

Purpose: to learn the lowest limit of LVDS current below 6mA for the safe operation.

23年5月5日LVDS Scanを行った際のintt1データをもとに決定する。

測定項目

- 1 bad channelの数
- 2 half entryの数

から LVDS currentを設定する値を決定する。

現在は6mAに設定されている。

→今回の結果では2mAもOKだと考えている。

1 bad channelの数

データについて

```
B1L105S_bad_ch.txt
ファイル 編集 表示

Total 3
U1 0
U2 0
U3 0
U4 0
U5 1
U6 0
U7 1
U8 0
U9 0
U10 0
U11 0
U12 0
U13 0
U14 0
U15 1
U16 0
U17 0
U18 0
U19 0
U20 0
U21 0
U22 0
U23 0
U24 0
U25 0
U26 0
```

The entry of the "*_ch_property.root" is 3328 corresponding to the number of channel one half-ladder has (128 * 26). The entry of the "*_bad_ch.root" is given by the number of channels that failed the first 4 bits check.

Channel profile [\[edit\]](#) [\[edit source\]](#)

Each channel has it's own channel-property. It's just like the ID card. Currently the channel property has 8 digits, each digit represents one status of that channel.

```
1 -> channel entry, if the channel entry is within the criteria range -> 1
10 -> channel entry with an ampl cut, if the value is within the criteria range -> 1
100 -> threshold position, if the value is within the criteria range -> 1
1000 -> ampl dist turn on width, if the value is within the criteria range -> 1
1 0000 -> channel half-entry indication (classification not considered) if the channel has only half entry -> 1
10 0000 -> empty channel indication (classification not considered) if entry < 10. -> 1
100 0000 -> double entry indication (classification not considered) if entry is doubled -> 1
1000 0000 -> Noisy channel indication (classification not considered) if entry > 1000. -> 1
```

So for example, 00011100 -> If you see a channel with such profile, which means its threshold position and turn width are good, but the entry is only half -> It's a typical half-entry channel.

Example 01001100, its threshold position and turn width are good but the channel entry is doubled, it's known due to the problem of the optic-coupler, for detail, please check the INTT elog slot 20.

