

# ZDC EMCAL electronics and data acquisition

---

SHI HONG YAO

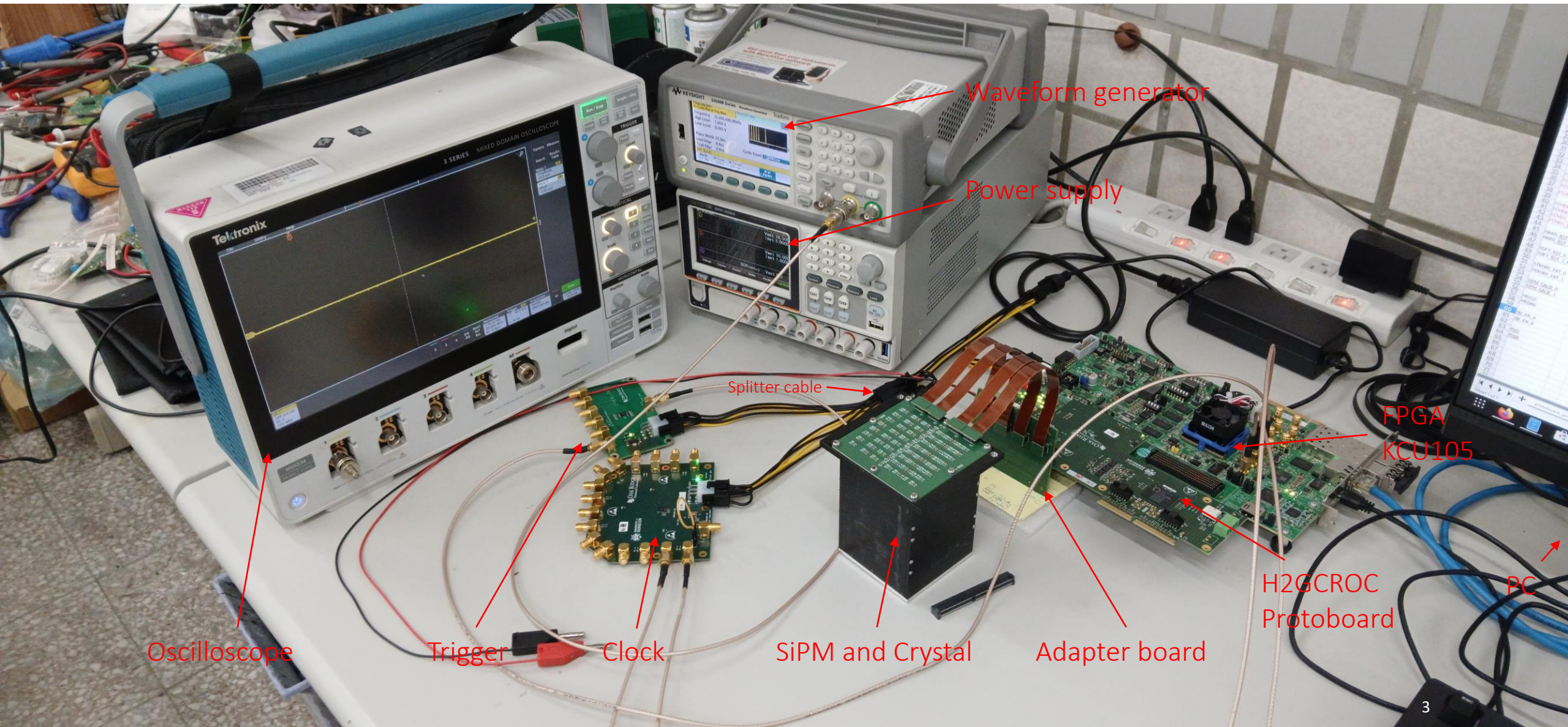
2026/05/01

# Outline

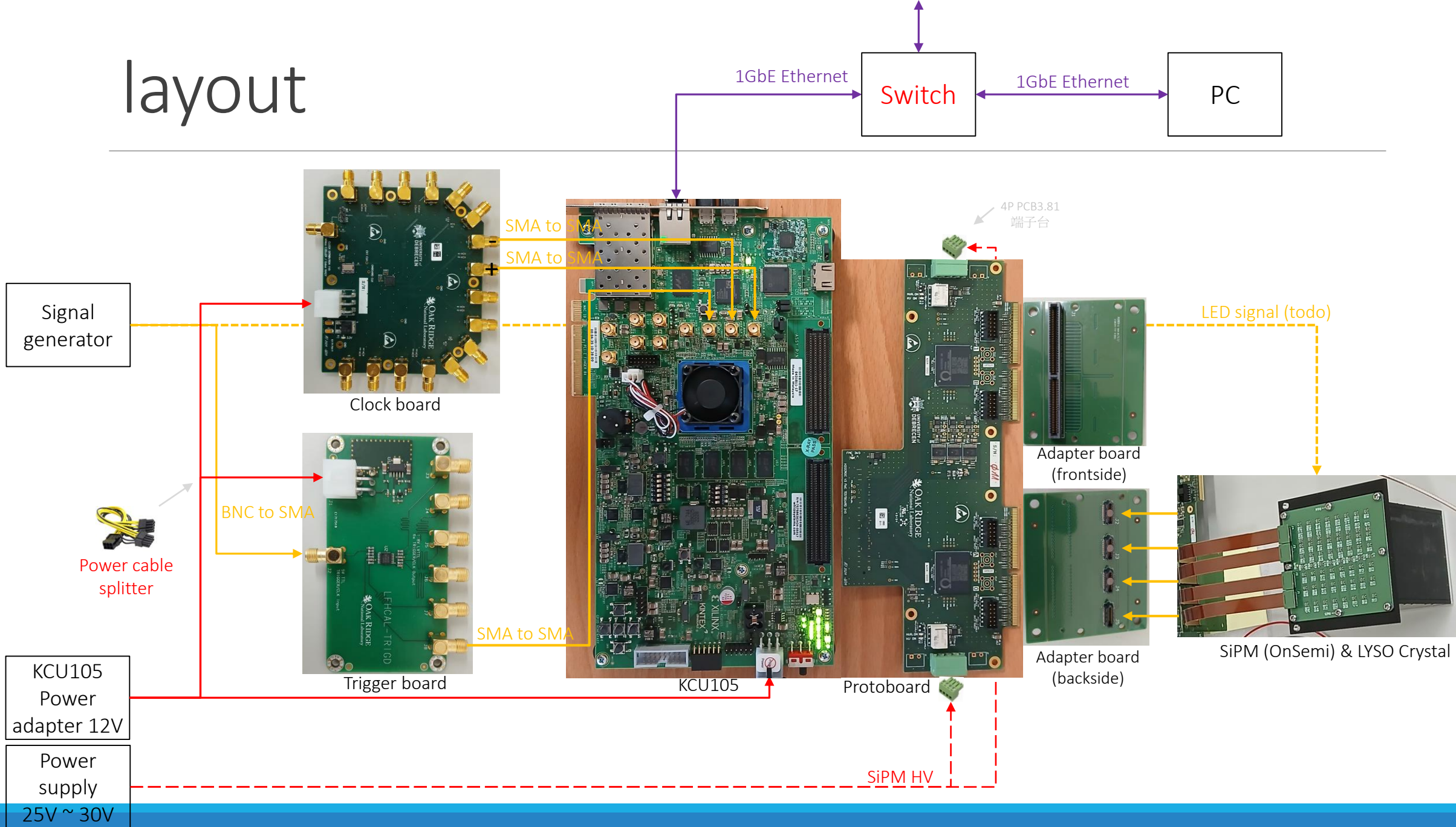
---

- ◇ Introduce electronics components and the DAQ system
- ◇ Test results using the old firmware
- ◇ Recent upgrade using the new firmware

