

sPHENIX INTT 日本語ミーティング、2020/09/25

INTT テストベンチ外部トリガー用 データ収集システム (DAQ) の構築

ぬかづか

糠塚元気 (RBRC)

スペシャルサンクス：奈良女の皆さん

もくじ

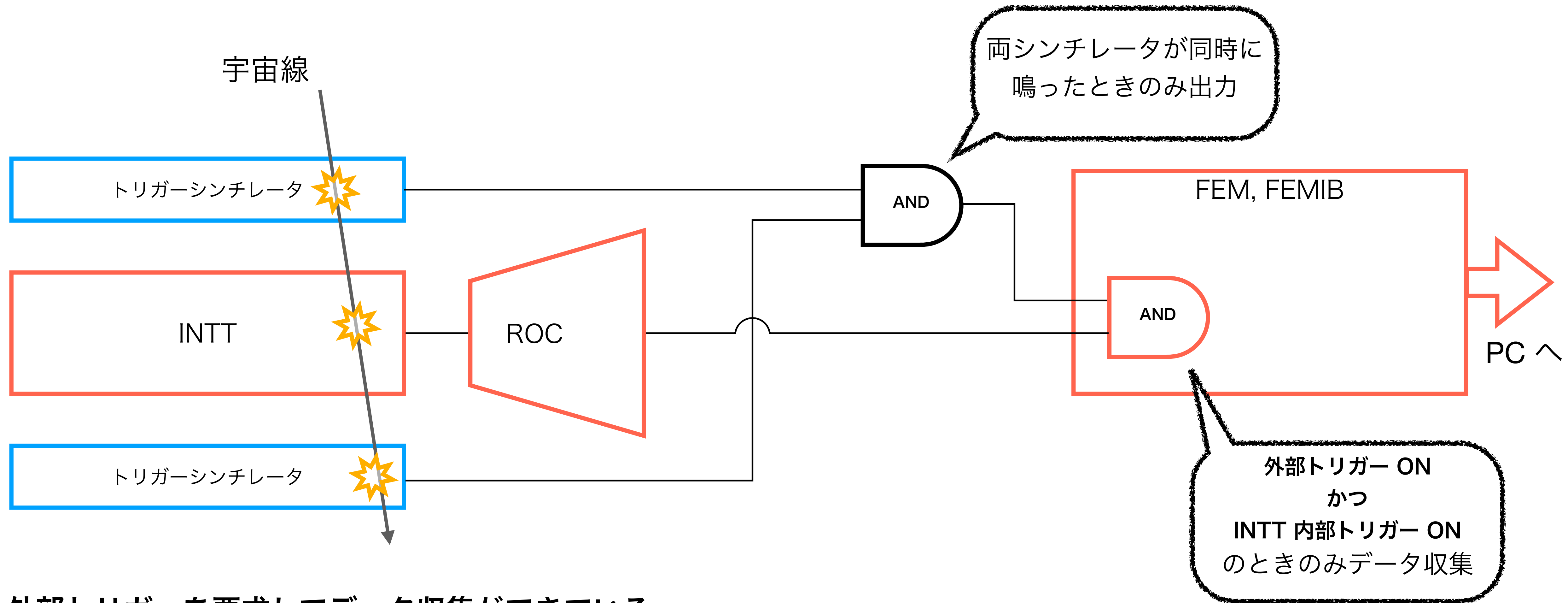
INTT テストベンチ, 外部トリガーモード

DAQ システムのセットアップ

動作確認

これから

テストベンチ、外部トリガーモード



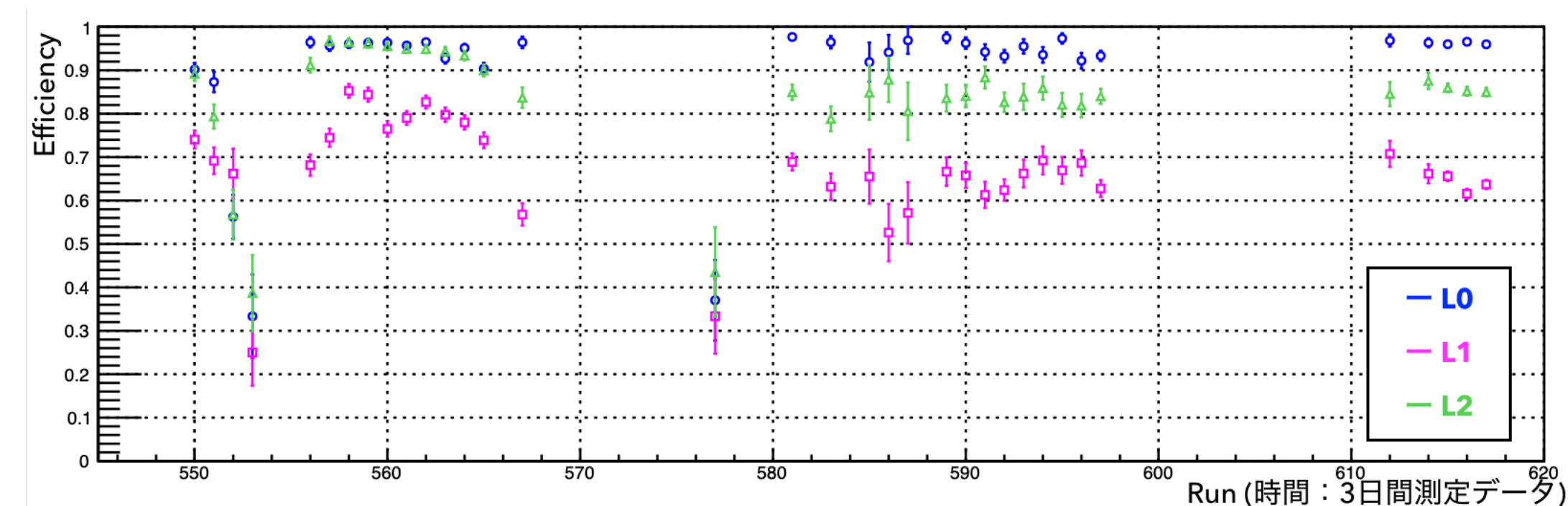
外部トリガーを要求してデータ収集ができている
外部トリガーの細かい情報（波高、タイミング等）は記録していない

テストベンチ、外部トリガーモードの実績

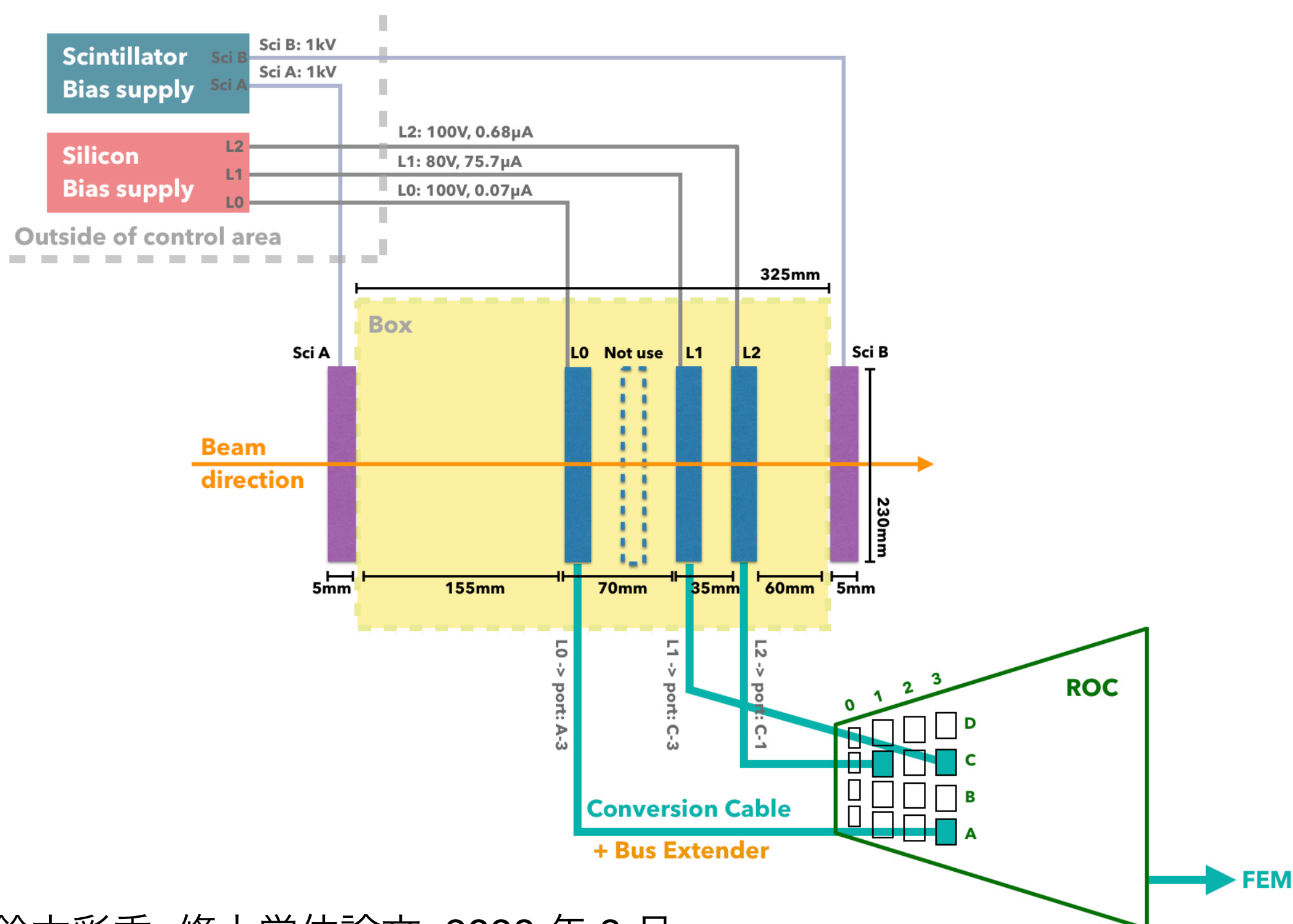


2019年6月テストビーム実験@FNAL

- ラダー L の検出効率 = $\frac{\text{全ラダーにヒットがあったイベント数}}{\text{L以外のラダーにヒットがあったイベント数}}$
- 検出効率 96% を確認した