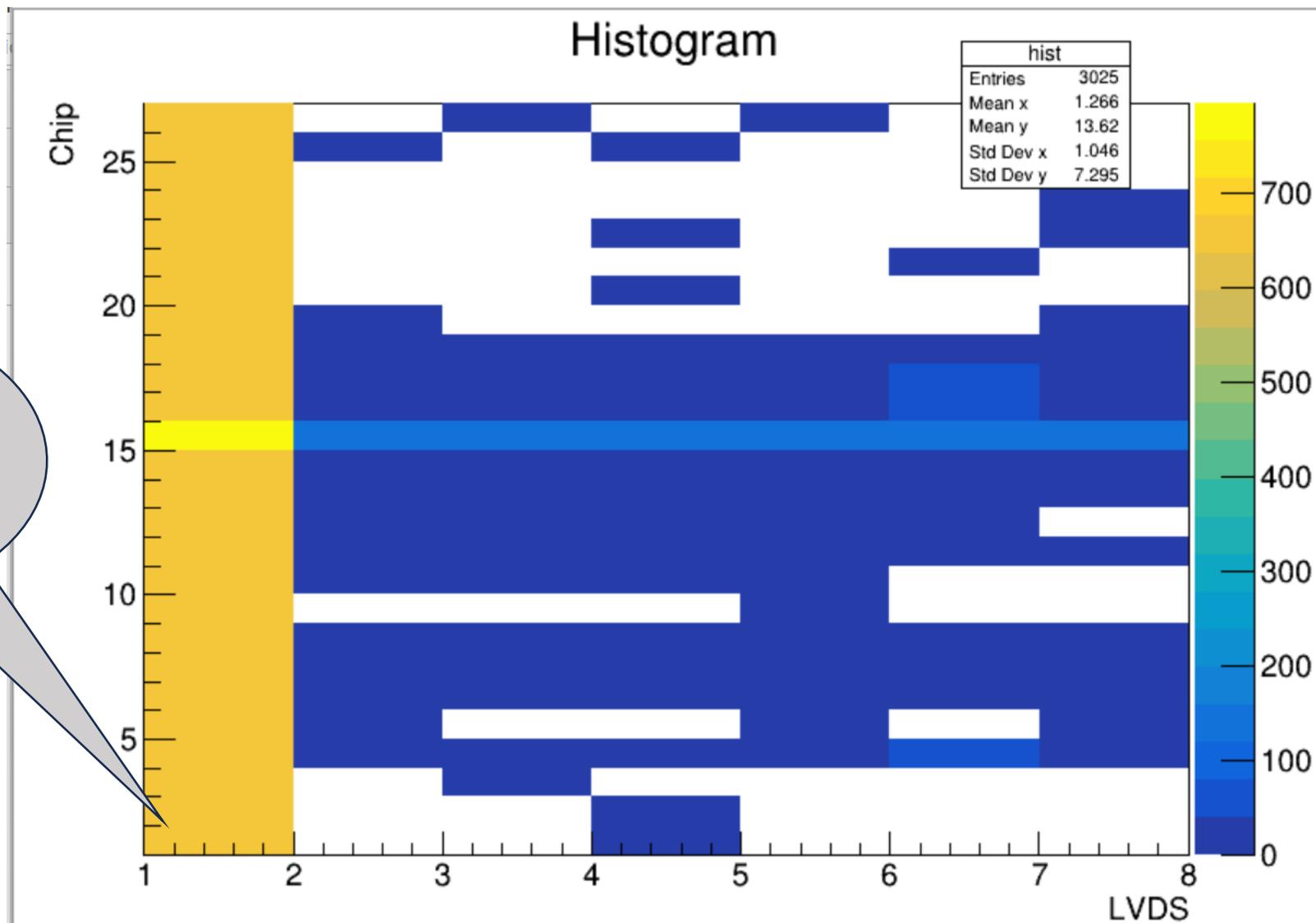
Example 10000000 -> Noisy channel

Example 00100000 -> empty channel or channel entry less than 10.

- ハーフエントリーか否かの情報は含まれていない。
- Ampl<threshold or Ampl distribution(ノイズによって惑わされる)<criticalでもカウントされる。

LVDSによるBad ch数を見る。

全ラダーを組み合わせたもの



レギュレーター交換によって改善されている。

- レギュレーター交換前 chipによる依存性があった。

- 交換後 Chipによる依存性なし

- レギュレーター交換前のテスト

210324_LVDS_Scan.xlsx

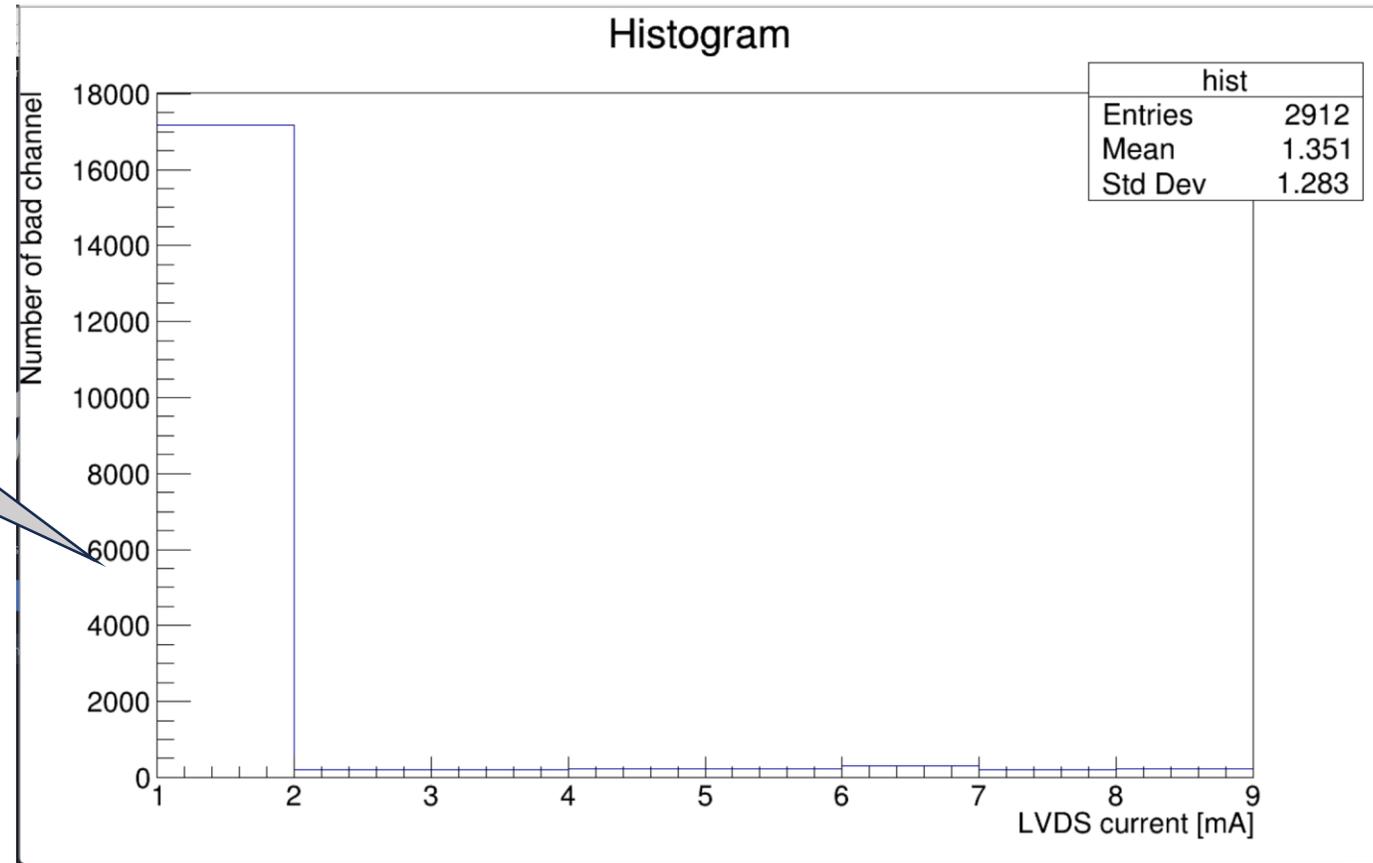
| LVDS Current | | Calibration Results | | | | | | | | | | | | | | | | | | | | | | | | Data | | |
|--------------|-------------|---------------------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|------|
| [mA] | GUI Setting | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | |
| 8mA | 255 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 1738 |
| 7mA | 127 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 1751 |
| 6mA | 63 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 1755 |
| 5mA | 31 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 1800 |
| 4mA | 15 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 1807 |
| 3mA | 7 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 1813 |
| 2mA | 3 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 1818 |
| 1mA | 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 1824 |
| 1mA | 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 1829 |