# electronics



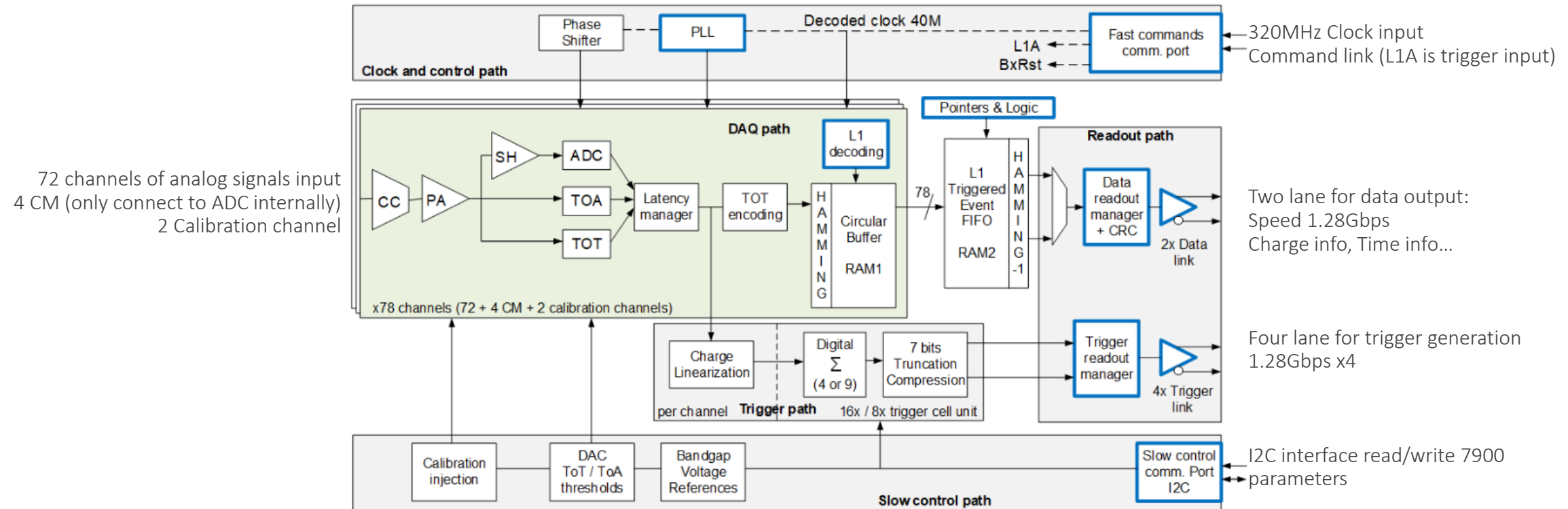
# layout



# H2GCROC

## 1 H2GCROC3b: architectural overview

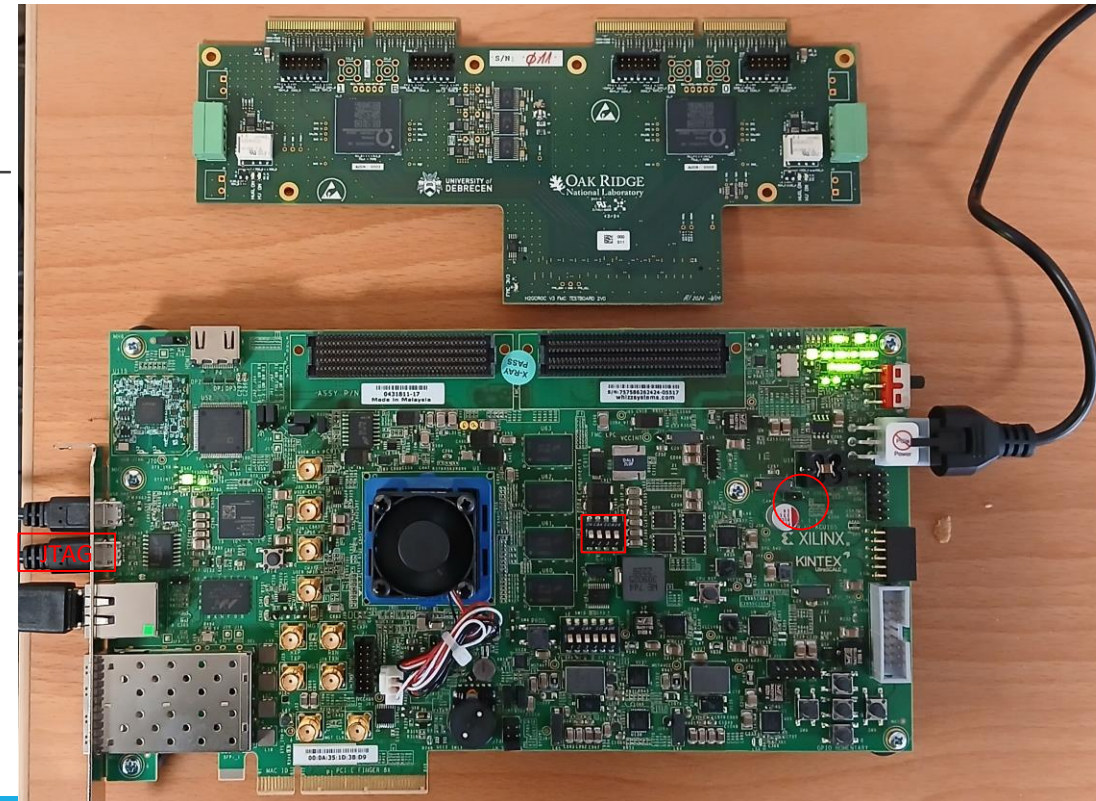
The overall block diagram is described as per below.



CC: Current conveyor (VFE), calibration SiPM gain variation  
 PA: Preamplifier  
 TOT: Time Over Threshold, dedicated discriminator and TDC (50ps), 12 bit  
 TOA: Time Of Arrival, dedicated discriminator and TDC (24ps), 12 bit  
 SH: Shaper  
 ADC: 10bit 40MHz SARADC

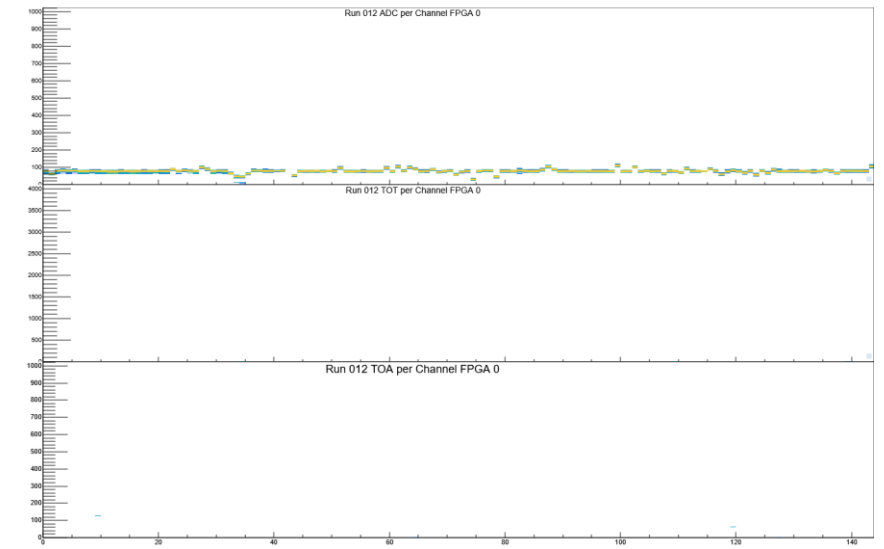
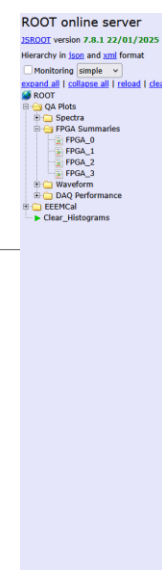
# FPGA & Firmware

- ◇ Firmware: v4.11
- ◇ Use Vivado to programming it
- ◇ FPGA IP address = 10.1.2.208 + DIP SWITCH[2:0]
- ◇ Must insert jumper J14 on the FPGA board
- ◇ <https://web.tresorit.com/l/86zmA#BE8fq4of3N8pFNz-cXpgg&ZZ8blfjDltJSr5qvdDoZYuGsgz6LDm0Q>

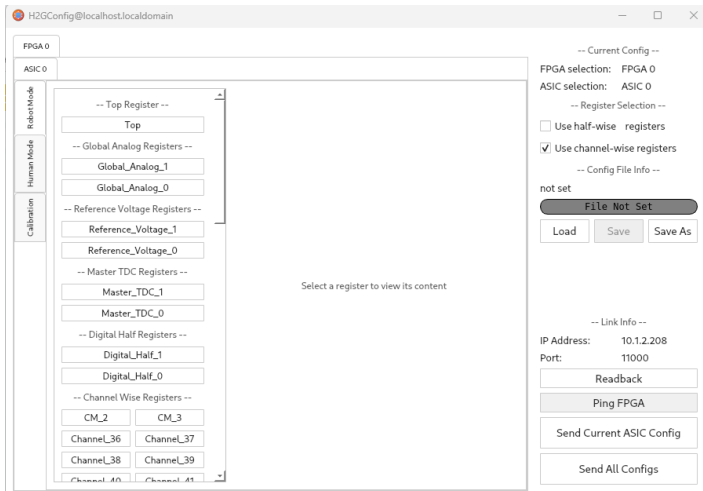


# Software

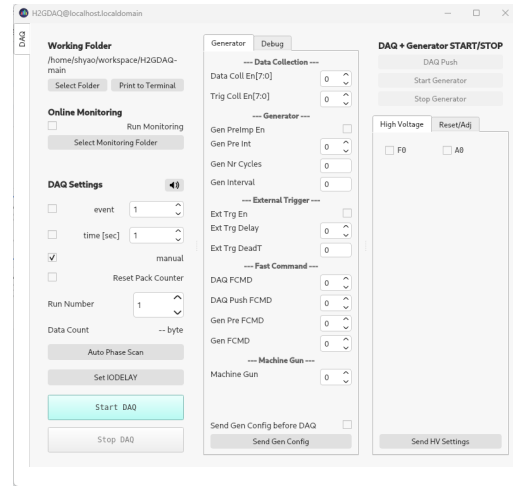
- ◇ <https://gitlab.cern.ch/sjia/H2GConfig.git>
- ◇ <https://gitlab.cern.ch/sjia/H2GDAQ.git>
- ◇ [https://gitlab.cern.ch/sjia/H2GCalib\\_3B](https://gitlab.cern.ch/sjia/H2GCalib_3B.git)
- ◇ [https://github.com/tlprotzman/h2g\\_online\\_monitoring](https://github.com/tlprotzman/h2g_online_monitoring)
- ◇ [https://github.com/tlprotzman/h2g\\_decode](https://github.com/tlprotzman/h2g_decode)