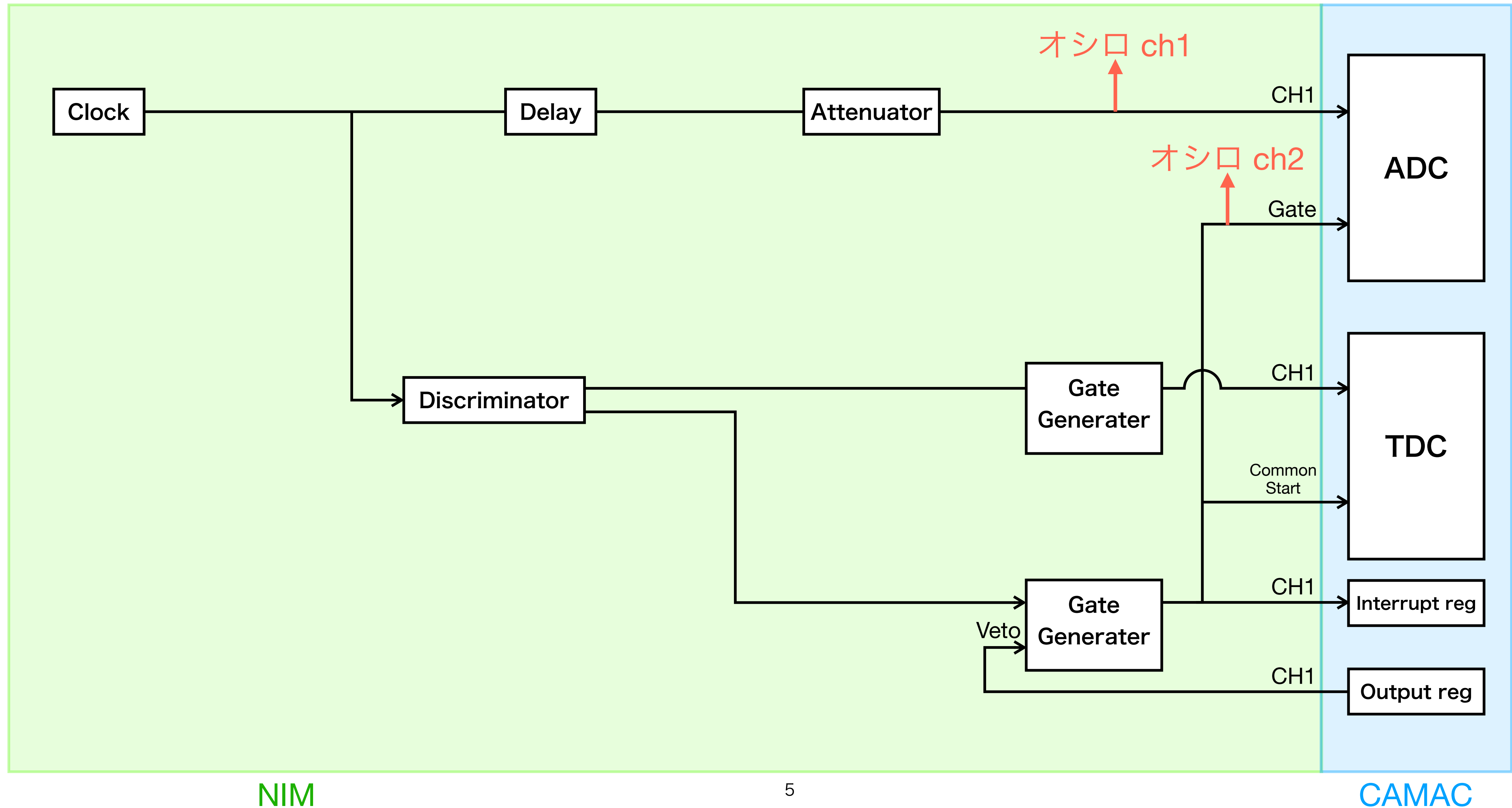


- システムが理想的に動作すれば検出効率は 100% になるはず...
- 外部トリガー情報を解析に取り込むと検出効率向上につながる見込み



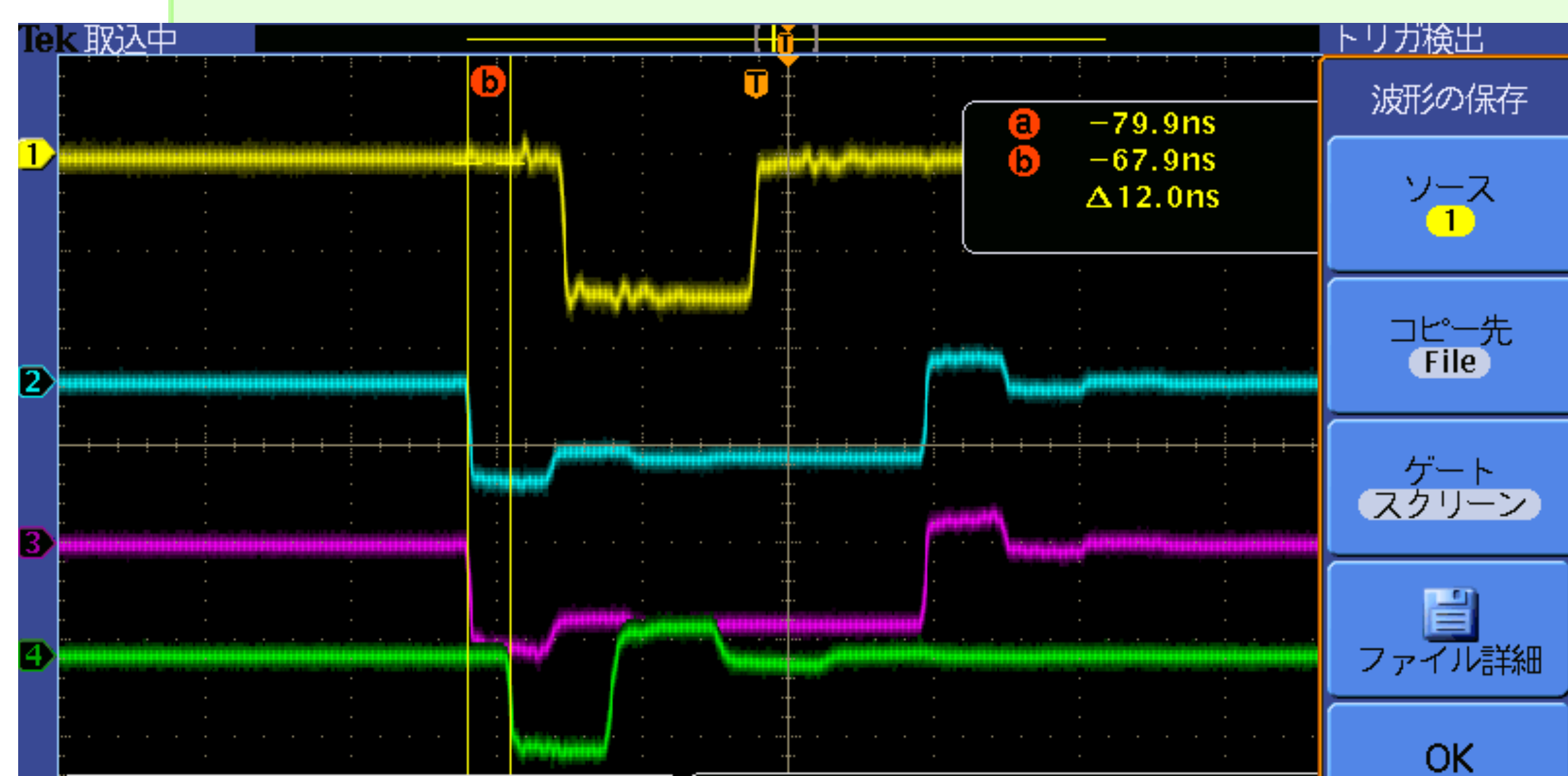
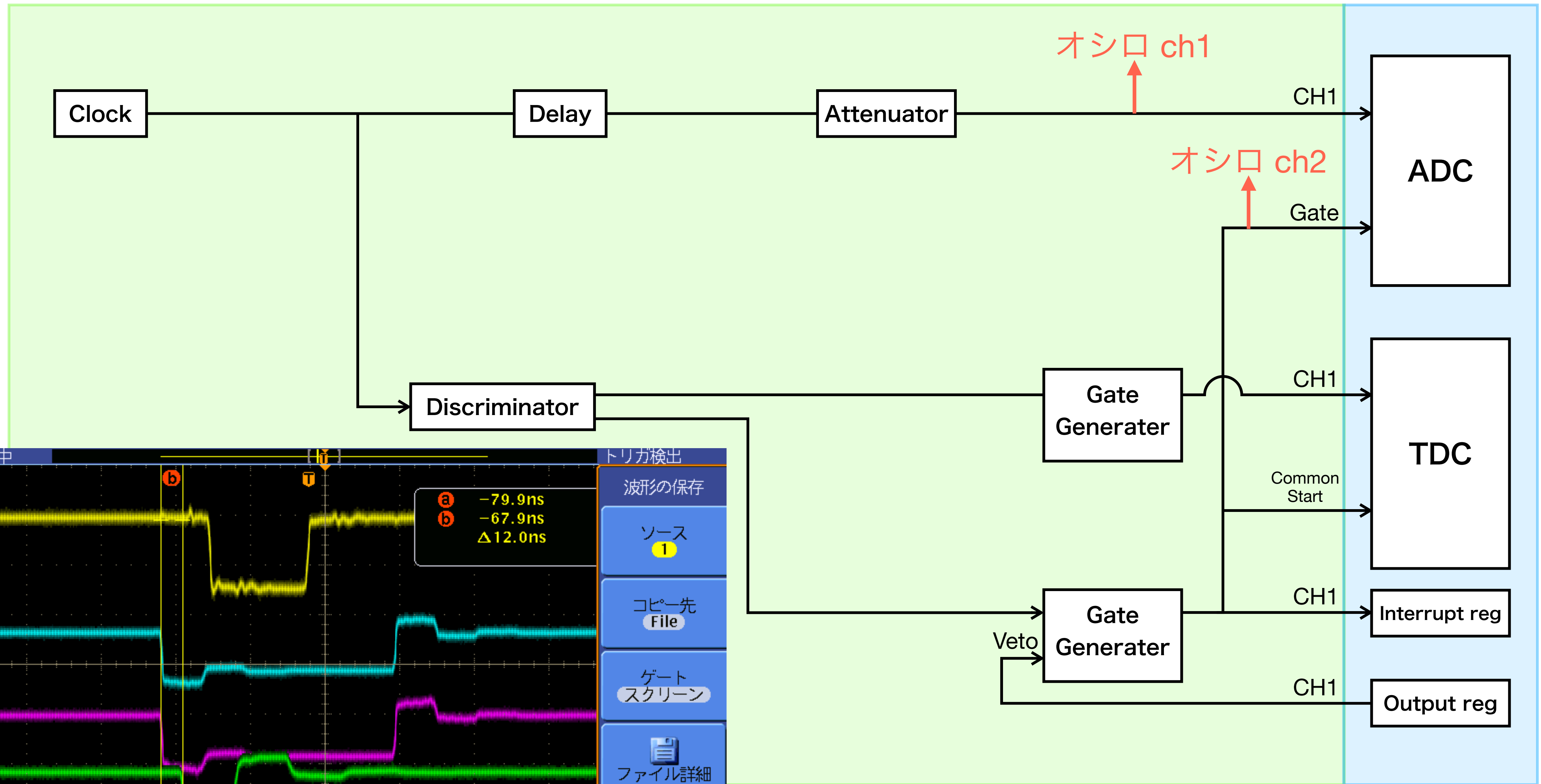
テストベンチ用外部トリガーデータ収集システム

テストセットアップ1

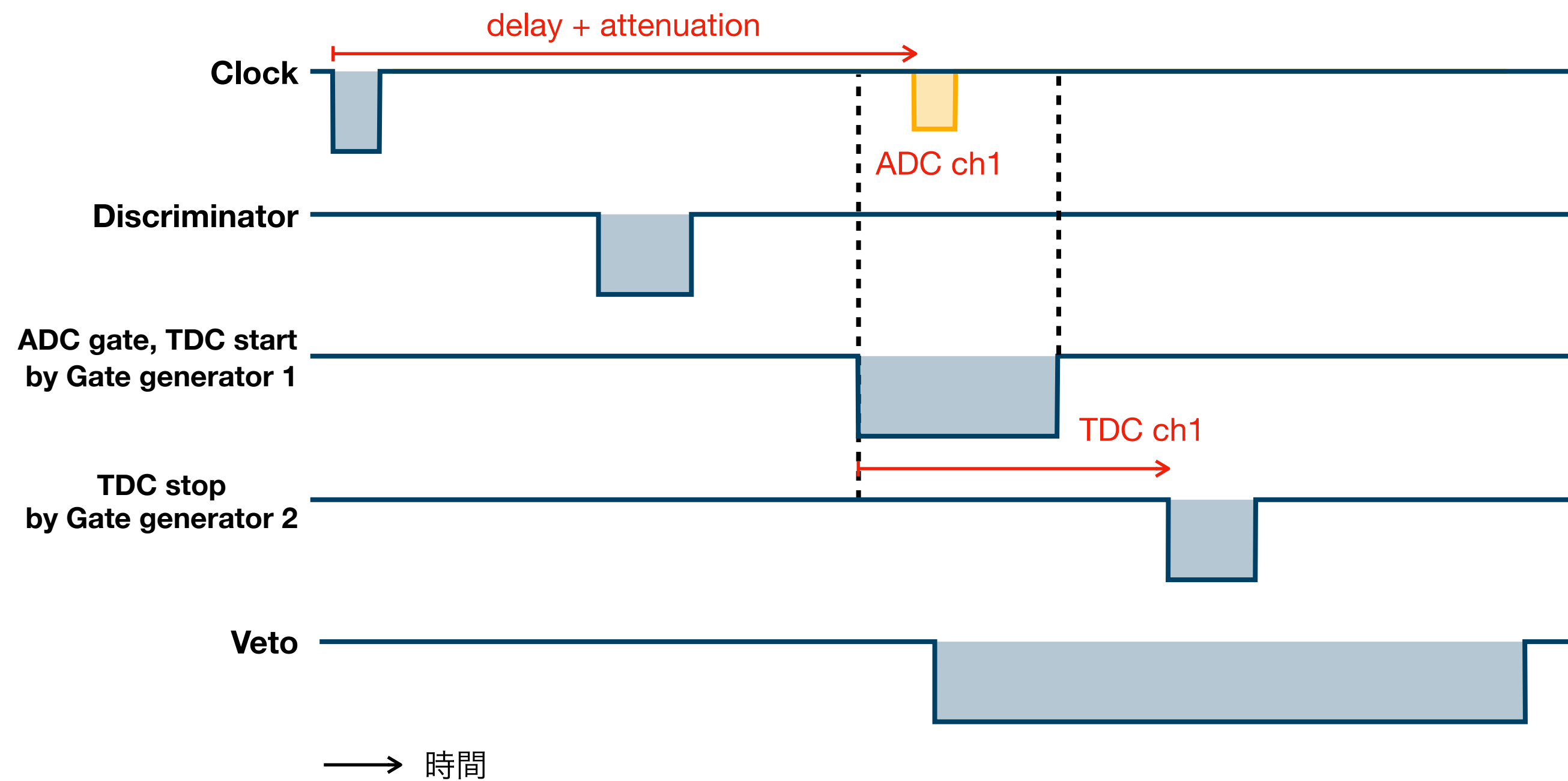
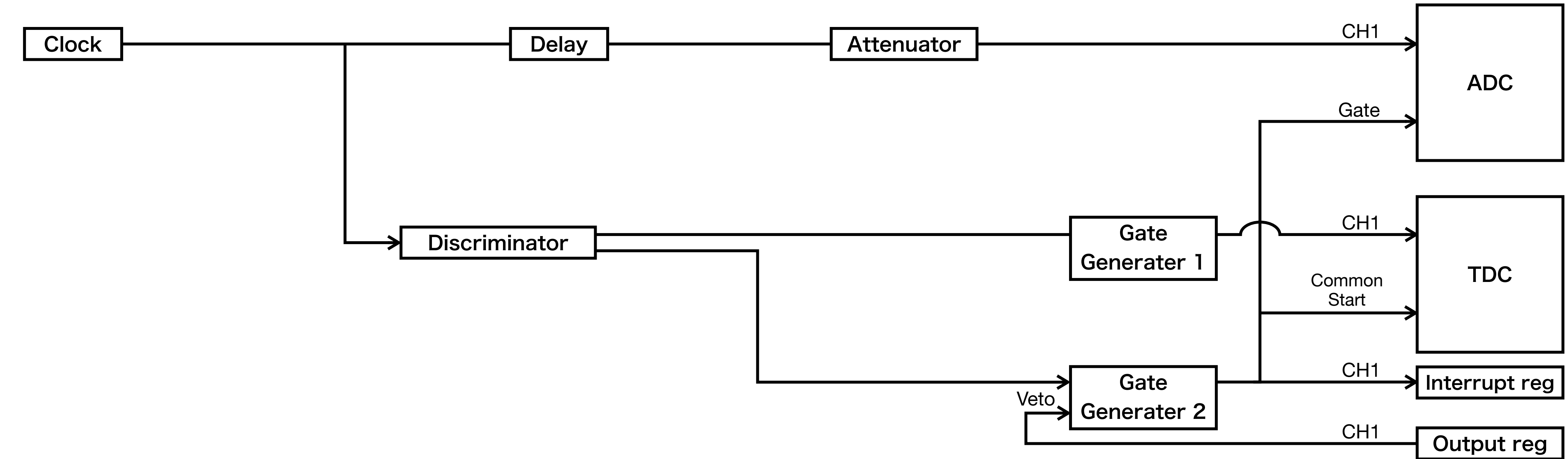