Legend: Good (Green), Half Entry (Cyan), Empty (Red)

This measurement was done in NWU before we upgrade regulators for FPHX power on ROCs.

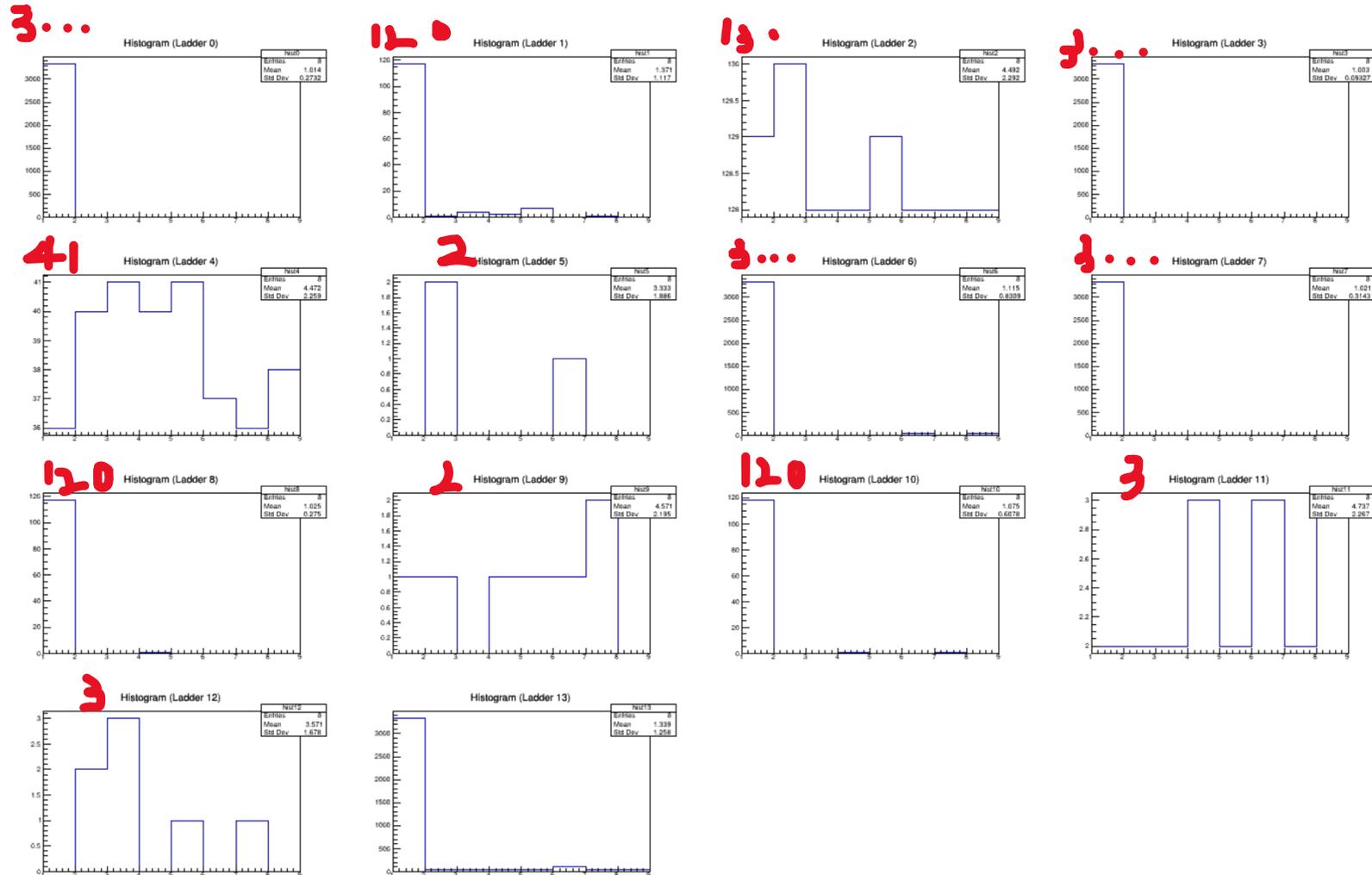
全ラダーを足したプロット

LVDS=1mA
だけbad ch数
が多い



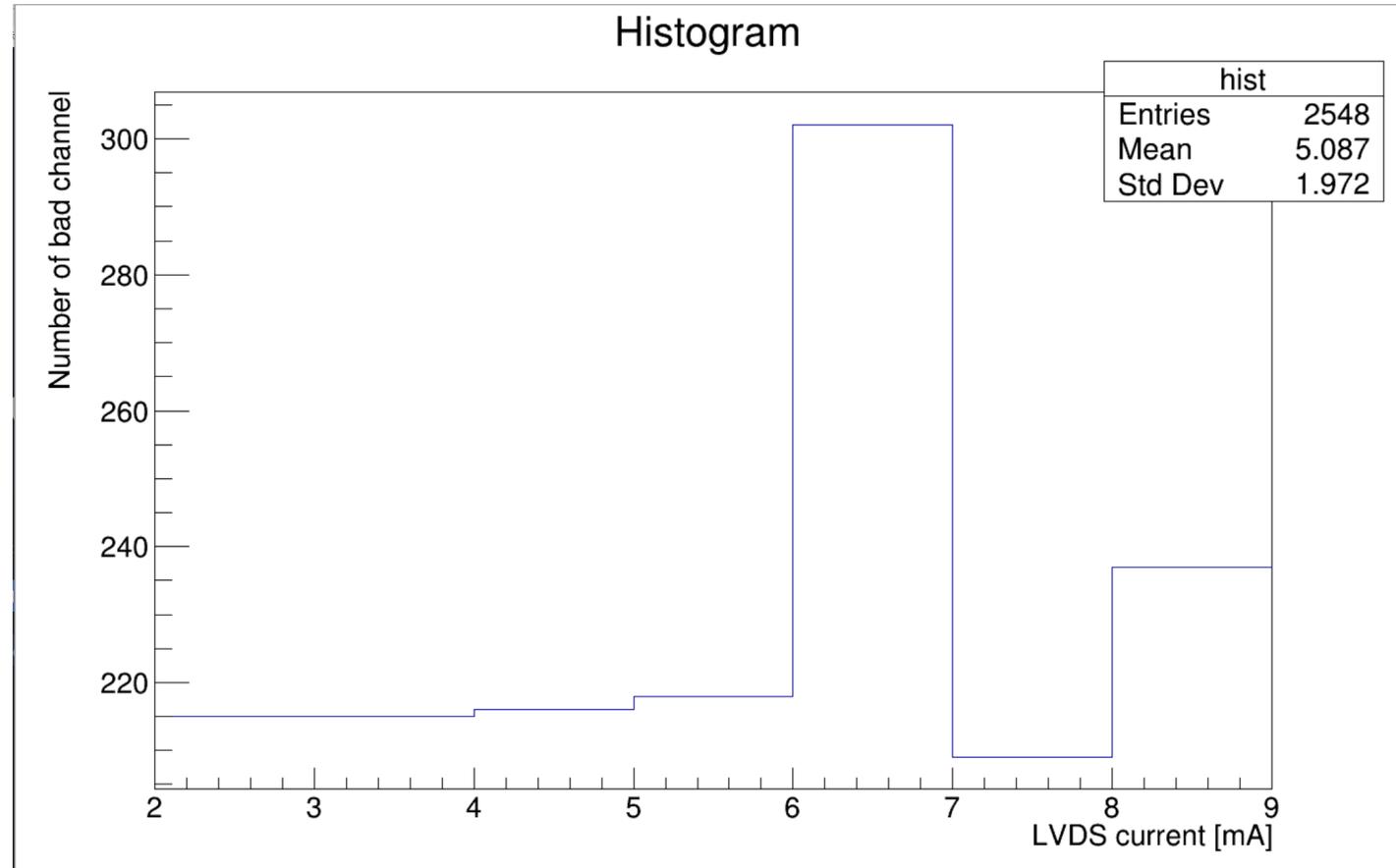
各ラダーごとにプロットし直した。

- intt1



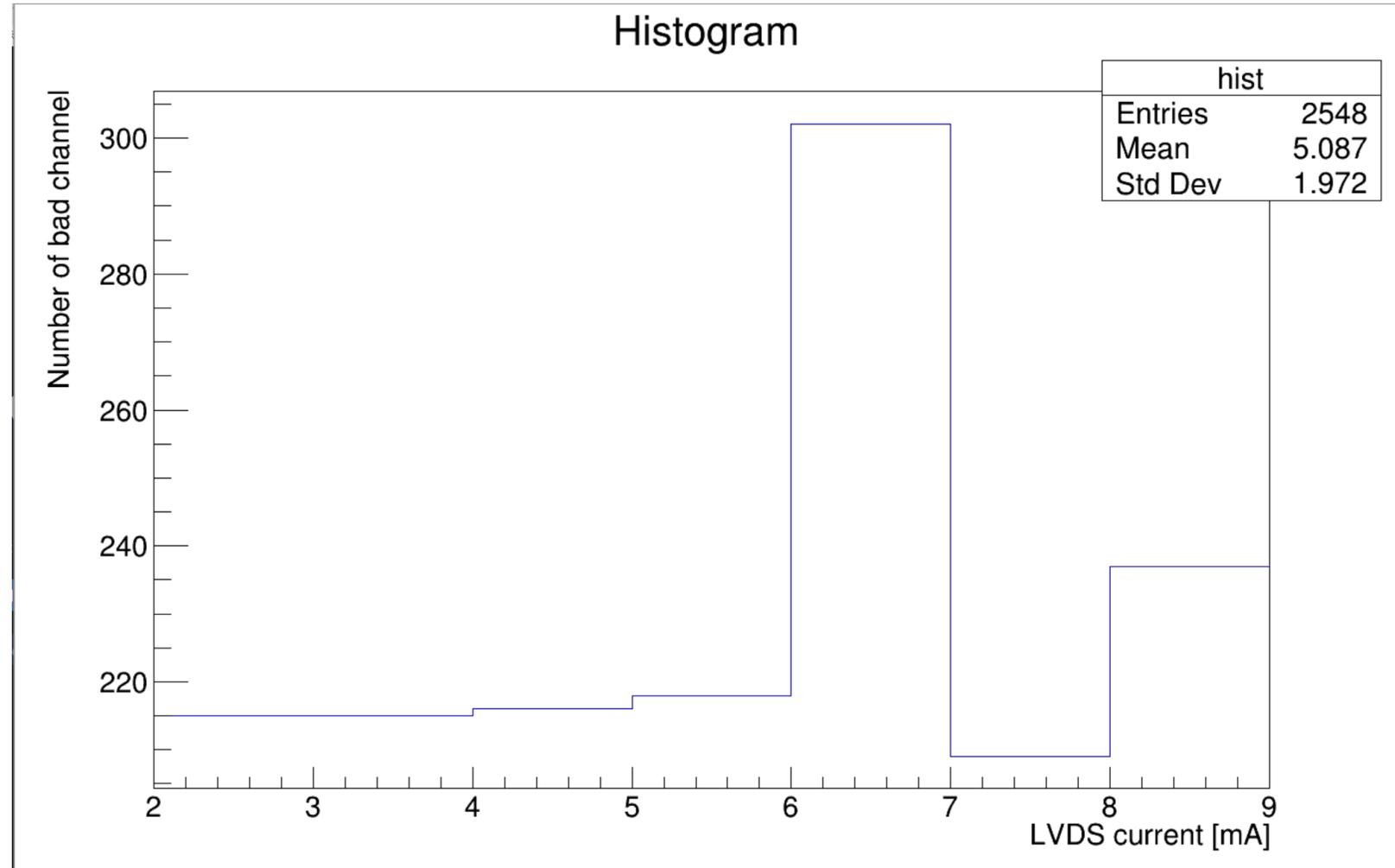
LVDS $\geq 2\text{mA}$

- Intt1のデータ
- chの数が正しくプロットされていることを確認した。
- →方法は、LVDS=6mAでtextファイルを手計算で足し合わせた。ものと比較。同じ値に



LVDS $\geq 2\text{mA}$

- Intt1のデータ
- LVDS $\geq 2\text{mA}$ では、相関関係がない。
- Bad 数から言えることは、LVDS=2mAでも良い。



どうやってハーフエントリーの数を調べる？

- 1 ch_property.root と tree を使う
- 2 tree名はch_property
- 3 profile は Half entry数
- 4 Profile(integer) をconvert binaryする。
- 5 binaryからhalfかをはんだんする。

The ch_property tree contains :

```
*ROC ID
*Ladder name (direction included)
*Felix channel
*Port ID
*chip_id
*chan_id
*ch_entry
*ch_entry_ampl_cut
*turn_on
*width
*profile (It's decimal (Integer), convert to binary in C++ std::bitset<8>(profile), in Python3 bin(profile)), the introduction of the profile is shown above.
*adc0_width
*adc1_width
*adc2_width
```

Channel profile [\[edit\]](#) [\[edit source\]](#)

Each channel has its own channel-property. It's just like the ID card. Currently the channel property has 8 digits, each digit represents one status of that channel.

