H2GCROC

2025 - 2026

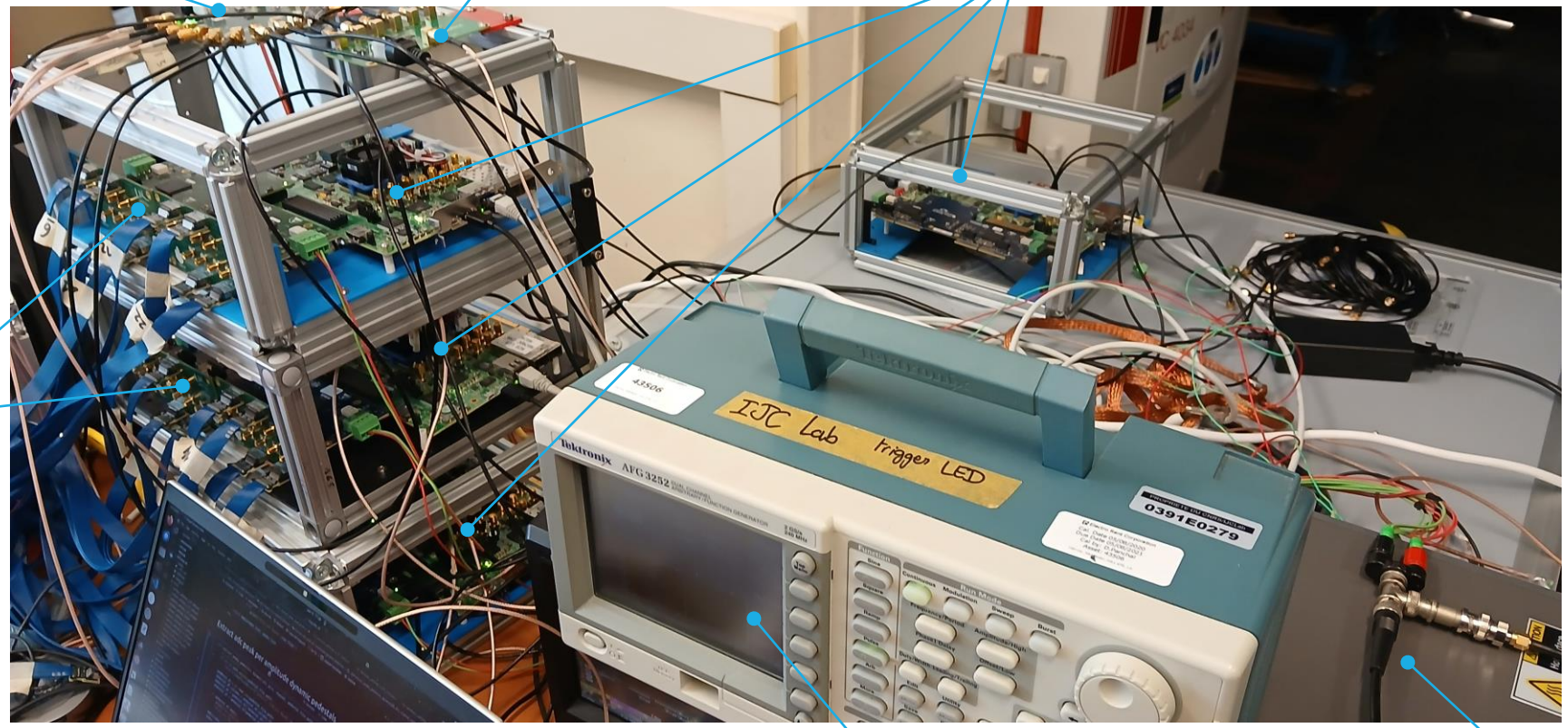
Hardware

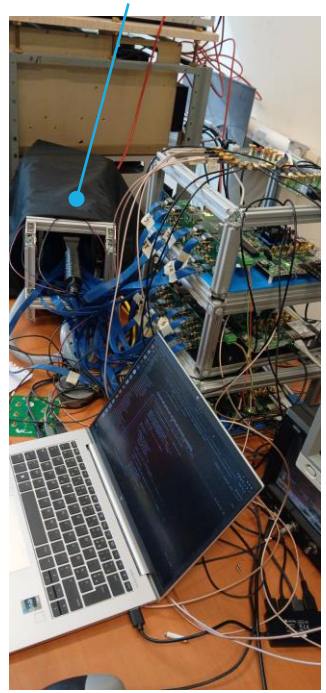
Common clock board

Trigger fan out board

KCU105 & protoboard x4

Crystal





Adapter board

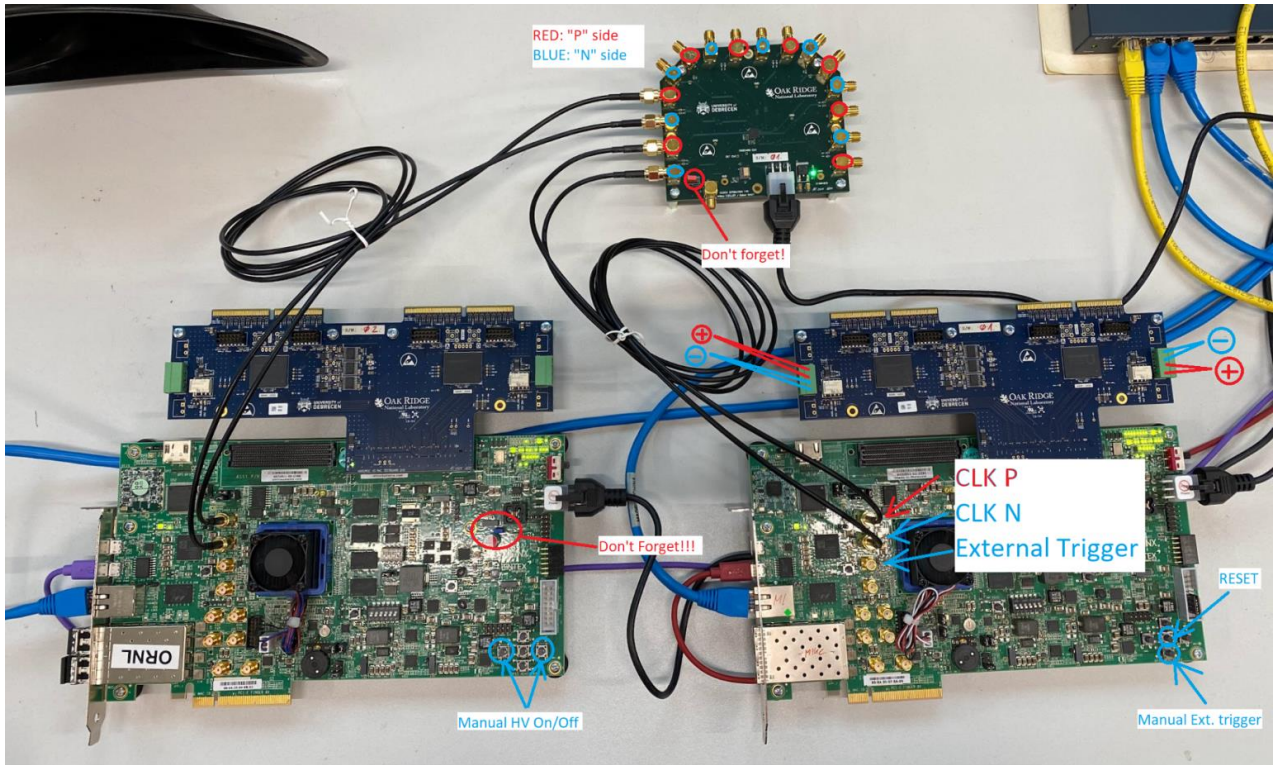
◇ Clock & Trigger & Power cable xN

Signal generator x1

High Voltage Supply

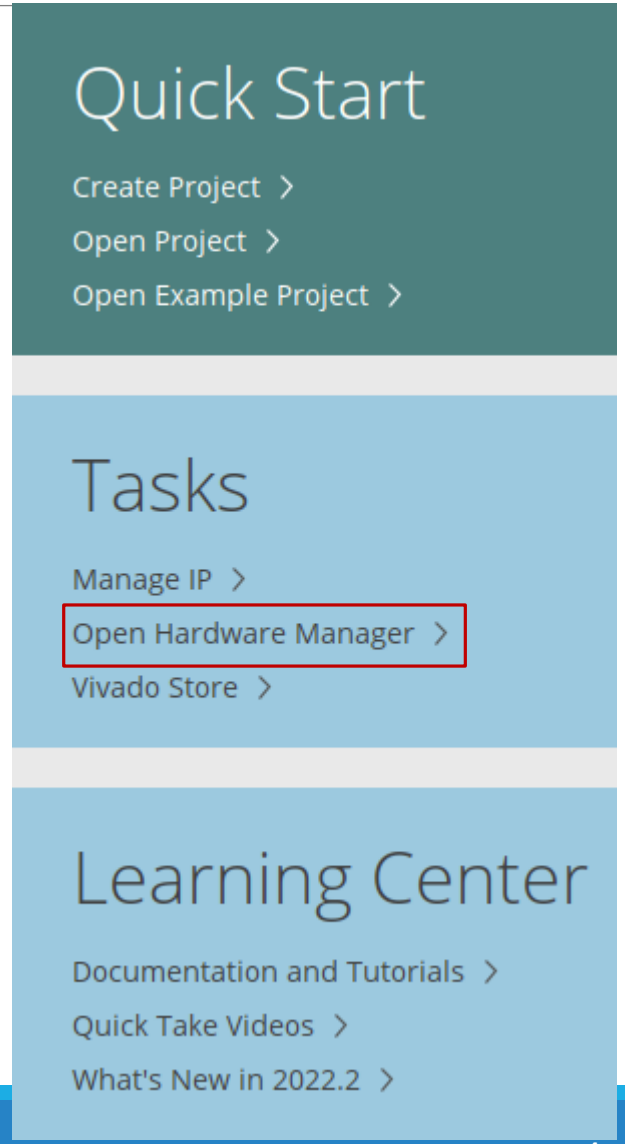
FAQ

- Must insert jumper J14 on the FPGA board
- If you have multiple protoboard and FPGA board, you need a common clock input



Programming FPGA

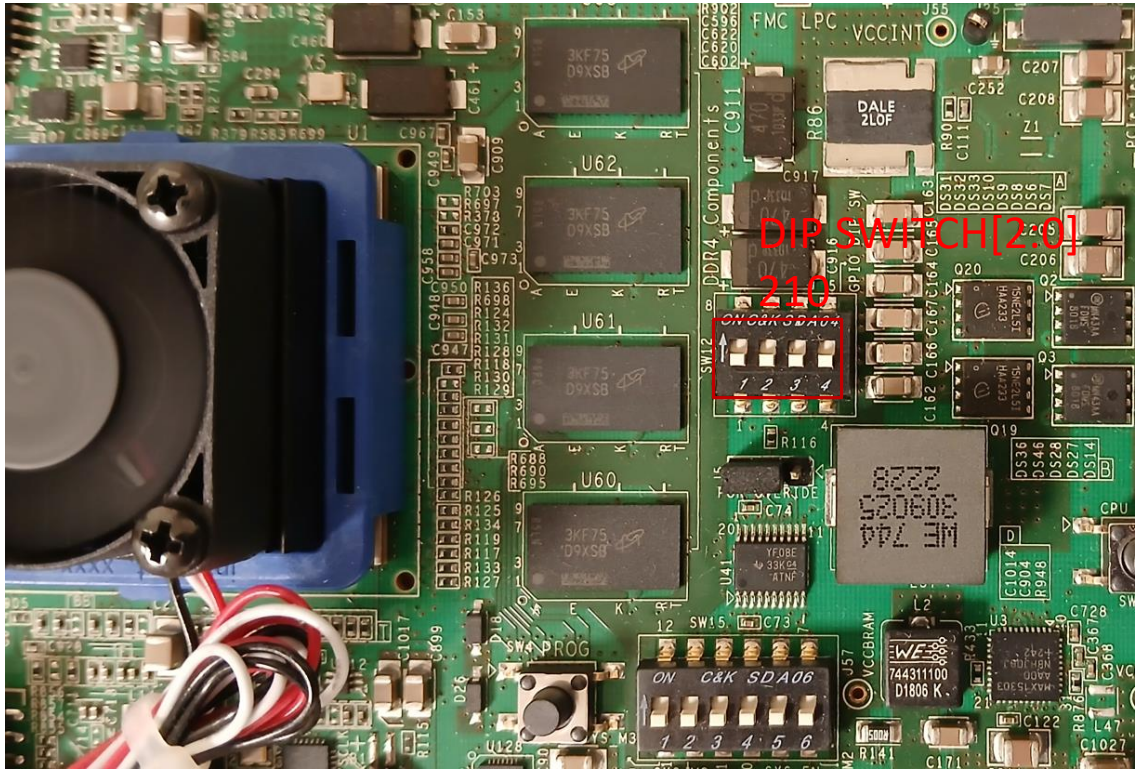
- Install & Launch Vivado (no need license key)
- Open Hardware Manager
- Connect JTAG to PC and power on FPGA board
  - Click Open target on the left Panel
  - Open the device
  - Right click on it and Click “Add Configuration Memory Device”
  - Select device “mt25qu256-spi-x1_x2_x4” and continue
  - Select file “frame_protoboard_v2.mcs” (version v4.11)
  - Program it



- Firmware: v4.11
  - https://web.tresorit.com/l/86zmA#BE8fqp4of3N8pFNz-cXpgg&ZZ8bIfjDltJSr5qvdDoZYuGsgz6LDm0Q
- Programming guide: KCU_Flashing.pdf
  - https://web.tresorit.com/l/86zmA#BE8fqp4of3N8pFNz-cXpgg&GYtZ9jQtioHk1QwqwK5nizBt2rLRRpSt

FPGA IP address

- DIP Switch on KCU105
- FPGA IP address = 10.1.2.208 + DIP SWITCH[2:0]



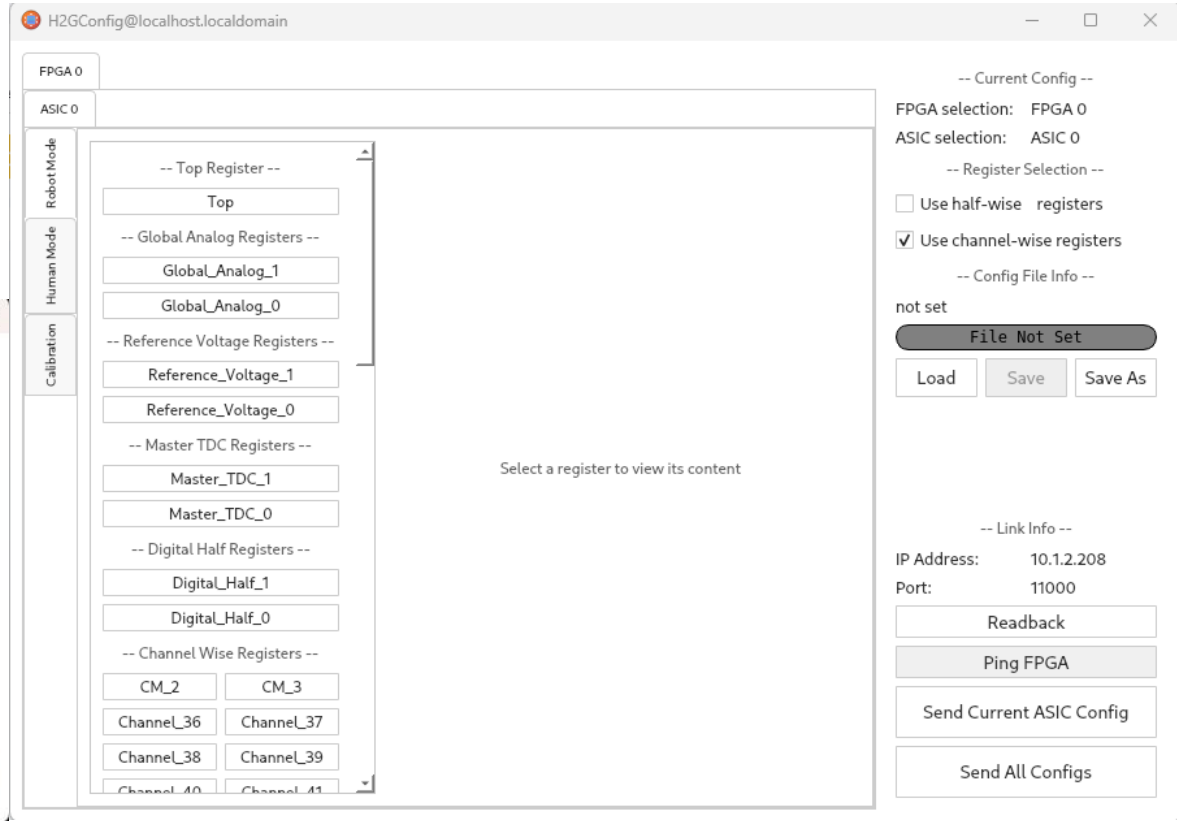
Software environ

- AlmaLinux 9.5, Use AnaConda
  - conda create -n hgc
  - conda activate hgc
  - conda install -c conda-forge pygame pyside6 numpy matplotlib loguru pandas tqdm
- Ubuntu 24.04, Use python venv
  - sudo apt-get install python3-pip python3-venv libxcb-cursor-dev net-tools
  - mkdir myenv
  - python3 -m venv myenv
  - source myenv/bin/activate
  - pip install pyside6 numpy matplotlib loguru pygame pandas tqdm
- Windows 11, Use AnaConda
  - the same as AlmaLinux 9.5, Use AnaConda, you can install git with conda
- baremetal, other ...

Software H2GConfig

- H2GConfig
  - conda activate hgc
  - cd workspace
  - git clone https://gitlab.cern.ch/sjia/H2GConfig.git
  - cd ~/workspace/H2GConfig
  - python H2GConfig.py

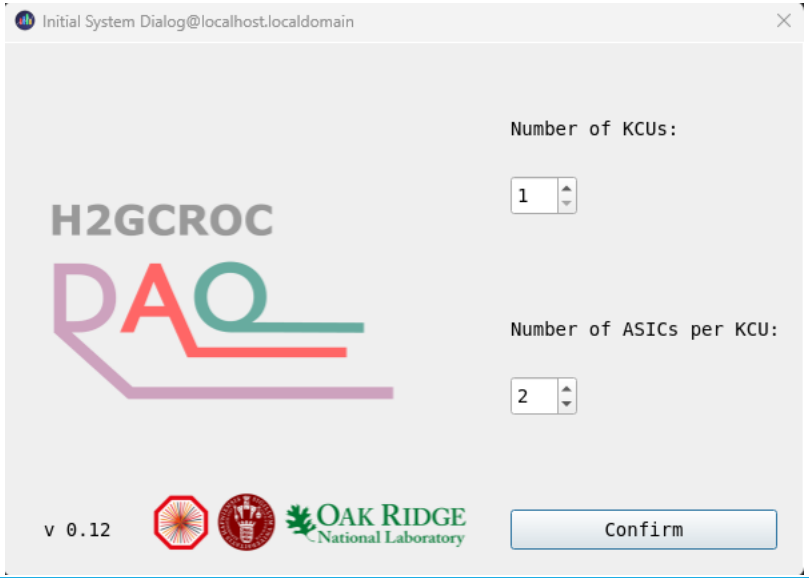
5e478442b 2025/2/26

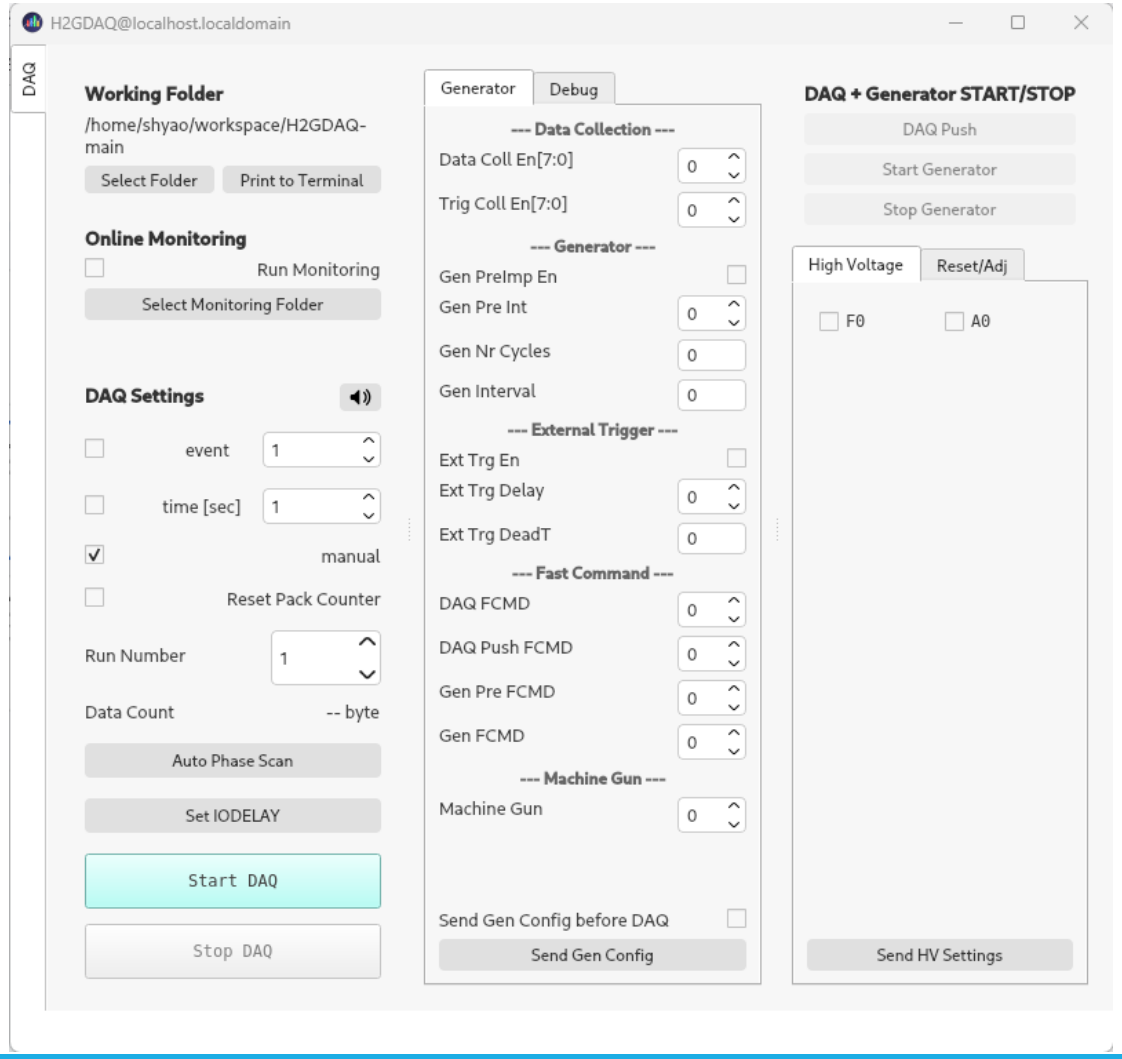


Software H2GDAQ

- The same virtual environ work for H2GDAQ
  - conda activate hgc
  - cd workspace
  - git clone https://gitlab.cern.ch/sjia/H2GDAQ.git
  - cd ~/workspace/H2GDAQ
  - git checkout dev-0v12
  - python H2GDAQ.py