テストベンチ用外部トリガーデータ収集システム

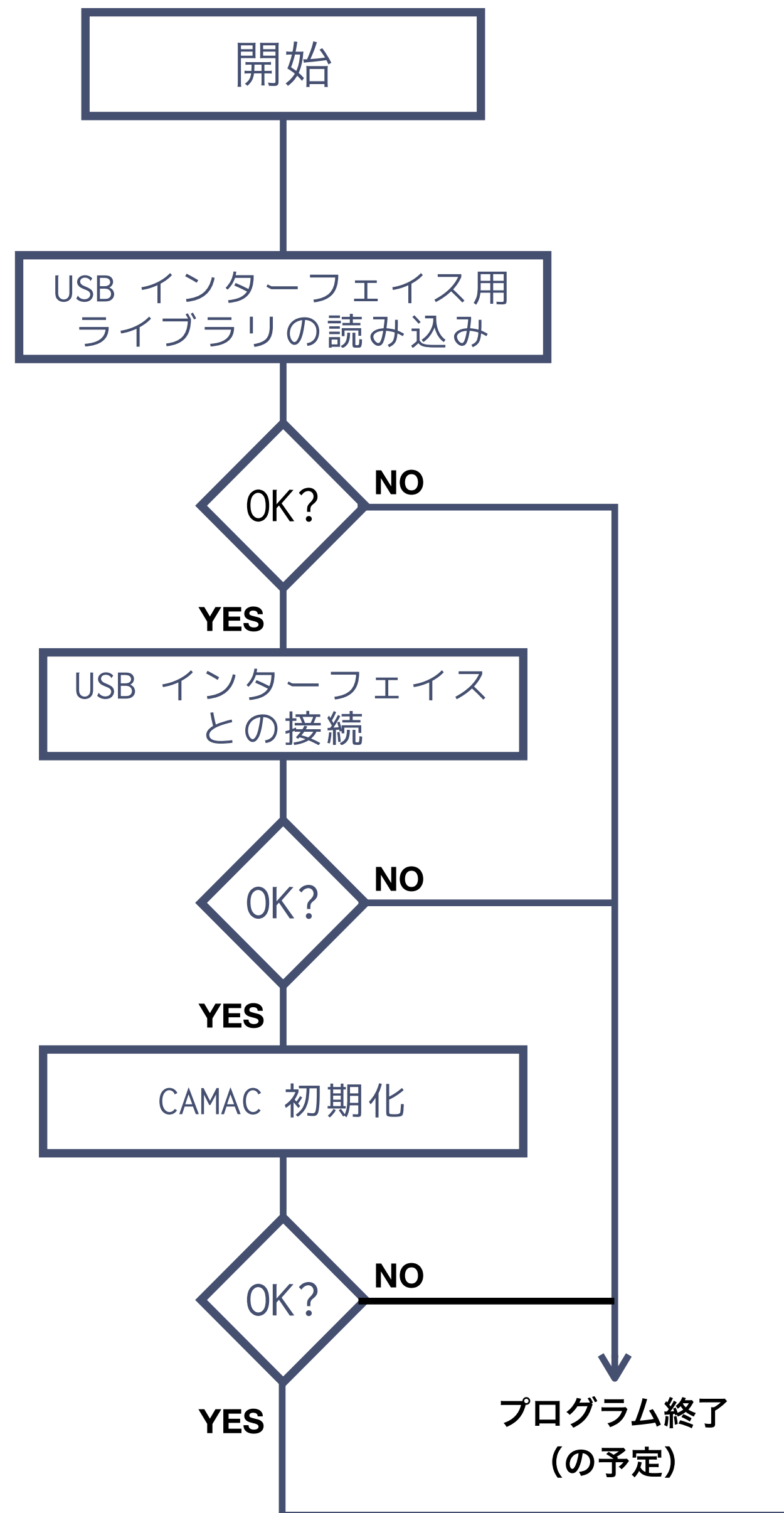
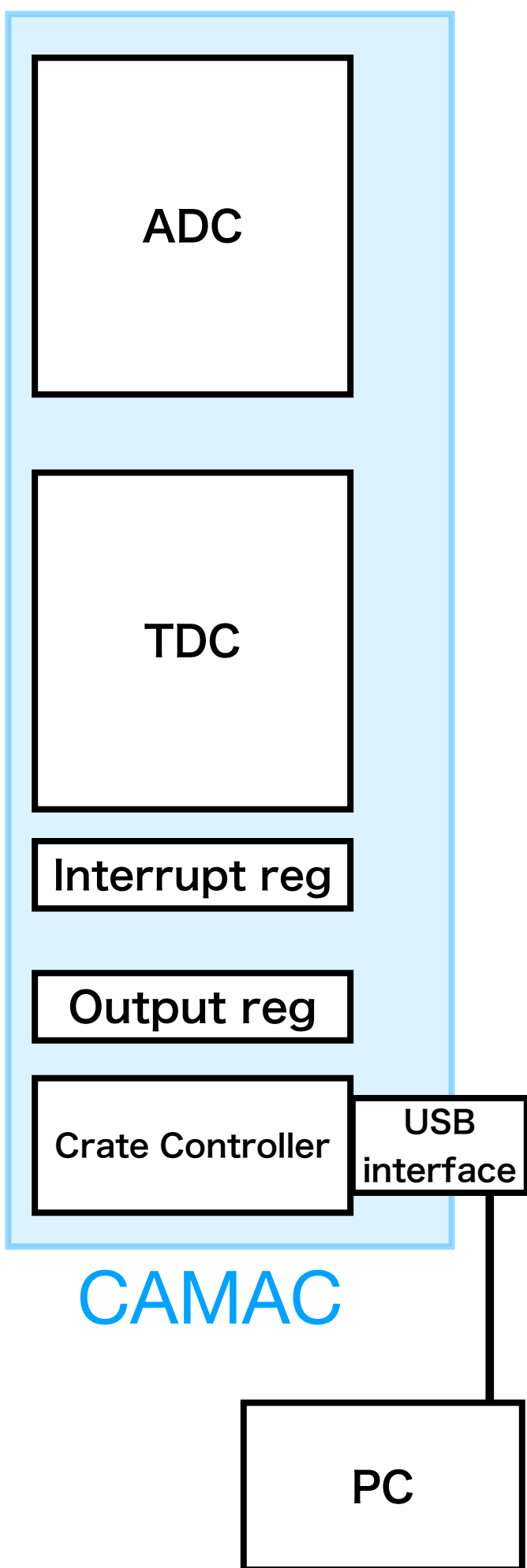
テストセットアップ1



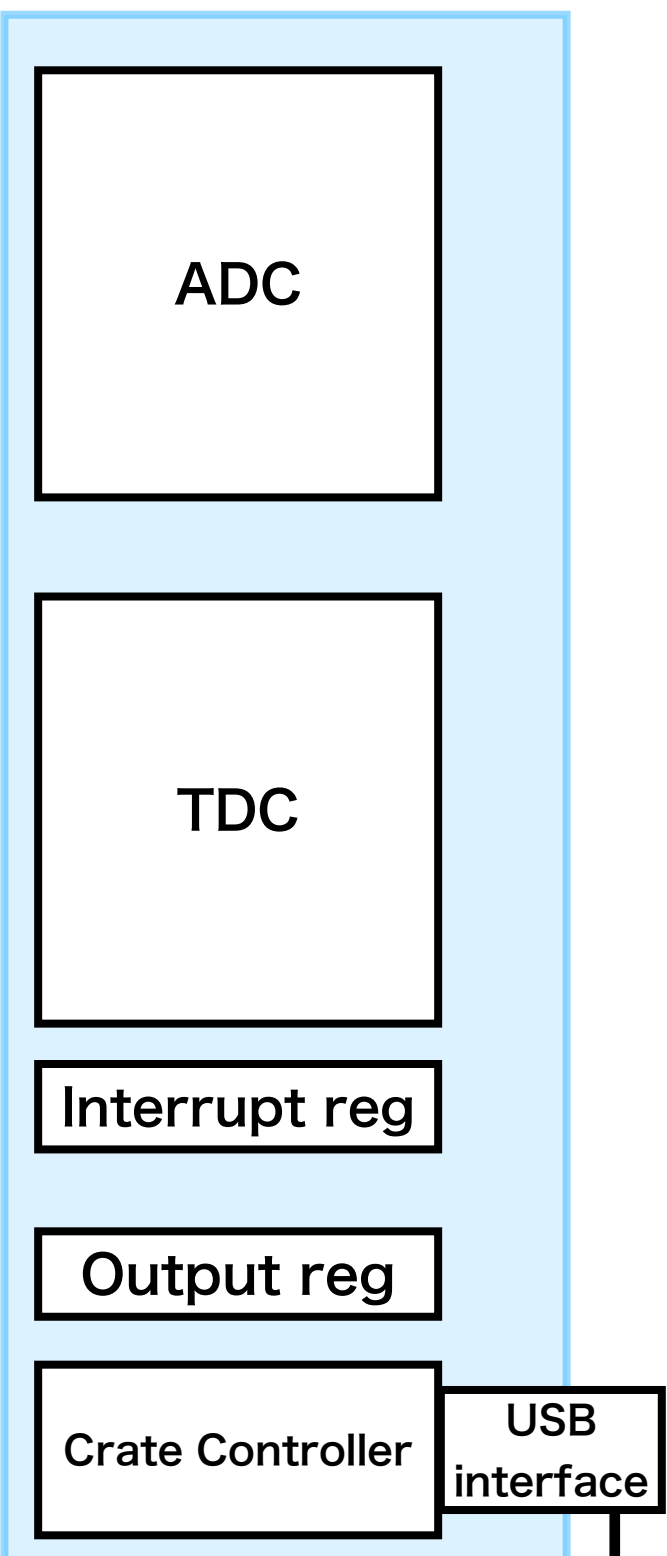
テストベンチ用外部トリガデータ収集システム



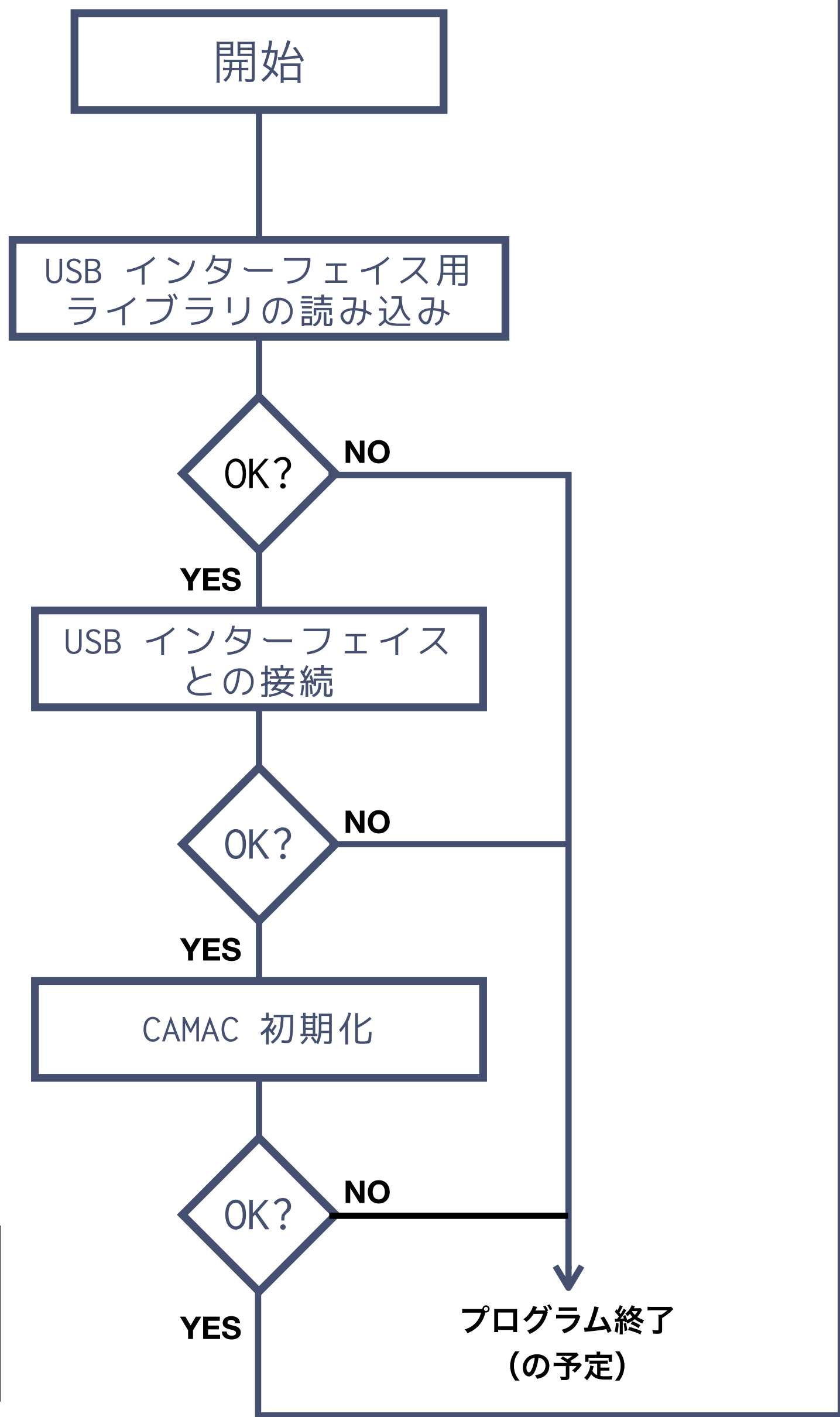
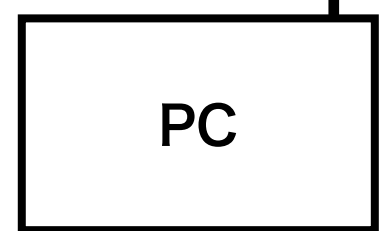
DAQ の動作, CAMAC との通信準備