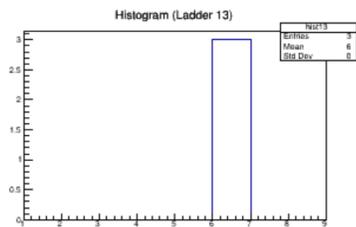
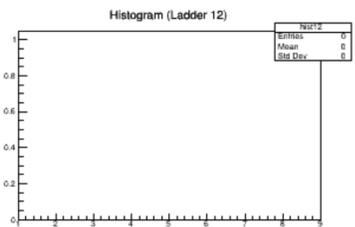
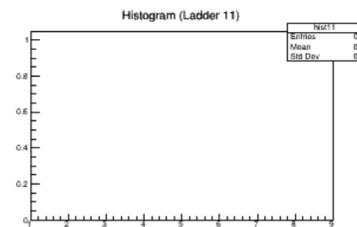
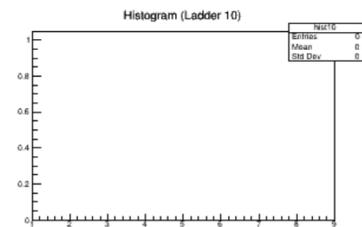
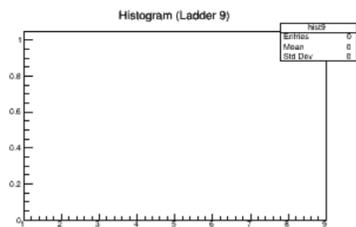
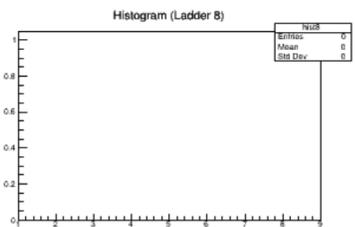
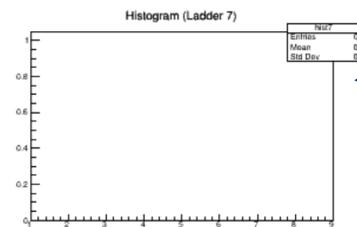
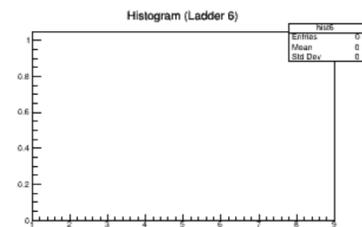
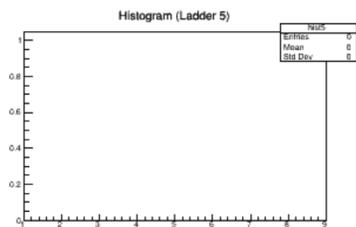
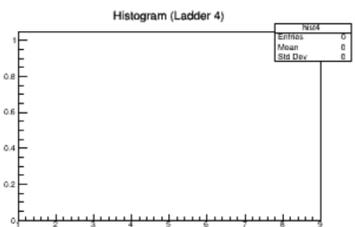
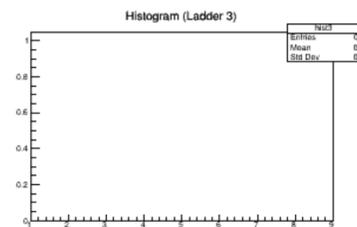
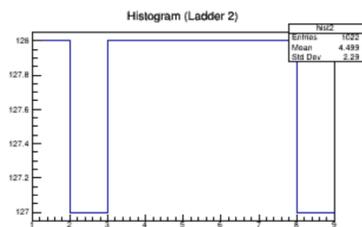
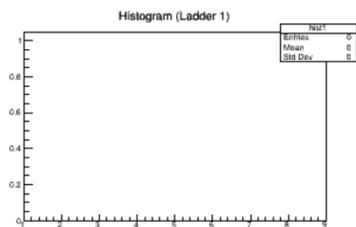
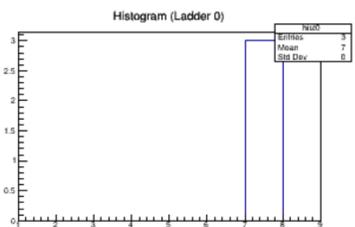
```
1 -> channel entry, if the channel entry is within the criteria range -> 1
10 -> channel entry with an ampl cut, if the value is within the criteria range -> 1
100 -> threshold position, if the value is within the criteria range -> 1
1000 -> ampl dist turn on width, if the value is within the criteria range -> 1
10000 -> channel half-entry indication (classification not considered) if the channel has only half entry -> 1
10 0000 -> empty channel indication (classification not considered) if entry < 10. -> 1
100 0000 -> double entry indication (classification not considered) if entry is doubled -> 1
1000 0000 -> Noisy channel indication (classification not considered) if entry > 1000. -> 1
```

So for example, 00011100 -> If you see a channel with such profile, which means its threshold position and turn width are good, but the entry is only half -> It's a typical half-entry channel.

