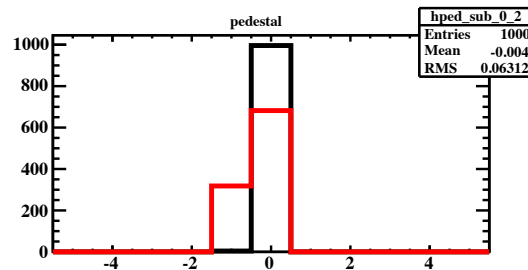
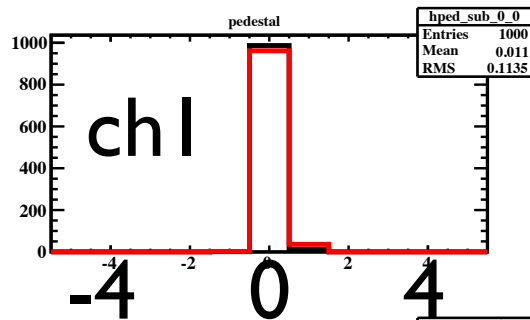


Pedestal 測定

横軸:QDC [count]

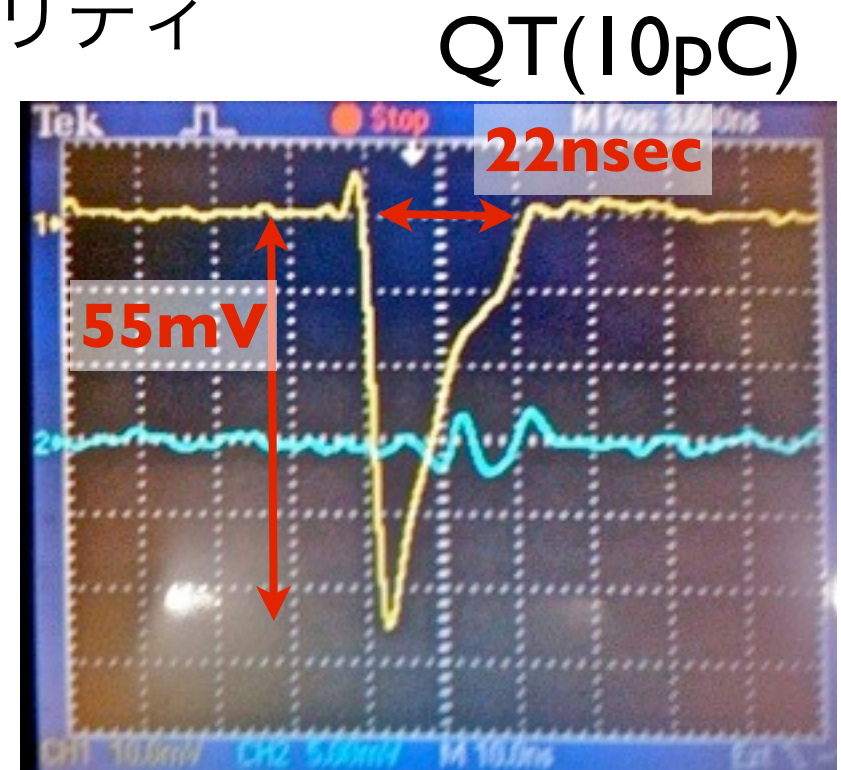
黒:A系、赤:B系

平均値からのばらつきは1count程度.



Charge測定

- 試しに 10pC, 50pC, 100pC のQを測定.
 - ある1channelに対して各Chargeで2000イベント.
- TRGからのEvent counterによる同期
- Chargeに対するADCのリニアリティ
 - QTは新品の物を使用.
- HIT Threshold : 500mV に設定.
→ QTの出力(10pC)を積分すると、10.9pC (オシロのサンプリングデータを使用)



TRG vs Event counter

TRGで記録した数とGONG,ATMでの数を比較

→ 同じイベントでズれていないかを確認.

```
root [22] mizu_p1->Scan("trg:gong", "(trg+1)-gong!=0")
```

```
*****
```

```
*      Row      *          trg *          gong *
```

```
*****
```

```
*****
```

```
==> 0 selected entries
```

```
(Long64_t)0
```

```
root [19] mizu_p1->Scan("trg:etag", "((trg+1)&0x00FF)-etag!=0")
```

```
*****
```

```
*      Row      * Instance *          trg *          etag *
```

```
*****
```

```
*****
```

```
==> 0 selected entries
```