DAQ の動作, データ収集準備



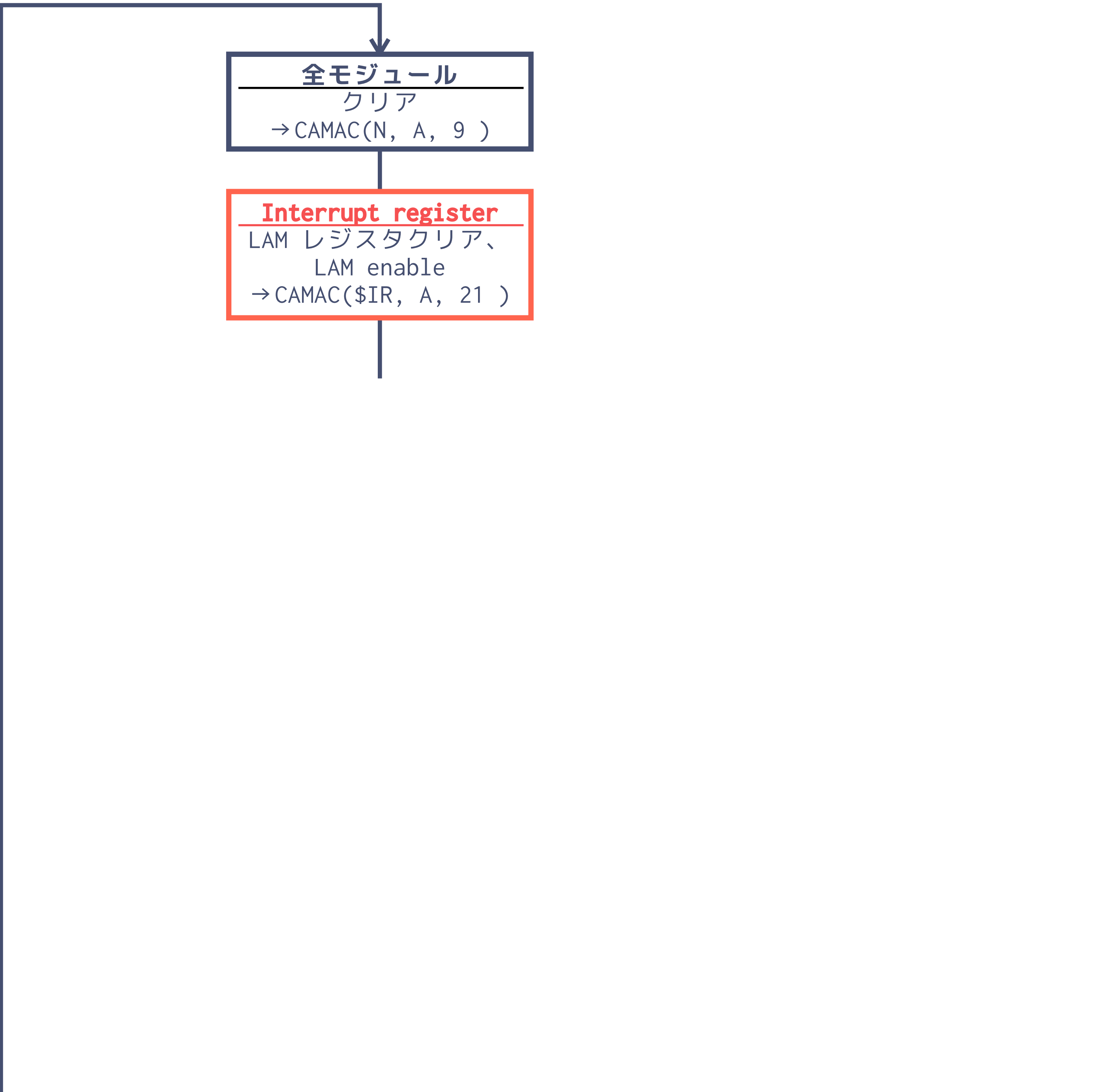
CAMAC



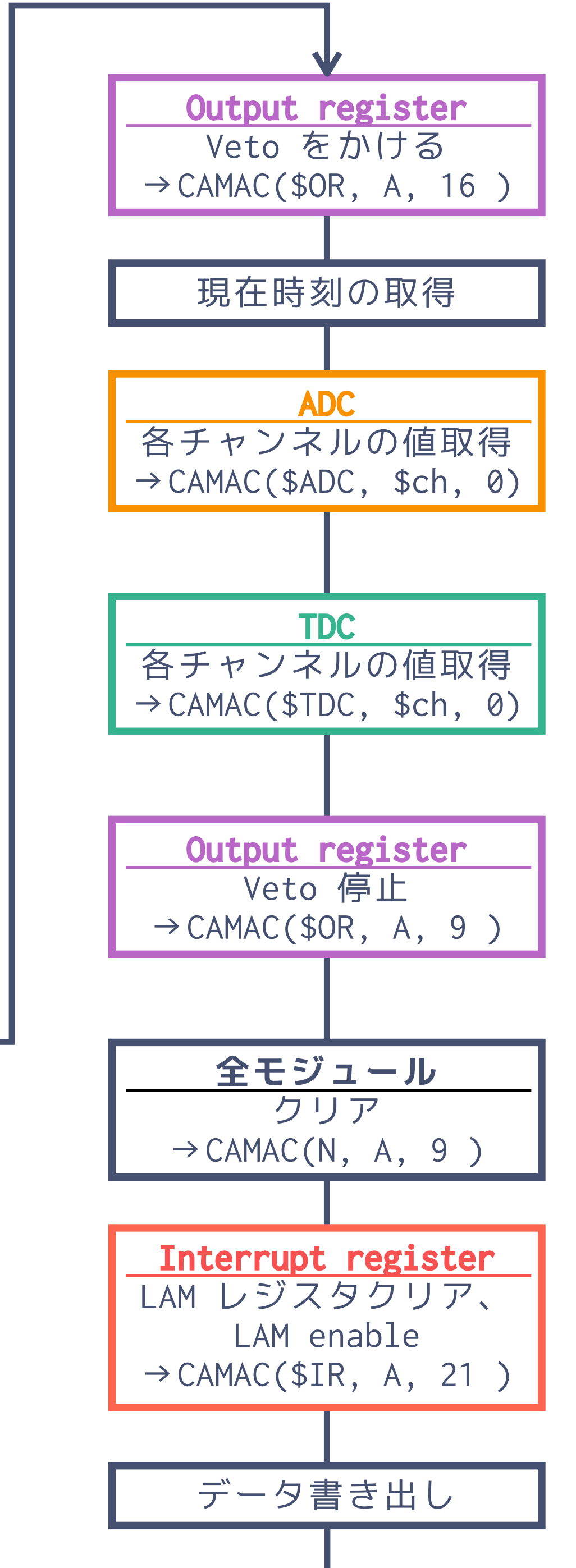
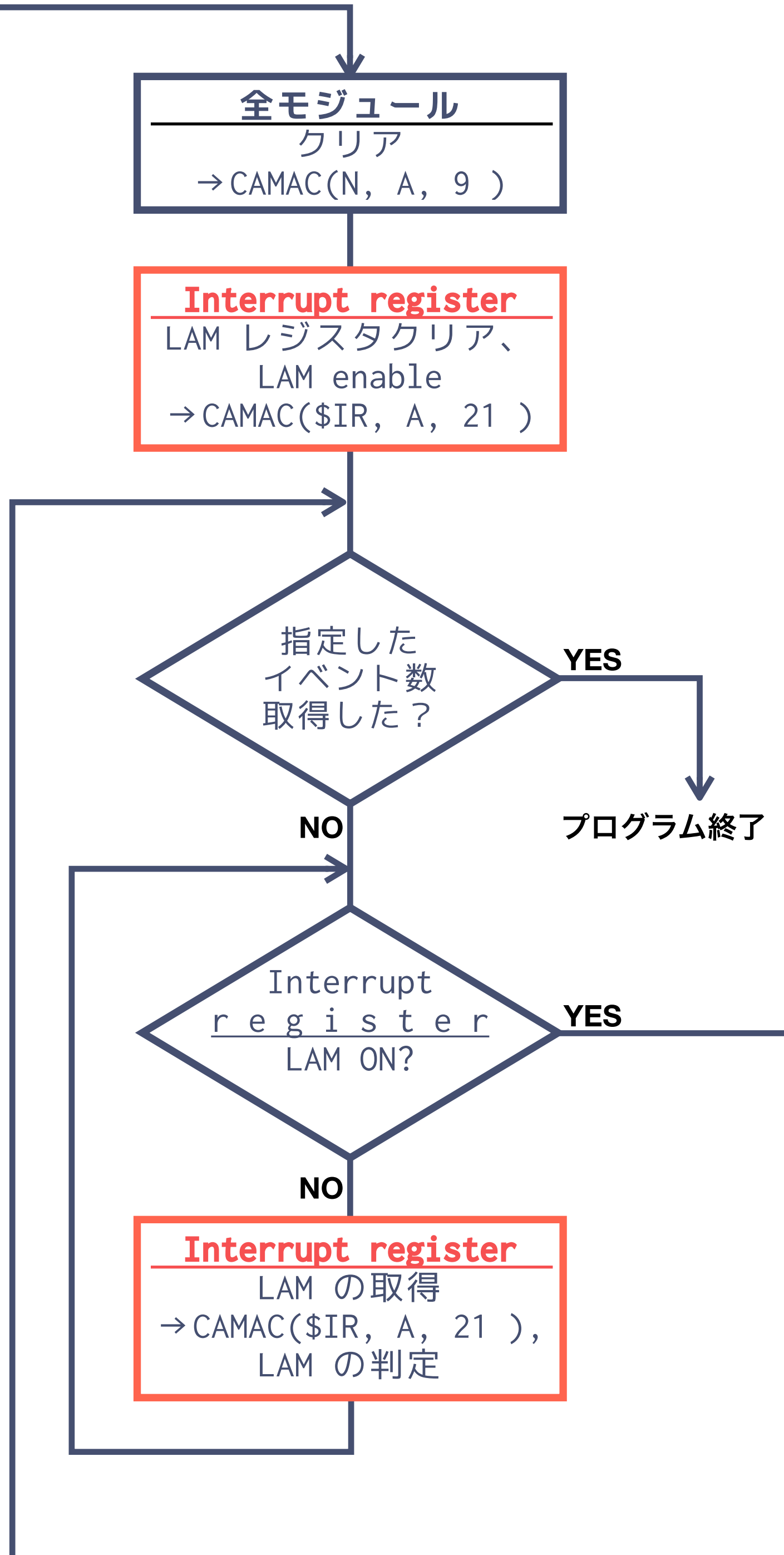
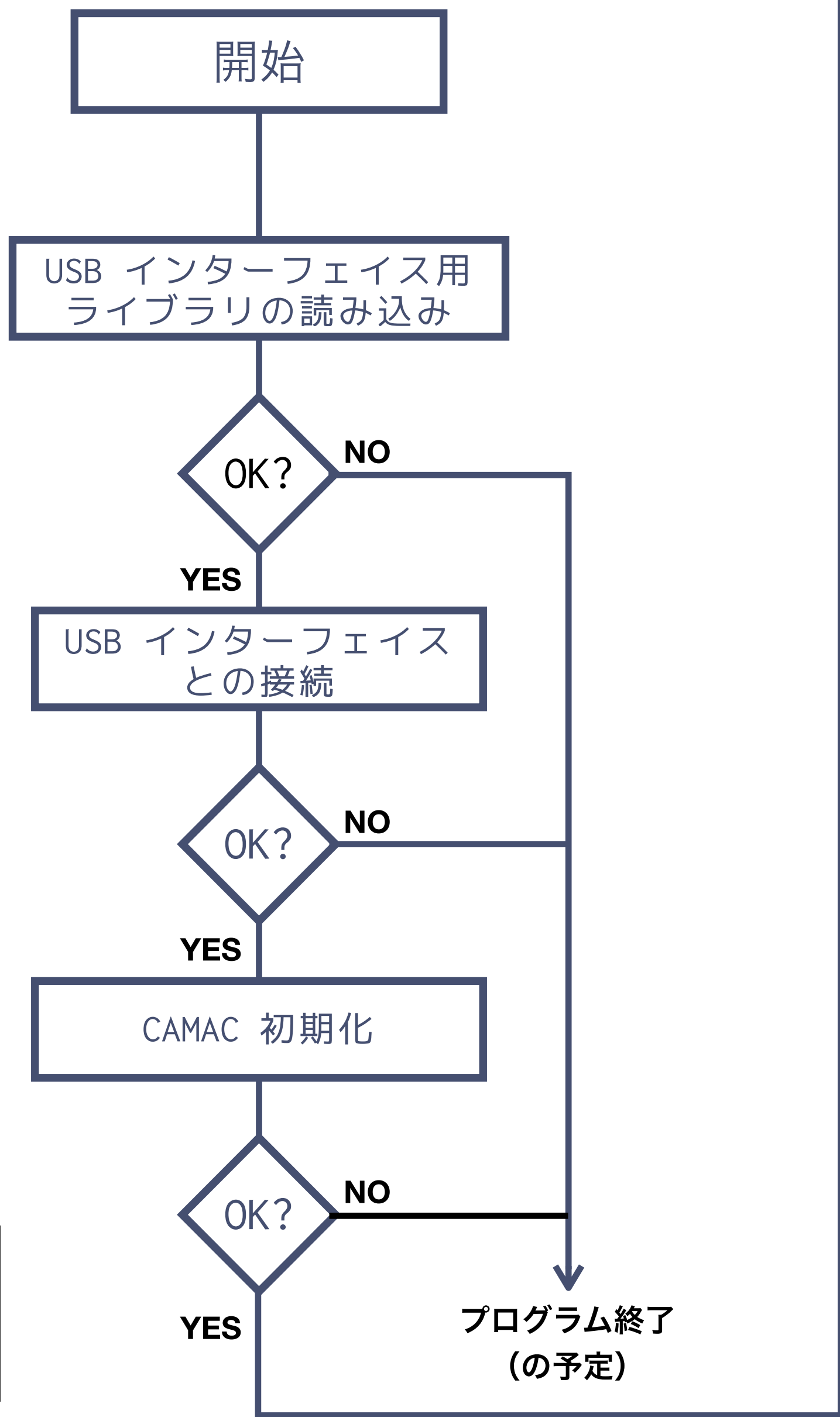
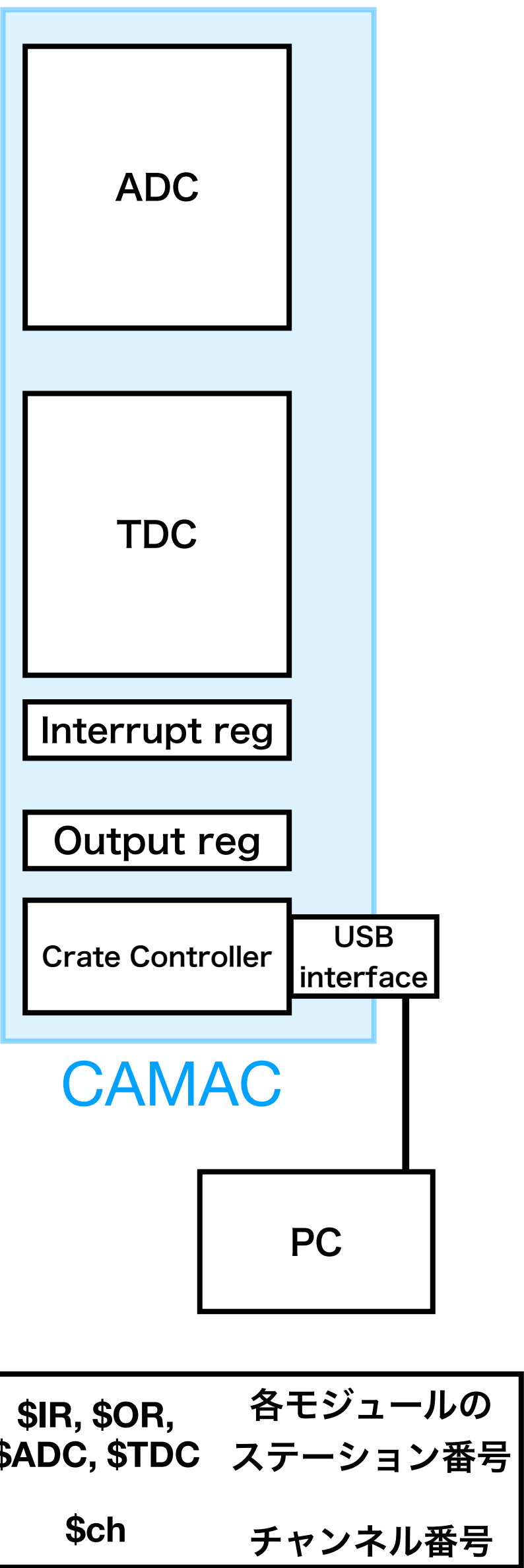
\$IR, \$OR, 各モジュールの
\$ADC, \$TDC ステーション番号
\$ch チャンネル番号

全モジュール
クリア
→CAMAC(N, A, 9)

Interrupt register
LAM レジスタクリア、
LAM enable
→CAMAC(\$IR, A, 21)



DAQ の動作, データ収集



DAQ プログラム

OS : Windows 7

言語 : Visual C++

GUI インターフェイス : 未実装

依存関係 : 豊伸電子 CCP-USB のライブラリ HECMUSB3.dll

ファイル :

CAMAC_DAQ_backend.cpp : main 関数

DAQ.cpp, .h : 設定ファイル読み込み, データ収集のクラス

Functions.cpp, .h : 便利な関数

LoadHECMUSB3.cpp, .h : ライブラリファイルの読み込み

Module.cpp, .h : CAMAC モジュール単位の設定を扱うクラス

resource.h : Visual C++ が自動生成したもの

データ収集の設定 : テキストファイルに書き込む

開発状況 : テスト PC で動いた。テストベンチ用 PC に未移植。

```
#####  
# 設定ファイルの例 (ただのテキストファイル) #  
#####  
# 収集するイベント数  
EventNum 1000  
  
# データ収集する時間と単位  
# MeasurementTime 10 min # 未実装  
  
# データを書き出すディレクトリ  
OutputDir ~/data  
  
# 使用する CAMAC クレートの番号  
Crate 0  
  
# 使用するモジュールのタイプ,  
# ステーション番号, チャンネル  
Module ADC 1 0 1  
Module TDC 2 0 1  
Module InterruptRegister 3 0  
Module OutputRegister 4 0  
  
# LAM を監視するモジュール番号とチャンネル番号  
LAM 2 1  
  
# Veto をかけるモジュール番号とチャンネル番号  
Veto 3 1  
  
# 実行中にターミナルに表示する情報量 (0, 1, 2)  
Verbosity 1
```