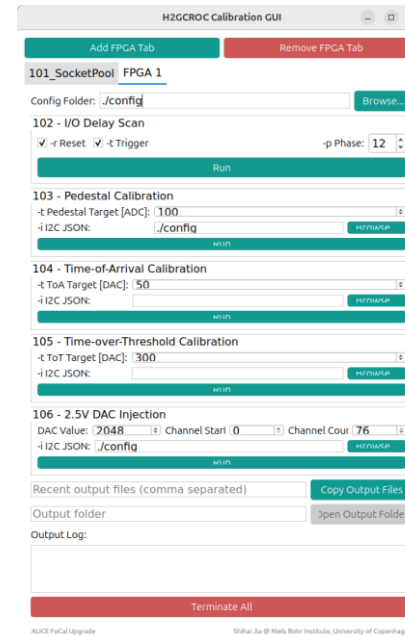
online monitoring (Web)



H2GConfig



H2GDAQ



H2GCalib\_3B

```

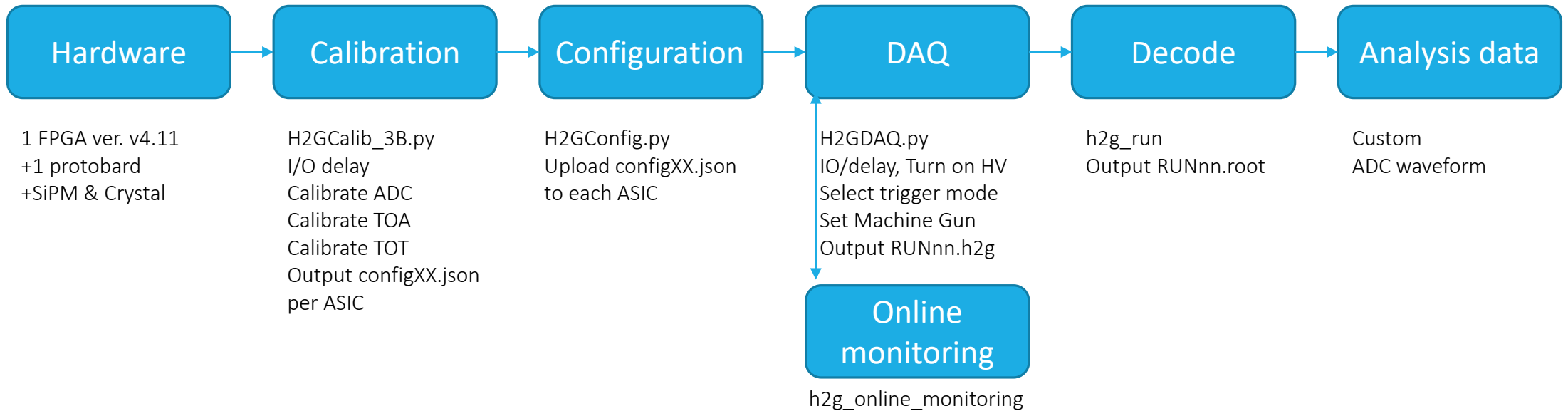
./build/h2g_run -r 14 -d 0 -n 1
$ root
root [1] TFile f("Run014.root")
root [2] events->Print()
*****
*Tree :events : Events
*Entries : 98258 : Total = 7360953537 bytes File Size = 492028194 *
* : : Tree compression factor = 14.96
*****
*Br 0 :event_number : event_number/i
*Entries : 98258 : Total Size= 395183 bytes File Size = 138994 *
*Baskets : 18 : Basket Size= 51200 bytes Compression= 2.84 *
*.....*
*Br 1 :timestamps : timestamps[1]/i
*Entries : 98258 : Total Size= 395145 bytes File Size = 114179 *
*Baskets : 18 : Basket Size= 51200 bytes Compression= 3.45 *
*.....*
*Br 2 :num_samples : num_samples/i
*Entries : 98258 : Total Size= 395161 bytes File Size = 4019 *
*Baskets : 18 : Basket Size= 51200 bytes Compression= 98.15 *
*.....*
*Br 3 :adc : adc[144][32]/i
*Entries : 98258 : Total Size= 1811620311 bytes File Size = 408563585 *
*Baskets : 5745 : Basket Size= 7088128 bytes Compression= 4.43 *
*.....*
*Br 4 :toa : toa[144][32]/i
*Entries : 98258 : Total Size= 1811620311 bytes File Size = 22647543 *
*Baskets : 5745 : Basket Size= 7088128 bytes Compression= 79.99 *
*.....*
*Br 5 :tot : tot[144][32]/i
*Entries : 98258 : Total Size= 1811620311 bytes File Size = 8514786 *
*Baskets : 5745 : Basket Size= 7088128 bytes Compression= 212.75 *
*.....*
*Br 6 :hamming : hamming[144][32]/i
*Entries : 98258 : Total Size= 1811643307 bytes File Size = 8529147 *
*Baskets : 5745 : Basket Size= 7089664 bytes Compression= 212.39 *
*.....*
*Br 7 :hit_max : hit_max[144]/i
*Entries : 98258 : Total Size= 56630819 bytes File Size = 21343750 *
*Baskets : 353 : Basket Size= 221184 bytes Compression= 2.65 *
*.....*
*Br 8 :hit_pedestal : hit_pedestal[144]/i
*Entries : 98258 : Total Size= 56632604 bytes File Size = 22016214 *
*Baskets : 353 : Basket Size= 221696 bytes Compression= 2.57 *
*****