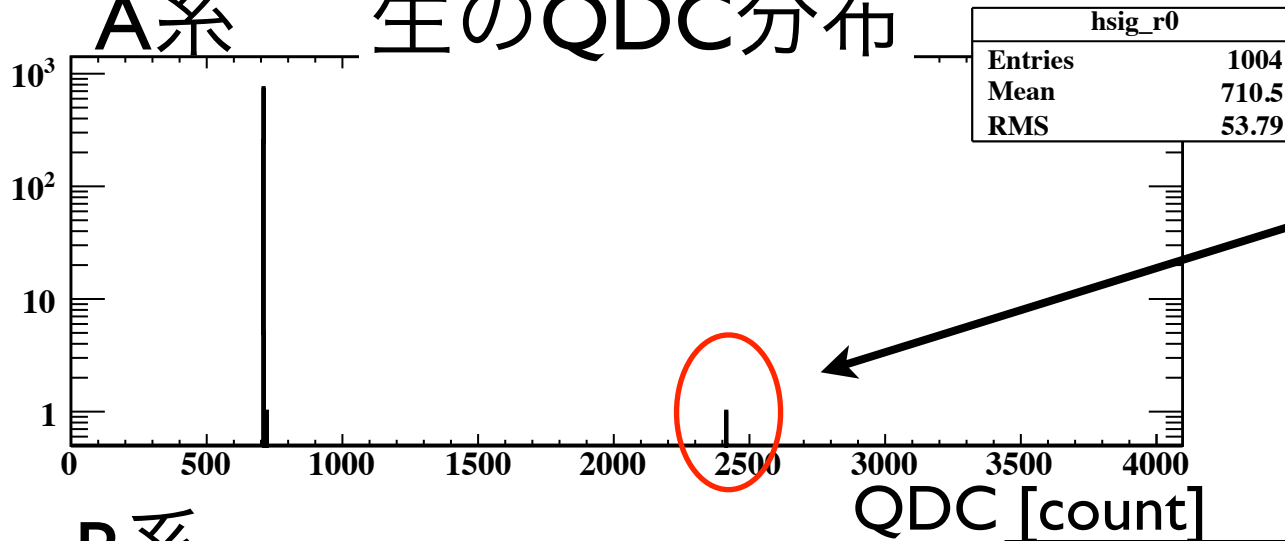
```
(Long64_t)0
```

etag :ATM' event counter

→ Event counterによる同期は取れている.

Charge測定 (10pC)