8a17963 2025/4/25



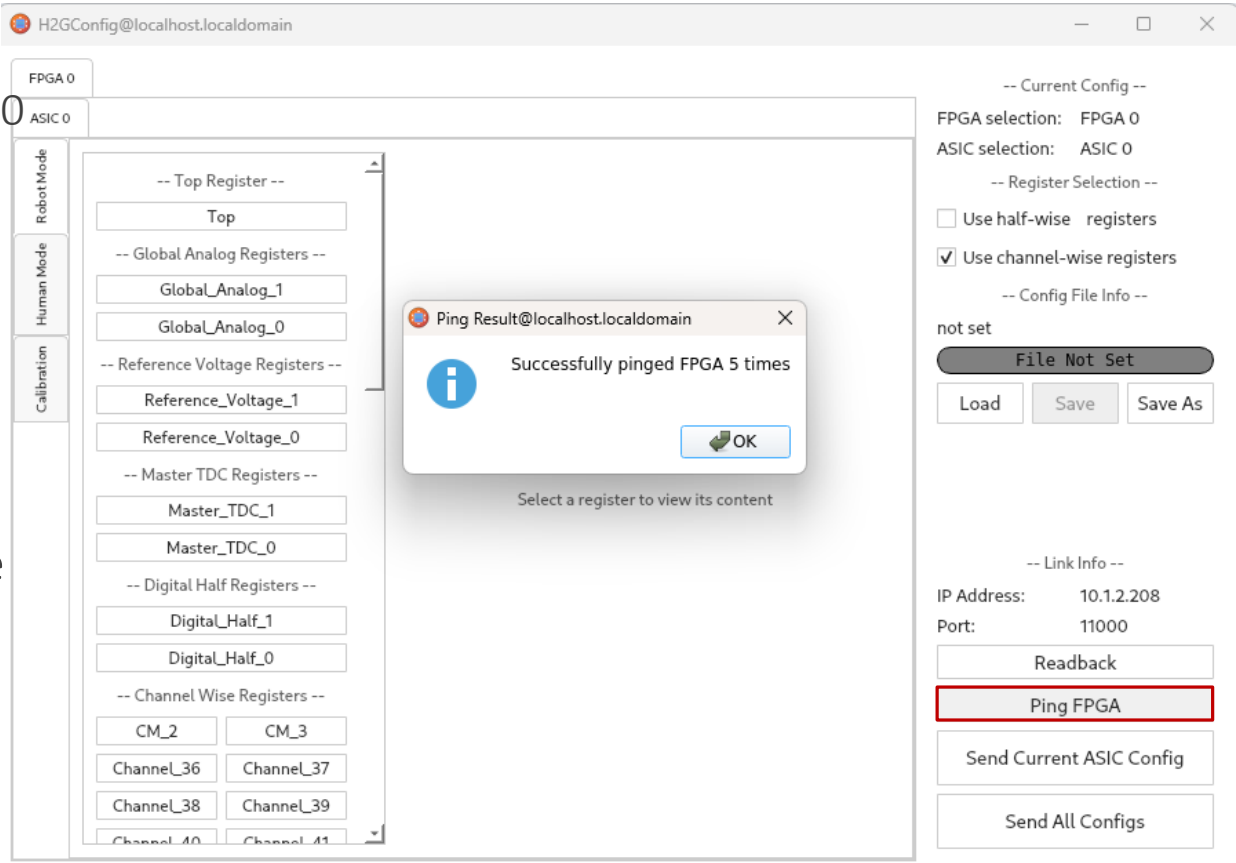


Software H2GCalib

- Create a virtual environ for H2GCalib or add package to hgc environ
  - conda activate hgc
  - conda install -c conda-forge pygame pyside6 numpy matplotlib loguru tqdm pandas
  - cd workspace
  - git clone https://gitlab.cern.ch/sjia/H2GCalib_3B 8dce2723fb 2025/5/9
  - cd ~/workspace/H2GCalib_3B
- Usage 1: GUI mode
  - python 100_UI.py
- Usage 2: console mode
  - Python 000_SocketPool.py
  - ...
  - Python 010_ToTCalib.py

Ethernet communication

- Connect ethernet cable between PC & FPGA board
- Set PC IP address to 10.1.2.207
  - $ IF=enp4s2
  - $ sudo ifconfig ${IF} 10.1.2.207 netmask 255.255.255.0
  - $ sudo ifconfig ${IF} mtu 9000 up
- make sure your switch, cable and network card support 1 Gbps speed, 100 Mbps will not work
  - $ sudo ethtool ${IF} | grep Speed
  - Speed: 1000Mb/s
- H2GConfig has a button “Ping FPGA” can test the connection, it don’t support under windows
- Ping
  - On Linux: ping 10.1.2.208 -s 32
  - On Windows: ping 10.1.2.208 -l 32



Ethernet communication

- Network Optimization (optional)
- Create config file 50-h2gcroc.conf and add the following lines:
  - net.core.rmem_max=8388608
  - net.core.wmem_max=8388608
  - net.core.rmem_default=8388608
  - net.core.wmem_default=8388608
  - net.core.netdev_max_backlog=250000
  - net.ipv4.udp_mem=8388608 8388608 8388608
  - net.ipv4.udp_rmem_min=16384
  - net.ipv4.udp_wmem_min=16384
- Then run the following command to apply the changes:
  - sudo cp 50-h2gcroc.conf /etc/sysctl.d/
  - sudo sysctl -p

Ethernet Firewall

- AlmaLinux 9.5
- DAQ and Calibration programs use UDP port 11000 ~

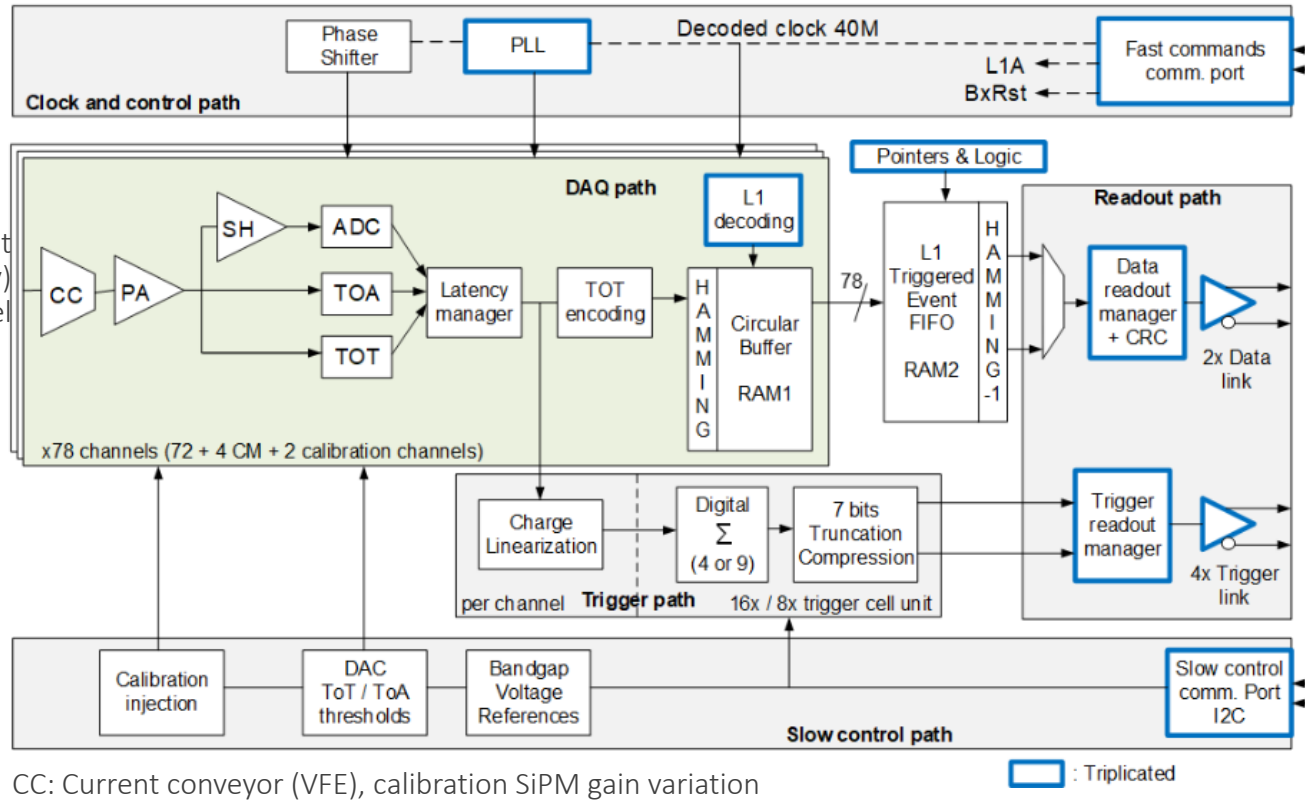
```
[root@localhost ~]# firewall-cmd --zone=public --add-interface=enp87s0 --permanent
success
[root@localhost ~]# firewall-cmd --zone=public --add-port=11000-11100/udp --permanent
success
[root@localhost ~]# firewall-cmd --reload
success [root@localhost ~]# firewall-cmd --zone=public --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp86s0 enp87s0
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 11000-11100/udp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Note interface enp86s0 connect to office network
interface enp87s0 is for h2gc data transfer

Architectural

1 H2GCROC3b: architectural overview

The overall block diagram is described as per below.



320MHz Clock input
Command link (L1A is trigger input)

72 channels of analog signals input
4 CM (only connect to ADC internally)
2 Calibration channel

Two lane for data output:
Speed 1.28Gbps
Charge info, Time info…

Four lane for trigger generation
1.28Gbps x4

I2C interface read/write 7900 parameters

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

Default register value

The default config value of H2GCROC after power up

```
"Channel_0-71           ": "00 00 00 00 00 00 00 00 00 00 00 00 00 00 00",
"CM_0-3                 ": "00 00 00 00 00 00 00 00 00 00 00 00 00 00 00",
"CALIB_0-1              ": "00 00 00 00 00 00 00 00 00 00 00 00 00 00 00",

"Global_Analog_0-1      ": "cf d3 83 28 28 00 9a 9a a2 8a 50 4b 4b 68 6f",
"Reference_Voltage_0-1  ": "dc 00 3c 1c 58 d0 00 00 00 00 00 00 00",
"Master_TDC_0-1         ": "37 d4 40 80 08 b0 02 00 00 80 08 b0 02 00 00 00",
"Digital_Half_0-1       ": "00 00 00 00 80 00 00 00 00 00 00 00 00 19 00 08 cc cc cc 0c cc cc cc cc 0f 01 00",
"HalfWise_0             ": "00 00 00 00 00 00 00 00 00 00 00 00 00 00 00",

"Top                    ": "08 0f 40 31 00 03 85 00 00 00 00 00 00 00 00 00 00 00 00 00 00"
```

Total 1363 Bytes

Operation Procedure

- H2GCROC can generate a pulse signal internally and inject it to input channel, with adjustable amplitude
- H2GCalib_3B
  - Set IODELAY – adjust the data/trigger line delay between FPGA and H2GCROC, this parameter is provided for FPGA, not for H2GCROC.
  - ADC pedestal
  - TOA calibration – adjust the channel threshold to make the output value close to the user's target value
  - TOT calibration – adjust the channel threshold to make the output value close to the user's target value
- Repeat this procedure for all FPGA and H2GCROC combination and generate configuration files

Operation Procedure

- H2GDAQ
  - Set IODELAY – this step is the same as calibration
- H2GConfig
  - Send All configs – upload all calibration values to all H2GCROC
- H2GDAQ
  - Start DAQ
- Decode data or Online Monitoring

H2GCalib_3B

- Modify file H2GCalib_3B/config/common_settings_3B.json

```
{
    "udp": {
        "h2gcroc_ip": "10.1.2.208",
        "pc_ip": "10.1.2.207",
        "h2gcroc_port": 11000,
        "pc_cmd_port":  11000,
        "pc_data_port": 11001
    }
}
```

H2GCalib_3B – Socket Pool

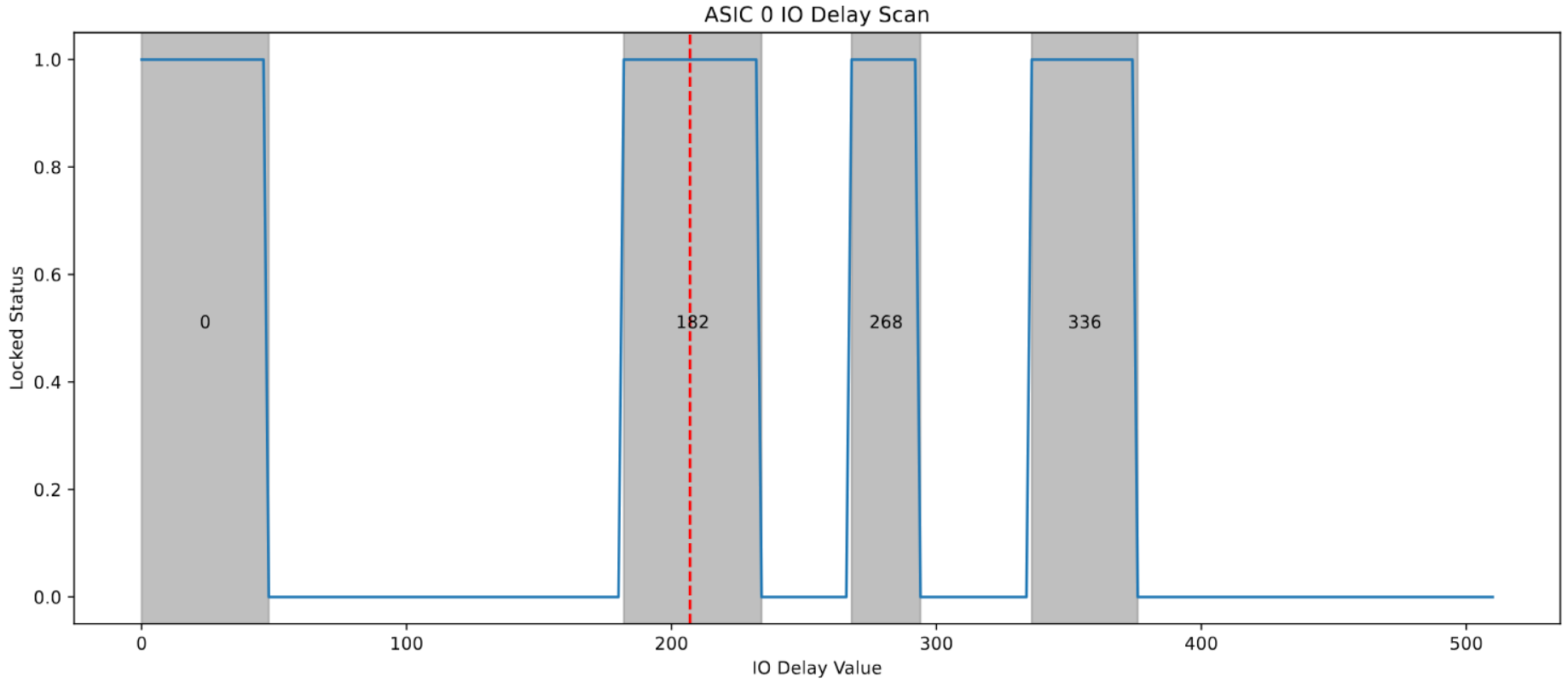
- python H2GCalib_3B/ 000_SocketPool.py