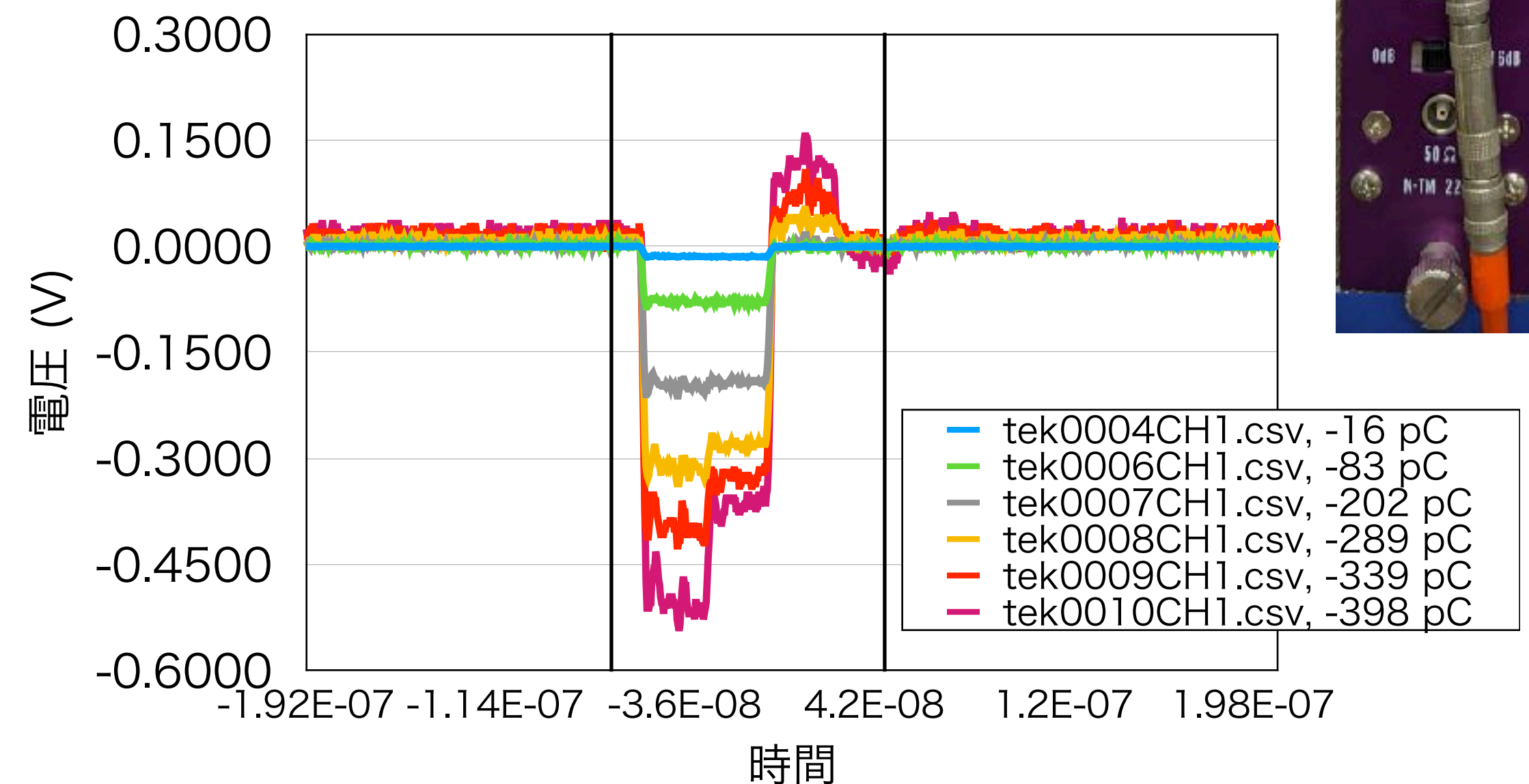

動作確認, NIM Clock で ADC の挙動を見る

LeCroy 2249W

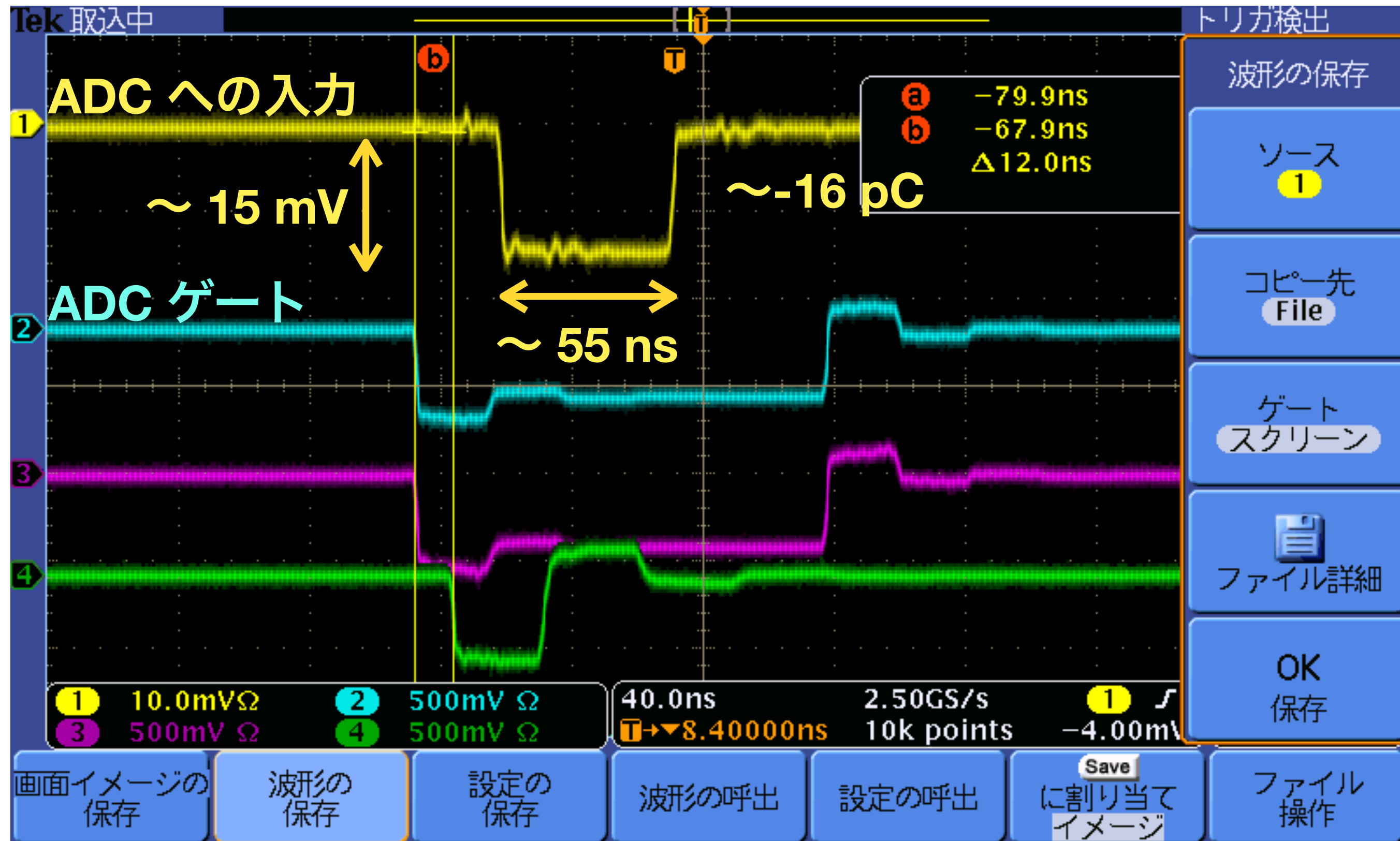
タイプ	: 積分形
入力チャンネル	: 12ch
出力	: 1980 チャンネル 11 bit
ゲイン	: -0.25 pC/ch \pm 5%
レンジ	: 最大 -500 pC
ゲート信号	: 30 ns ~ 10 μ s, 内部 delay 7ns あり
ペDESTAL	: 本体側面のつまみで 100 ch ほど調整可能
処理時間	: 106 μ s

測定する信号の作り方

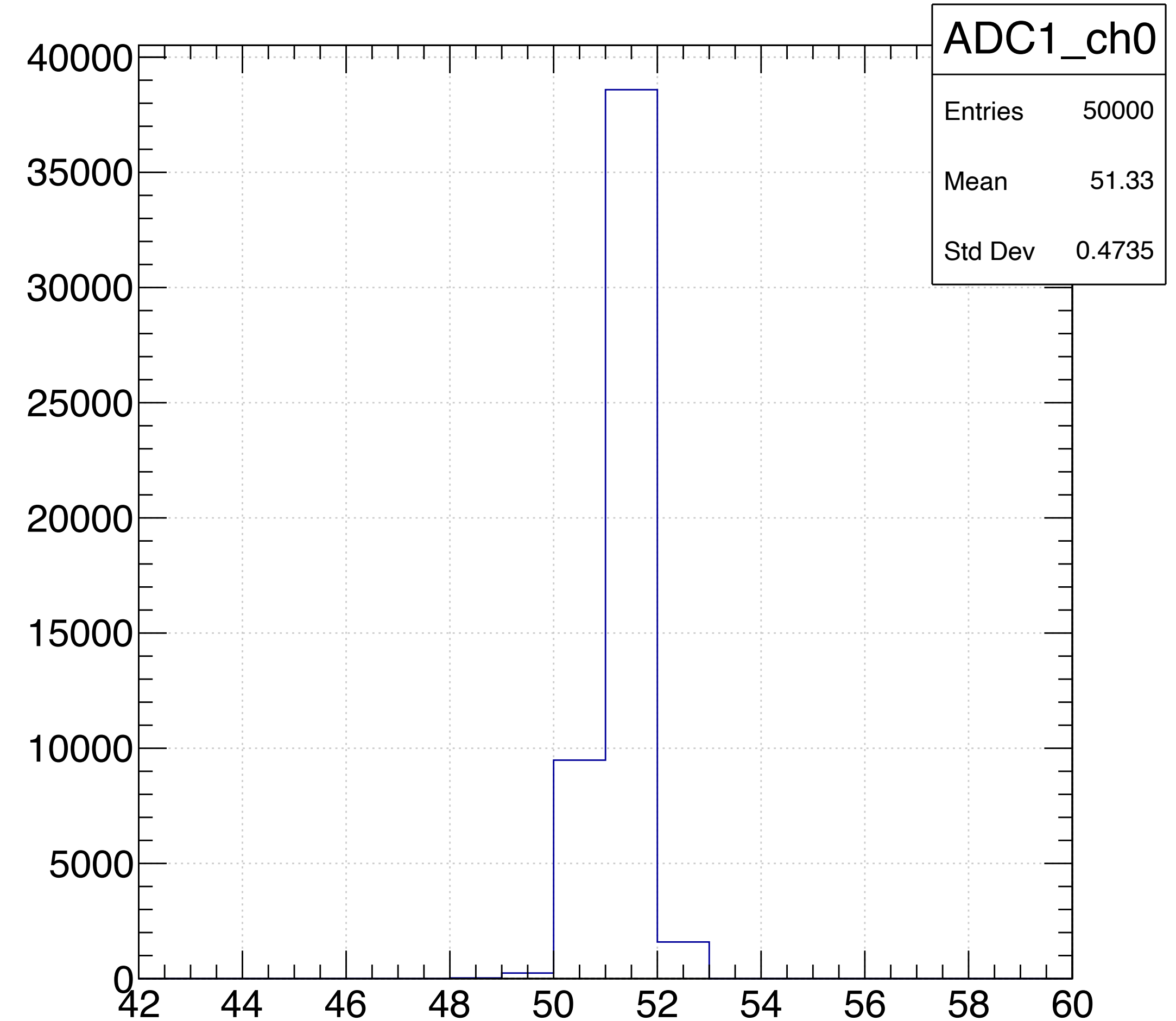
1. NIM クロックで信号生成
2. Gate generator で信号幅、タイミング調整
3. Attenuator で波高調整