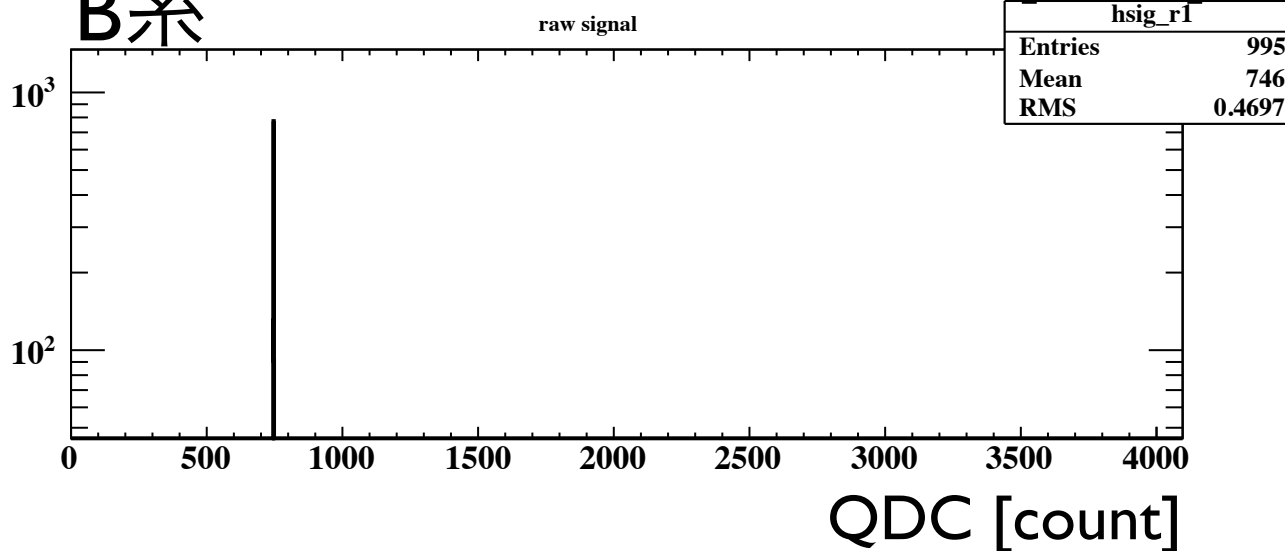
A系 生のQDC分布



1イベントだけ変に高い
→ 他のチャージでもほ
ぼ位置に出る.

→ 必ず一番最初のイベ
ント.