```
[Pool] Control → 127.0.0.1:6002
[Pool] Data    → 127.0.0.1:6001
...
```

- This program maintains multiple network connections, and other calibration programs use this as a relay, keep this program running in the background

H2GCalib_3B – IO delay

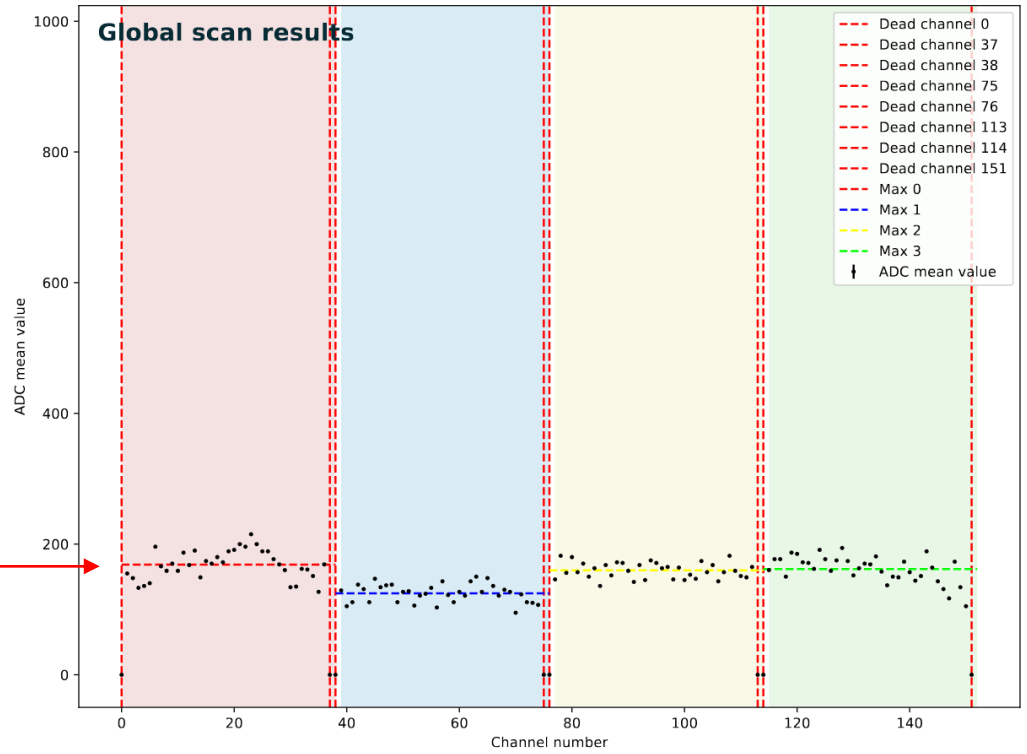
- python 007_IODelay.py
- You can also use H2GDAQ's IO Delay button to do this, but it don't generate report



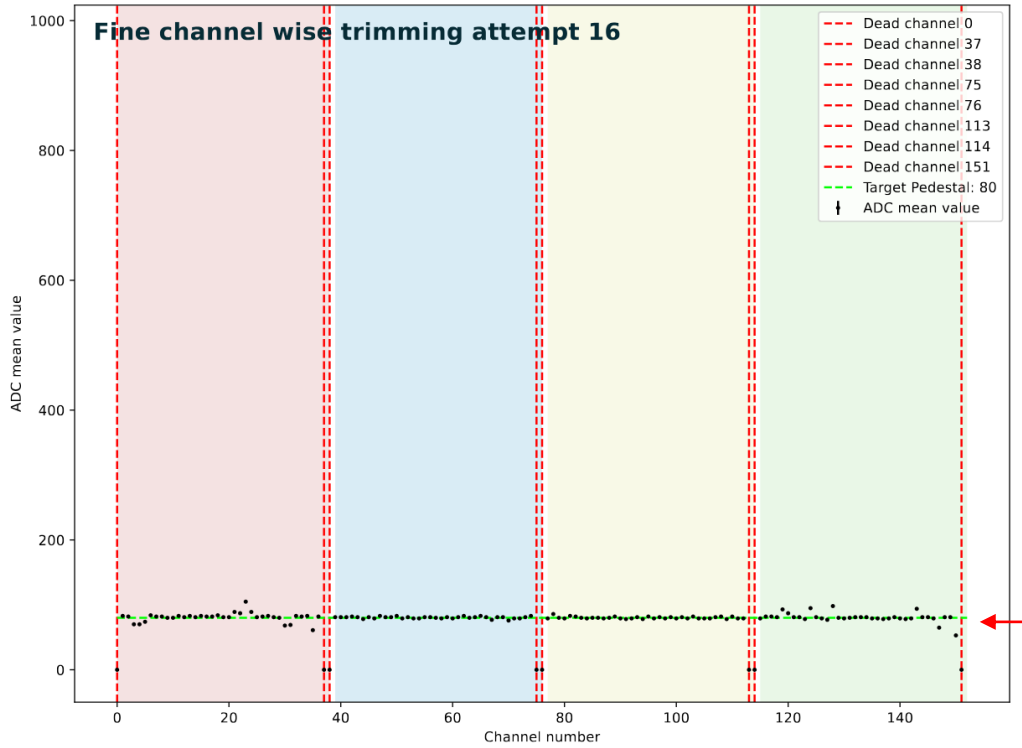
H2GCalib_3B – ADC pedestal calibration

- This program try to get the best pedestal value, it start from config “default_2024Aug_config.json”
- python 005_PedestalCalib.py

avg



Before calibration ADC pedestal



target

After calibration ADC pedestal

H2GCalib_3B – ADC pedestal calibration

- Run calibration again but base on the previous output but it don't work, TODO
- python 005_PedestalCalib.py -i python 005_PedestalCalib.py -i dump/005_PedestalCalib_data_20250625_135655/config_pede_a0_20250625_135817.json,dump/005_PedestalCalib_data_20250625_135655/config_pede_a1_20250625_135817.json

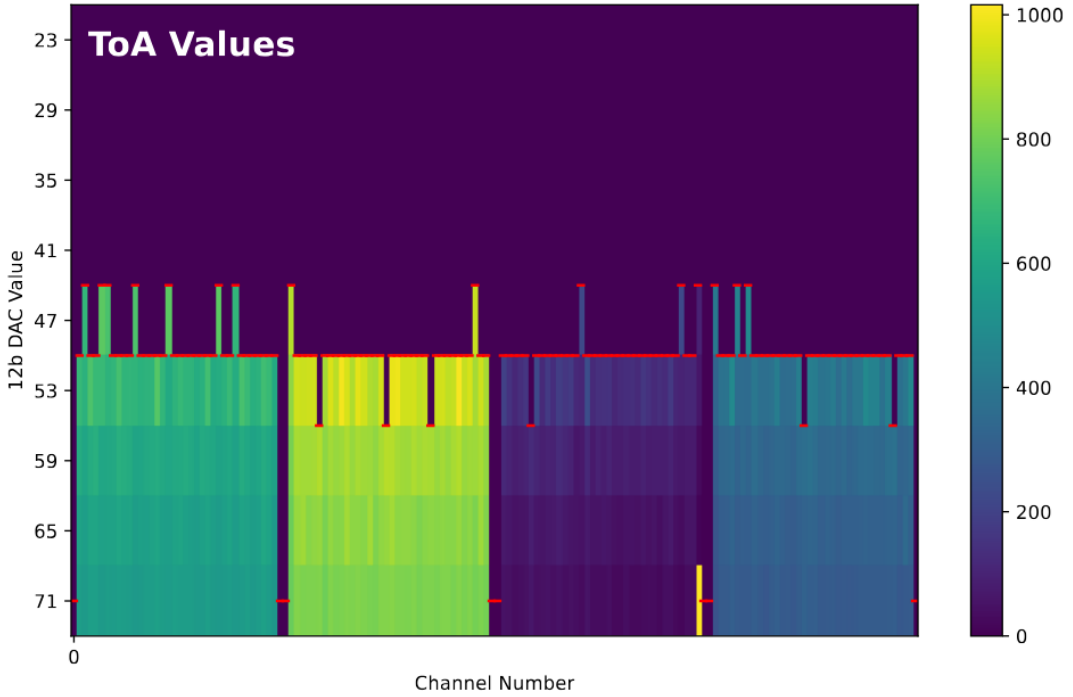
previous ADC pedestal calibration result

final ADC pedestal calibration result

H2GCalib_3B – ToA Calibration

- python 006_ToACalib.py -t 50 -i dump/005_PedestalCalib_data_20250625_135655/config_pede_a0_20250625_135817.json,dump/005_PedestalCalib_data_20250625_135655/config_pede_a1_20250625_135817.json

Before calibration ToA



After calibration ToA

H2GCalib_3B – ToA Calibration

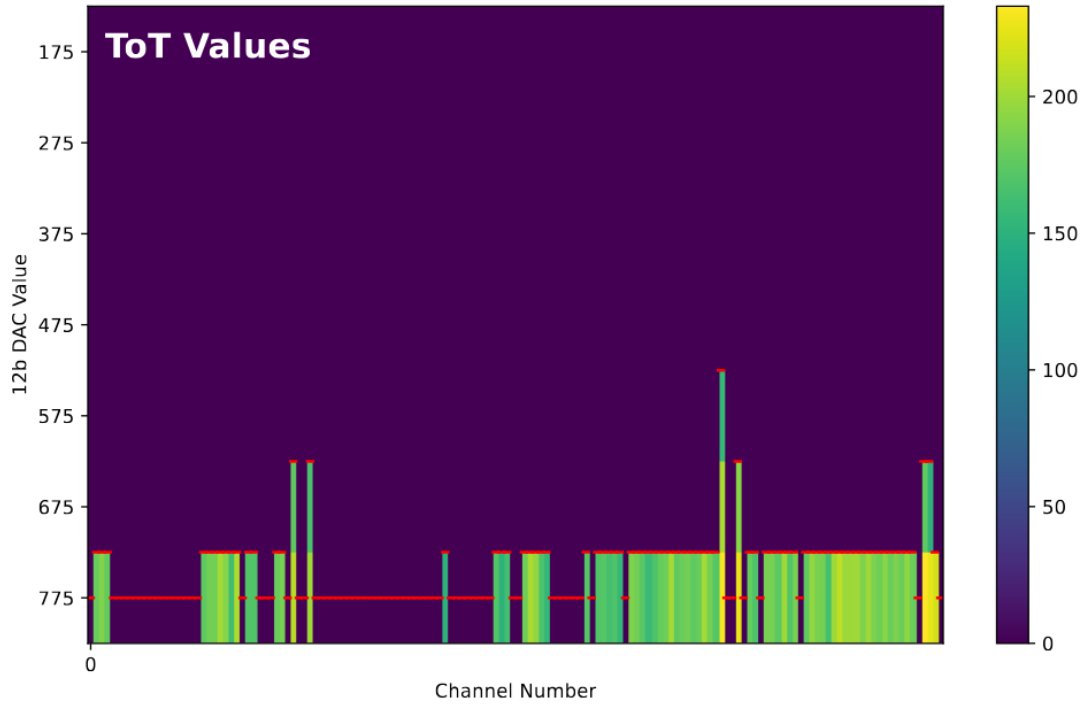
- Run calibration again but base on the previous output TODO
- python 006_ToACalib.py -t 50 -i dump/006_ToACalib_data_20250625_143708/config_toa_a0_20250625_151028.json,dump/006_ToACalib_data_20250625_143708/config_toa_a1_20250625_151028.json

previous ToA calibration result

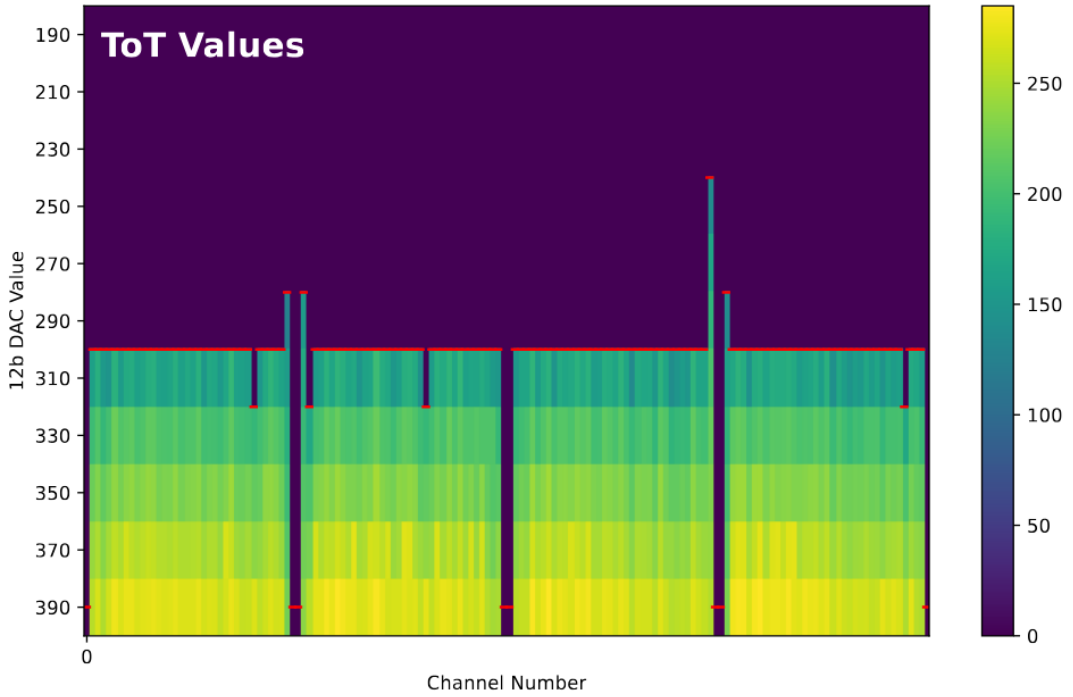
final ToA calibration result

H2GCalib_3B – ToT Calibration

- python 010_ToTCalib.py -t 300 -i dump/006_ToACalib_data_20250625_143708/config_toa_a0_20250625_151028.json,dump/006_ToACalib_data_20250625_143708/config_toa_a1_20250625_151028.json



Before calibration ToT



After calibration ToT

H2GCalib_3B – ToT Calibration

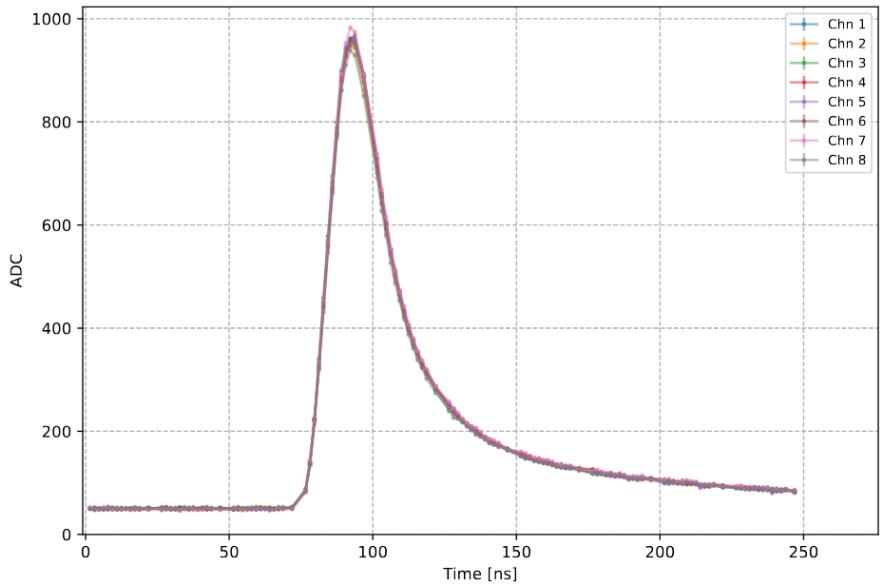
- Run calibration again but base on the previous output TODO
- python 010_ToTCalib.py -t 300 -i dump/006_ToACalib_data_20250625_143708/config_toa_a0_20250625_151028.json,dump/006_ToACalib_data_20250625_143708/config_toa_a1_20250625_151028.json

previous ToT calibration result

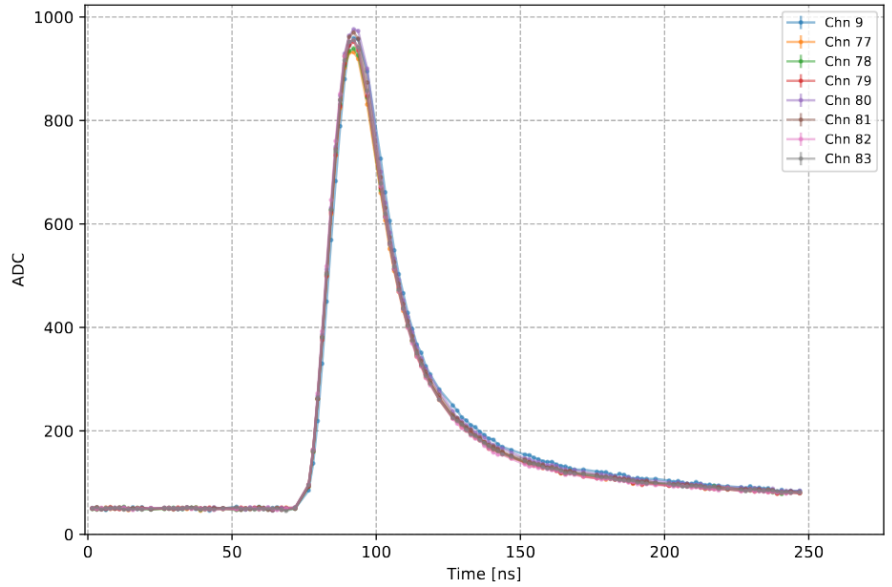
final ToT calibration result

H2GCalib_3B – inject 2V5 signal

- modify file , line 282 `phase_settings = [0, 1, 2, 3, 5, 8, 9, 10, 11, 12, 13, 14, 15]`
- python 009_IntInjection_2V5.py -i dump/010_ToTCalib_data_20250625_152005/config_tot_a0_20250625_155055.json,dump/010_ToTCalib_data_20250625_152005/config_tot_a1_20250625_155055.json



ASIC 0 (ch 1 ~ 9)



ASIC 1 is fine (ch 77 ~ 85)

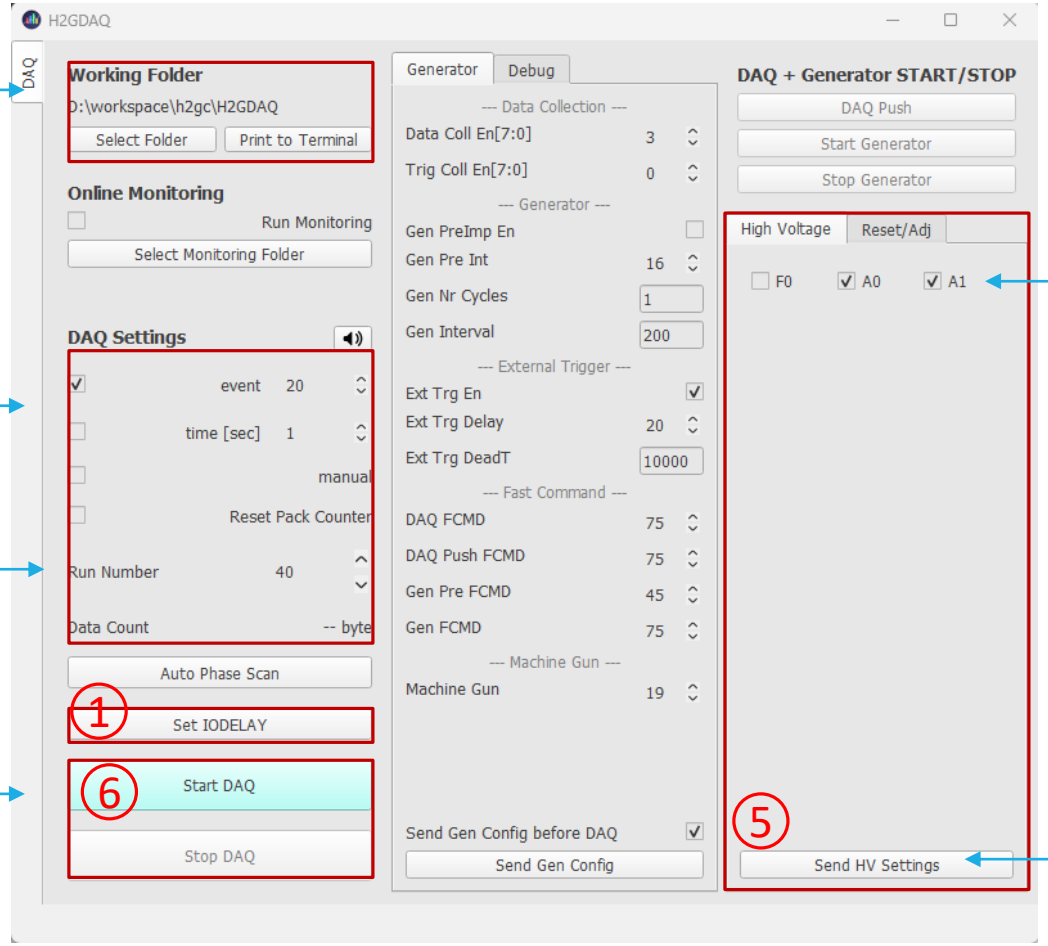
Operation Procedure - DAQ

Directory to store data

DAQ stop condition: By event count, by time or manual

Output Run040.h2g

Start/Stop DAQ button



high voltage relay setting

A0 = ASIC 0 only

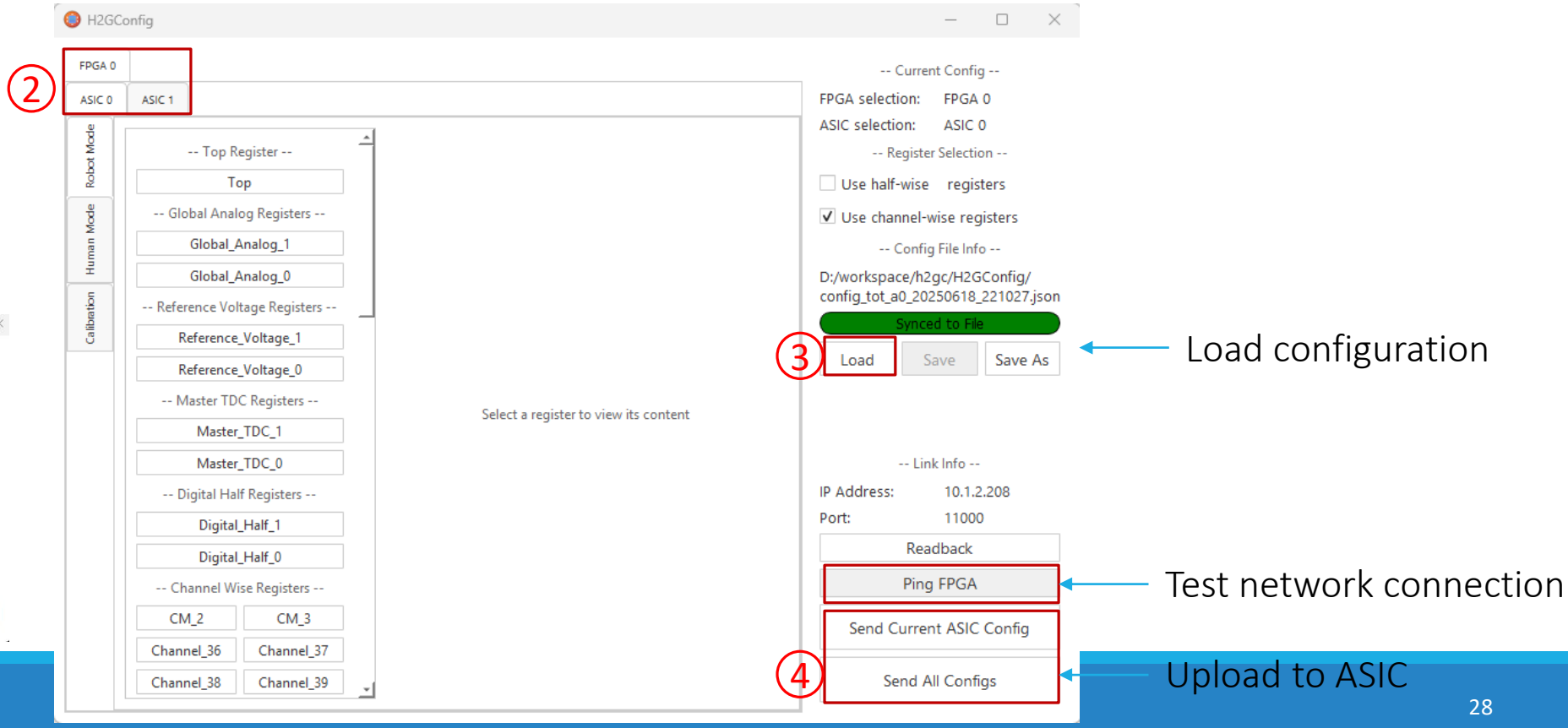
A1 = ASIC1 only

F0 = ASIC0 and ASIC1 both

Send command to turn On/Off the relay base on the settings above. You also need to turn on your power supply

Operation Procedure - Config

- H2GConfig
  - The GUI can read the configuration file and display it graphically. It allow you to modify register bit by bit
  - It can upload configuration to the ASIC
  - There is also a Ping FPGA button for you to test the ethernet connection



Operation Procedure - online_monitoring

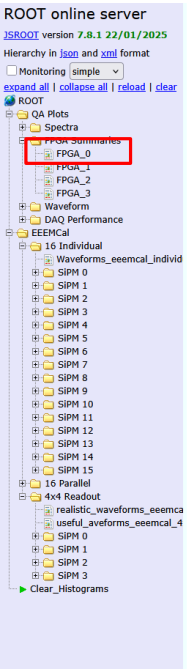
- https://github.com/tlprotzman/h2g_online_monitoring
  - dependence: ROOT
  - developed by Tristan Protzman from Lehigh University

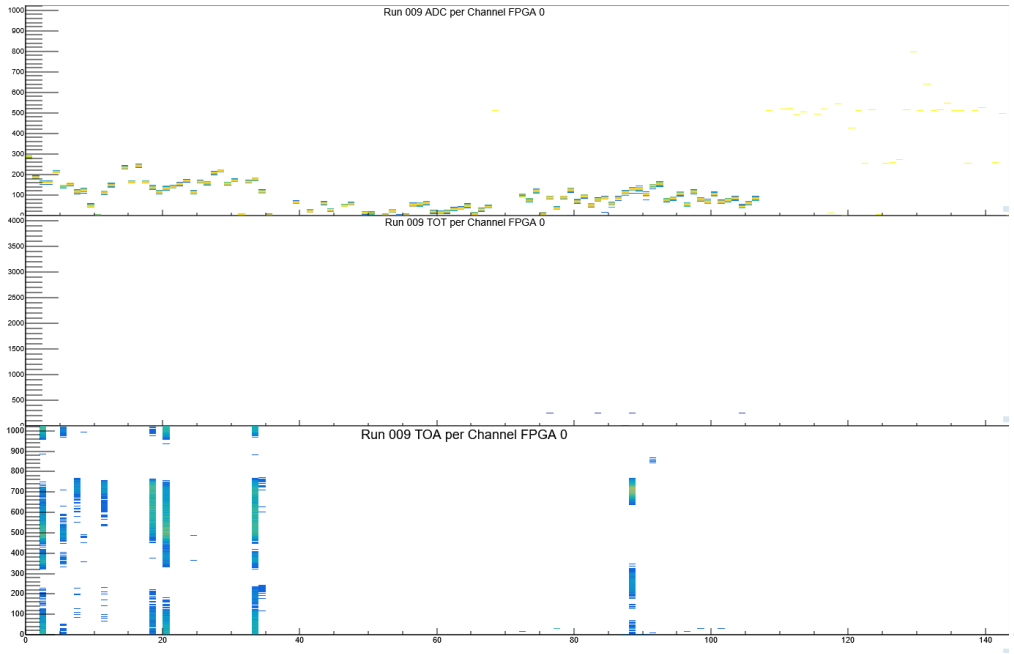
```
$ cat eeemcal_20sample.cfg
MAX_SAMPLES=20
NUM_FPGA=4
DETECTOR_ID=2
```

```
$ git clone https://github.com/tlprotzman/h2g_online_monitoring.git
$ cd ~/workspace/h2g_online_monitoring
$ make
$ export DATA_DIRECTORY=~/workspace/H2GDAQ
$ export MONITORING_PORT=12345
$ root -q -b -x -l RunMonitoring.c\(RunNumber\)

Processing RunMonitoring.c(9)...
Parsing config file eeemcal_20sample.cfg
Key: MAX_SAMPLES, Value: 20
Key: NUM_FPGA, Value: 4
Key: DETECTOR_ID, Value: 2