```

decode

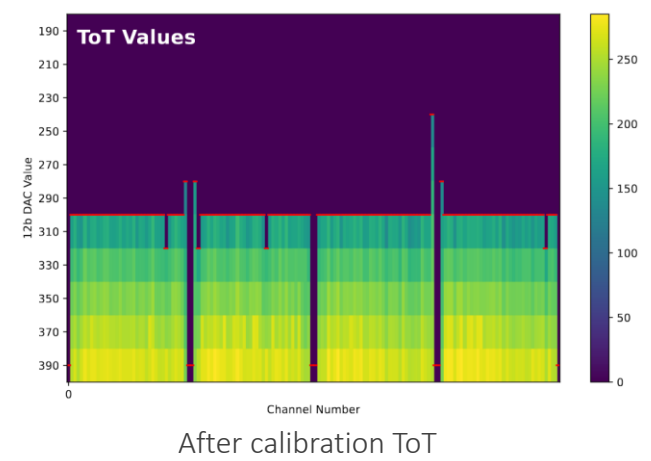
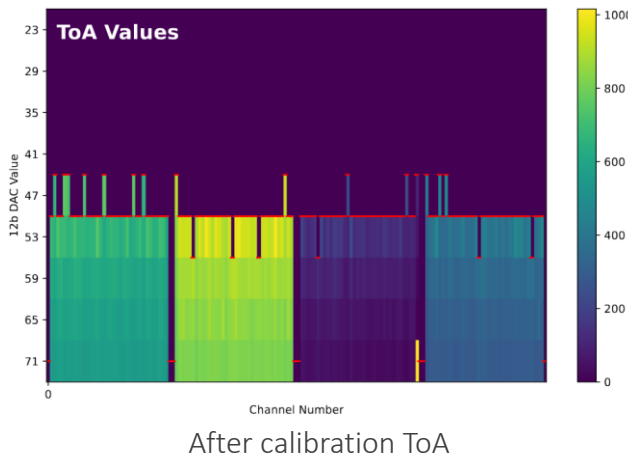
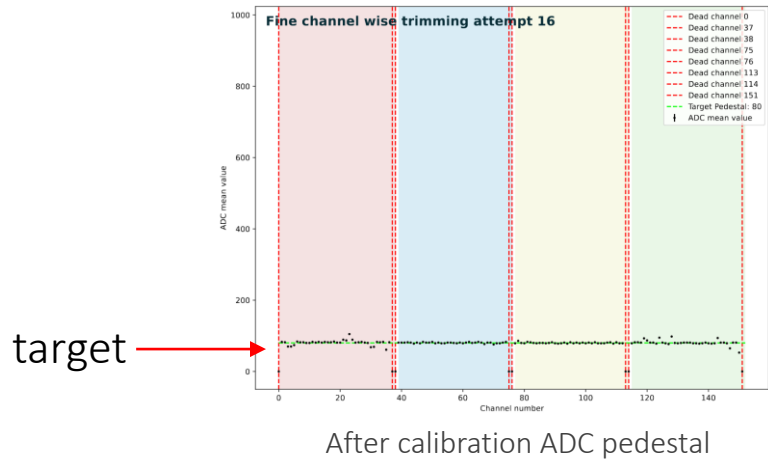
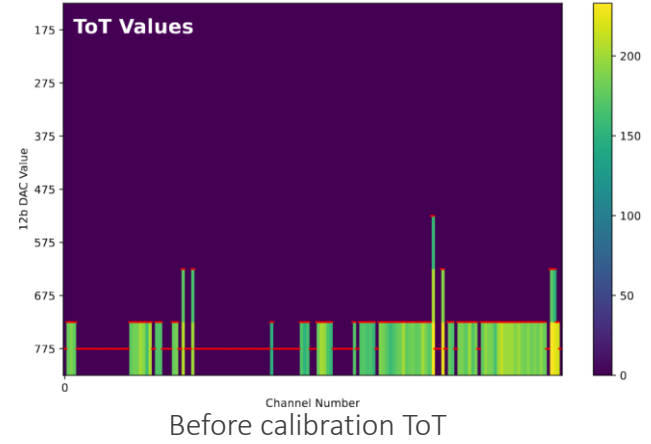
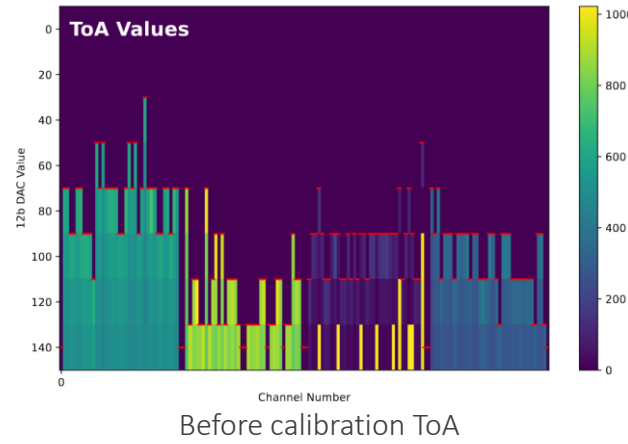
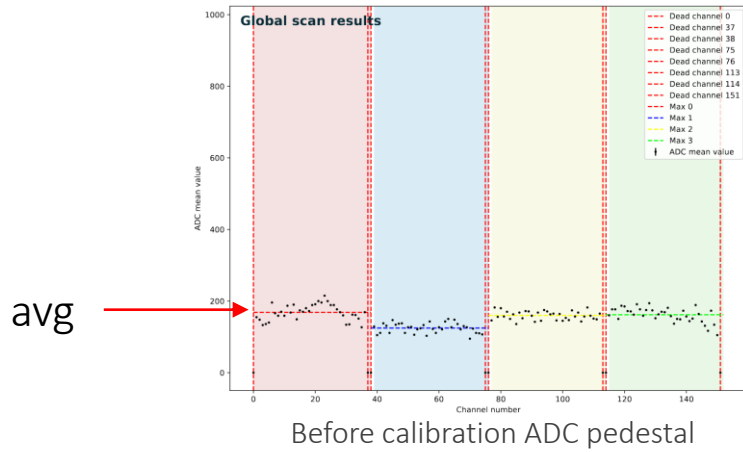
# Steps

◇ The entire process is roughly as follows:



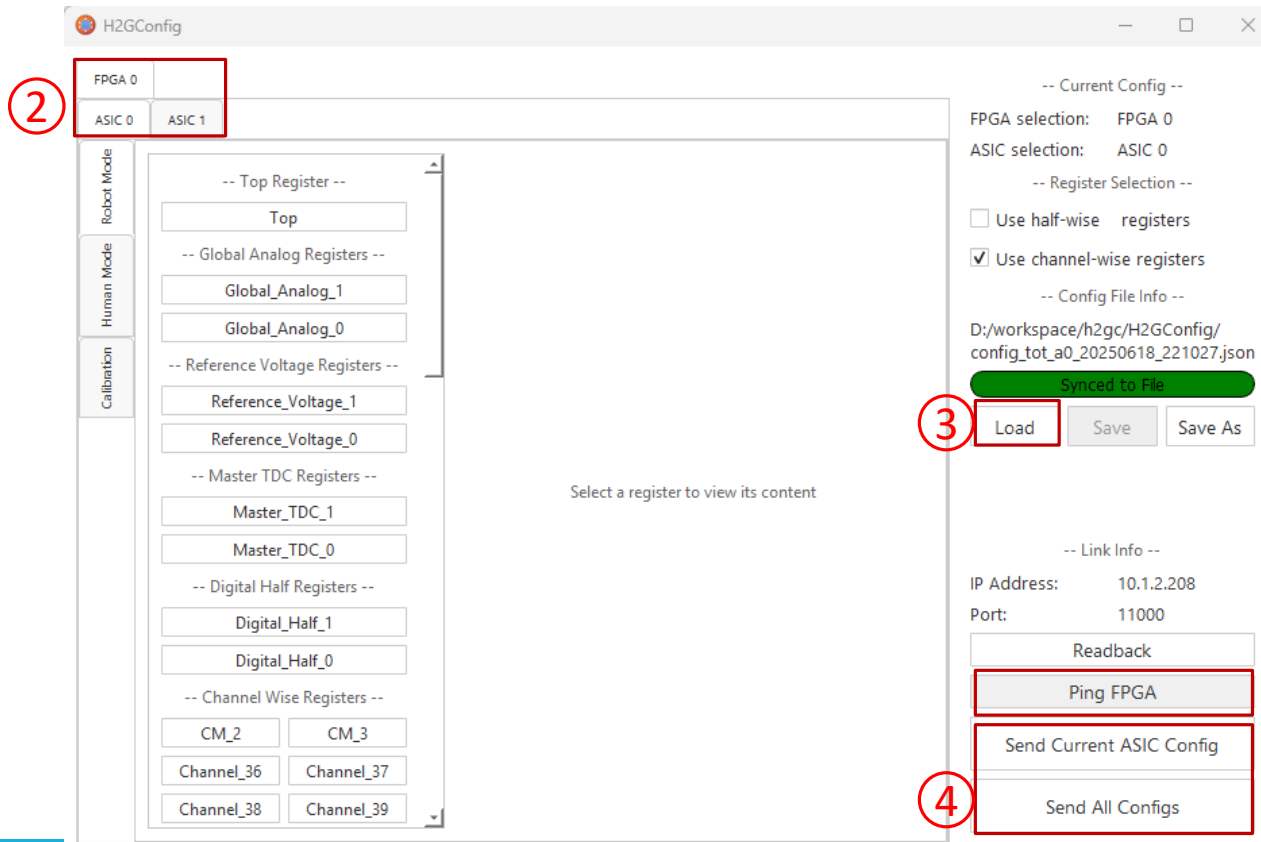
# Calibration

## ◇ H2GCalib\_3B



# H2GConfig & H2GDAQ

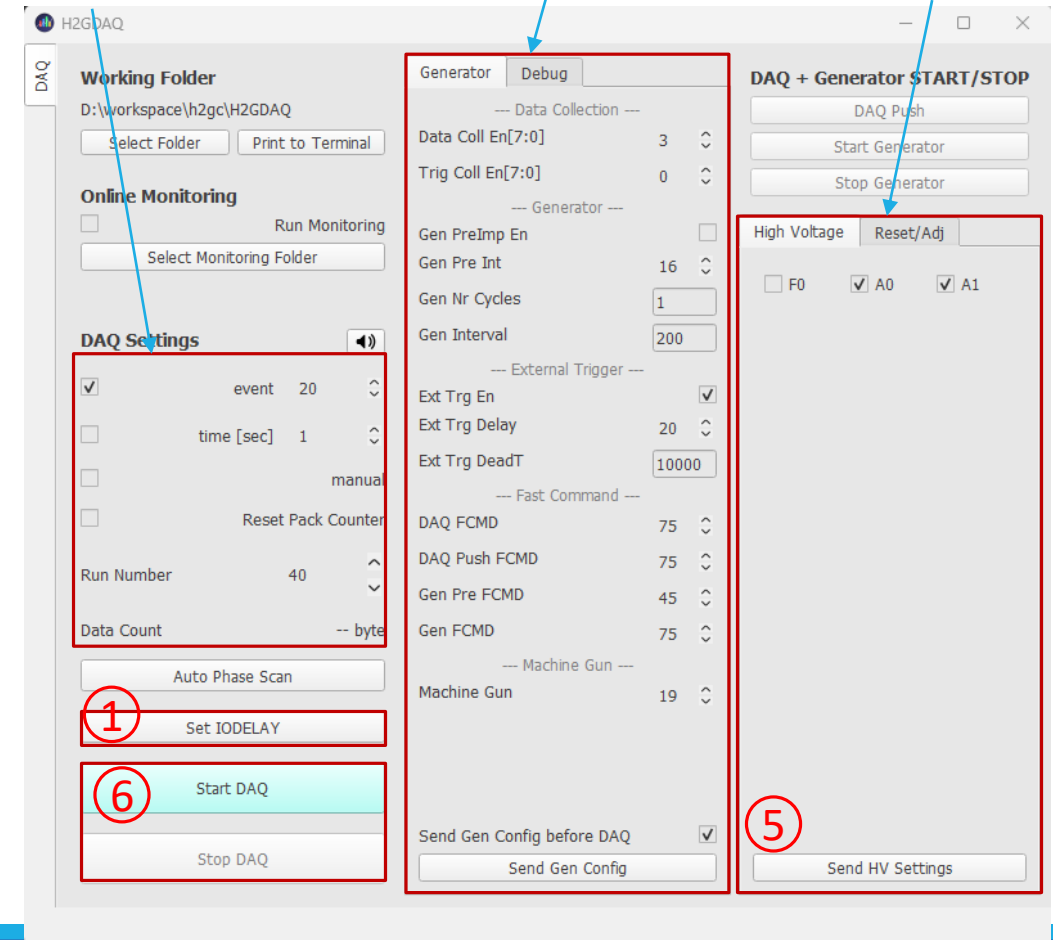
- ◇ H2GConfig - upload configuration to the ASIC
- ◇ H2GDAQ - Start/Stop DAQ



DAQ stop condition:  
By event count, by  
time or manual.  
Output Run040.h2g

FPGA & trigger  
related setting

high voltage relay setting  
A0 = ASIC 0 only  
A1 = ASIC1 only  
F0 = ASIC0 and ASIC1 both

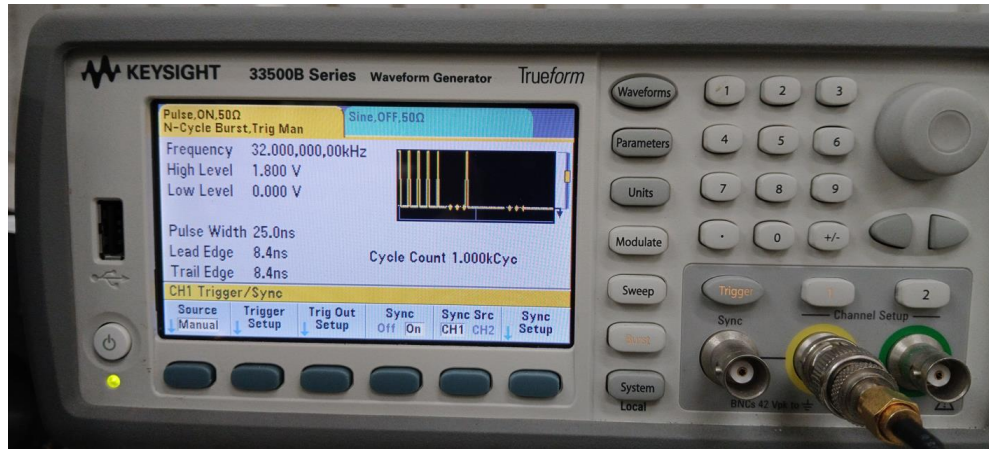


# Trigger mode

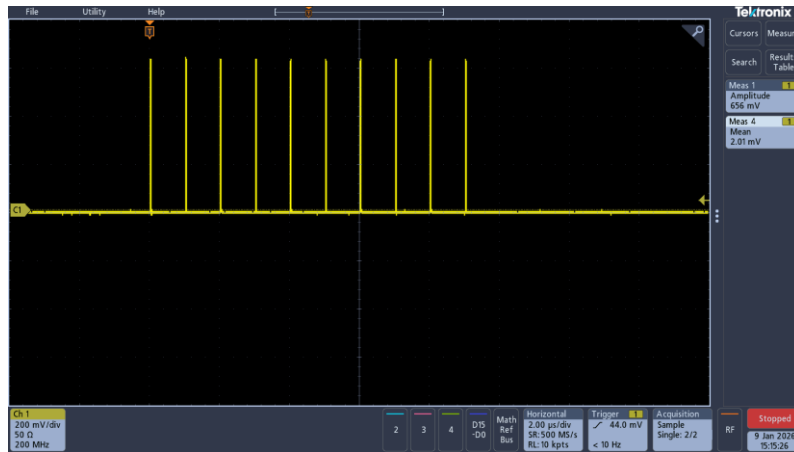
---

- ◇ In practice, the external trigger signal is sent to the FPGA.
- ◇ Four possible mode
  - ◇ DAQ Push
  - ◇ Generator
  - ◇ Internal trigger
  - ◇ External trigger
- ◇ Common parameter - **Machine Gun**
  - ◇ The FPGA firmware generates a series of read commands and sends them to H2GCROC.
  - ◇ The real number of fast read commands sent will be the setting value plus 1.
  - ◇ This field has 8 digits, but the reasonable value range is 1 to 31.

# External trigger



test signal : 32KHz, 1000 pulse



test signal : 1MHz, 10 pulse

H2GDAQ@localhost.localdomain

Working Folder  
/home/shyao/workspace/H2GDAQ-main  
Select Folder Print to Terminal

Online Monitoring  
 Run Monitoring  
Select Monitoring Folder

DAQ Settings  
 event 1  
 time [sec] 1  
 manual  
 Reset Pack Counter  
Run Number 1  
Data Count -- byte  
Auto Phase Scan  
Set IODELAY  
Start DAQ  
Stop DAQ

Generator Debug

--- Data Collection ---  
Data Coll En[7:0] 0  
Trig Coll En[7:0] 0

--- Generator ---  
Gen PreImp En   
Gen Pre Int 0  
Gen Nr Cycles 0  
Gen Interval 0

--- External Trigger ---  
Ext Trg En   
Ext Trg Delay 0  
Ext Trg DeadT 0

--- Fast Command ---  
DAQ FCMD 0  
DAQ Push FCMD 0  
Gen Pre FCMD 0  
Gen FCMD 0

--- Machine Gun ---  
Machine Gun 0

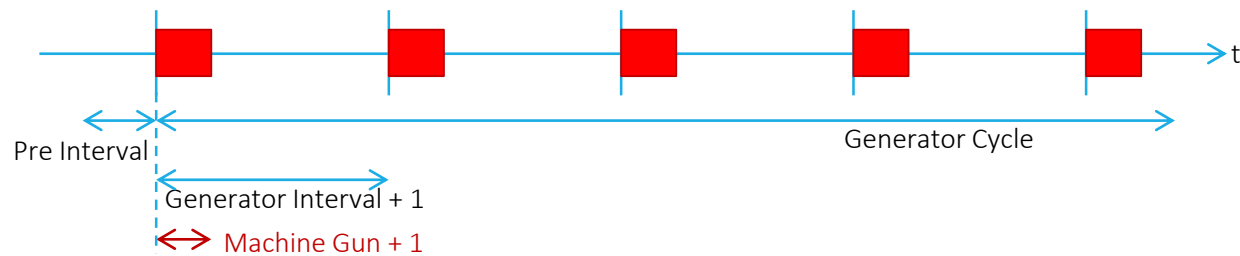
Send Gen Config before DAQ   
Send Gen Config