B系



Charge測定

- 最初のイベントにHITがない (今回のCharge測定では必ず起こっている) → Pedestal測定では起きていない。

以下 10pC の時のデコード結果

```
0  0x000001d8 = h:0x0000 (m:0,s:0), l:0x01d8 (t:0) -> Data size of this entry : 472 [bytes]

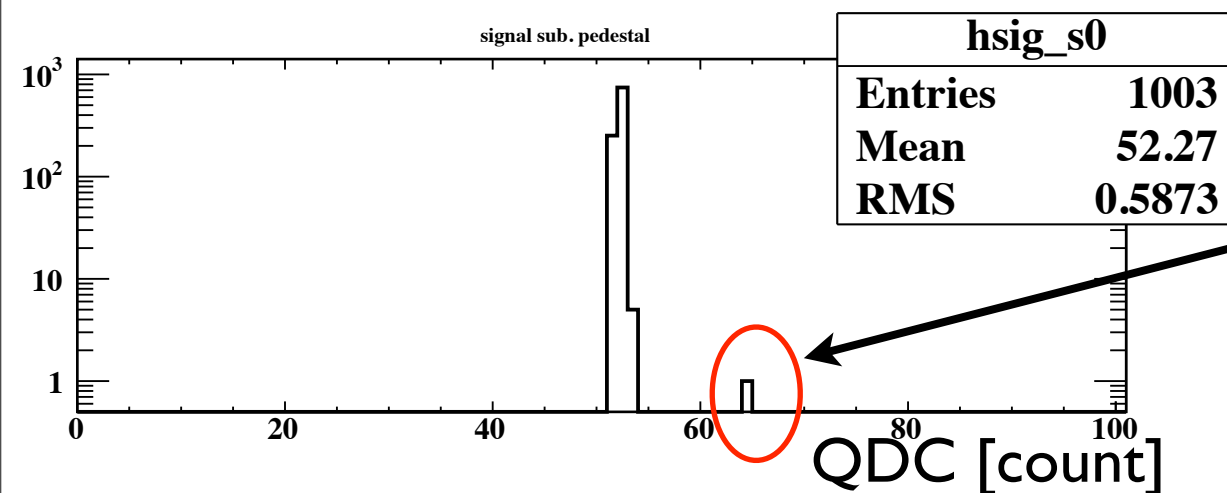
1  0x00010002 = h:0x0001 (m:0,s:1), l:0x0002 (t:0) -> Serial#: 1 word# / event : 2
2  0x00000001 = h:0x0000 (m:0,s:0), l:0x0001 (t:0) -> Event count : 1
3  0xff000000 = h:0xff00 (m:31,s:1792), l:0x0000 (t:0) -> module down bit (end of event)

4  0x00020005 = h:0x0002 (m:0,s:2), l:0x0005 (t:0) -> Serial#: 2 word# / event : 5
5  0x00000002 = h:0x0000 (m:0,s:0), l:0x0002 (t:0) -> Event count : 2
6  0x08000085 = h:0x0800 (m:1,s:0), l:0x0085 (t:0) -> Event#:2, type:0 ch:5
7  0x0801473f = h:0x0801 (m:1,s:1), l:0x473f (t:1) -> TDC : 1855
8  0x0802896e = h:0x0802 (m:1,s:2), l:0x896e (t:2) -> QDC : 2414
9  0xff000000 = h:0xff00 (m:31,s:1792), l:0x0000 (t:0) -> module down bit (end of event)

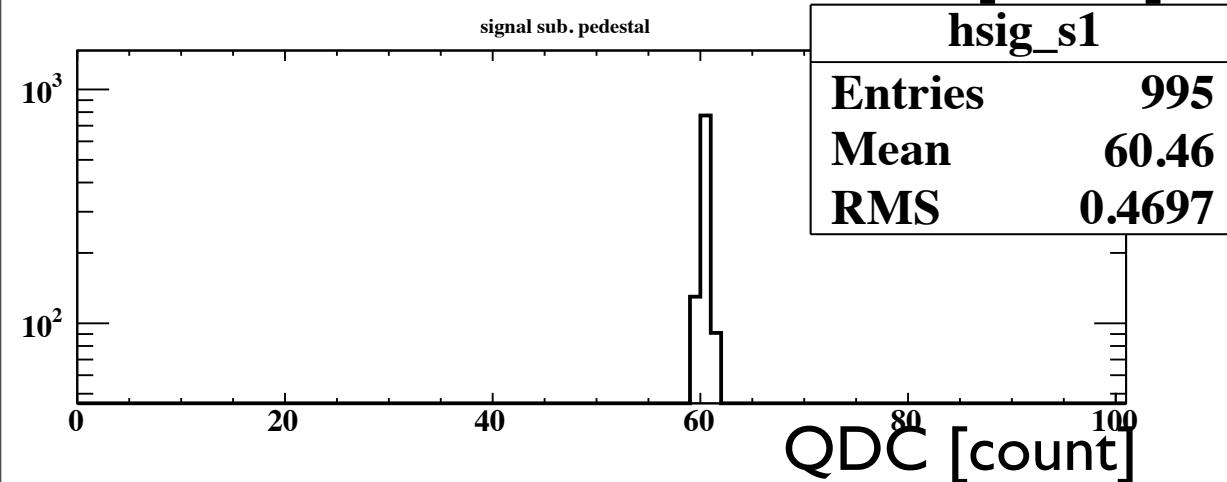
10 0x00030005 = h:0x0003 (m:0,s:3), l:0x0005 (t:0) -> Serial#: 3 word# / event : 5
11 0x00000003 = h:0x0000 (m:0,s:0), l:0x0003 (t:0) -> Event count : 3
12 0x080000c5 = h:0x0800 (m:1,s:0), l:0x00c5 (t:0) -> Event#:3, type:0 ch:5
13 0x08014744 = h:0x0801 (m:1,s:1), l:0x4744 (t:1) -> TDC : 1860
14 0x080282d1 = h:0x0802 (m:1,s:2), l:0x82d1 (t:2) -> QDC : 721
15 0xff000000 = h:0xff00 (m:31,s:1792), l:0x0000 (t:0) -> module down bit (end of event)
```

Charge測定(10pC)

- 問題のなさそうな2 event以降のデータを使用.
- Pedestalを引く際には、先ほどの平均値を使用.