動作確認, NIM Clock で ADC の挙動を見る

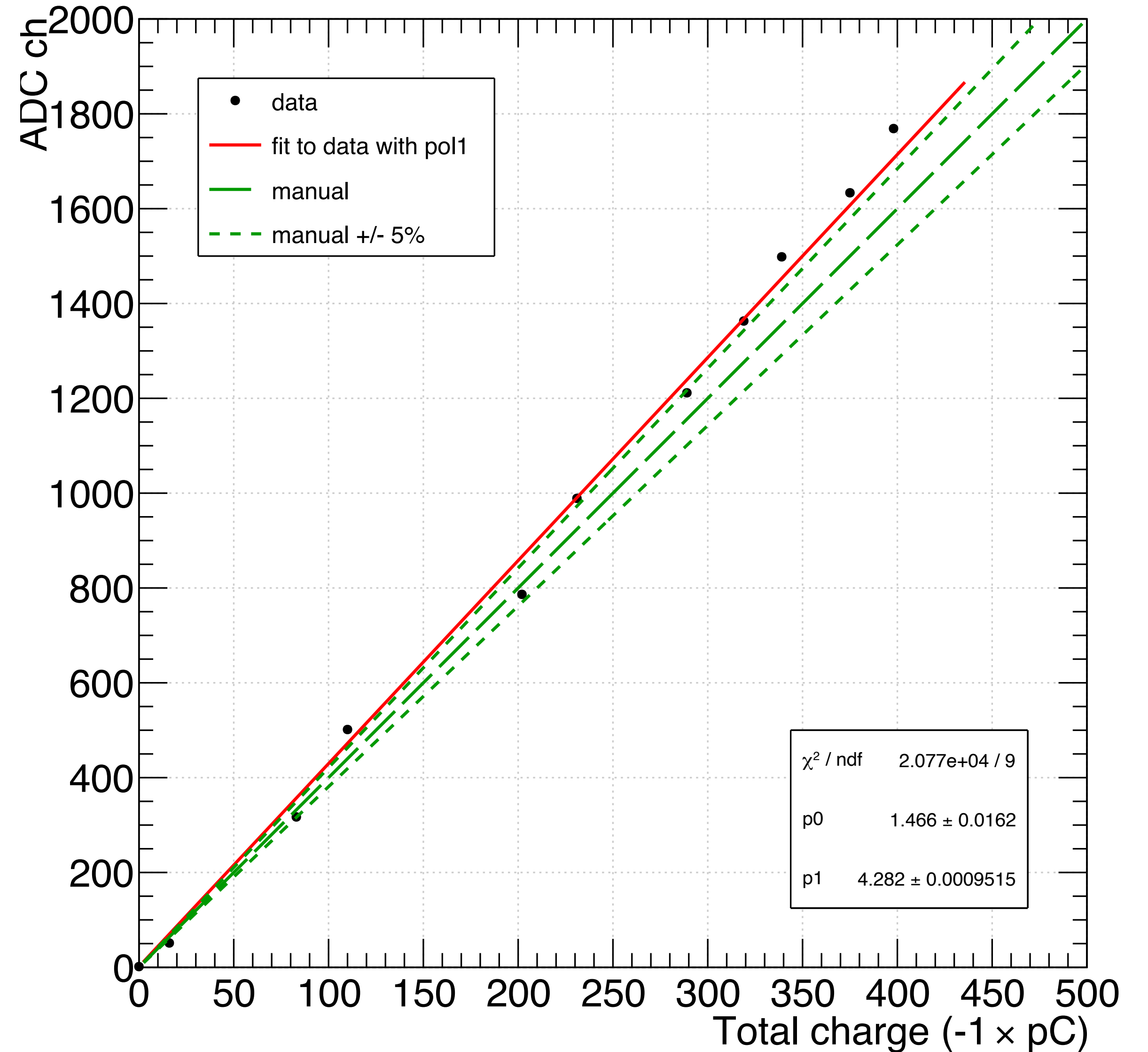
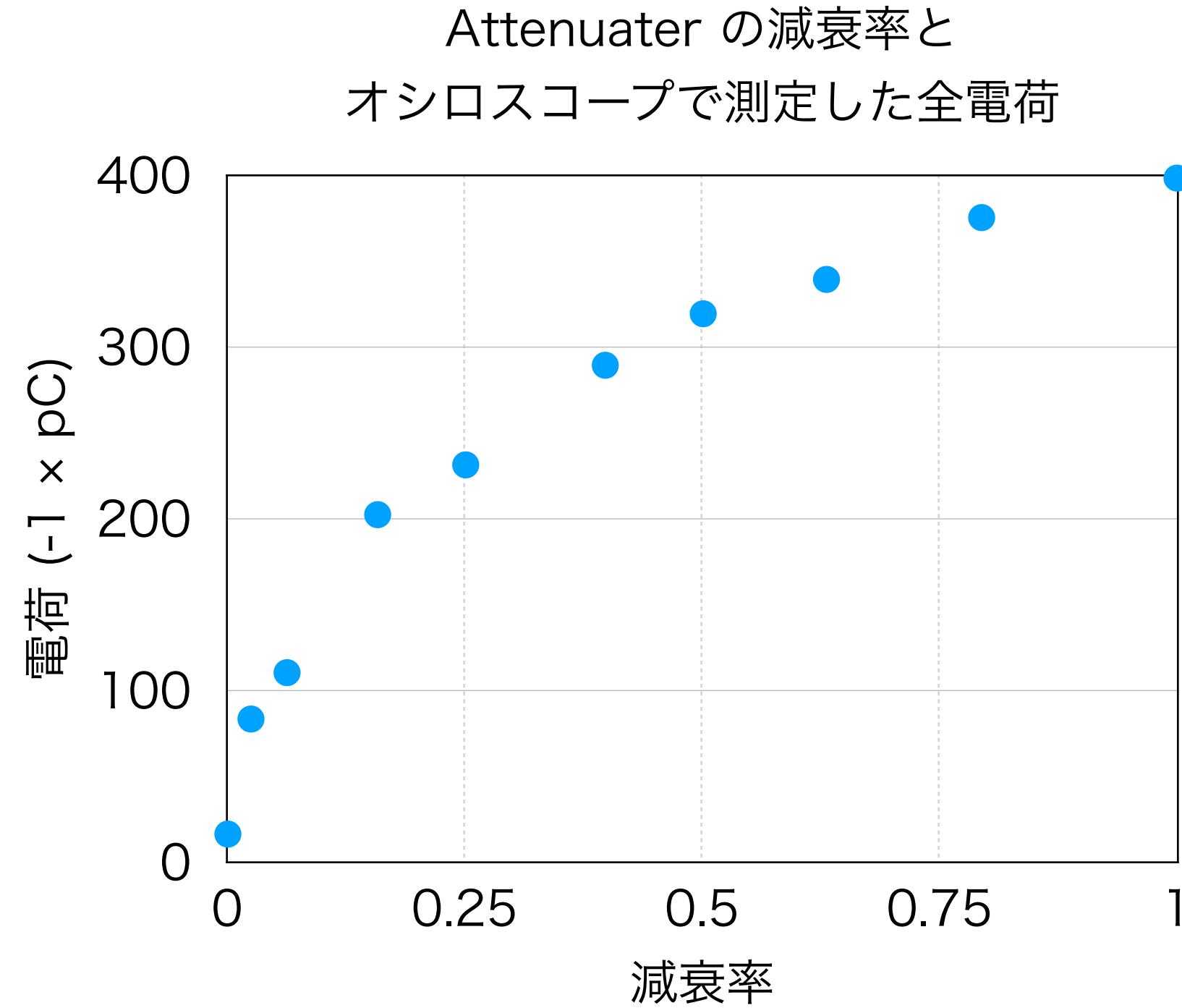


ADC1_ch0, zoomed



動作確認, NIM Clock で ADC の挙動を見る

Tota charge vs ADC ch

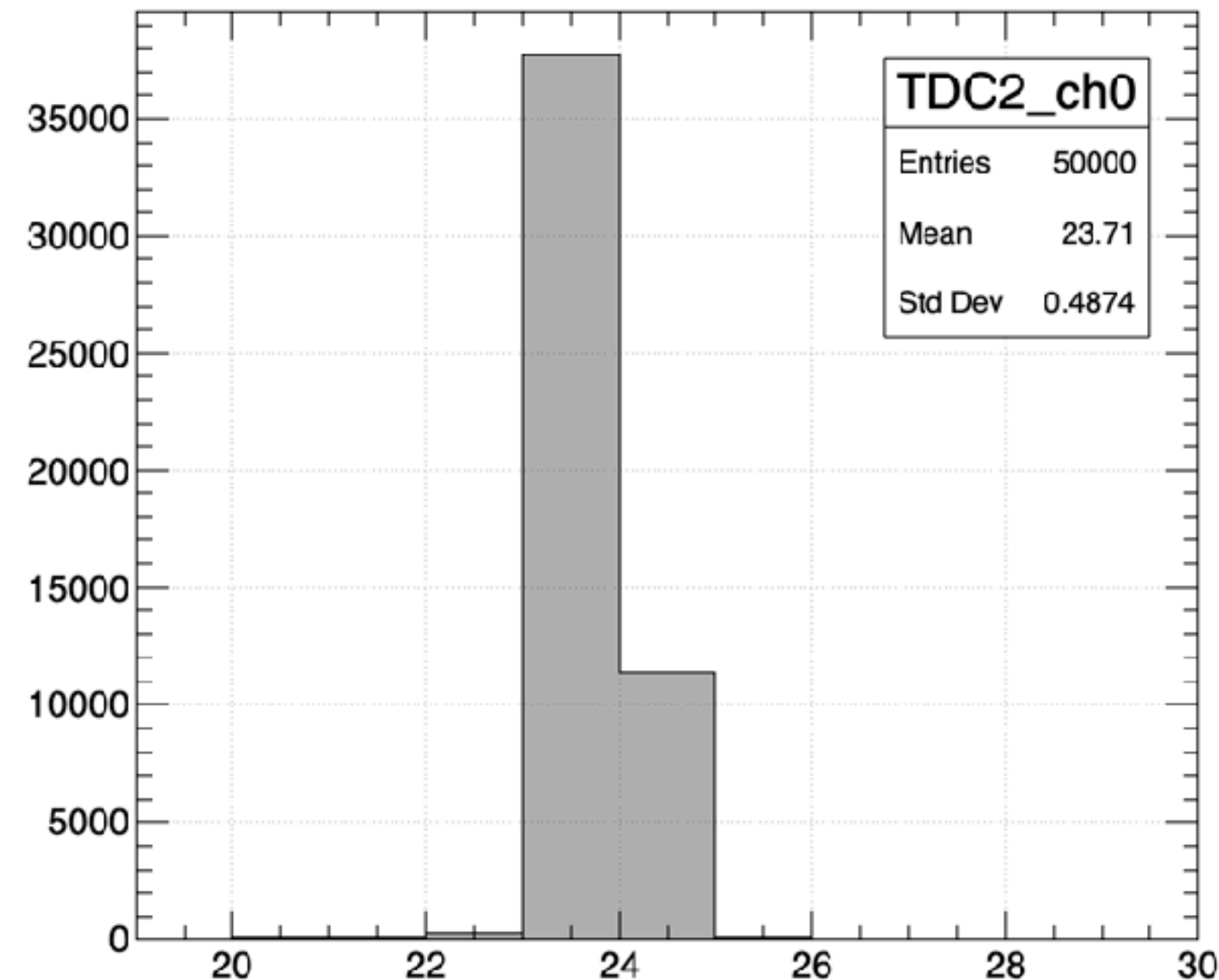


動作確認, TDC で一定の時間差を測定する

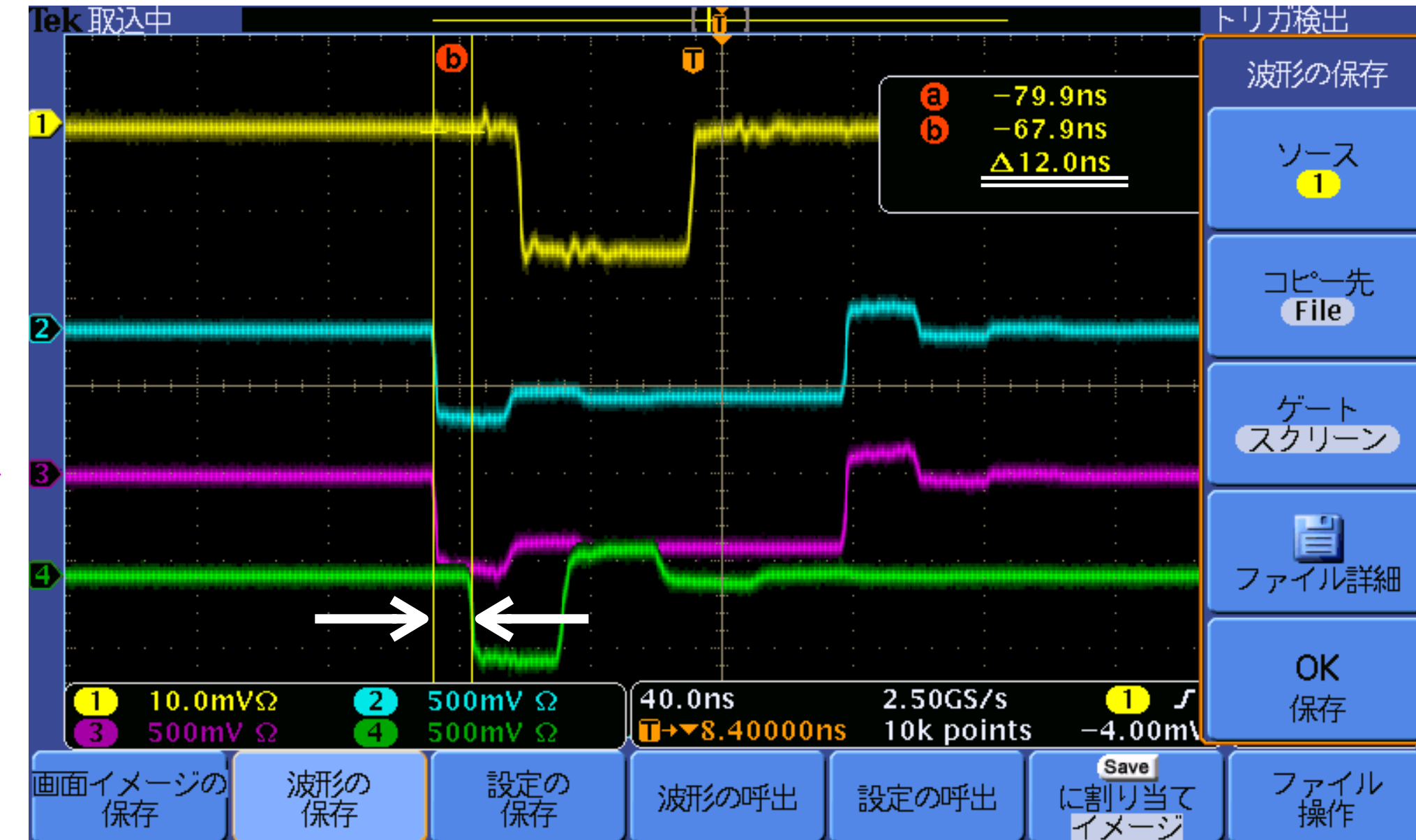
LeCroy 2228A

入力チャンネル : 8ch
出力 : 2048 チャンネル 11 bit
分解能 : 50, 100, 200 ps (フルスケールに依存)
フルスケール : 100, 200, 500 ns (調整可能)
処理時間 : 100 μ s