DAQ + Generator START/STOP  
DAQ Push  
Start Generator  
Stop Generator

High Voltage Reset/Adj  
 F0  A0  
Send HV Settings

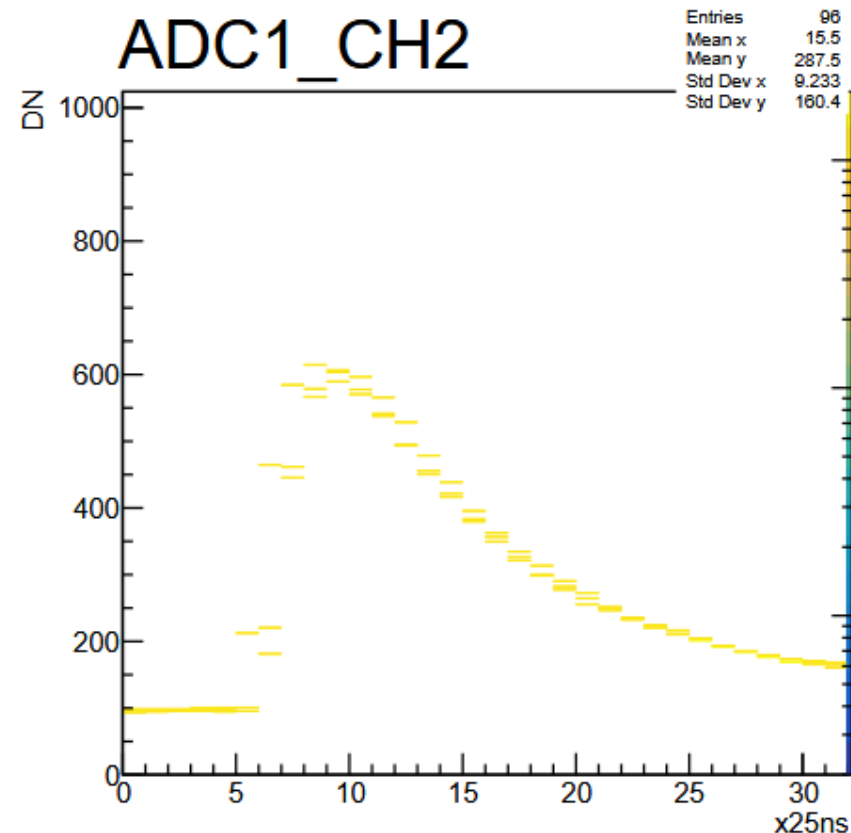
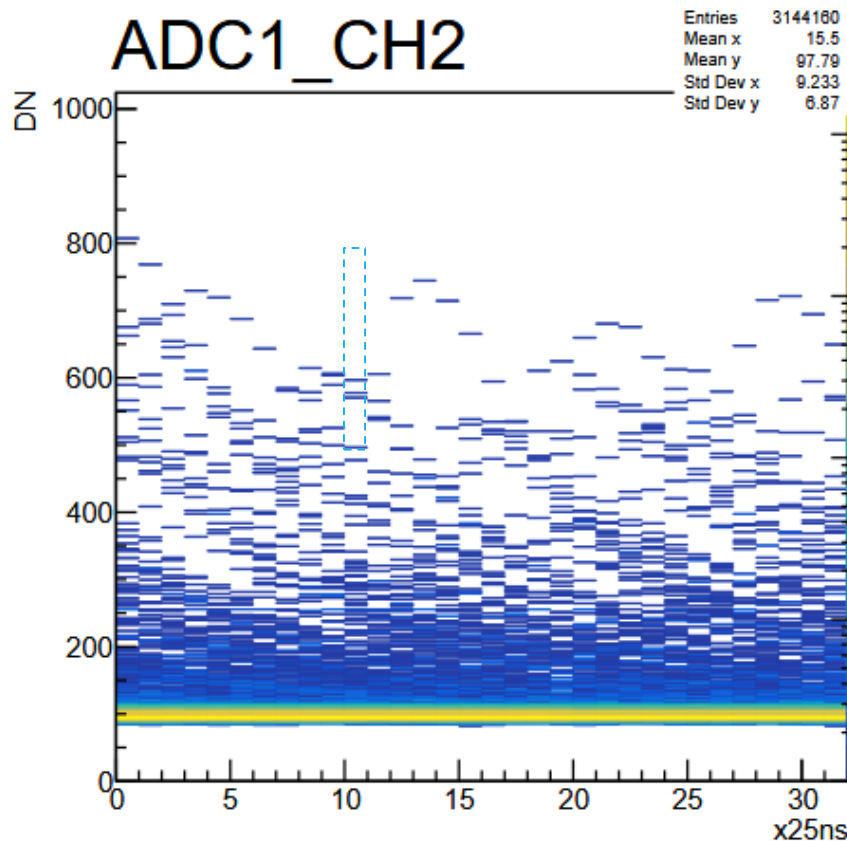
# Generator

- ◇ The generator function is enabled by user commands and is executed once after being enabled.
- ◇ Three parameters
  - ◇ Pre Interval – delay before the first fire, 16 bit, Unit 25ns
  - ◇ Generator Cycle - fire how many time, 32 bit
  - ◇ Generator Interval – Time between two fires, 32 bit, Unit 25ns
- ◇ Number of fast read command send = (Generator Cycle) \* (Machine Gun + 1)



# DAQ test

- ◇ Set generator parameter to 1KHz rate, 100k cycles. Set machine gun to 32 read
- ◇ Set High voltage to **28V**. Start DAQ then decode datafile and analysis.

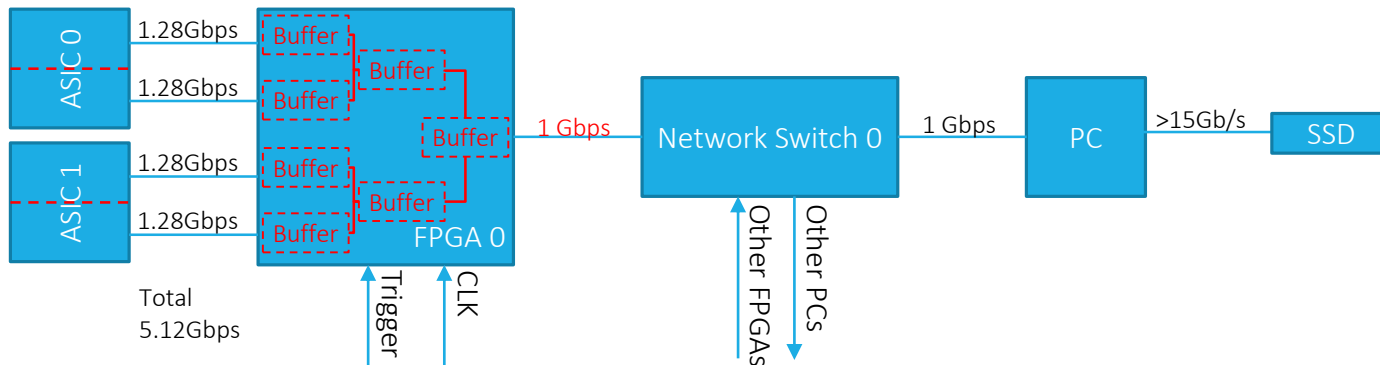


ADC waveform  
x-axis: time  
y-axis: amplitude

All other connected 63 channels is similar.

# Data rate / trigger rate

- ◇ Each channel generates 4 bytes of data (include TOA, TOT, ADC) every 25 nanoseconds.
- ◇ A DAQ Frame is 1280+32 bits, 38 channels, it takes 1.025 microseconds to transmit data to FPGA.
- ◇ If maximum consecutively read (Machine Gun = 31, 800ns window)
  - ◇ H2GCROC takes 32.8 microseconds to transmit data to FPGA.
  - ◇ FPGA takes 163.8 microseconds to transmit data to PC.
  - ◇ Maximum trigger rate is about 6kHz



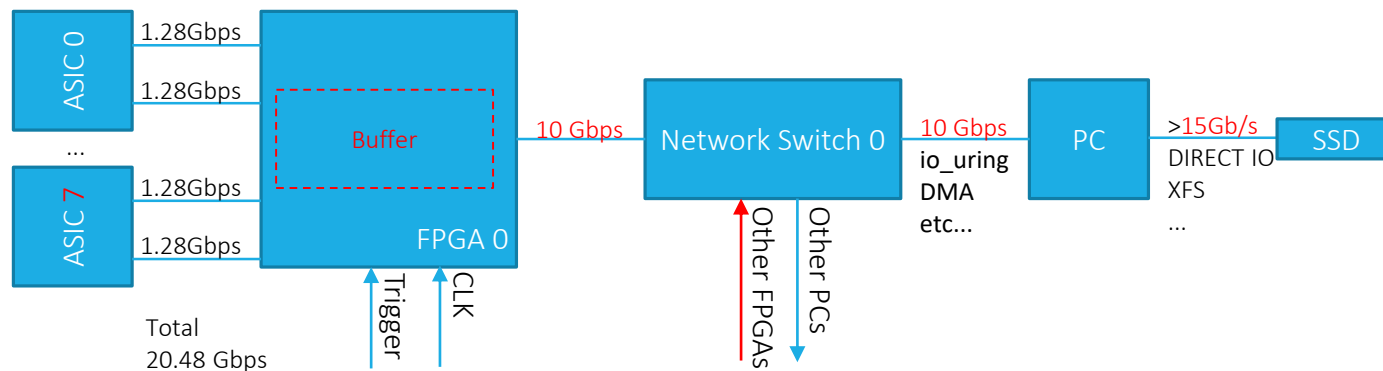
Data transfer in parallel, Each lane handle half channels. 5.12 Gbps

Any bottleneck will lead to longer dead time or potential data loss.