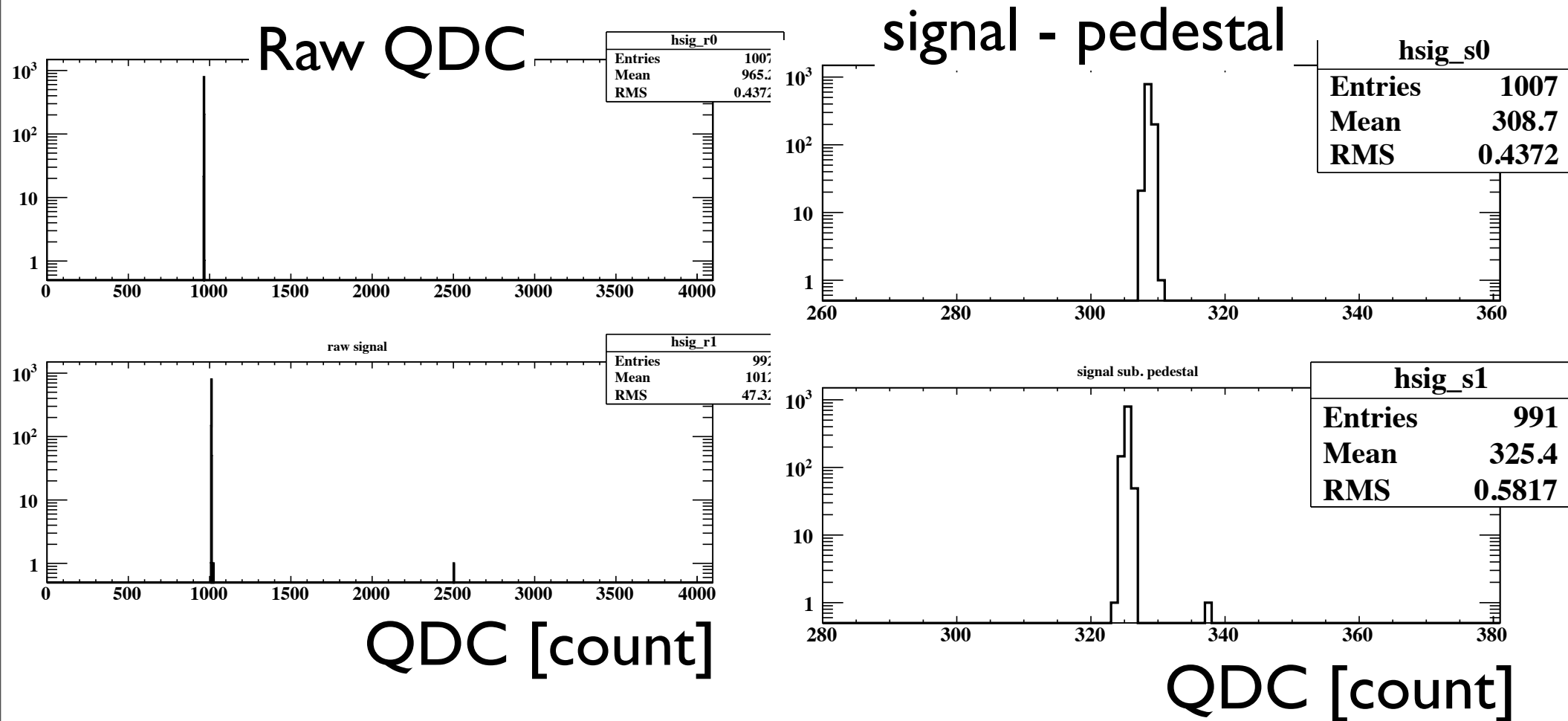


若干高い→ 必ず2event目

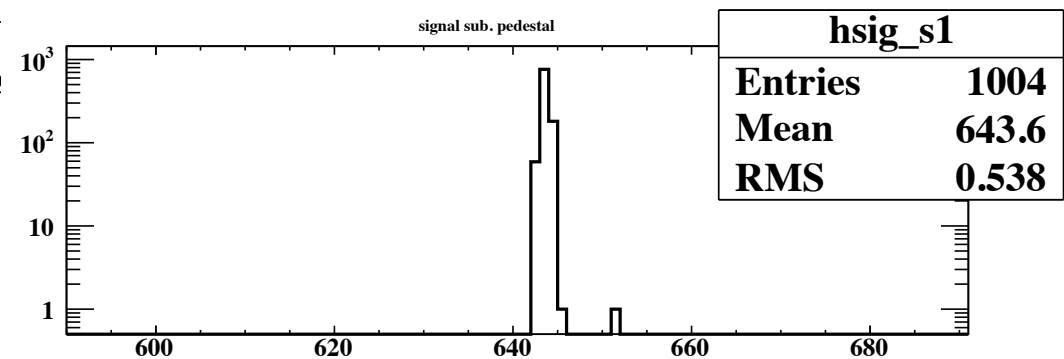
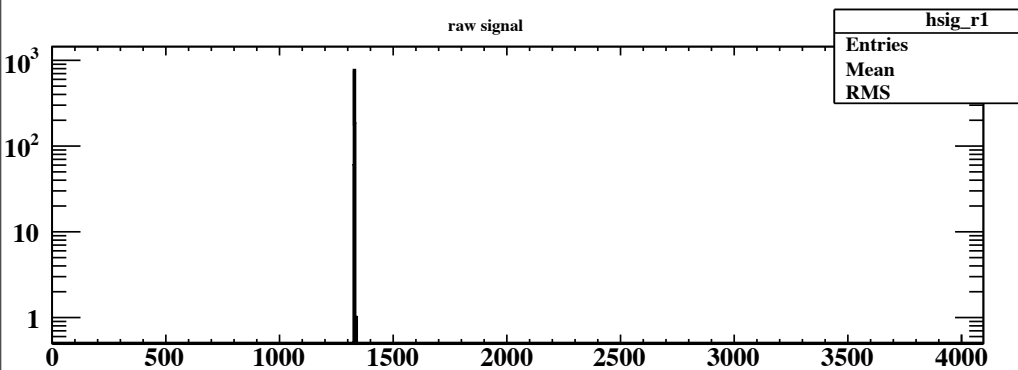