Configuring for detector ID 2: EEEMCal
Configuration values:
NUM_LINES: 5
NUM_FPGA: 4
NUM_CHANNELS: 72
NUM_LINES: 5
MAX_SAMPLES: 20
MACHINE_GUN_MAX_TIME: 600
PACKET_SIZE: 1452

Configuring for detector ID 2: EEEMCal
Configuration values:
NUM_LINES: 5
NUM_FPGA: 4
NUM_CHANNELS: 72
NUM_LINES: 5
MAX_SAMPLES: 20
MACHINE_GUN_MAX_TIME: 600
PACKET_SIZE: 1452
huh??
Info in <THttpEngine::Create>: Starting HTTP server on port 12345
Attempting to open file /home/shyao/workspace/H2GDAQ/Run009.h2g
Starting at byte 828
Building events... done!
Updating canvases... done!
.....
```

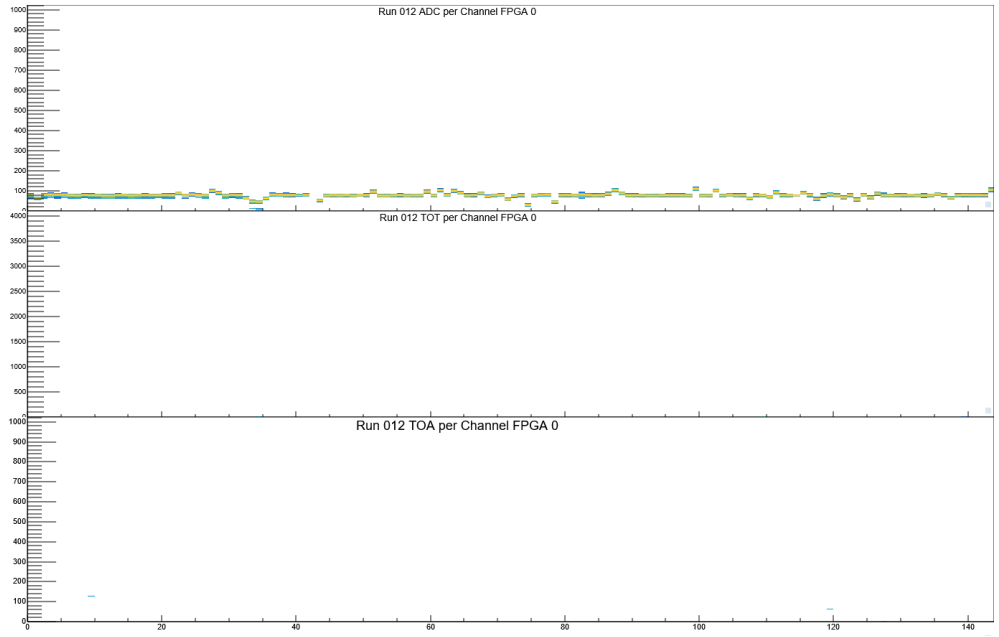
Operation Procedure - online_monitoring

- Launch web browser and connect to localhost:12345





before calibration



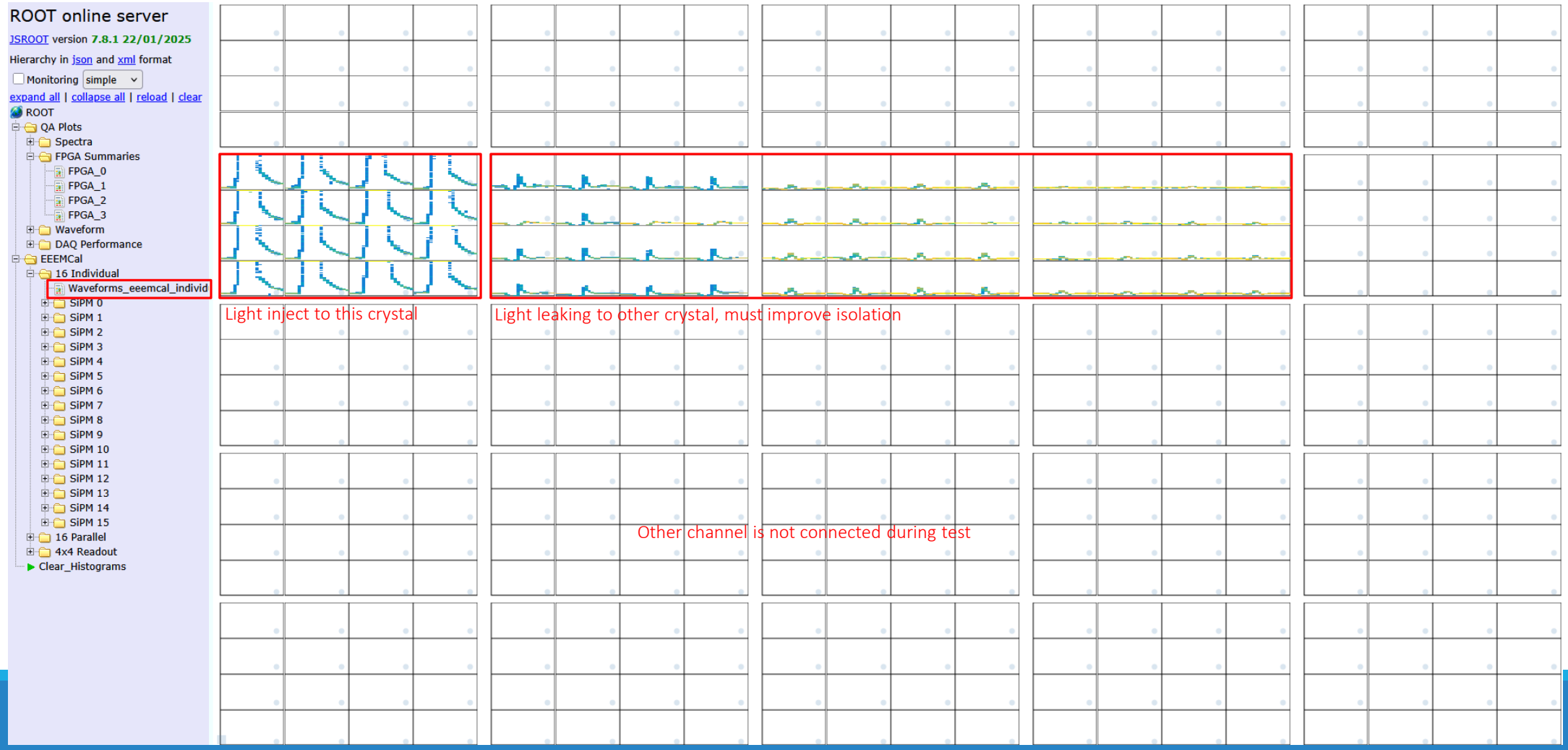
After calibration

Operation Procedure - online_monitoring

- Inject light to crystal, each crystal use 4x4 SiPM. The light comes from an LED with a pulse width of 100ns

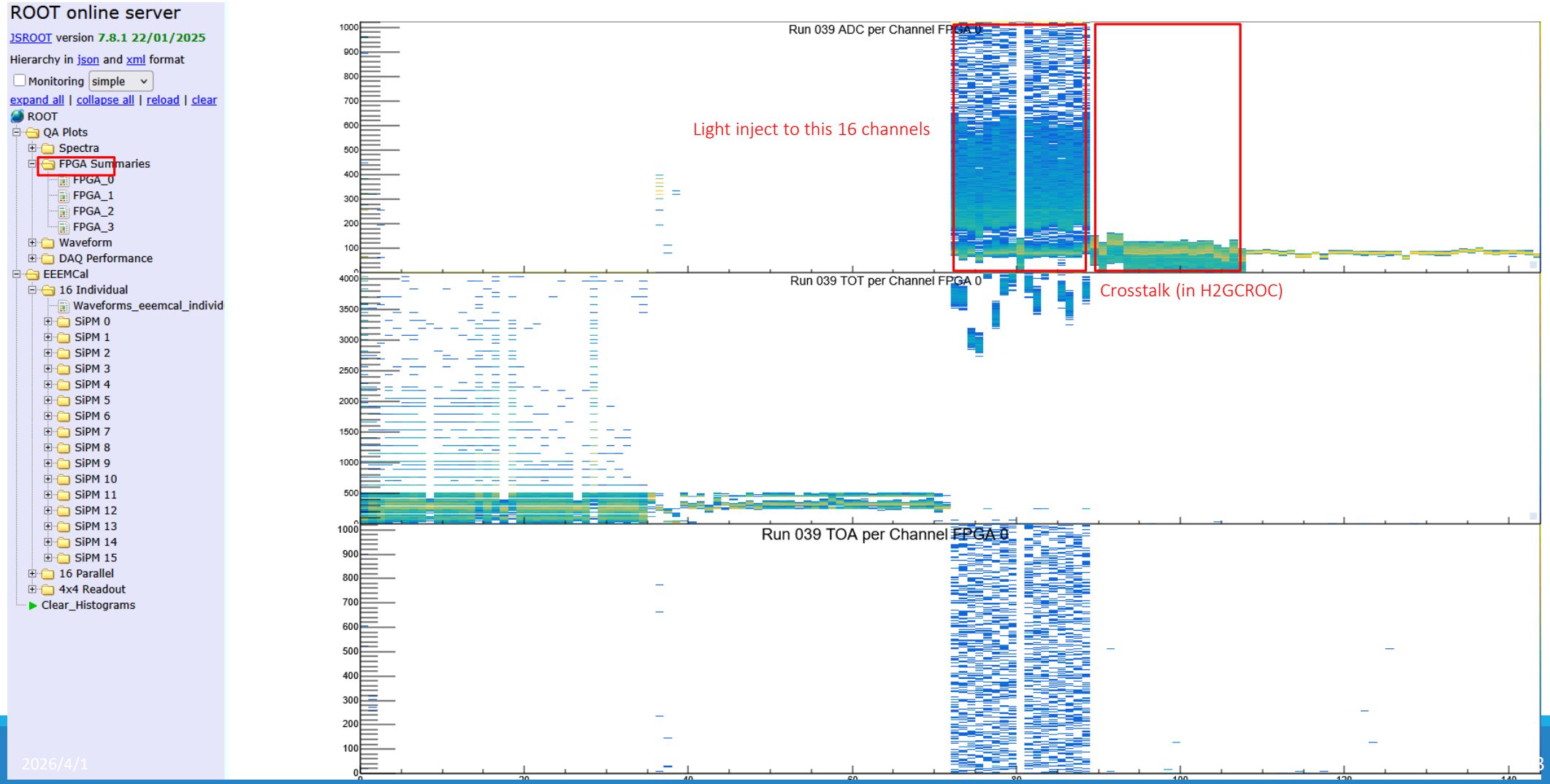
Operation Procedure - online_monitoring

- Draw in crystal Array, it depend on your channel mapping



Operation Procedure - online_monitoring

- Inject light to crystal, without leaking



Operation Procedure - decode

- https://github.com/tlprotzman/h2g_decode
  - dependent ROOT , developped by Tristan Protzman from Lehigh University
  - It provide a library h2g_decode and a executable h2g_run

```
$ cd ~/workspace
$ git clone https://github.com/tlprotzman/h2g_decode.git
$ cd ~/workspace/h2g_decode
$ mkdir data build; cd build; cmake ..; make
$ export DATA_DIRECTORY=~/workspace/H2GDAQ
$ export OUTPUT_DIRECTORY=~/workspace/h2g_decode

$ ./build/h2g_run
Usage: h2g_decode -r <run_number> [-d <detector_id>] [-n <num_kcu>] [-g] [-G LEVEL] [-T]
  -r, --run        Run number (required)
  -d, --detector    Detector ID (default: 0, LFHCAL: 1, EEEMCAL: 2)
  -n, --num-kcu     Number of KCUs (default: 4)
  -g, --debug       Enable debug output with INFO level
  -G, --debug-level Set debug level explicitly:
                    0: OFF, 1: ERROR, 2: WARNING, 3: INFO, 4: DEBUG, 5: TRACE
  -T, --truncate    Enable ADC truncation
  -h, --help        Show this help message
```

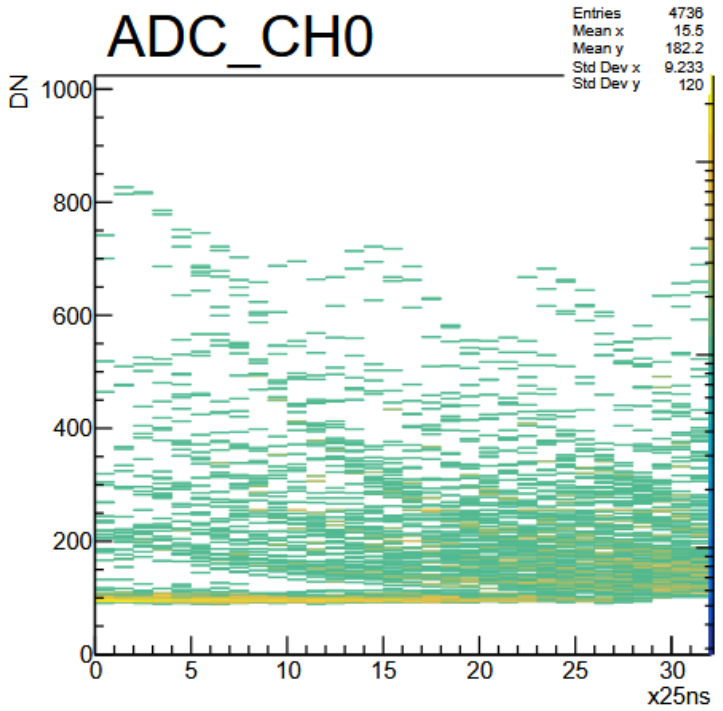
** Data file is named with “Run%03d.h2g”

Operation Procedure - decode

- Convert rawdata to root file for future analysis.

```
$ ./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     *
*.............................................................................*
```

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



Q/A

- WWW

Operation Procedure - online_monitoring

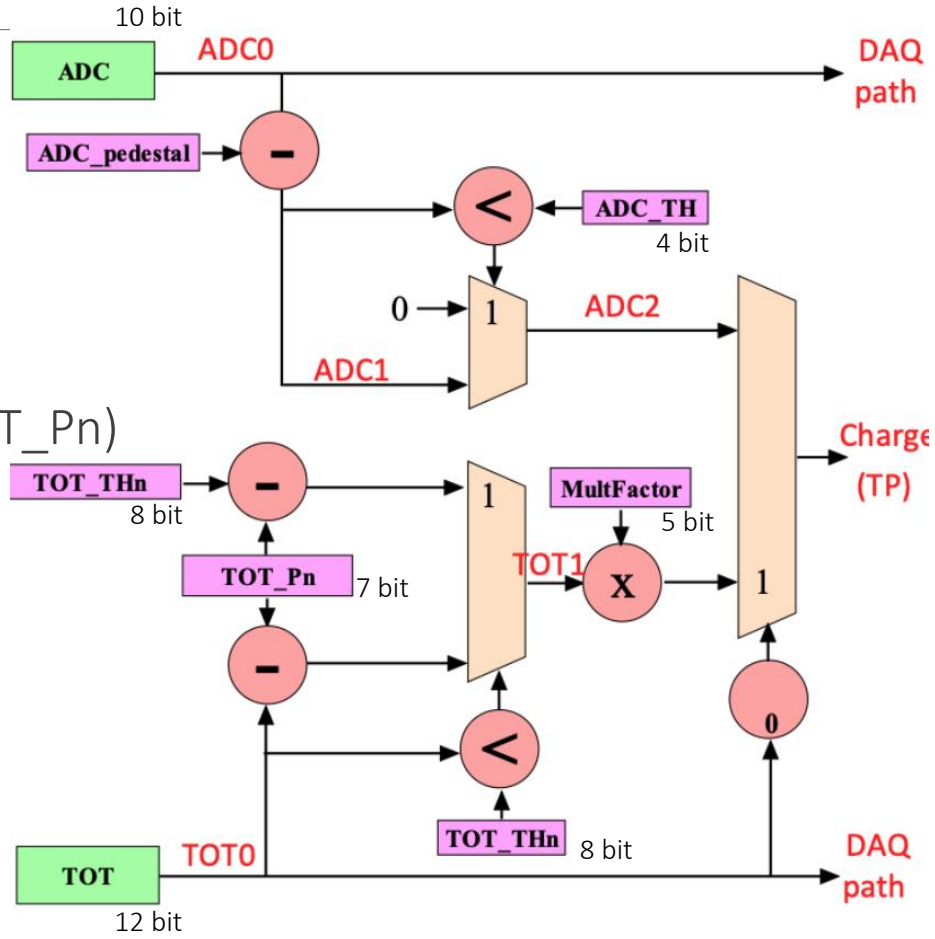
- To modify channel mapping: online_monitor.cxx

```
// EEEMCal mapping - instead of "layers", we have a single plane, where each crystal
is one connector
// FPGA IP | ID
// 208     | 0
// 209     | 1
// 210     | 2
// 211     | 3
int eeemcal_fpga_map[25] = {0, 0, 0, 0, 2,
                            0, 0, 0, 0, 2,
                            1, 1, 1, 1, 2,
                            1, 1, 1, 1, 2,
                            2, 2, 2, 2, 3};


// ASIC | ID
// 0    | 0
// 1    | 1
int eeemcal_asic_map[25] = { 0, 0, 0, 0, 1,
                             1, 1, 1, 1, 1,
                             0, 0, 0, 0, 1,
                             1, 1, 1, 1, 1,
                             0, 0, 0, 0, 0};


// Connector | ID
// A         | 0
// B         | 1
// C         | 2
// D         | 3
int eeemcal_connector_map[25] = { 0,  1,  2,  3,  0,
                                  0,  1,  2,  3,  1,
                                  0,  1,  2,  3,  2,
                                  0,  1,  2,  3,  3,
                                  0,  1,  2,  3,  0};
```

```
int eeemcal_16i_channel_a_map[16] = { 0,  1,  2,  3,  4,  5,  6,  7,
                                      9, 10, 11, 12, 13, 14, 15, 16};

int eeemcal_16i_channel_b_map[16] = {19, 20, 21, 22, 23, 24, 25, 26,
                                     27, 28, 29, 30, 31, 32, 33, 34};

int eeemcal_16i_channel_c_map[16] = {55, 56, 57, 58, 59, 60, 61, 62,
                                     63, 64, 65, 66, 67, 68, 69, 70};

int eeemcal_16i_channel_d_map[16] = {36, 37, 38, 39, 40, 41, 42, 43,
                                     45, 46, 47, 48, 49, 50, 51, 52};
/*
int eeemcal_16i_channel_a_map[16] = { 2,  6, 11, 15,  0,  4,  9, 13,
                                      1,  5, 10, 14,  3,  7, 12, 16};

int eeemcal_16i_channel_b_map[16] = {20, 24, 29, 33, 18, 22, 27, 31,
                                     19, 23, 28, 32, 21, 25, 30, 34};

int eeemcal_16i_channel_c_map[16] = {67, 63, 59, 55, 69, 65, 61, 57,
                                     70, 66, 60, 56, 68, 64, 58, 54};