1312 bits = header + half channel + crc + idle

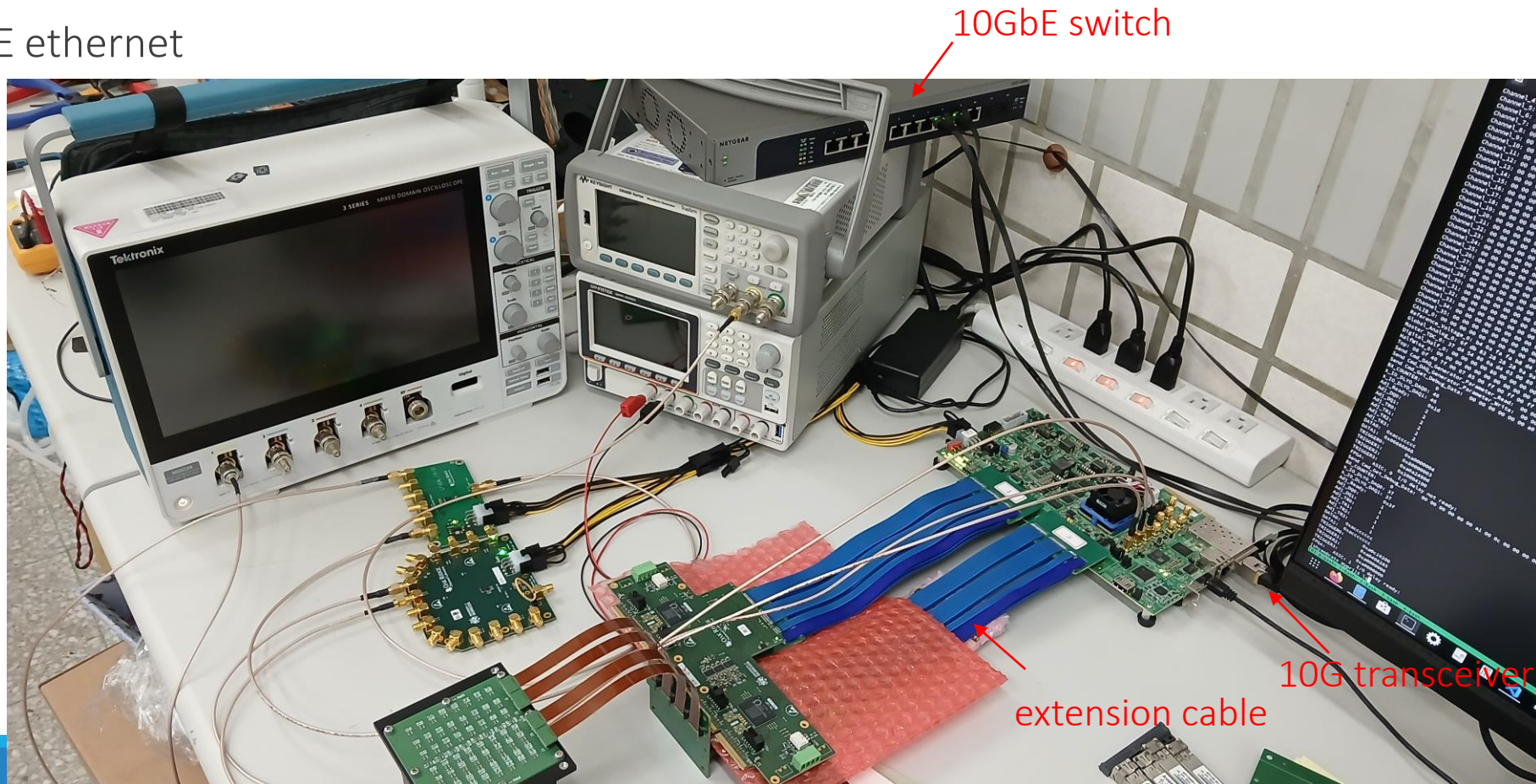
# Upgrade

- ◇ for large crystal with 10000+ channel, it require 100+ ASIC; 20+ FPGA; 200Gb+ bandwidth (worst case)
- ◇ Update FPGA Firmware to v5.069.
  - ◇ support more H2GCROC (up to eight) and 10GbE hardware
  - ◇ The data packet format changed, making it more efficient and convenient.
  - ◇ New functions (under study)
- ◇ Software
  - ◇ Faster calibration



# Hardware upgrade

- ◇ Use both FMC connectors on the FPGA evaluation board.
- ◇ Connect the two Protoboard using an extension cable.
- ◇ 10GbE ethernet



# Software upgrade

- ◇ Newer calibration software support this firmware
- ◇ git clone <https://gitlab.cern.ch/sjia/h2gcalibx>
- ◇ It also replace H2GConfig

This user interface renders all content in the terminal and is responsive to mouse input.

ssh remote

The screenshot shows the CalibX terminal interface. At the top, there are tabs for 'FPGA 1', 'FPGA 2', 'FPGA 3', and 'FPGA 4', with 'FPGA 3' selected. The main content area is titled 'FPGA Settings' and contains a form with the following fields:

- UDP JSON File: `/home/shyao/workspace/h2gcalibx/conf...` (with a 'Browse UDP JSON' button)
- ASIC Number: `2` (with an 'ASIC number' label)

Annotations with blue arrows point to various elements:

- 'Top Tab' points to the 'FPGA 3' tab.
- 'Side Tab' points to the 'FPGA Settings' tab in the left sidebar.
- 'config/common\_setting\_4\_11\_208.json' points to the 'UDP JSON File' field.
- 'input 4, press ENTER' points to the 'ASIC number' label.
- 'Start Socket Pool' points to the terminal output.
- 'Close Socket Pool' points to a button in the terminal.

The terminal output at the bottom shows:

```
▶ Starting UDP Socket Pool...
▶ Running command: python3 -u ./101_SocketPool.py
[Pool] Control → 127.0.0.1:6002
[Pool] Data → 127.0.0.1:6001
```

At the bottom of the terminal, there are keyboard shortcuts: `^n` New FPGA tab, `^w` Close FPGA tab, `q` Quit application, and `^p` palette.

# 202 PedestalX

- ◇ Pre-amp reset default setting
  - ◇  $R_f = 8 \Rightarrow 80K \text{ Ohm}$
  - ◇  $C_f = 10 \Rightarrow 500 \text{ fF}$
  - ◇  $C_{f\_Comp} = 10 \Rightarrow 500 \text{ fF}$
  - ◇  $\tau = R_f * (C_f + C_{f\_Comp}) = 80\text{ns} = 3 \text{ samples}$
- ◇ Conveyor Gain default setting
  - ◇  $CC = 8 \Rightarrow \text{Gain} = 0.2$

CC	0.025, 0.05, 0.1, 0.2	
$C_d$ (pF)	5, 10, 20	At the conveyor output and at the preamplifier input. To ensure the preamp stability
$R_f$ (Ohm)	10K, 20K, 40K, 80K	In parallel, these resistors provide 15 values to be adjusted with the $C_f$ and $C_{f\_comp}$ values
$C_f$ (fF)	50, 100, 200, 400	Added with the $C_{f\_comp}$ capacitors, provide the preamplifier constant time with $R_f$
$C_{f\_comp}$ (fF)	50, 100, 200, 400	Same purpose than $C_f$ capacitors but connected differently to improve the preamplifier stability.

The screenshot shows the CalibX configuration tool for the 202 PedestalX target. The 'Target pedestal' is set to 100. The 'Template JSON File' is `/home/shyao/workspace/h2gcalibx/config/default_2024Aug_config.json`. The following settings are configured:

- Feedback Resistor (RF): 8 (Unit: 10K (0-15))
- Feedback Capacitor (CF): 10 (Unit: 15fF (0-15))
- Current Conveyor Gain (CC): 8 (Unit: 0.025 (0-15))
- CF Compensation (CFComp): 10 (Unit: 15fF (0-15))