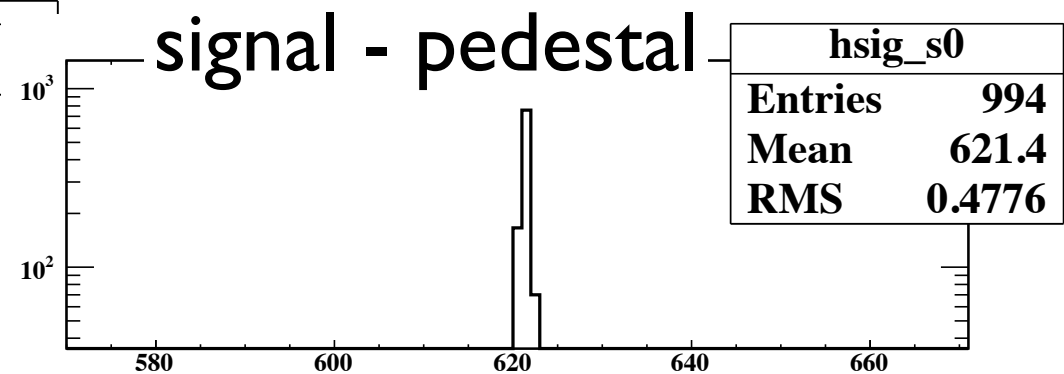
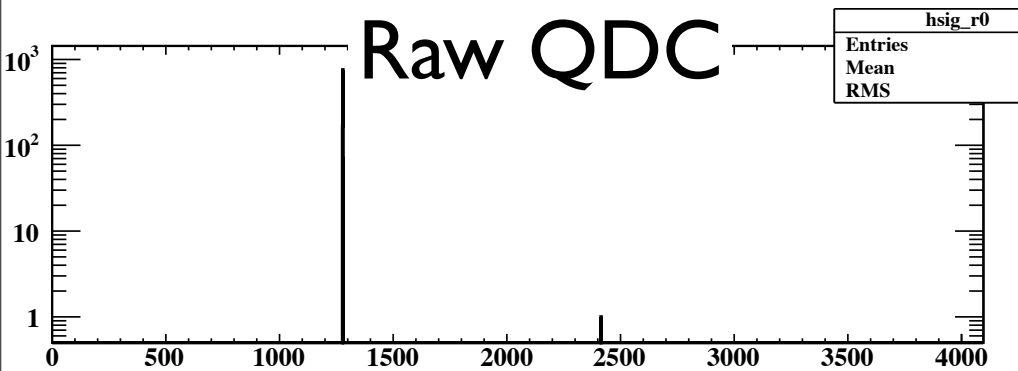


A/Bでイベント数が違う
→ Signalが入ってきたときに、どちらともREADYであればどちらが使われるかは決まっていない.