int eeemcal_16i_channel_d_map[16] = {50, 46, 40, 36, 52, 48, 42, 38,
                                     51, 47, 43, 39, 49, 45, 41, 37};
*/
```

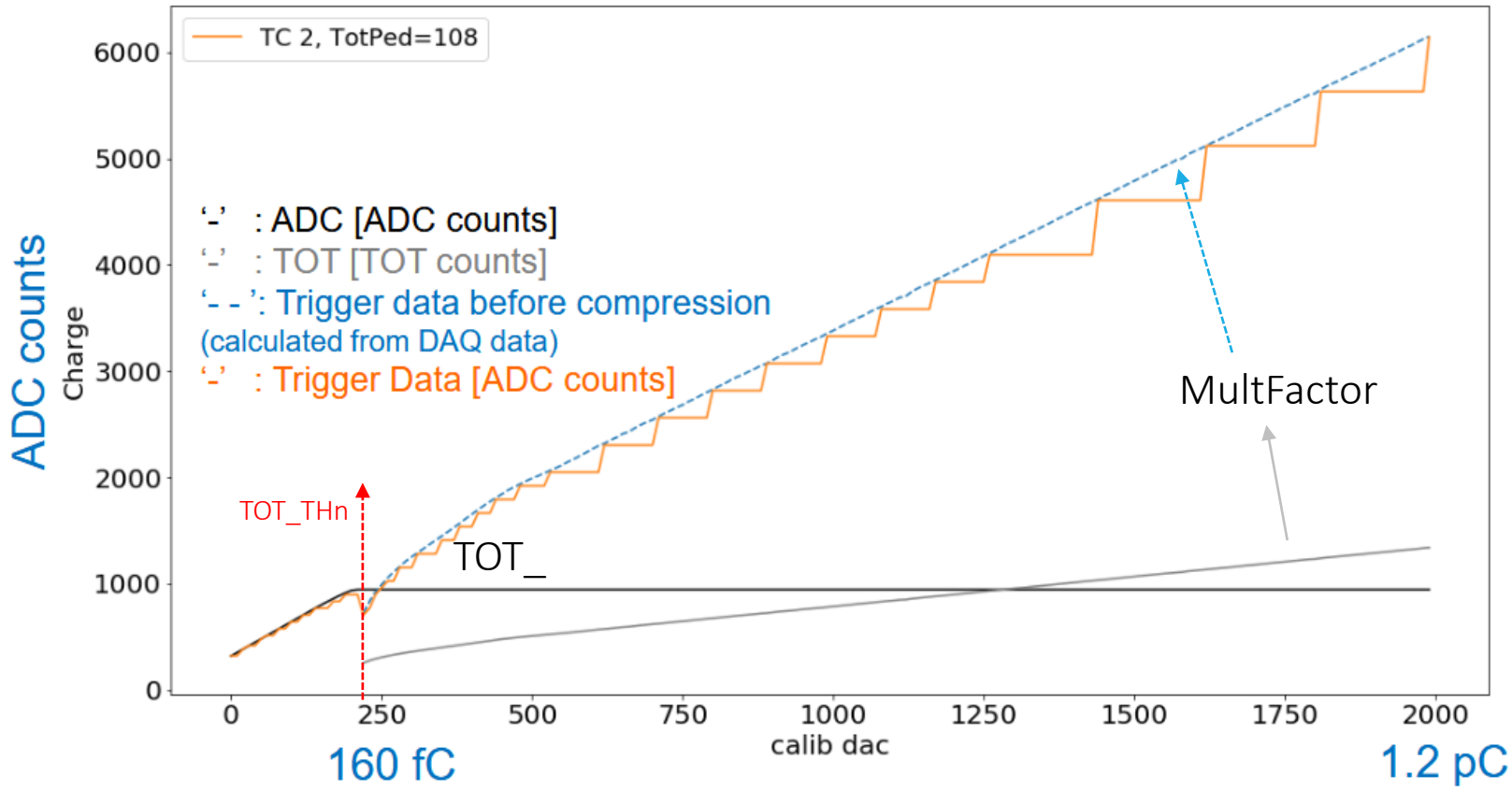
Trigger path

- ADC1 = ADC0 – ADC_pedestal
- ADC2 = (ADC1 < ADC_TH) ? 0: ADC1
  - ***if set ADC_pedestal = ADC_TH = 0
  - ***ADC2 = ADC0
- TOT1 = (TOT0 < TOT_THn) ? (TOT_THn – TOT_Pn) : (TOT0 – TOT_Pn)
  - *** if set TOT_THn = TOT_Pn = 0
  - *** TOT1 = TOT0
- Charge = (TOT0 != 0) ? TOT1 * MultFactor : ADC2



Trigger path

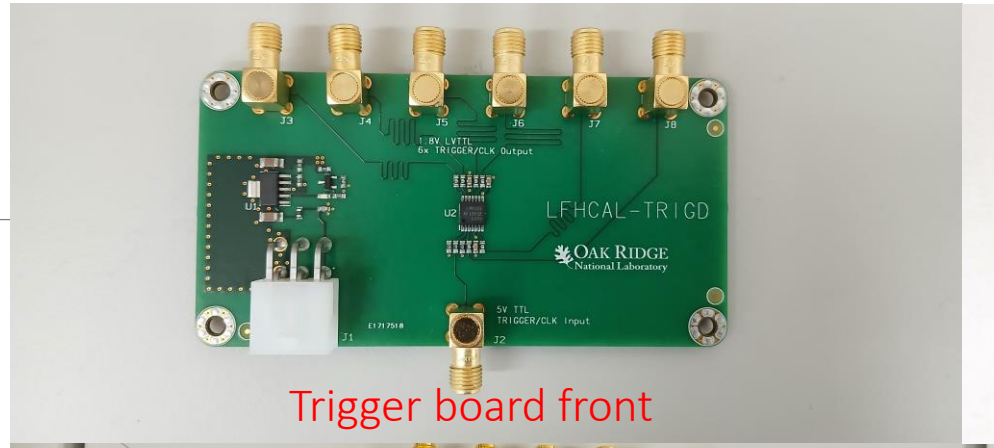
- ADC [ADC counts] is 10 bit
- TOT [TOT counts] is 12 bit
- linearization



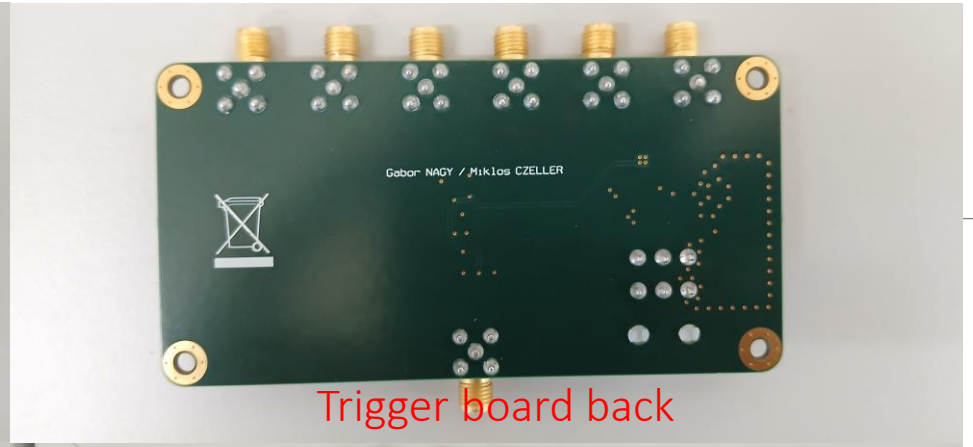
Ref. TIPP21_HGCROC.pdf

Local R&D

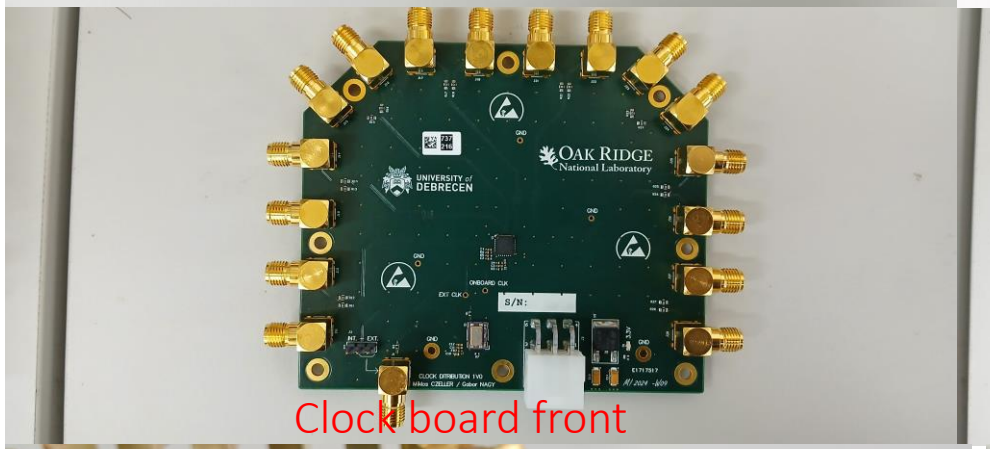
- www



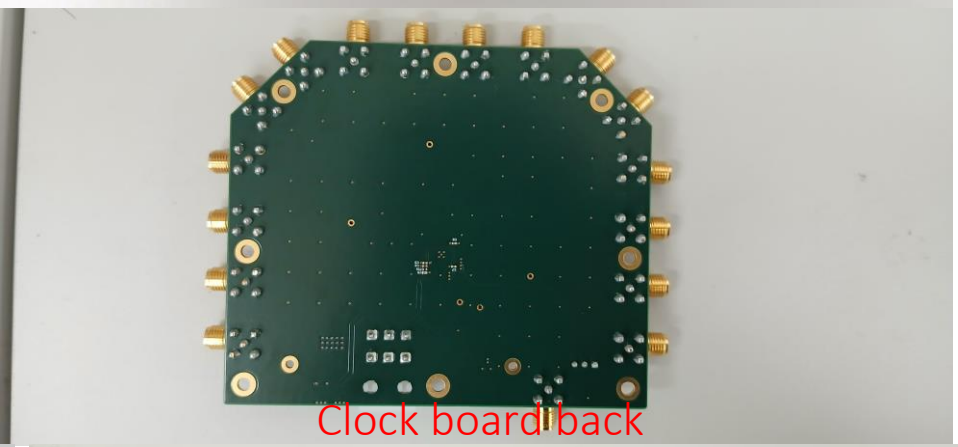
Trigger board front



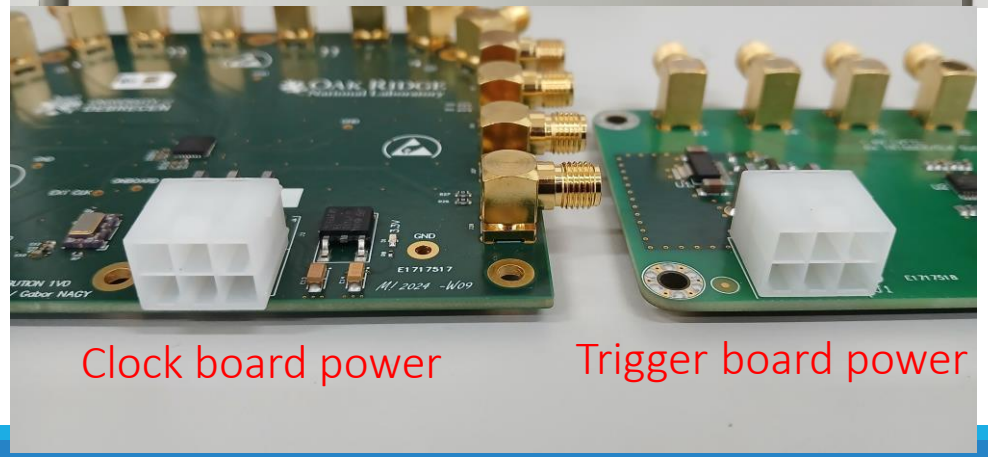
Trigger board back



Clock board front

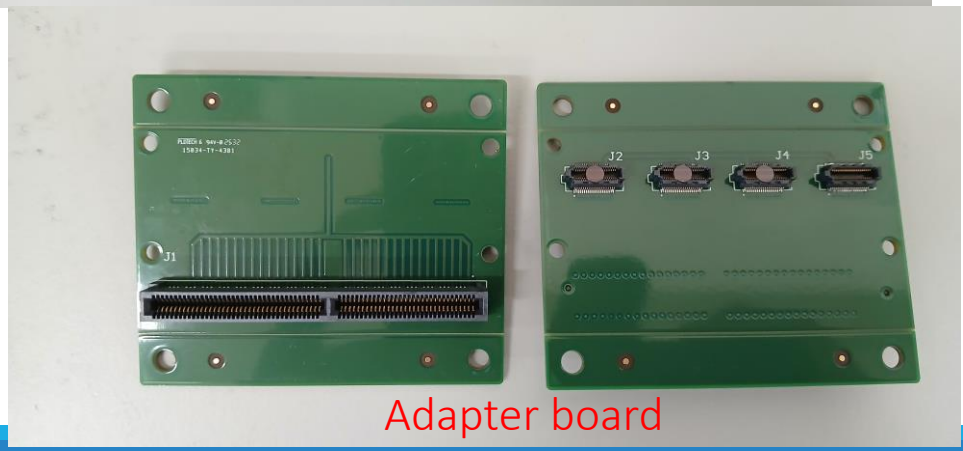


Clock board back



Clock board power

Trigger board power



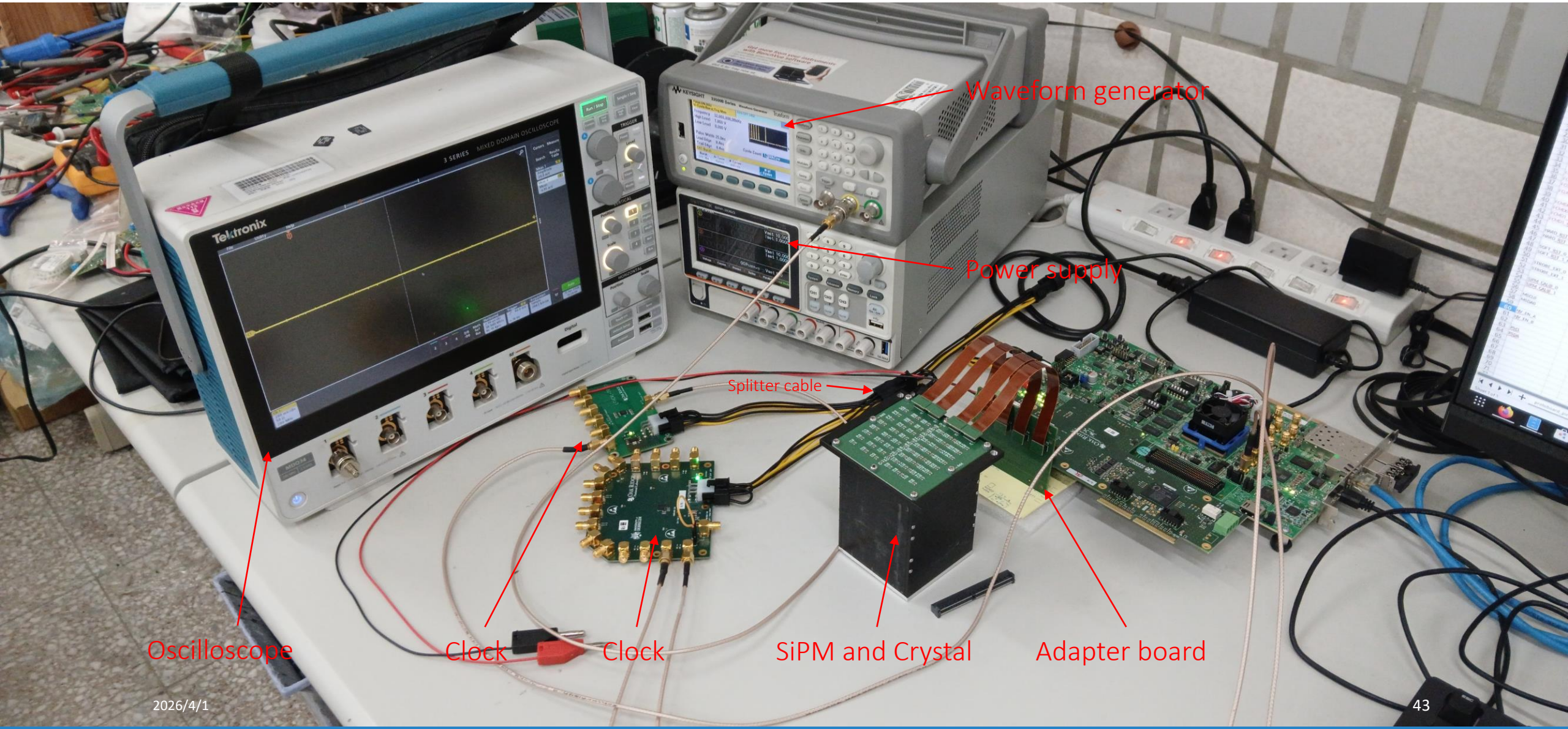
Adapter board

Hardware @NCU

- Device
  - PbWO/LYSO Crystal 1/1
  - SiPM board 1/1
  - Adapter board 2/4
  - protoboard 3/4
  - KCU105 1/2
  - Clock board 1/1
  - Trigger board 1/1
- Equip
  - PC 1/1
  - Power supply (for Clock distribution board and Trigger board) 1/1
  - Signal generator 1/1
  - Network Switch 1GbE 1/1
- Cable
  - SMA male to male cable 3/8
  - SMA to BNC connector 2/6
  - SMA to <unknown> cable for clock board connect to <unknown input> 0/1
  - SMA to <unknown> cable for trigger board connect to signal generator 0/1
  - Power cable for Clock distribution board 1/1
  - Power cable for Trigger board 1/1
  - SiPM board to Adapter board ?/? not sure
  - Network Cable 1GbE 1/2

Assume 2 KCU105 and 4 protoboard setup

Current environment



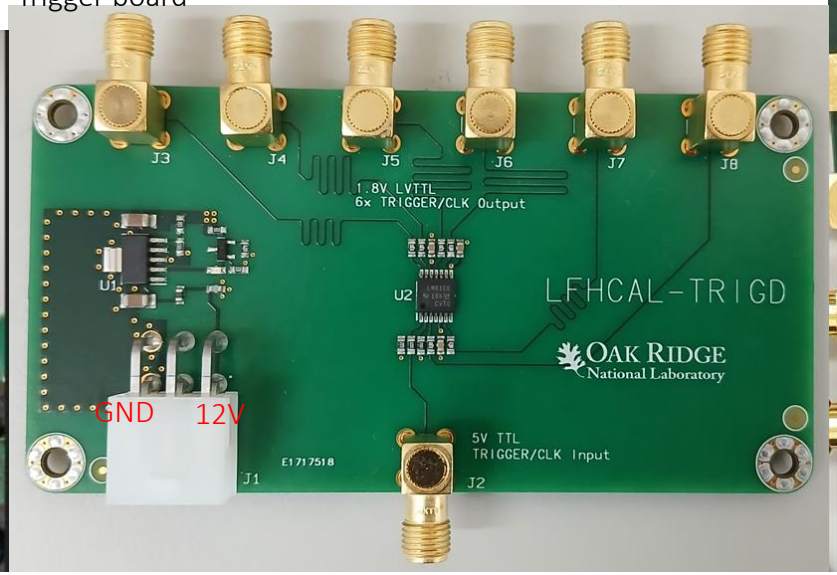
About 12V Power

- Shared FPGA evaluation board power supply via splitter

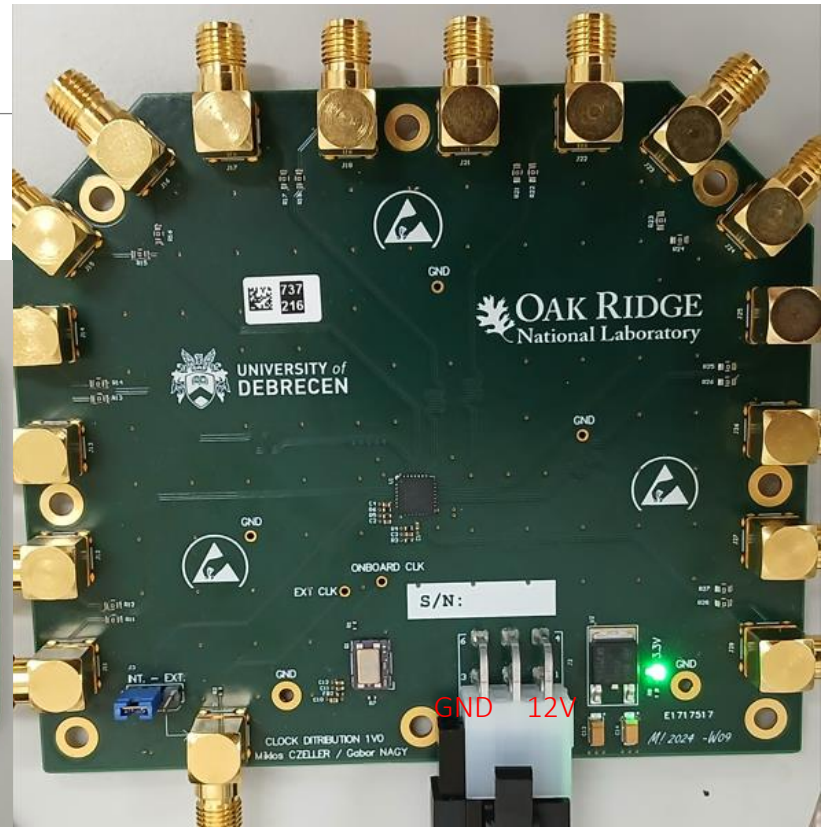
KCU105 Power supply

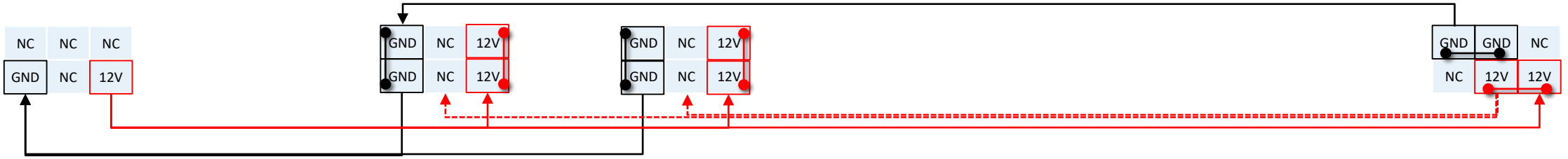
KCU 105

Trigger board



Clock board



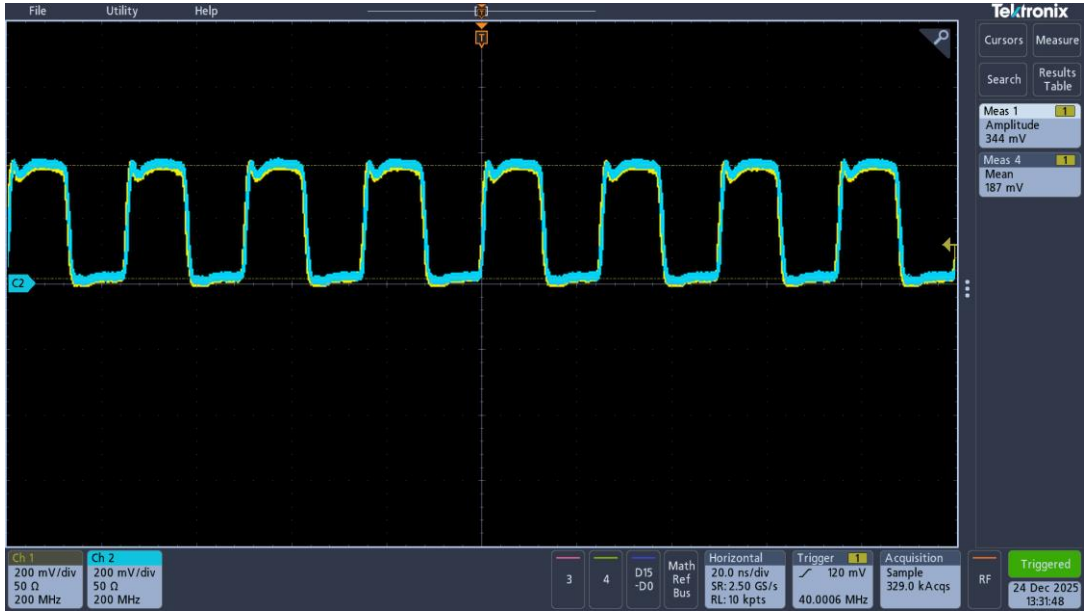


GND	GND	12V
GND	12V	12V

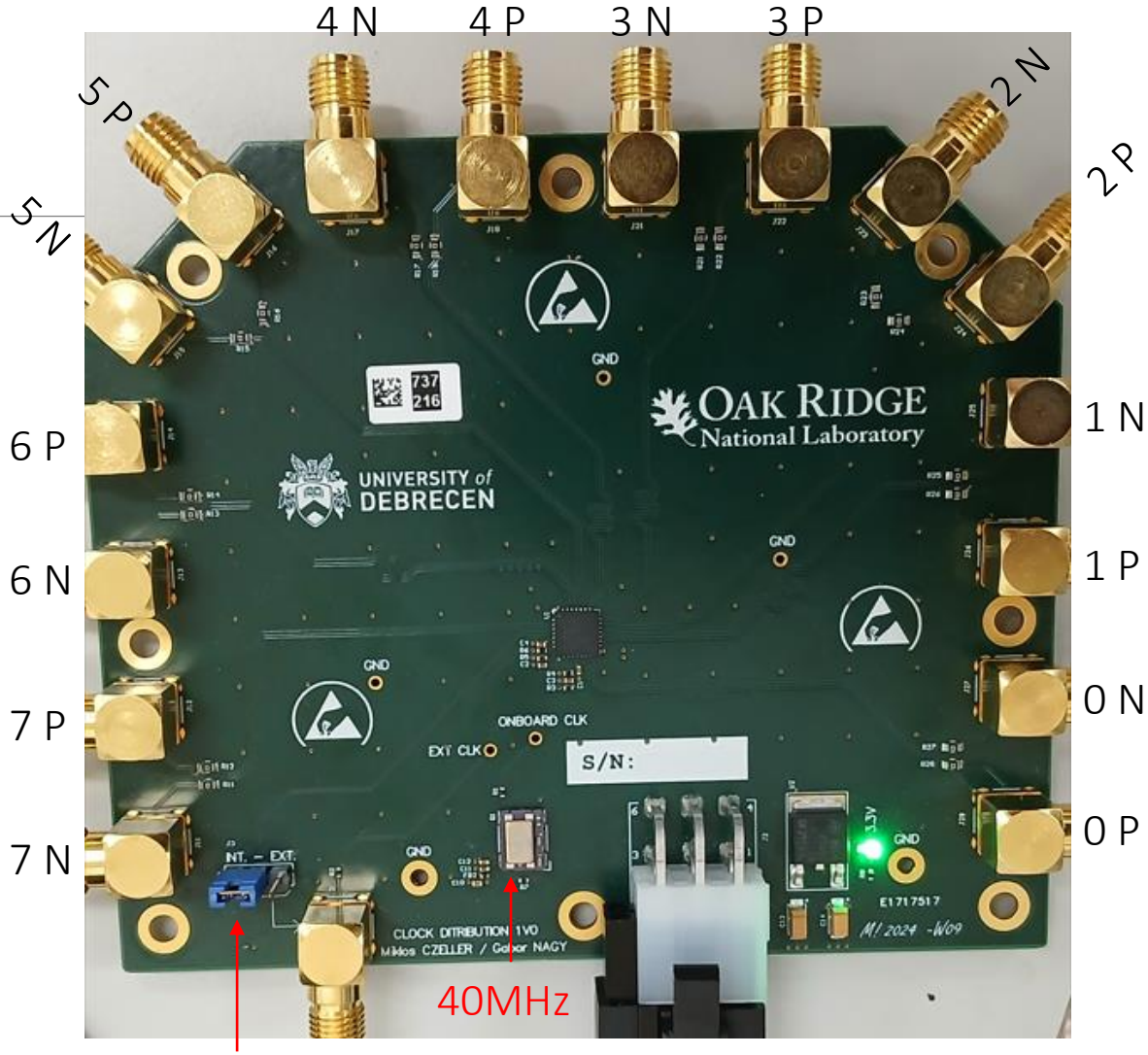
← Connect all them together, your **cable** will be

Clock board

- .



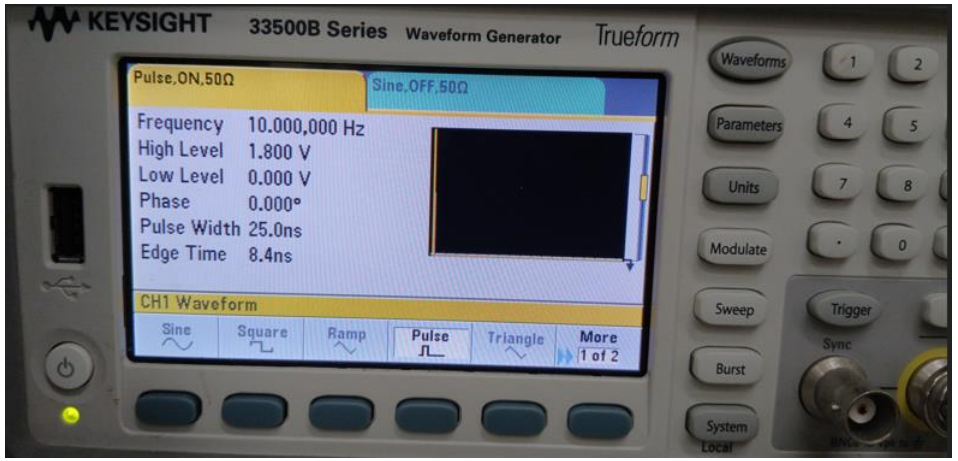
The output of 7P and 6P



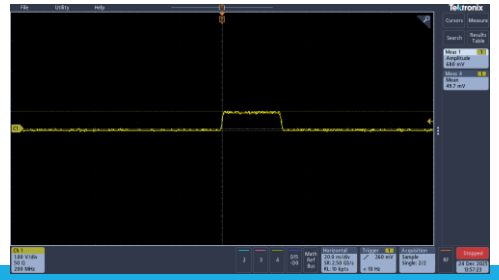
Jumper select:
Int: On board 40MHz 0.5ppm clock
Ext: External input

external trigger signal

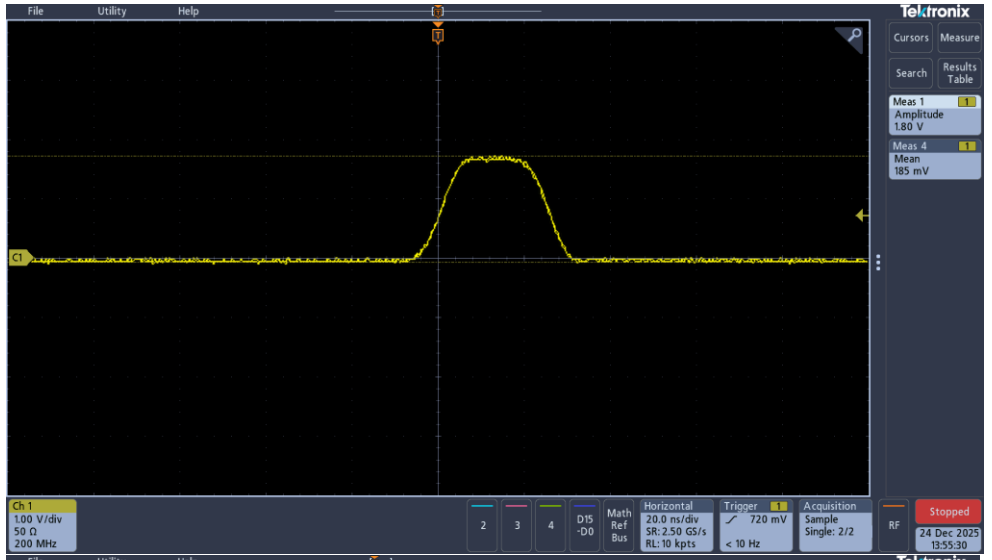
- 1.8V output, 10Hz, Pulse width 25ns
- Send to Trigger input



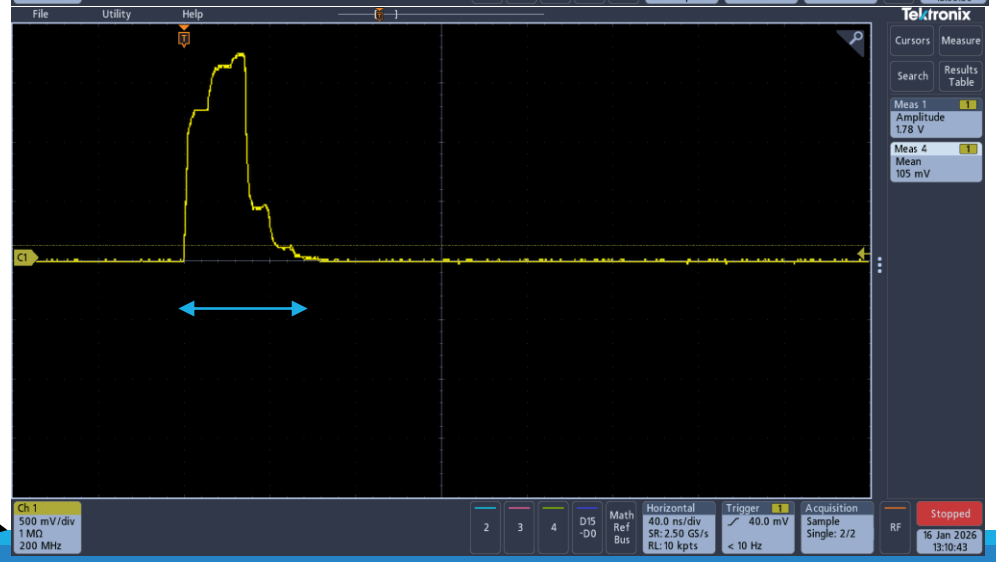
If the input signal voltage decreases, the output width will decrease. Smaller than 0.8V output disappears



1MΩ



Input



Output

UDP packet format (FPGA -> PC)

◇ a UDP packet

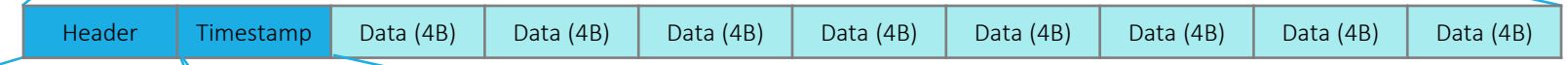
1452 or 8972 Bytes

Normal frame N = 0 ~ 36
JUMBO frame N = 0 ~ 224

UDP header:

2B	Byte Counter	(0 ~ 36) *40
2B	Packet Counter	0 ~ 65535
8B	String (“0”)	

Data (40B):



Data format header:

1B	Header	b0-3: Asic address. b4-7: 0xA0
1B	Fpga board address	0x00 ~ 0x07
1B	packet type	TR0: 0x20, TR1: 0x21, TR2: 0x22, TR3: 0x23, DQ0: 0x24, DQ1: 0x25
1B	packet id	0 ~ 4

Timestamp:

32B	Timestamp	Guess: Unit 25ns, for time sync between FPGAs

UDP packet format (FPGA -> PC)

- One read command generate ten of 40B packet (400Bytes), Two ASIC is 800 Bytes
- Timestamp field is duplicated

	Header	Timestamp	Data (4B)	Data (4B)	Data (4B)	Data (4B)	Data (4B)	Data (4B)	Data (4B)	Data (4B)
DQ0, Half H2GCROC	0, 0, 0x24, 0	Timestamp	DaqH	CM	CH0	CH1	CH2	CH3	CH4	CH5
	0, 0, 0x24, 1	Timestamp	CH6	CH7	CH8	CH9	CH10	CH11	CH12	CH13
	0, 0, 0x24, 2	Timestamp	CH14	CH15	CH16	CH17	Calib	CH18	CH19	CH20
	0, 0, 0x24, 3	Timestamp	CH21	CH22	CH23	CH24	CH25	CH26	CH27	CH28
	0, 0, 0x24, 4	Timestamp	CH29	CH30	CH31	CH32	CH33	CH34	CH35	CRC-32
DQ1, Half H2GCROC	0, 0, 0x25, 0	Timestamp	DaqH	CM	CH0	CH1	CH2	CH3	CH4	CH5
	0, 0, 0x25, 1	Timestamp	CH6	CH7	CH8	CH9	CH10	CH11	CH12	CH13
	0, 0, 0x25, 2	Timestamp	CH14	CH15	CH16	CH17	Calib	CH18	CH19	CH20
	0, 0, 0x25, 3	Timestamp	CH21	CH22	CH23	CH24	CH25	CH26	CH27	CH28
	0, 0, 0x25, 4	Timestamp	CH29	CH30	CH31	CH32	CH33	CH34	CH35	CRC-32

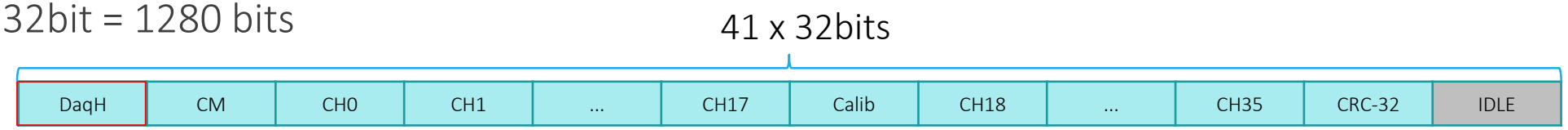
Possibly crosses UDP packet boundaries

A DAQ frame (Next page)

(fpga id, asic id, packet type, packet id)

DAQ frame (H2GCROC->FPGA)

- A read command, send a DAQ frame (From datasheet)
- 40 * 32bit = 1280 bits



DaqH:

4b	hd	0b1111
12b	BxCounter	Value of the Bunch Crossing Counter (BCD) on 12 bits. (40MHz). Range: 1 - 3564. Reset value: 1
6b	Event	Value of the Event Counter (EC) , to detect if a L1 trigger has been sent but related data wasn't saved (RAM2 full). Range: 0 - 63. Reset value: 1
3b	Orbit	Value of the Orbit Counter (OC). Range: 0 - 7. Reset value: 0. Increase 1 when BCD wrapping
1b	H1	Error during hamming decoding in Header
1b	H2	Error during hamming decoding in channels from CM to CH17 (1st Quarter)
1b	H3	Error during hamming decoding in channels from Calib to CH35 (2nd Quarter)
4b	tr	0b0101: no first event flag 0b0010: first event flag

CM:

10b	CM1 ADC
10b	CM0 ADC
12b	0b000

~ 89us BxCounter wrapping

~ 712us OC wrapping

CH#, Calib:

	CASE1, CASE2	CASE3	CASE4
10b	TOA	TOA	TOA
10b	ADC	TOT	TOT
10b	ADC-1	ADC-1	ADC
1b	Tp	Tp	Tp
1b	0	1	Tc

CASE1: Charge Q < TOA Threshold, TOA = 0

CASE2: Charge Q < TOT Threshold, TOA > 0

CASE3: Charge Q > TOT Threshold

CASE4: Debug mode

Tp: TOT-In-Progress

Tc: TOT Complete

-1: Previous BX ?

Decode