TDC2_ch0



TDC start →
TDC stop →

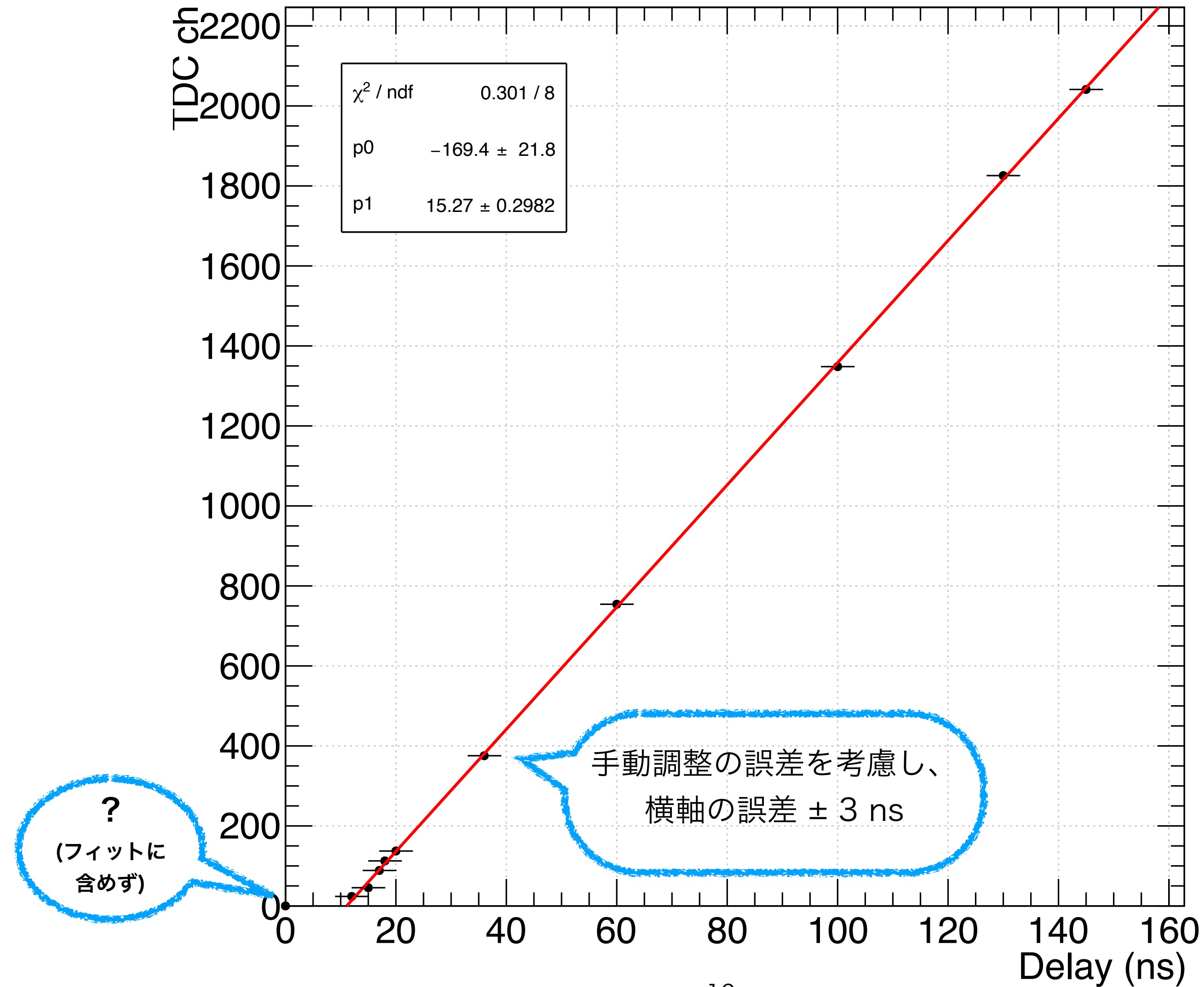


測定する時間差の作り方

1. NIM クロックで信号生成、2 系統に枝分け
2. Gate generator1 スタート信号生成
3. Gate generator2 で delay を設定しストップ信号生成

動作確認, TDC で一定の時間差を測定する

Delay vs TDC ch

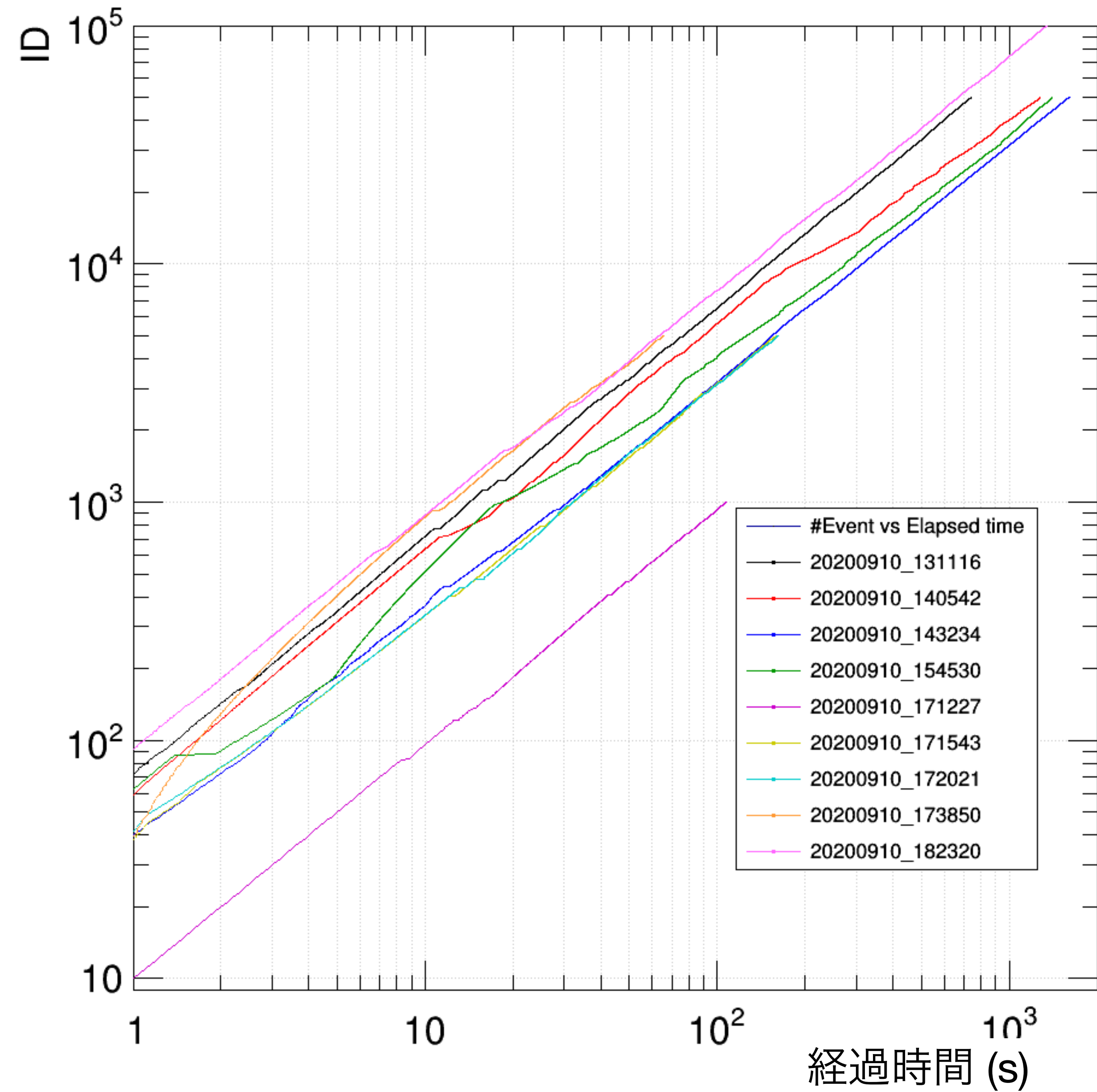


~15 ch/ns
→ 67 ps/ch

動作確認, 測定レート

#Event vs Elapsed time

ファイル	f (Hz)	イベント数	所要時間	レート (Hz)	待ち時間
20200910_131116	1k	50k	742	67	0
20200910_140542	1k	50k	1275	39	1 ms
20200910_143234	1k	50k	1610	31	1 ms
20200910_154530	1k	50k	1403	36	1 ms
20200910_171227	10	1k	108	9	1 ms
20200910_171543	100	5k	160	31	1 ms
20200910_172021	500	5k	162	31	1 ms
20200910_173850	1k	5k	66	76	300 μ s
20200910_182320	^{137}Cs	100k	1344	74	300 μ s