自作treeに入れてみた。

- intt1の全データを自作treeに入る。
- 今回はhalf entryのときのみ、Treeにfillした。

```
tree->getEntry(1);
binary = std::bitset<8>(profile);
if (binary[4])//左が0
{
std::cout <<"p="<<profile <<"chip="<<chip_id<<"chan="<<
std::cout << "The 4th bit is 1." << std::endl;
std::cout <<binary <<std::endl;
LVDS = extractedDigits1[number_of_LVSD];
ladder = extractedDigits2[number_of_ladder];
harf=1;
ch=chan_id;
chip=chip_id;
tree->Fill();
}
}file2->Close();
}
```



Half entryは
ladder 3つにしか
ない。

LVDS currentは
関係ない。

またLadder2 Chip15はhalf en

- 完成したrootファイルを見たら、
- LVDS currentによらずladder2 chip15がprofile=28
- つまりhalf entry

```
*****
* Row * LVDS.LVDS * ladder.la * profile.p * chip.chip * ch.ch * harf.harf *
*****
* 0 * 2 * 2 * 28 * 15 * 1 * 1 *
* 1 * 2 * 2 * 28 * 15 * 2 * 1 *
* 2 * 2 * 2 * 28 * 15 * 3 * 1 *
* 3 * 2 * 2 * 28 * 15 * 4 * 1 *
* 4 * 2 * 2 * 28 * 15 * 5 * 1 *
* 5 * 2 * 2 * 28 * 15 * 6 * 1 *
* 6 * 2 * 2 * 28 * 15 * 7 * 1 *
* 7 * 2 * 2 * 28 * 15 * 8 * 1 *
* 8 * 2 * 2 * 28 * 15 * 9 * 1 *
* 9 * 2 * 2 * 28 * 15 * 10 * 1 *
* 10 * 2 * 2 * 28 * 15 * 11 * 1 *
* 11 * 2 * 2 * 28 * 15 * 12 * 1 *
* 12 * 2 * 2 * 28 * 15 * 13 * 1 *
* 13 * 2 * 2 * 28 * 15 * 14 * 1 *
* 14 * 2 * 2 * 28 * 15 * 15 * 1 *
* 15 * 2 * 2 * 28 * 15 * 16 * 1 *
* 16 * 2 * 2 * 28 * 15 * 17 * 1 *
* 17 * 2 * 2 * 28 * 15 * 18 * 1 *
* 18 * 2 * 2 * 28 * 15 * 19 * 1 *
* 19 * 2 * 2 * 28 * 15 * 20 * 1 *
* 20 * 2 * 2 * 28 * 15 * 21 * 1 *
* 21 * 2 * 2 * 28 * 15 * 22 * 1 *
* 22 * 2 * 2 * 28 * 15 * 23 * 1 *
* 23 * 2 * 2 * 28 * 15 * 24 * 1 *
* 24 * 2 * 2 * 28 * 15 * 25 * 1 *
Type <CR> to continue or q to quit ==>
* 25 * 2 * 2 * 28 * 15 * 26 * 1 *
* 26 * 2 * 2 * 28 * 15 * 27 * 1 *
* 27 * 2 * 2 * 28 * 15 * 28 * 1 *
* 240 * 3 * 2 * 28 * 15 * 113 * 1 *
* 241 * 3 * 2 * 28 * 15 * 114 * 1 *
* 242 * 3 * 2 * 28 * 15 * 115 * 1 *
* 243 * 3 * 2 * 28 * 15 * 116 * 1 *
* 244 * 3 * 2 * 28 * 15 * 117 * 1 *
* 245 * 3 * 2 * 28 * 15 * 118 * 1 *
* 246 * 3 * 2 * 28 * 15 * 119 * 1 *
* 247 * 3 * 2 * 28 * 15 * 120 * 1 *
* 248 * 3 * 2 * 28 * 15 * 121 * 1 *
* 249 * 3 * 2 * 28 * 15 * 122 * 1 *
Type <CR> to continue or q to quit ==>
* 250 * 3 * 2 * 28 * 15 * 123 * 1 *
* 251 * 3 * 2 * 28 * 15 * 124 * 1 *
* 252 * 3 * 2 * 28 * 15 * 125 * 1 *
* 253 * 3 * 2 * 28 * 15 * 126 * 1 *
* 254 * 3 * 2 * 28 * 15 * 127 * 1 *
* 255 * 4 * 2 * 28 * 15 * 0 * 1 *
* 256 * 4 * 2 * 28 * 15 * 1 * 1 *
* 257 * 4 * 2 * 28 * 15 * 2 * 1 *
* 258 * 4 * 2 * 28 * 15 * 3 * 1 *
* 259 * 4 * 2 * 28 * 15 * 4 * 1 *
* 260 * 4 * 2 * 28 * 15 * 5 * 1 *
* 261 * 4 * 2 * 28 * 15 * 6 * 1 *
* 262 * 4 * 2 * 28 * 15 * 7 * 1 *
* 263 * 4 * 2 * 28 * 15 * 8 * 1 *
* 264 * 4 * 2 * 28 * 15 * 9 * 1 *
* 265 * 4 * 2 * 28 * 15 * 10 * 1 *
* 266 * 4 * 2 * 28 * 15 * 11 * 1 *
* 267 * 4 * 2 * 28 * 15 * 12 * 1 *
* 268 * 4 * 2 * 28 * 15 * 13 * 1 *
* 269 * 4 * 2 * 28 * 15 * 14 * 1 *
* 270 * 4 * 2 * 28 * 15 * 15 * 1 *
* 271 * 4 * 2 * 28 * 15 * 16 * 1 *
* 272 * 4 * 2 * 28 * 15 * 17 * 1 *
* 273 * 4 * 2 * 28 * 15 * 18 * 1 *
* 274 * 4 * 2 * 28 * 15 * 19 * 1 *
Type <CR> to continue or q to quit ==>
* 275 * 4 * 2 * 28 * 15 * 20 * 1 *
* 276 * 4 * 2 * 28 * 15 * 21 * 1 *
* 277 * 4 * 2 * 28 * 15 * 22 * 1 *
* 278 * 4 * 2 * 28 * 15 * 23 * 1 *
```