```
UDP_TX_Packet_Counter: 57  ByteCounter: 1440
hd: 15  H1: 0  H2: 0  H3: 0  tr: 5  BxCounter: 2827  Event: 13  Orbit: 0
FPGA: 0 ASIC: 0 packet_type: 25 packet_id: 00 timestamp: 0ad3daca Data: fb0b3405 00017000 06800000 09600000 08b00000 06600000 07800000 07800000
FPGA: 0 ASIC: 0 packet_type: 25 packet_id: 01 timestamp: 0ad3daca Data: 06500000 07100000 06200000 06800000 07900000 05f00000 06200000 07900000
FPGA: 0 ASIC: 0 packet_type: 25 packet_id: 02 timestamp: 0ad3daca Data: 08000000 05a00000 04600000 07a00000 05800000 06f00000 07a00000 07b00000
FPGA: 0 ASIC: 0 packet_type: 25 packet_id: 03 timestamp: 0ad3daca Data: 09e00000 06e00000 06700000 09200000 06f00000 05500000 09300000 07200000
FPGA: 0 ASIC: 0 packet_type: 25 packet_id: 04 timestamp: 0ad3daca Data: 07000000 08500000 05b00000 07d00000 06e00000 08b00000 05100000 ec0d330c
timestamp differ 181656266
hd: 15  H1: 0  H2: 0  H3: 0  tr: 5  BxCounter: 3130  Event: 13  Orbit: 7
FPGA: 0 ASIC: 1 packet_type: 24 packet_id: 00 timestamp: 0ad3daca Data: fc3a3785 0001c400 09200000 03b00000 07200000 05b00000 08100000 05c00000
FPGA: 0 ASIC: 1 packet_type: 24 packet_id: 01 timestamp: 0ad3daca Data: 06900000 08500000 08e00000 08400000 08200000 08600000 08000000 0830019b
FPGA: 0 ASIC: 1 packet_type: 24 packet_id: 02 timestamp: 0ad3daca Data: 09800000 08100000 09600000 08b00000 05900000 09000000 07800000 07d00000
FPGA: 0 ASIC: 1 packet_type: 24 packet_id: 03 timestamp: 0ad3daca Data: 05f00000 08500000 08600000 08c00000 06b00000 07800000 07400000 07900000
FPGA: 0 ASIC: 1 packet_type: 24 packet_id: 04 timestamp: 0ad3daca Data: 08900000 07200000 07b00000 08100000 07100000 07100000 06200000 5133d9f4
hd: 15  H1: 0  H2: 0  H3: 0  tr: 5  BxCounter: 2827  Event: 13  Orbit: 0
FPGA: 0 ASIC: 0 packet_type: 24 packet_id: 00 timestamp: 0ad3daca Data: fb0b3405 00013000 06100000 06400000 08d00000 07d00000 0a500000 06000000
FPGA: 0 ASIC: 0 packet_type: 24 packet_id: 01 timestamp: 0ad3daca Data: 06400000 07900000 05900000 07800000 03f00000 07c00000 04900000 07f00000
FPGA: 0 ASIC: 0 packet_type: 24 packet_id: 02 timestamp: 0ad3daca Data: 04e00000 06600000 05700000 05700000 05600000 03100000 04a00000 03200000
FPGA: 0 ASIC: 0 packet_type: 24 packet_id: 03 timestamp: 0ad3daca Data: 03100000 04700000 05200000 06300000 05800000 03e00000 04e00000 05a00000
FPGA: 0 ASIC: 0 packet_type: 24 packet_id: 04 timestamp: 0ad3daca Data: 06000000 07600000 3ff00000 06b00000 05e00000 05800000 04400000 ec98fb1f
hd: 15  H1: 0  H2: 0  H3: 0  tr: 5  BxCounter: 3130  Event: 13  Orbit: 7
FPGA: 0 ASIC: 1 packet_type: 25 packet_id: 00 timestamp: 0ad3daca Data: fc3a3785 00017800 06a00000 05500000 03600000 05f00000 06100000 07000000
FPGA: 0 ASIC: 1 packet_type: 25 packet_id: 01 timestamp: 0ad3daca Data: 05d00000 05a00000 06d00000 05300000 06d00000 06000000 05600000 04900000
FPGA: 0 ASIC: 1 packet_type: 25 packet_id: 02 timestamp: 0ad3daca Data: 04f00000 08000000 06900000 06400000 07000000 04600000 07500000 05600000
FPGA: 0 ASIC: 1 packet_type: 25 packet_id: 03 timestamp: 0ad3daca Data: 07600000 06300000 03f00000 05600000 05c00000 05c00000 04200000 03a00000
FPGA: 0 ASIC: 1 packet_type: 25 packet_id: 04 timestamp: 0ad3daca Data: 05d00000 08e00000 07a00000 01900000 06a00000 02b00000 05a00000 dedc3d7c
...
```

Raw Data

- A custom program can send start_daq and store UDP packet
- red boxed fields require byte swapping, i.e. Big-endian

For firmware V4.11, the last UDP packet is truncated. And it also send a full zero packet when stopping DAQ.

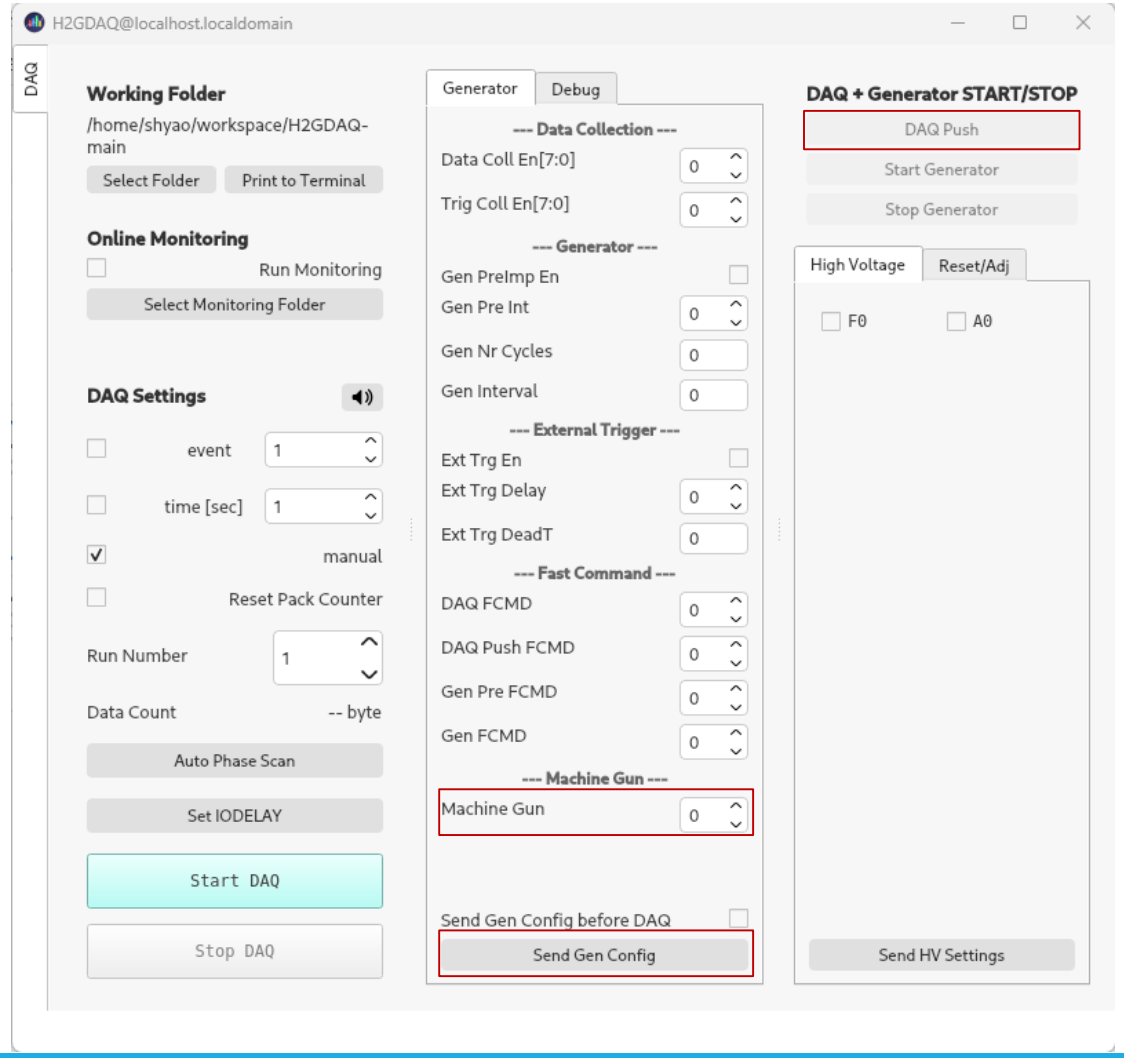
```
[shyao@cats myprj]$ od -Ax -t x4 -N 12 data/rawdata.10.bin
000000 5300a005 00000000 00000000     UDP Header
00000c
[shyao@cats myprj]$ od -Ax -t x4 -j 12 -w40 -N 1452 data/rawdata.10.bin
00000c 02400a0 9c626808 85ad3cf5 00280100 00009005 0000d005 00006008 00001001 0000d009 0000b005
000034 12400a0 9c626808 0000a005 0000a00f 0000b005 00004007 0000b003 00005007 0000f003 00008007
00005c 22400a0 9c626808 00006004 00002006 0000f03f 0000f03f 00004005 0000f03f 00002004 0000e002
000084 32400a0 9c626808 00008002 00005004 0000a004 0000f005 00002005 00009003 0000c004 00005005
0000ac 42400a0 9c626808 0000b005 0000f03f 0000f03f 00005006 00006005 0000f03f 0000d003 cb3a5347
0000d4 02500a1 9c626808 05ad6bf6 00780100 0000c006 00005005 00008003 0000e005 0000c005 00002007
0000fc 12500a1 9c626808 0000f005 0000a005 0000d006 00004005 0000e006 00001006 00009005 00009004
000124 22500a1 9c626808 0000f004 00004008 0000a006 00002006 00003007 00005004 00002007 00006005
00014c 32500a1 9c626808 00008007 00006006 00002004 0000a005 0000d005 0000d005 00001004 0000d003
000174 42500a1 9c626808 0000d005 00002009 0000a007 00009001 0000a006 0000d002 0000e005 da35dee7
00019c 002500a0 9c626808 85ad3cf5 41010400 00001010 00000000 00009000 00001000 00000010 00001000
0001c4 012500a0 9c626808 00001000 00000000 00001000 00000000 00001000 00009017 00001000 00006015
0001ec 022500a0 9c626808 00009018 00001015 00000000 00001000 00001010 00001000 00001010 0000f017
000214 032500a0 9c626808 00000000 00003018 00001000 00001001 00000000 00000000 00000019 00001010
00023c 042500a0 9c626808 0000d016 00001010 00000010 00008019 fc010000 00001019 00002010 625b12cb
000264 002400a1 9c626808 05ad6bf6 00c80100 00000009 0000d003 00002007 0000a005 00002008 00009005
00028c 012400a1 9c626808 0000a006 00004008 0000f008 00006008 00001008 00008008 0000f007 00004008
0002b4 022400a1 9c626808 00006009 00000008 00003009 0000a008 00007005 0000f008 00008007 0000b007
0002dc 032400a1 9c626808 00000006 00003008 00006008 00008008 0000a006 00007007 00005007 00009007
000304 042400a1 9c626808 0000a008 00001007 0000d007 0000f007 0000f006 0000f006 0000f005 ce6c3496
00032c 002400a0 c5626808 85b13df5 00300100 0000a005 0000c005 00008008 00001001 0000c009 0000b005
000354 012400a0 c5626808 0000c005 0000600f 0000a005 00004007 0000b003 00005007 00000004 00006007
00037c 022400a0 c5626808 00008004 00002006 0000f03f 0000f03f 00002005 0000f03f 00006004 0000f002
0003a4 032400a0 c5626808 0000b002 00004004 0000c004 0000e005 00003005 0000a003 0000f004 00008005
0003cc 042400a0 c5626808 0000e005 0000f03f 0000f03f 00005006 00006005 0000f03f 0000f003 808acca5
0003f4 002500a1 c5626808 05b16cf6 00780100 0000d006 00004005 00009003 0000e005 0000e005 00001007
00041c 012500a1 c5626808 00000006 00009005 0000d006 00004005 0000f006 00001006 0000a005 0000b004
000444 022500a1 c5626808 00000005 00004008 0000a006 00005006 00002007 00004004 00005007 00006005
00046c 032500a1 c5626808 00008007 00008006 0000f003 0000a005 0000f005 0000e005 0000f003 0000d003
000494 042500a1 c5626808 0000d005 00000009 00009007 00008001 0000c006 0000d002 0000d005 f1be7c7b
0004bc 002500a0 c5626808 85b13df5 41010400 00001010 00000000 00009000 00001000 00000010 00001000
0004e4 012500a0 c5626808 00001000 00000000 00001000 00000000 00001000 00009017 00001000 00006015
00050c 022500a0 c5626808 00009018 00001015 00000000 00001000 00001010 00001000 00001010 0000f017
000534 032500a0 c5626808 00000000 00003018 00001000 00001001 00000000 00000000 00000019 00001010
00055c 042500a0 c5626808 0000d016 00001010 00000010 00008019 00000000 00001019 00002010 70dc1c4d
000584 002400a1 c5626808 05b16cf6 00c40100 00001009 0000d003 00001007 0000b005 00002008 0000c005
0005ac 5400a005 00000000 00000000     Next UDP Header
0005b8
------ 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