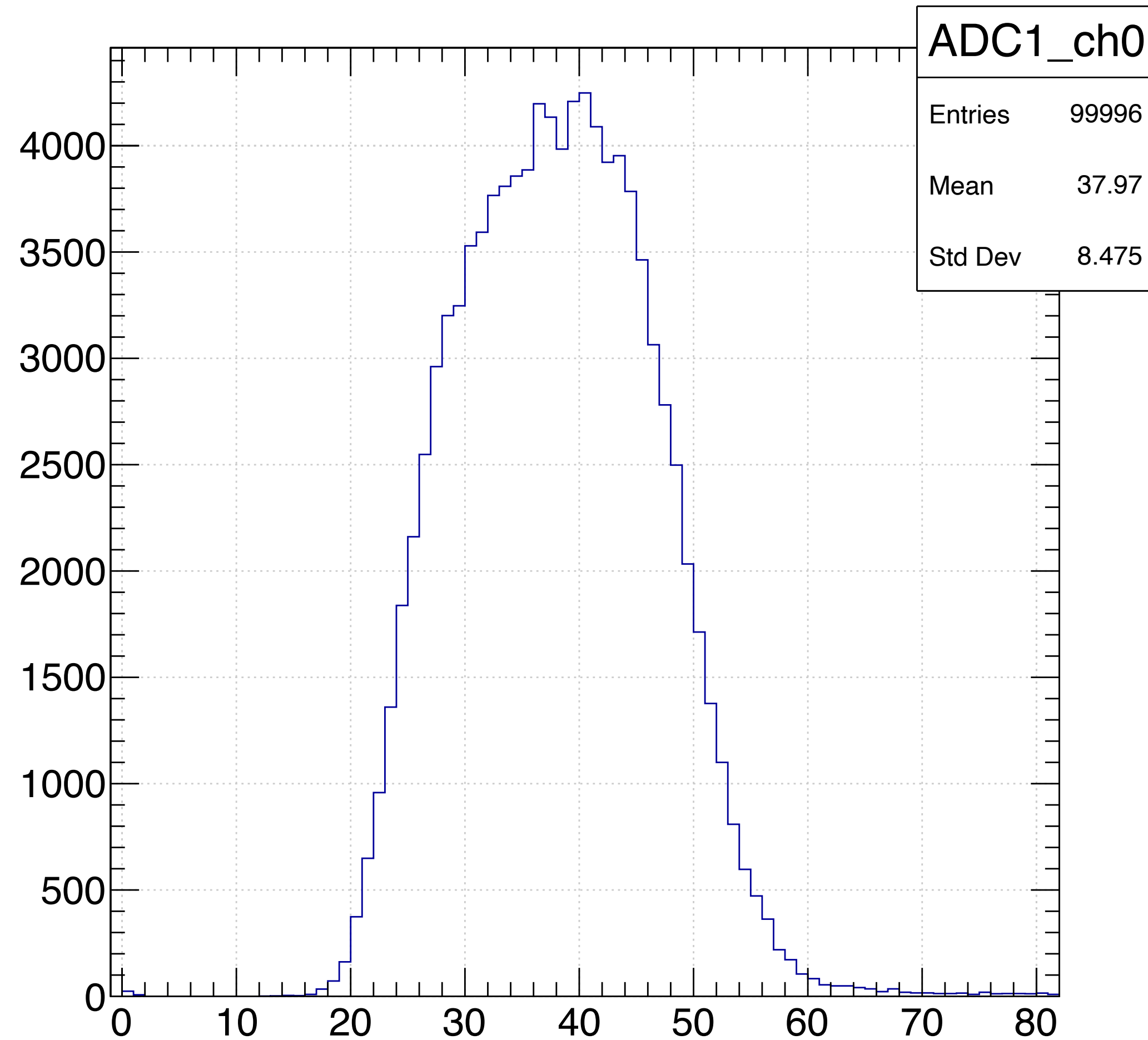


放射線源, 宇宙線で ADC 分布を見してみる

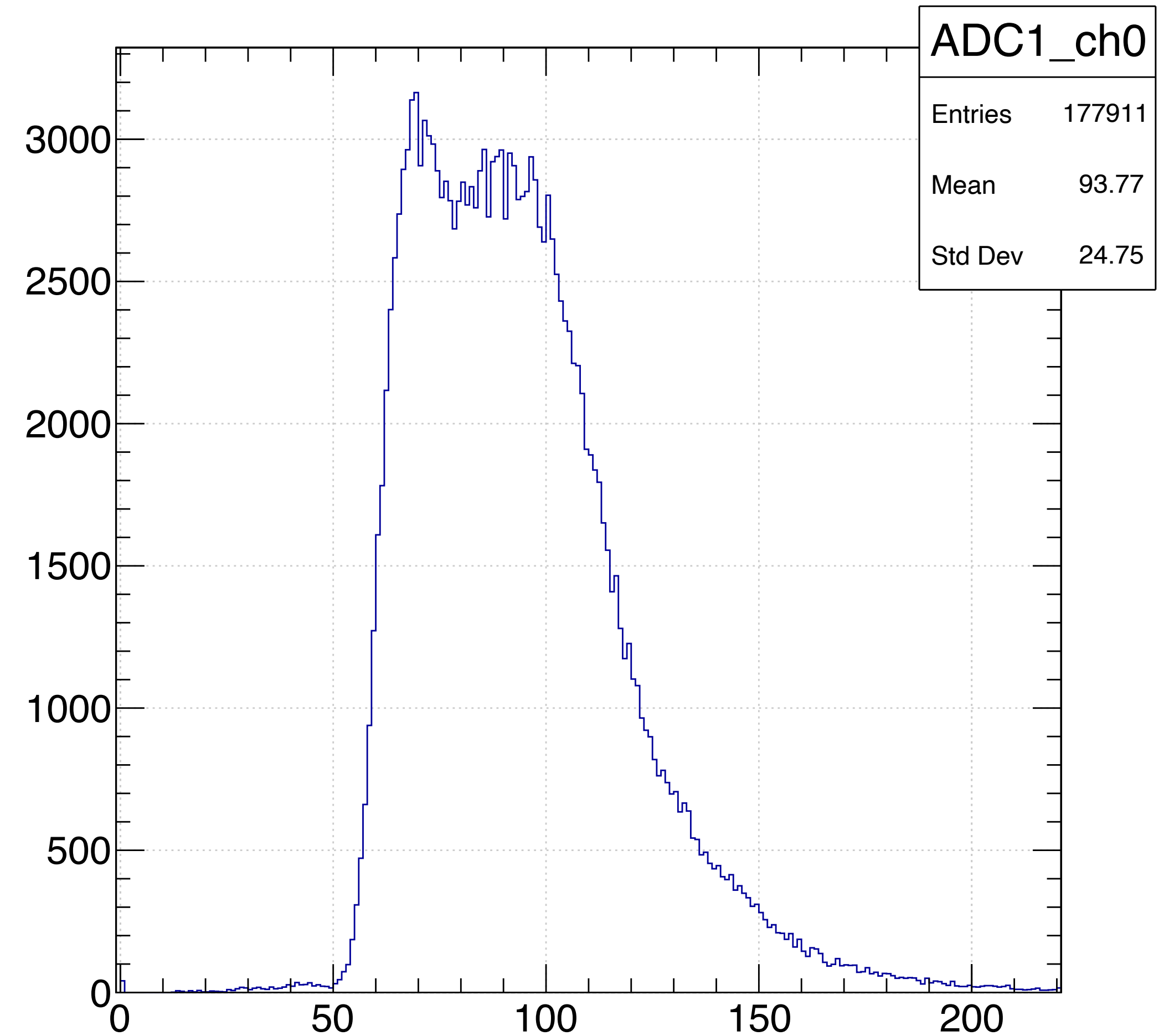
シンチレータ + PMT, ^{137}Cs 線源

シンチレータ + PMT, 宇宙線

ADC1_ch0 with cut, zoomed

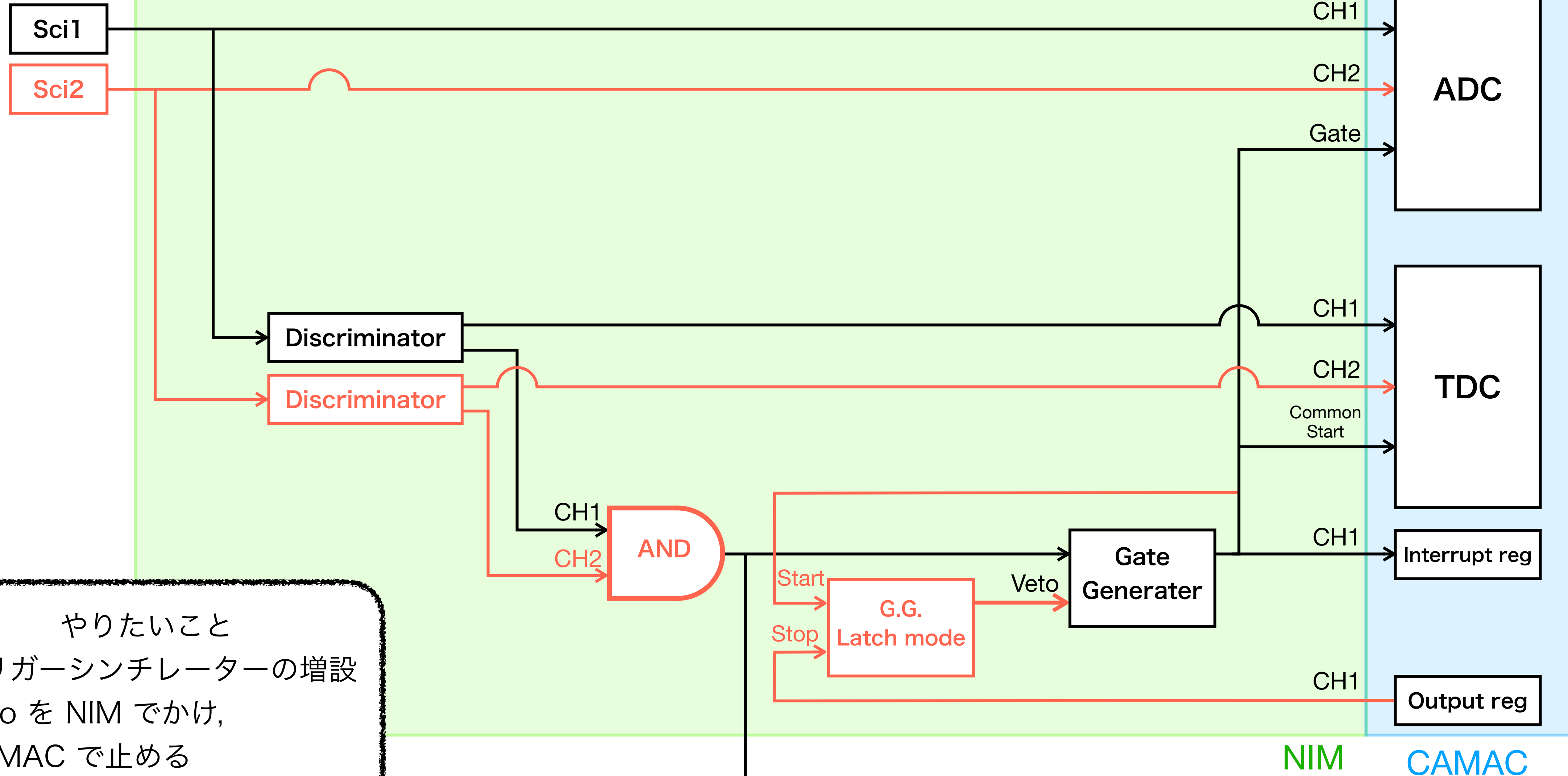


ADC1_ch0, zoomed



テストベンチ用外部トリガーデータ収集システムの予定

概念図、シンチレータ2つ



やりたいこと

- ・トリガーシンチレータの増設
- ・Veto を NIM でかけ, CAMAC で止める

NIM CAMAC

赤い部分を増設・変更予定

これから

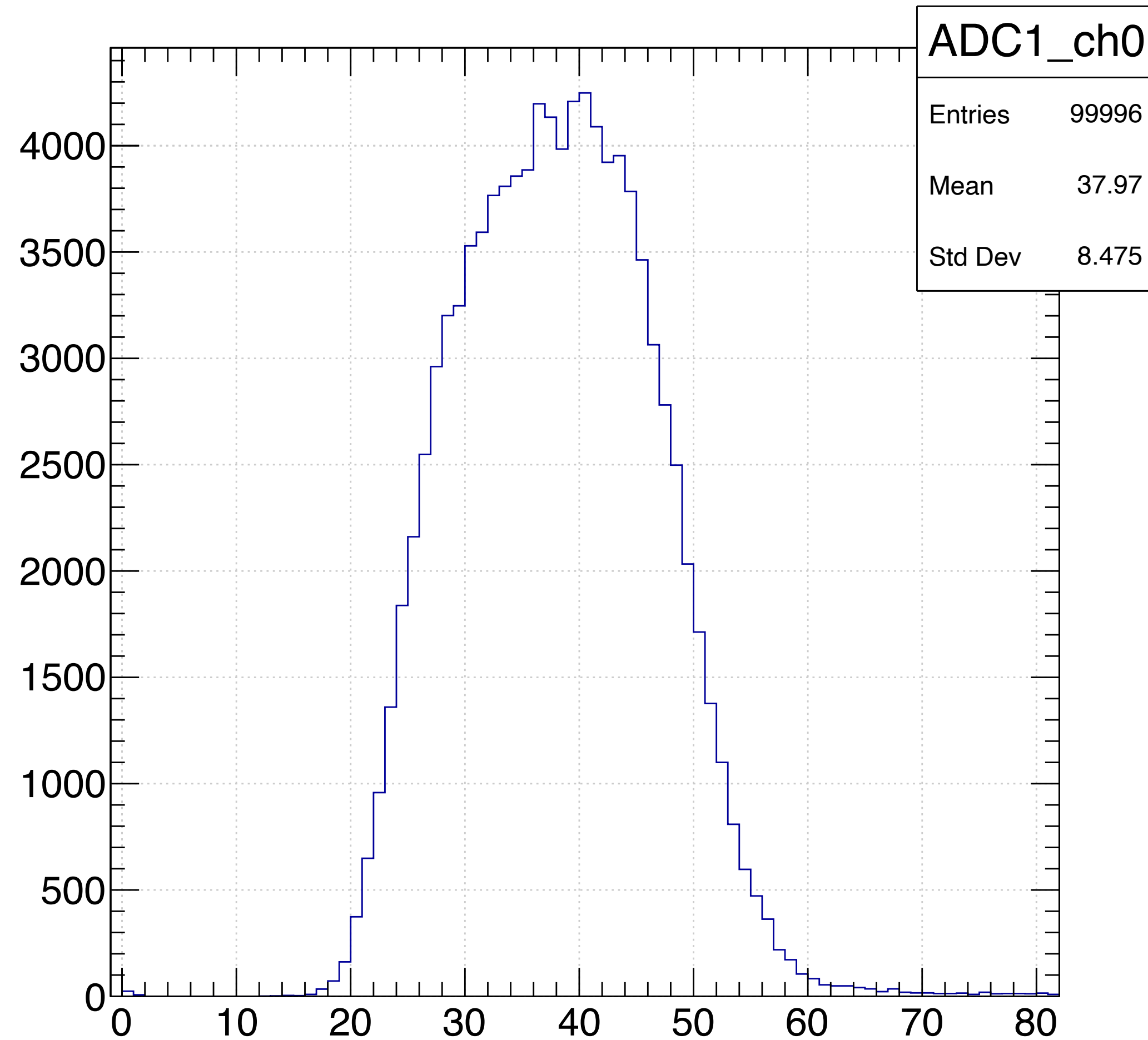
- INTT テストベンチ用 PC へ移植
- INTT DAQ との連動
- INTT DAQ の GUI 改良
- オフライン解析で INTT のイベントと外部トリガーのイベントが同期できるか確認
- INTT とトリガーシンチレーターで宇宙線の測定, 解析を試みる

バックアップ・作業中

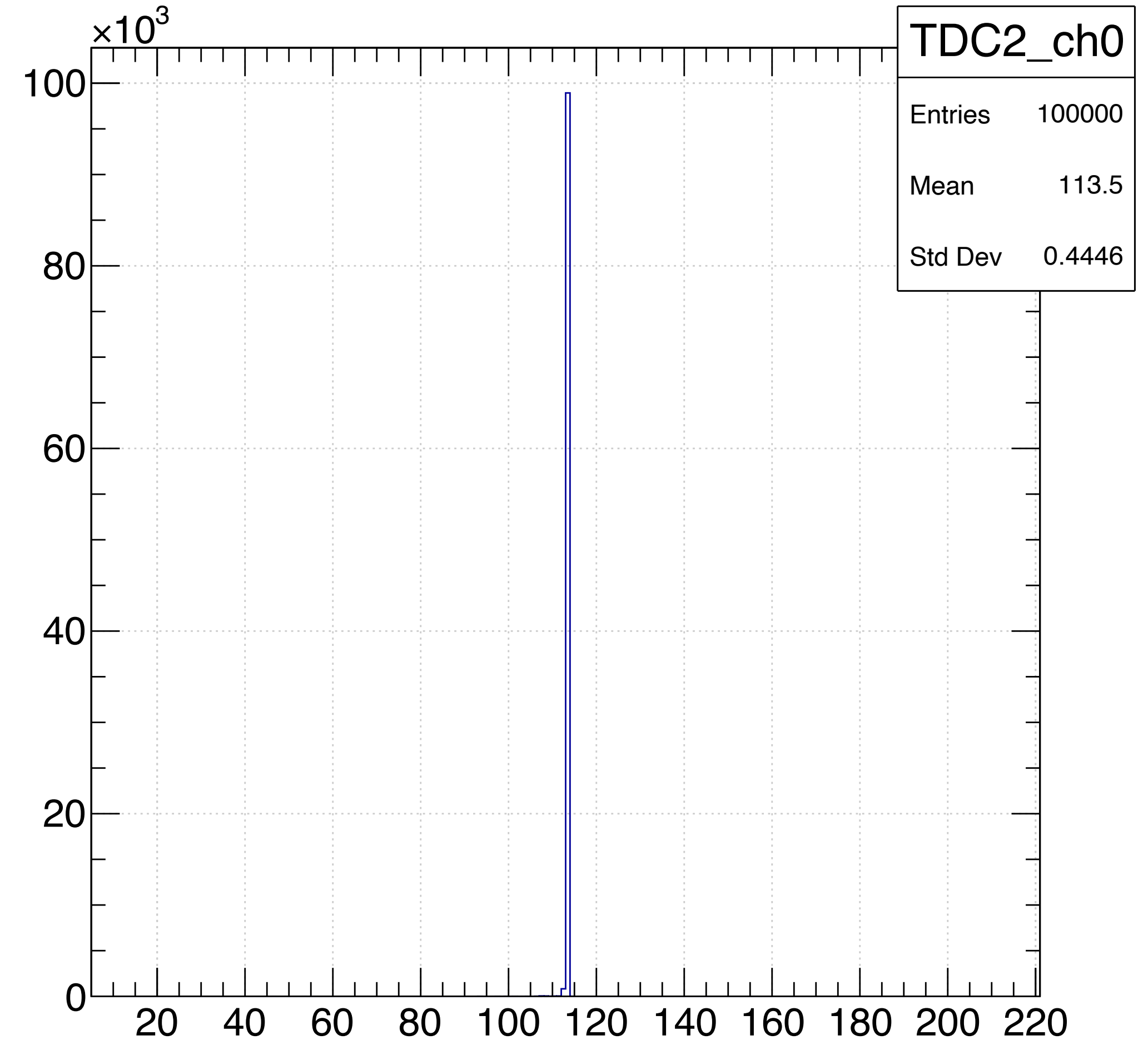
テストベンチ用外部トリガーデータ収集システム

シンチレータ + PMT, ^{137}Cs 線源

ADC1_ch0 with cut, zoomed



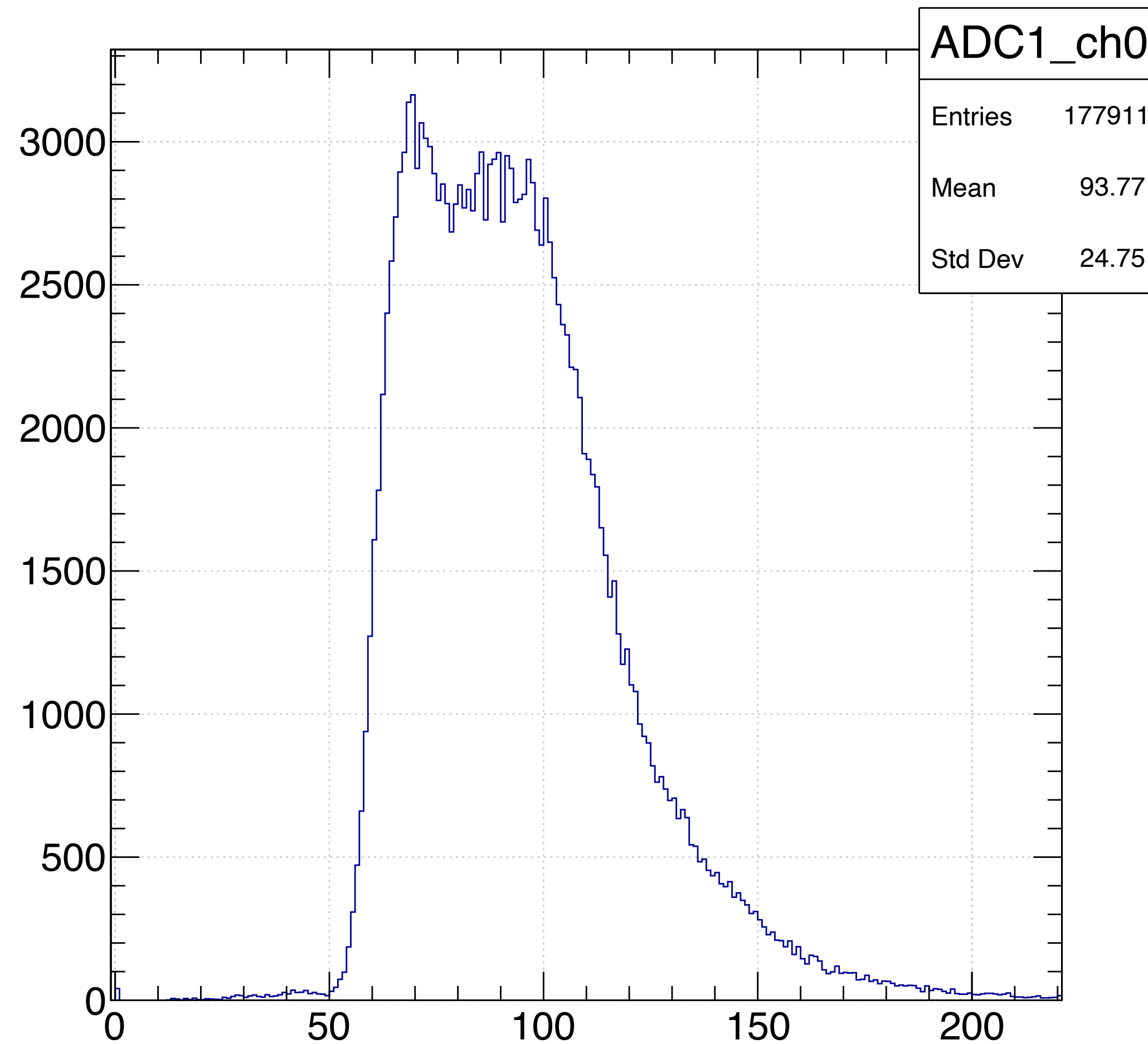
TDC2_ch0, zoomed



テストベンチ用外部トリガーデータ収集システム

シンチレータ + PMT, 宇宙線

ADC1_ch0, zoomed



TDC2_ch0, zoomed