Ladder=2がhalf entry 多い。

```
root [2] ch_property->Scan("chp_id:chan_id:profile")
*****
* Row * chp_id * chan_id * profile *
*****
0 * 1 * 0 * 15 *
1 * 1 * 1 * 15 *
2 * 1 * 2 * 15 *
3 * 1 * 3 * 15 *
4 * 1 * 4 * 15 *
5 * 1 * 5 * 15 *
6 * 1 * 6 * 15 *
7 * 1 * 7 * 15 *
8 * 1 * 8 * 15 *
9 * 1 * 9 * 15 *
10 * 1 * 10 * 15 *
11 * 1 * 11 * 15 *
12 * 1 * 12 * 15 *
13 * 1 * 13 * 15 *
14 * 1 * 14 * 15 *
15 * 1 * 15 * 15 *
16 * 1 * 16 * 15 *
17 * 1 * 17 * 15 *
18 * 1 * 18 * 15 *
19 * 1 * 19 * 15 *
20 * 1 * 20 * 15 *
21 * 1 * 21 * 15 *
22 * 1 * 22 * 15 *
23 * 1 * 23 * 15 *
24 * 1 * 24 * 15 *
Type <CR> to continue or q to quit ==>
25 * 1 * 25 * 15 *
26 * 1 * 26 * 15 *
27 * 1 * 27 * 15 *
28 * 1 * 28 * 15 *
29 * 1 * 29 * 15 *
30 * 1 * 30 * 15 *
31 * 1 * 31 * 15 *
32 * 1 * 32 * 15 *
33 * 1 * 33 * 15 *
34 * 1 * 34 * 15 *
35 * 1 * 35 * 15 *
36 * 1 * 36 * 15 *
37 * 1 * 37 * 15 *
38 * 1 * 38 * 15 *
39 * 1 * 39 * 15 *
40 * 1 * 40 * 15 *
41 * 1 * 41 * 15 *
42 * 1 * 42 * 15 *
43 * 1 * 43 * 15 *
44 * 1 * 44 * 15 *
45 * 1 * 45 * 15 *
46 * 1 * 46 * 15 *
47 * 1 * 47 * 15 *
```

L2
L2

```
14 * 116 * 15 *
1 * 117 * 15 *
1 * 118 * 15 *
1 * 119 * 15 *
14 * 120 * 15 *
1 * 121 * 15 *
1 * 122 * 15 *
1 * 123 * 15 *
14 * 124 * 15 *
14 * 125 * 15 *
14 * 126 * 15 *
14 * 127 * 15 *
15 * 0 * 12 *
15 * 1 * 28 *
15 * 2 * 28 *
15 * 3 * 28 *
15 * 4 * 28 *
15 * 5 * 28 *
15 * 6 * 28 *
15 * 7 * 28 *
15 * 8 * 28 *
15 * 9 * 28 *
15 * 10 * 28 *
15 * 11 * 28 *
15 * 12 * 28 *
15 * 13 * 28 *
15 * 14 * 28 *
15 * 15 * 28 *
15 * 16 * 28 *
15 * 17 * 28 *
15 * 18 * 28 *
15 * 19 * 28 *
15 * 20 * 28 *
15 * 21 * 28 *
15 * 22 * 28 *
15 * 23 * 28 *
15 * 24 * 28 *
15 * 25 * 28 *
15 * 26 * 28 *
```

L2
L2

- Ladder=2, LVDS=2mA の ch_property.root ファイルを目視確認
- その結果、本当に profile=28 だった。

まとめ

- 1 bad channelの数 →2mA以上
- 2 half entryの数→LVDSに依存しない。

以上から LVDS currentを設定する値を決定すると、

LVDS = 2mAでもよさそう。