DAQ mode

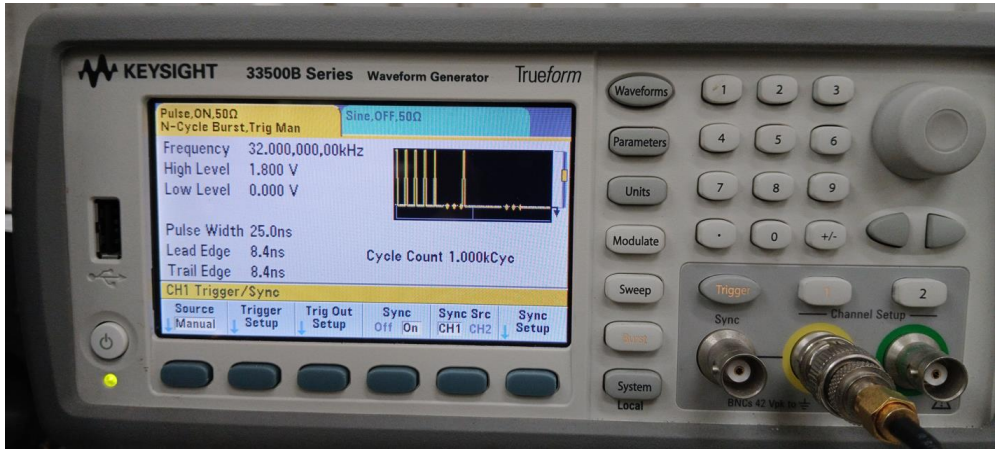
- In practice, the external trigger signal is sent to the FPGA.
- Four possible mode
  - DAQ Push
  - Generator
  - Internal trigger
  - External trigger
- Common parameter - Machine Gun
  - The FPGA firmware generates a series of read commands and sends them to H2GCROC.
  - The real number of fast read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

DAQ Push

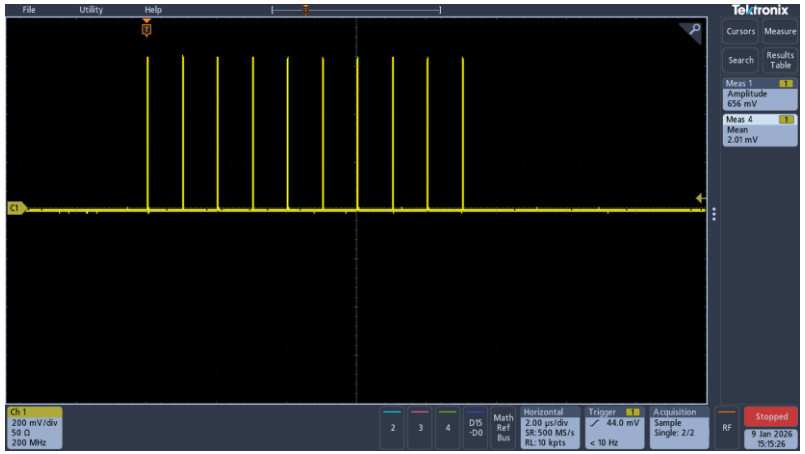
- Software trigger
- Send this trigger by user interface



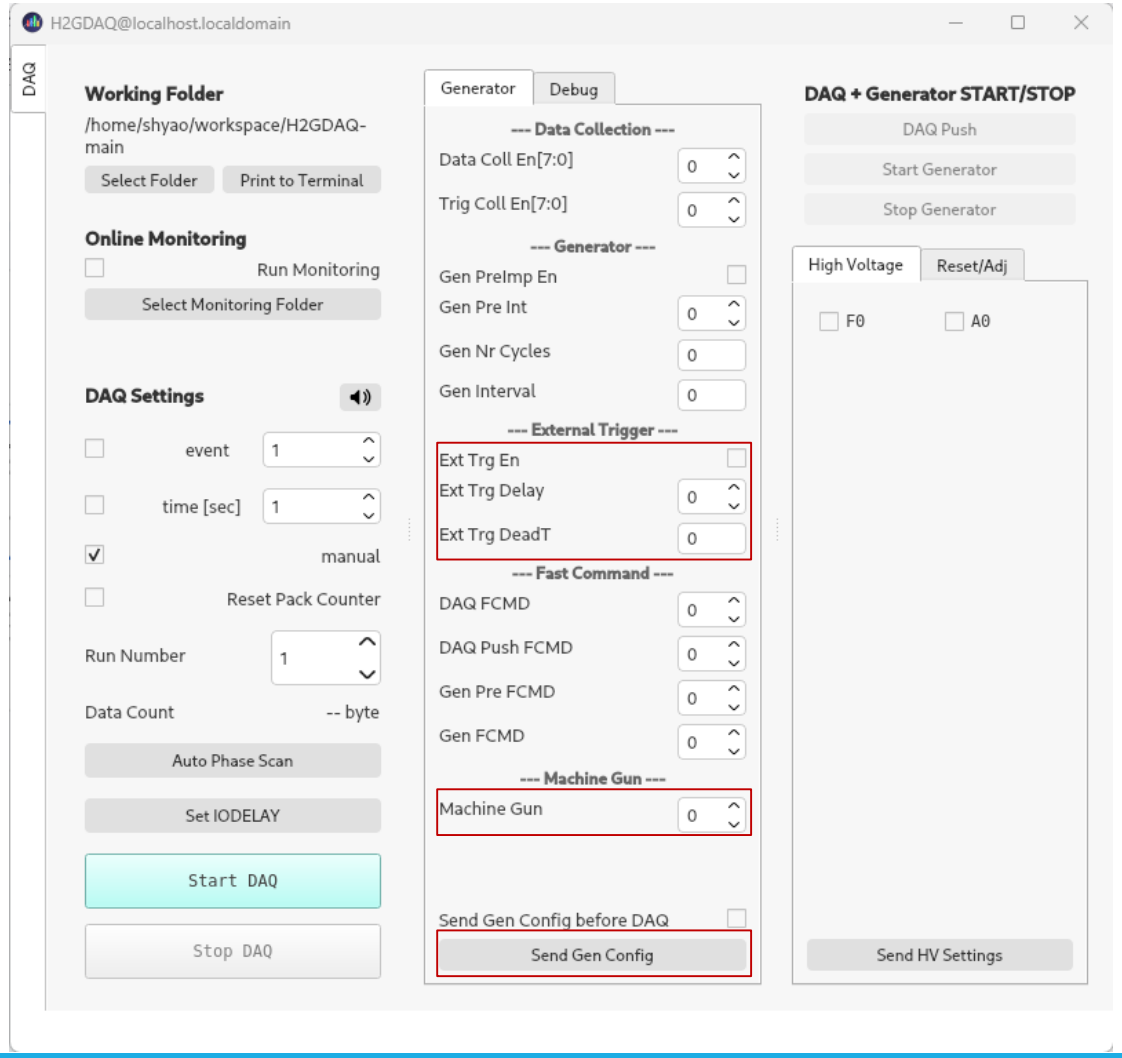
External trigger



test signal : 32KHz, 1000 pulse

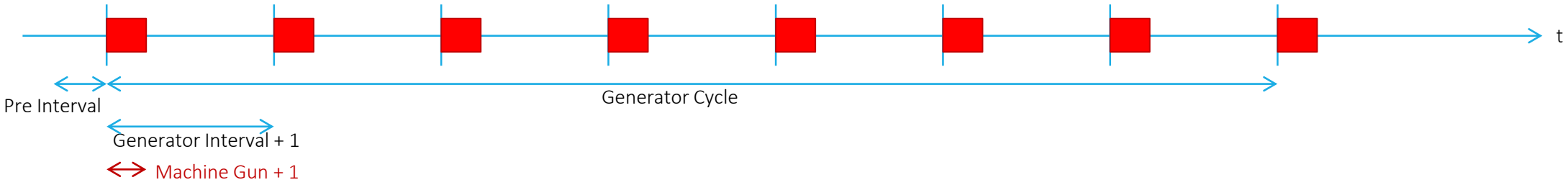


test signal : 1MHz, 10 pulse



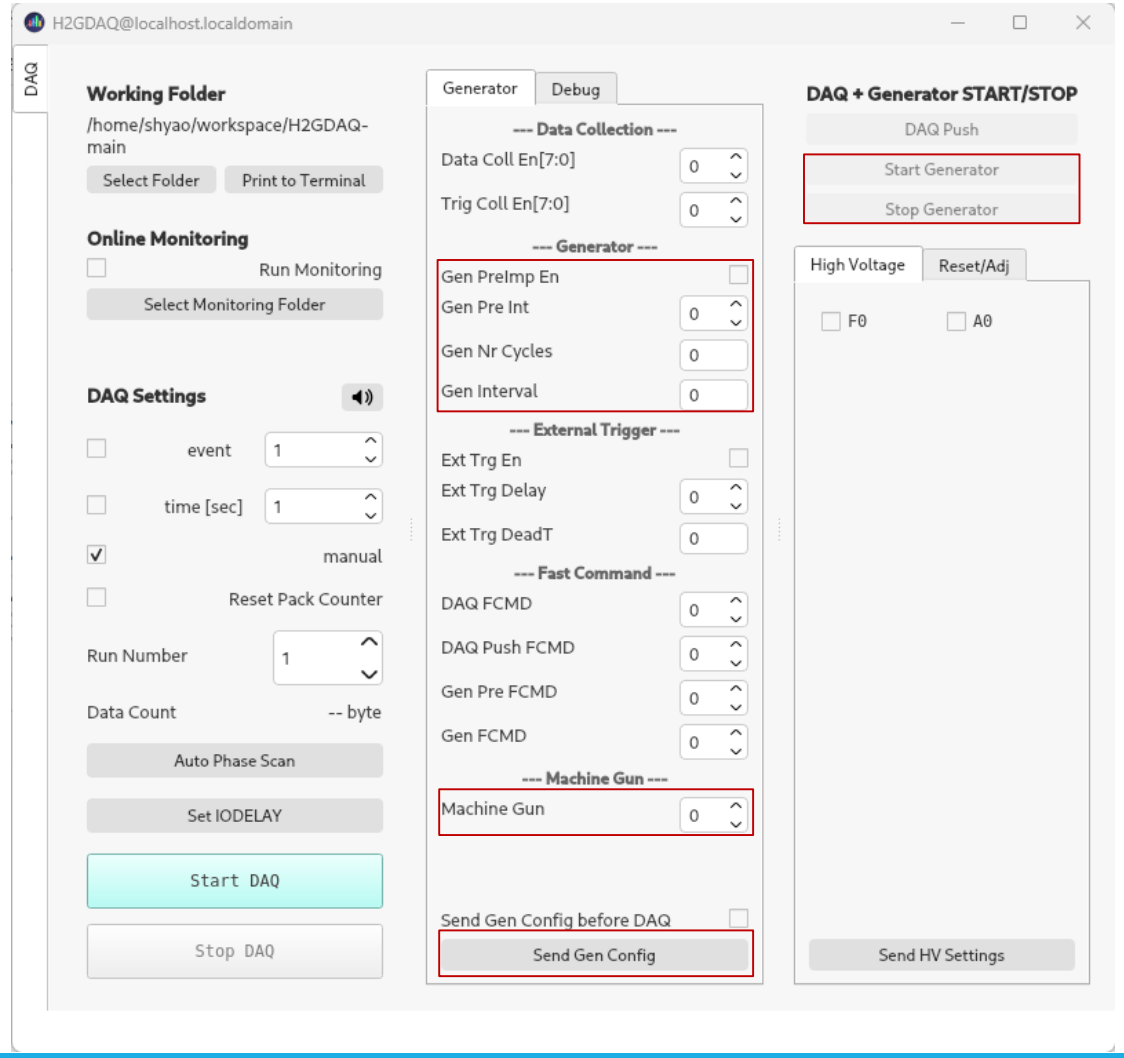
Generator

- 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)



Generator

- Send parameter by user interface

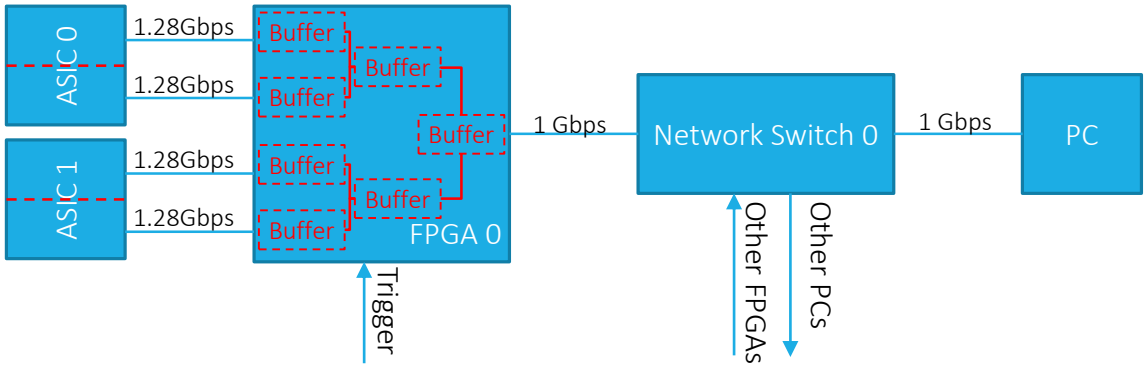


Internal trigger

- No way was found to enable this feature from the user interface. Try using custom code.
- FPGA firmware manual say it only support ASIC 0

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, it takes 1.025 microseconds to transmit data to FPGA.
- If maximum consecutively read (Machine Gun = 31)
  - It takes 32.8 microseconds to transmit data to FPGA.
  - Maximum trigger rate is about 6kHz

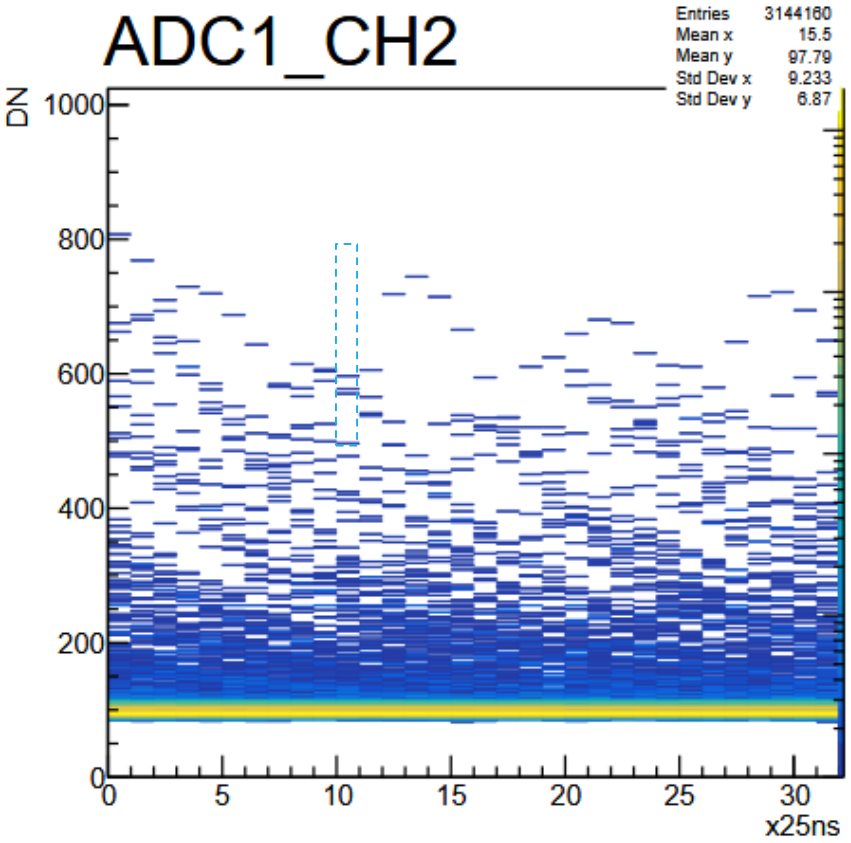


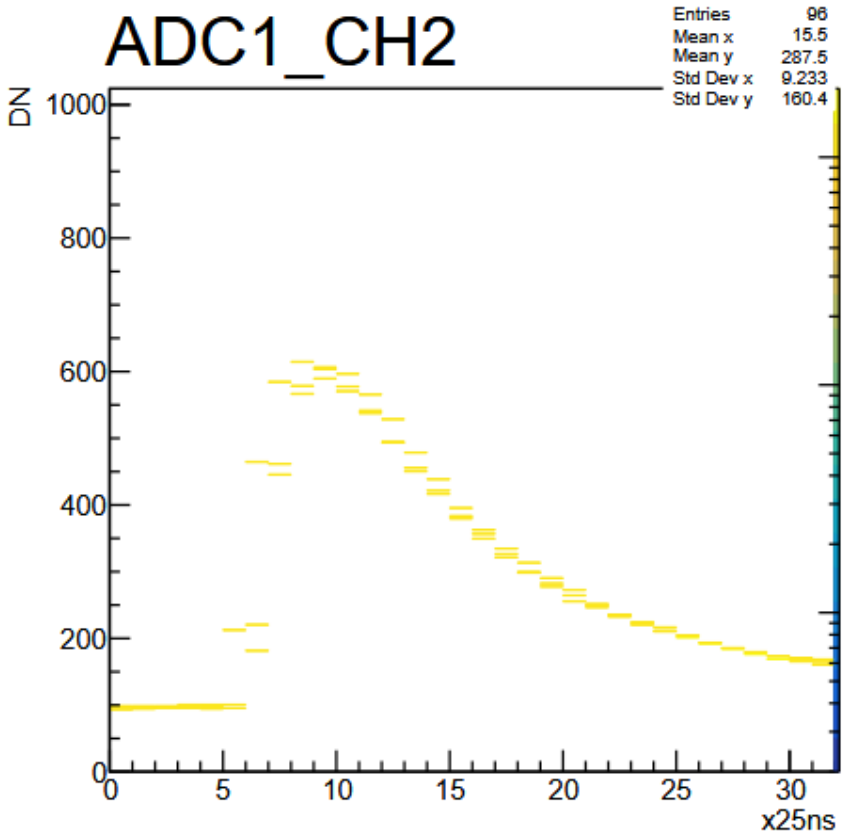
Data transfer in parallel, Each lane handle half channels. ~5 Gbps

1312 bits = header + half channel + crc + idle

DAQ test

- Set generator parameter to 1KHz rate, 100k cycles. Set machine gun to 32 read
- Set High voltage to 28V.