Charge測定(50pC)



Charge測定(100pC)



QDC [count]

QDC [count]

Result & ToDo

I QDC count = 0.1 pC

Input Charge [pC]	A-QDC mean [pC]	B-QDC mean [pC]
10	5.23	6.05
50	30.9	32.5
100	62.1	64.4

I イベント目にHITがない問題

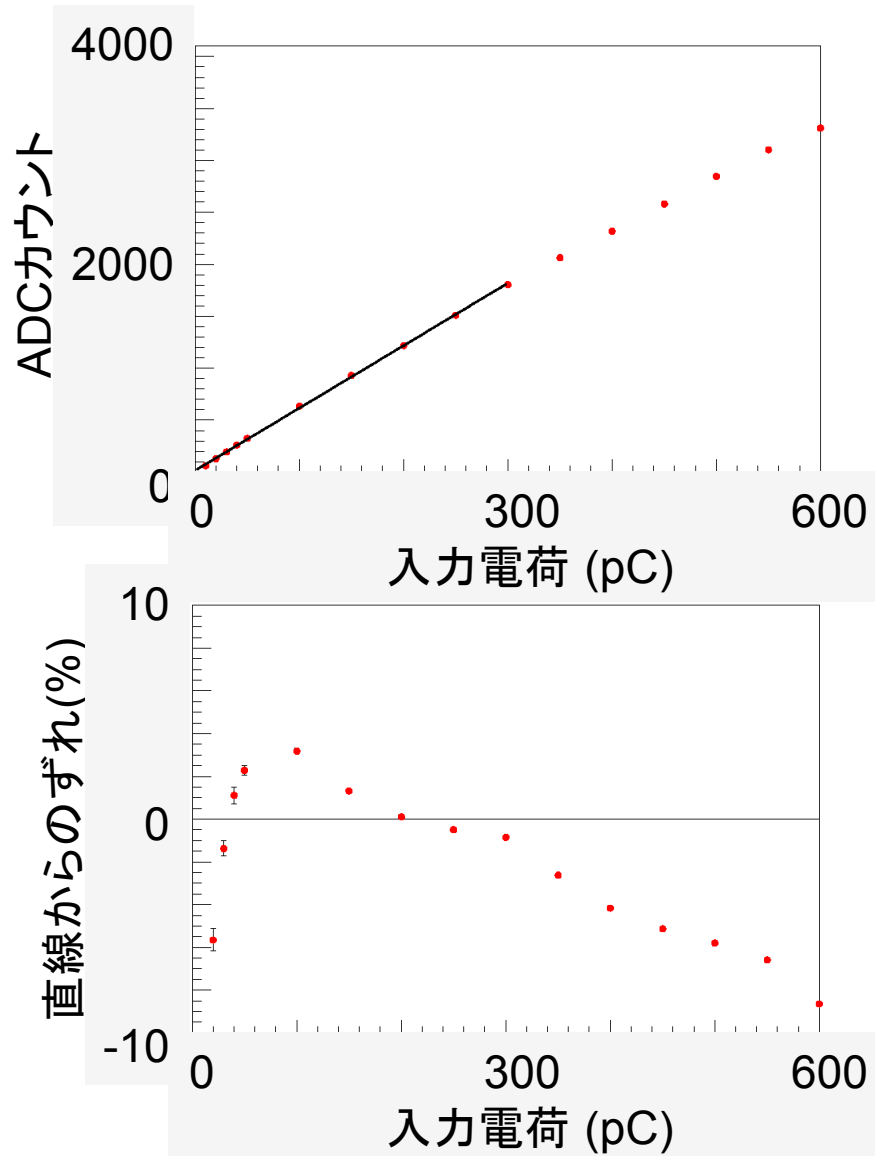
+ チャージがおかしい問題 の解決

→ プログラムの見直し

→ QTの安定性のチェック → FADCを使ってチェック.

X-MASSのCalibration (平出)

ADCリニアリティの測定



ATMへの入力電荷量を0~600pCまで変化させて、ADCの応答を測定する。

300pC以下を直線でフィットし、直線からのずれをみた。

- 500pCで6%
- 電荷量が小さいところでもずれる

➔ 実際の解析では、入力電荷量とADCカウントのルックアップテーブルを作成する