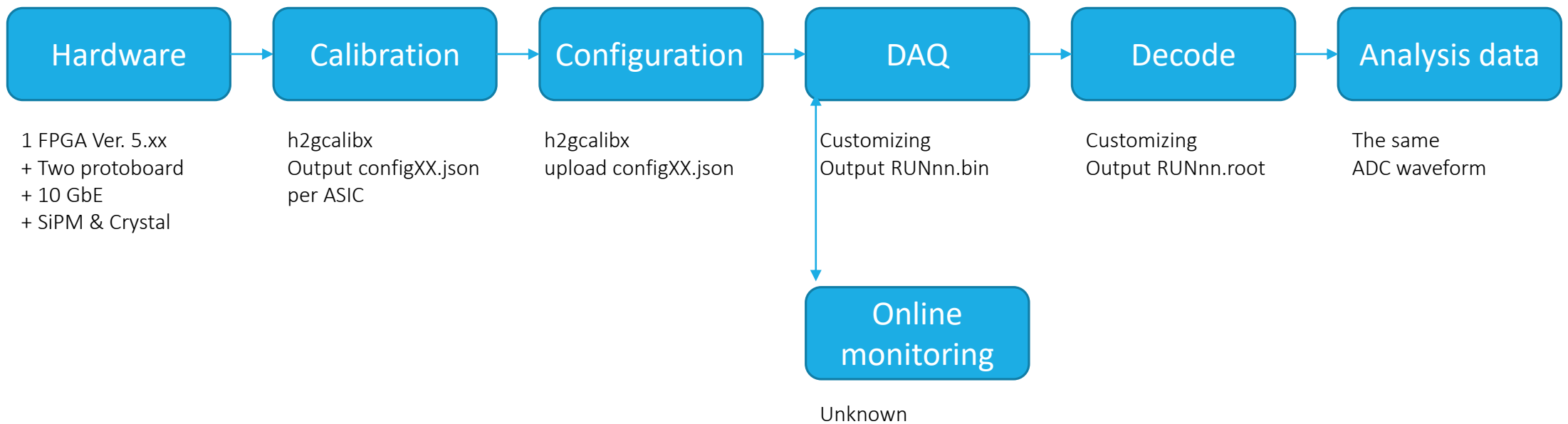
The 'Final Tune' section shows 15 attempts with channel values for Ch03, Ch04, Ch05, Ch39, Ch40, and Ch41. For example, in attempt 4, Ch03 is 115, Ch04 is 150, Ch05 is 125, Ch39 is 102, Ch40 is 99, and Ch41 is 97.

At the bottom, there are buttons for 'Start Socket Pool' and 'Close Socket Pool'. The status bar shows connection logs for the control and data sockets, and application controls like 'New FPGA tab', 'Close FPGA tab', and 'Quit application'.

# DAQ update?

---

- ◇ Unfortunately, I have no new DAQ software like H2GDAQ and decoder.
- ◇ Customizing one

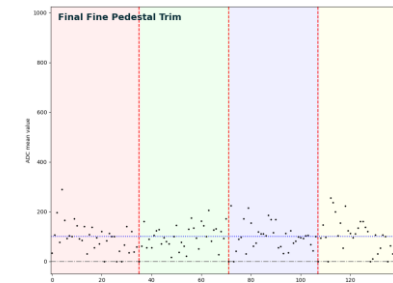
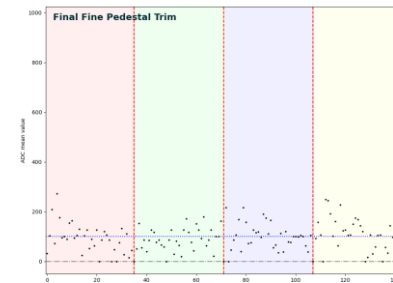
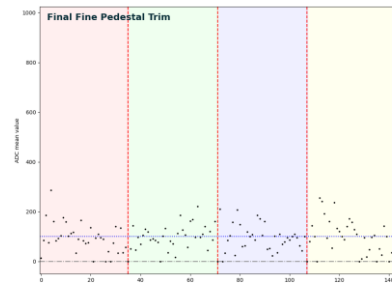
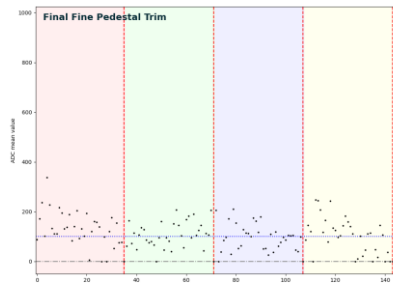


# Q/A

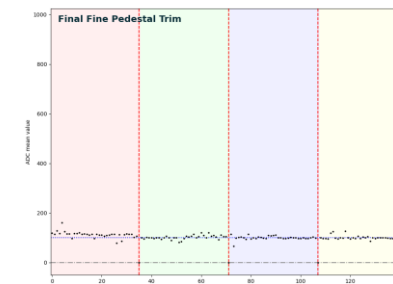
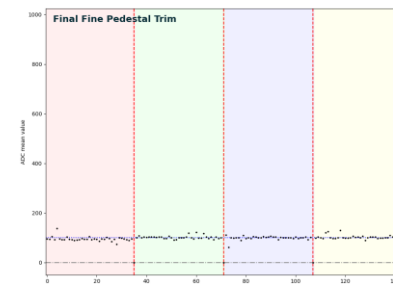
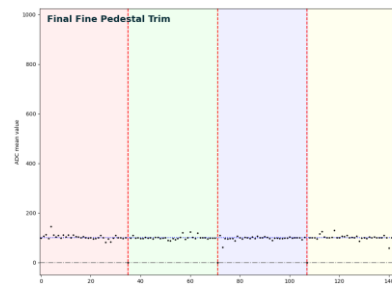
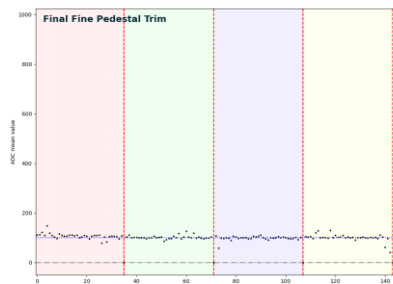
---

◇ Thank you for listening

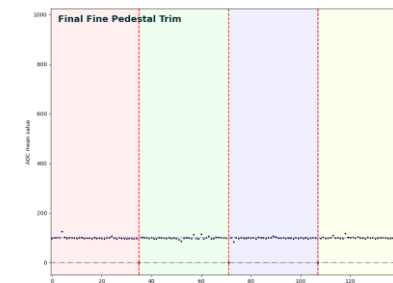
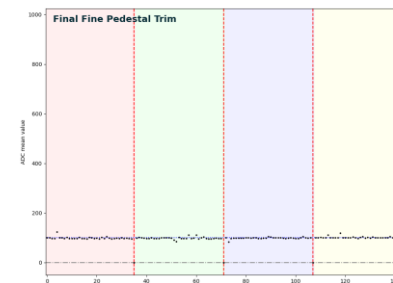
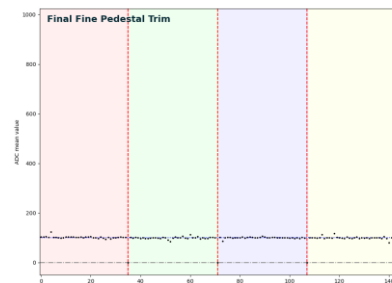
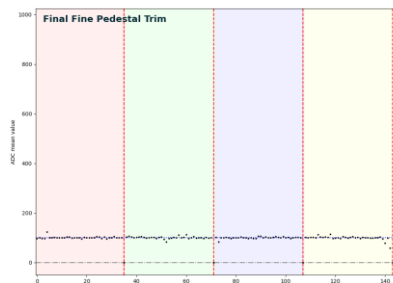
RF = 1



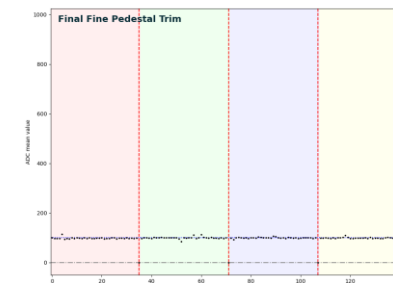
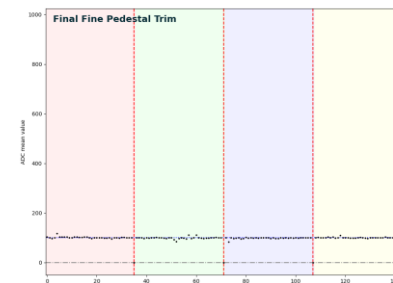
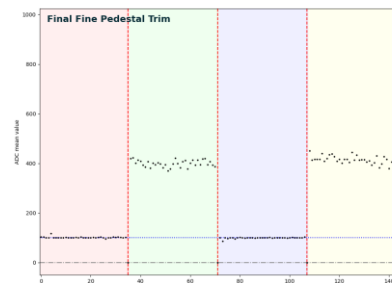
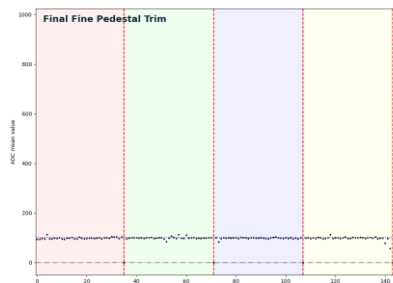
RF = 5



RF = 10



RF = 15



CF = CF\_Comp = 1

CF = CF\_Comp = 5

CF = CF\_Comp = 10

CF = CF\_Comp = 15