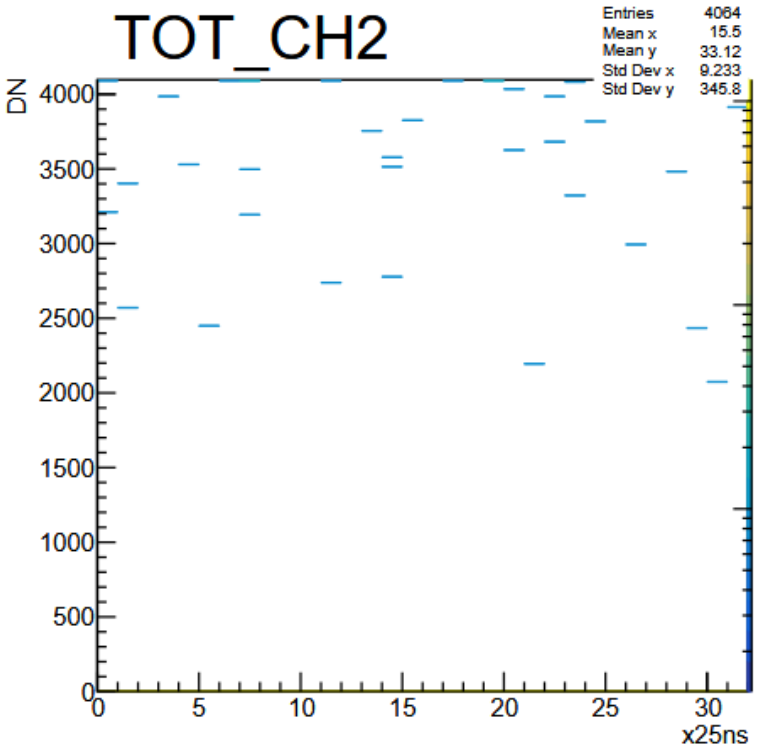


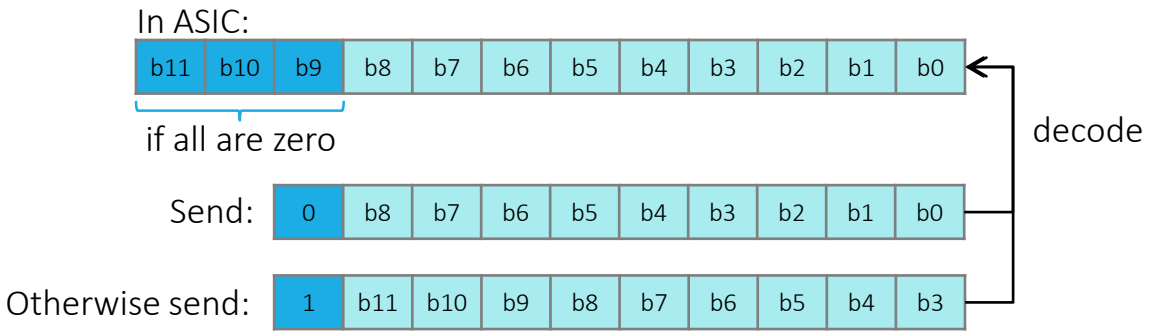


All other connected 63 channels is similar.

TOT

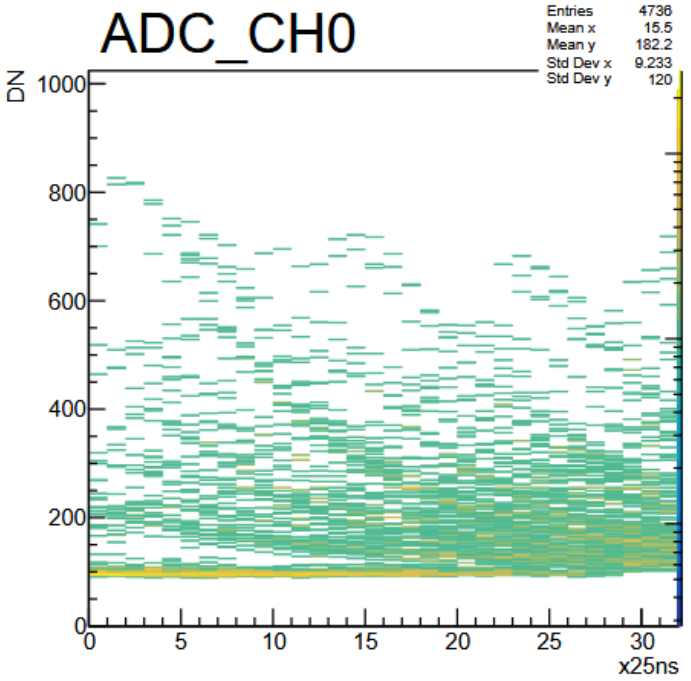
- TOT is 12 bit in ASIC, but send 10 bit only

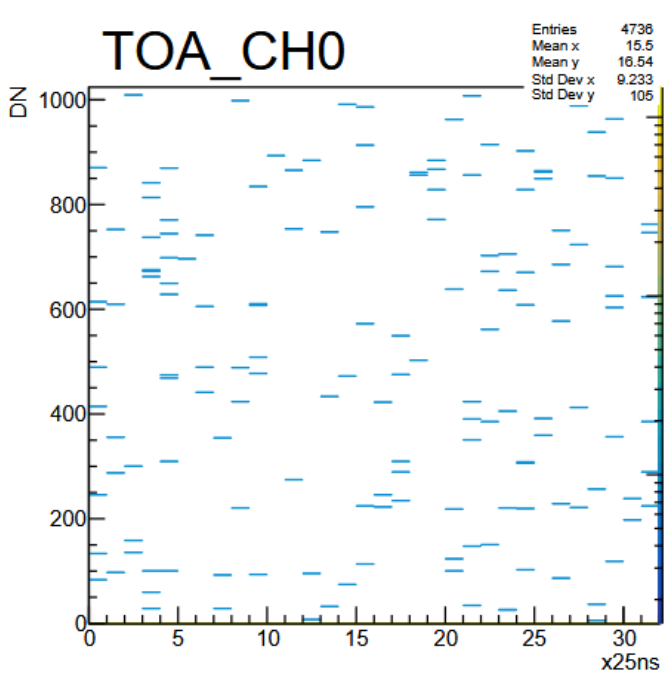


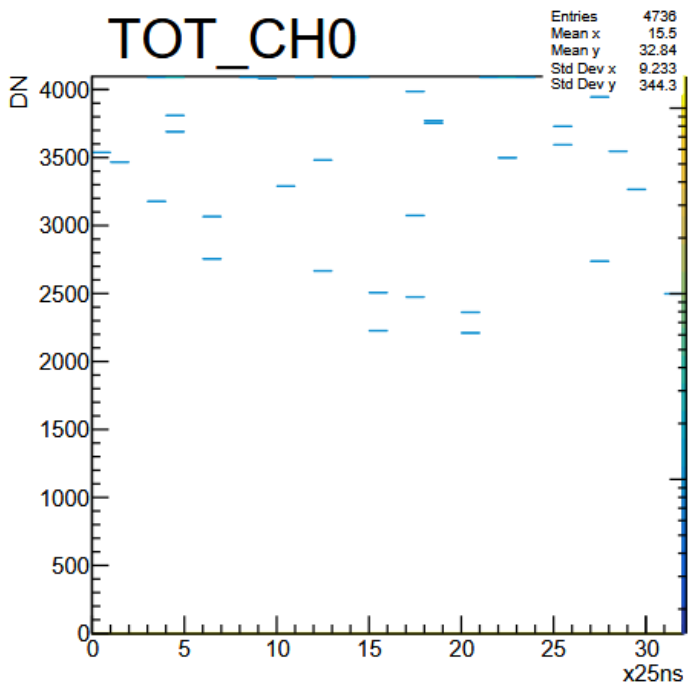


TOA

- TOA output is not zero when signal over the threshold
  - I use it as the filter
- 28V, Efficiency: 4736/32/98258 = ~ 0.1% ~ 0.3%

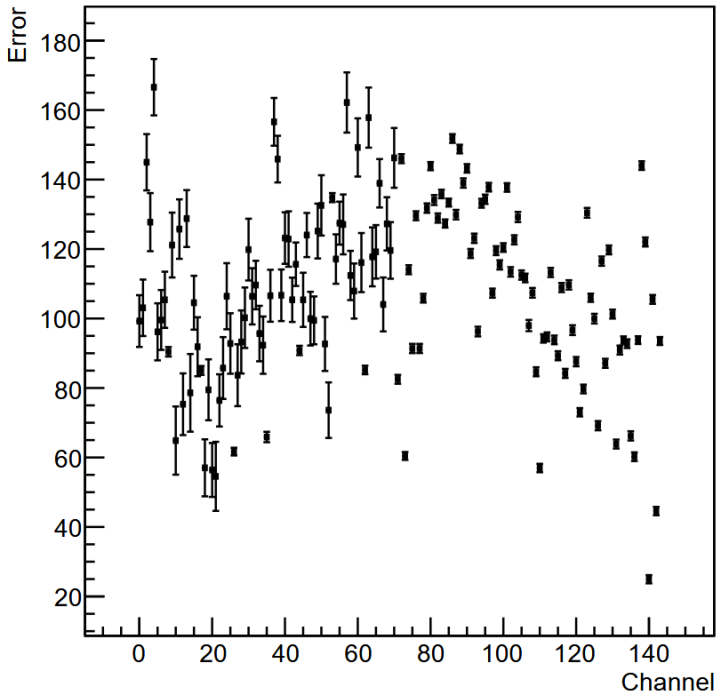






Noise

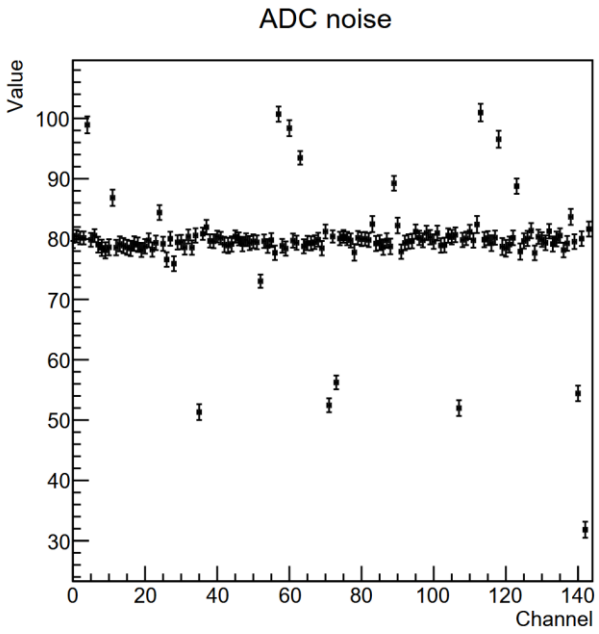
- 28V. Event Filter: (TOA == 0)
- pedestal calibration is wrong



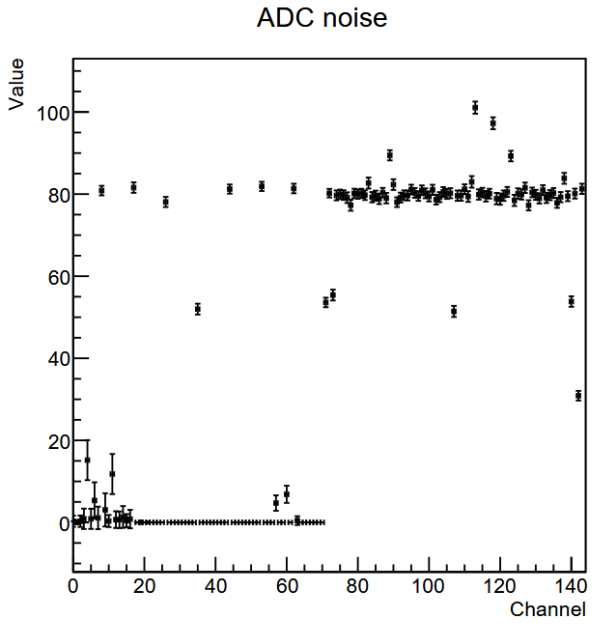
28V Noise

Noise

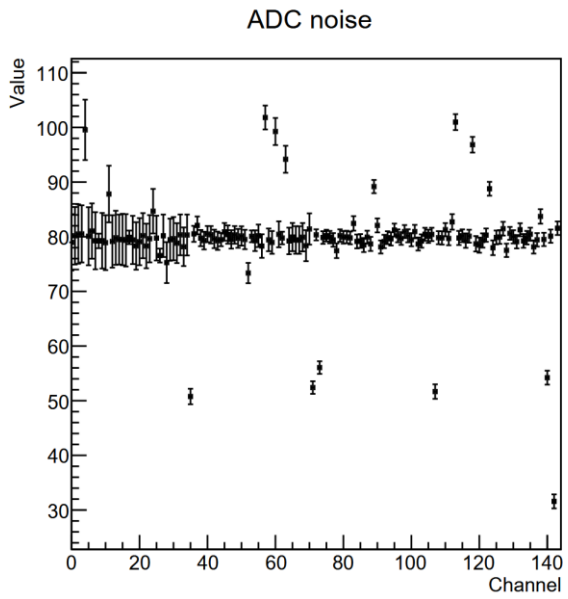
- Event Filter: (TOA == 0)



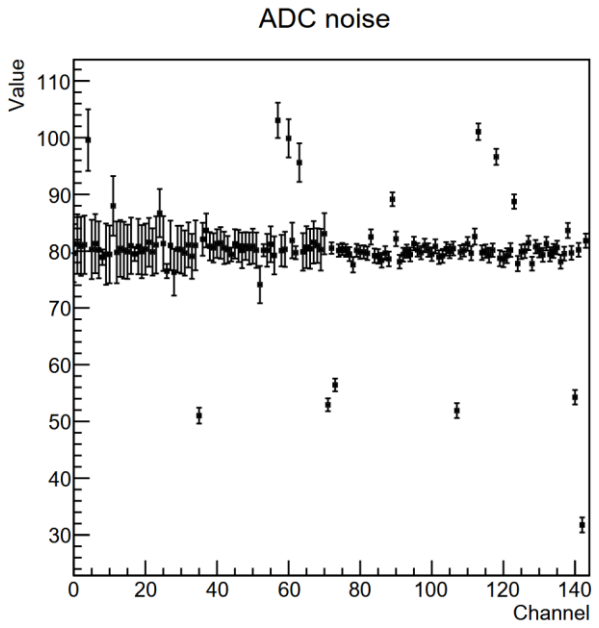
float



0V



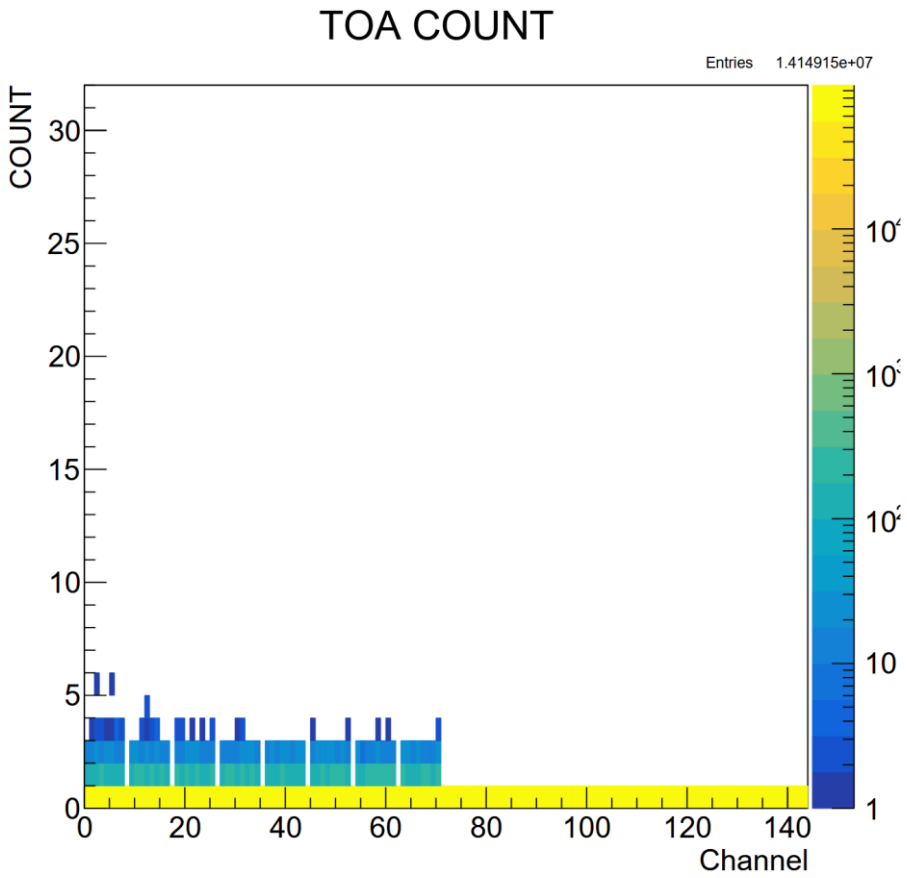
10V

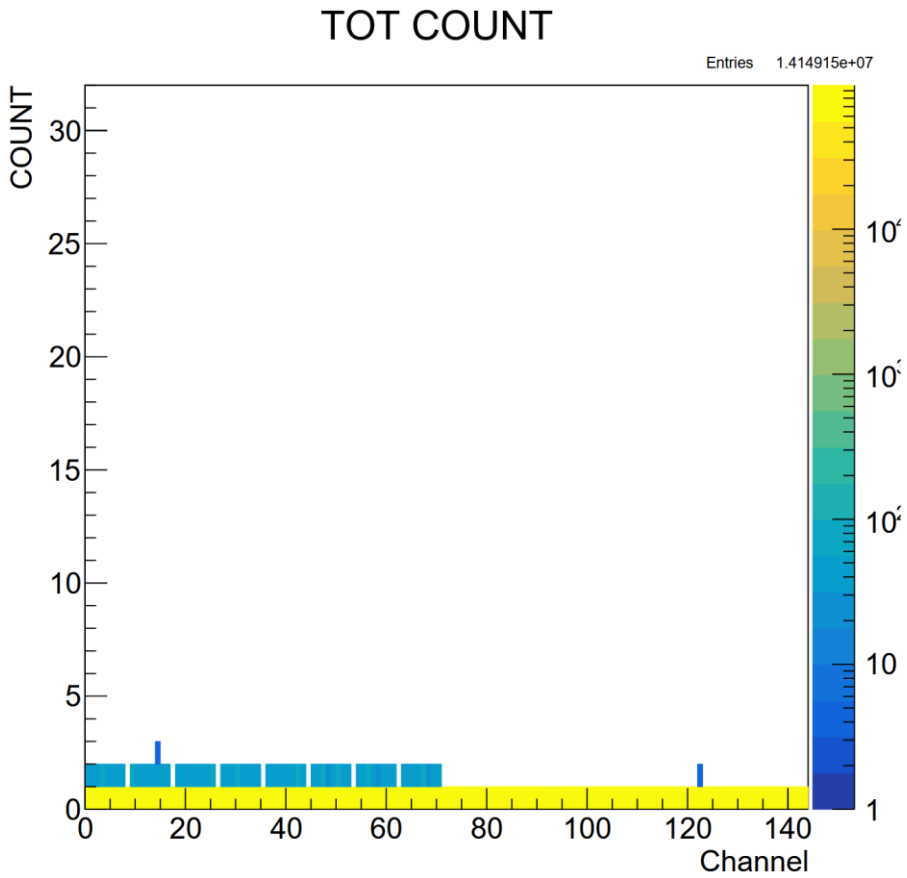


28V

Count

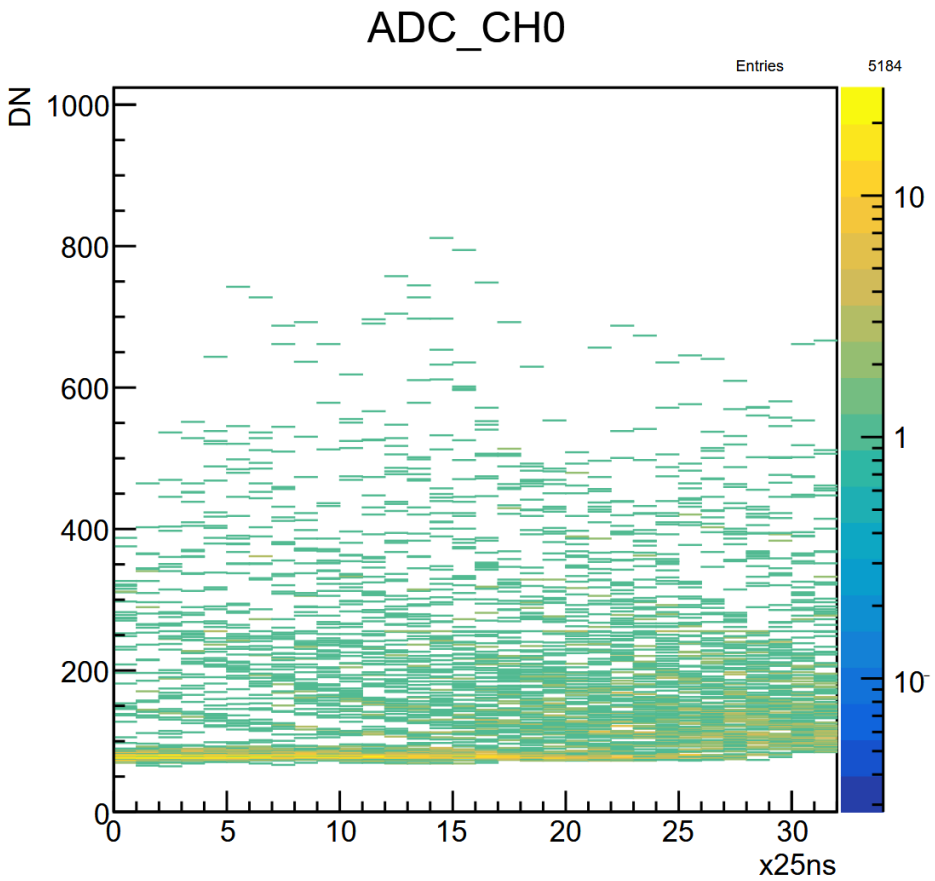
- 28V

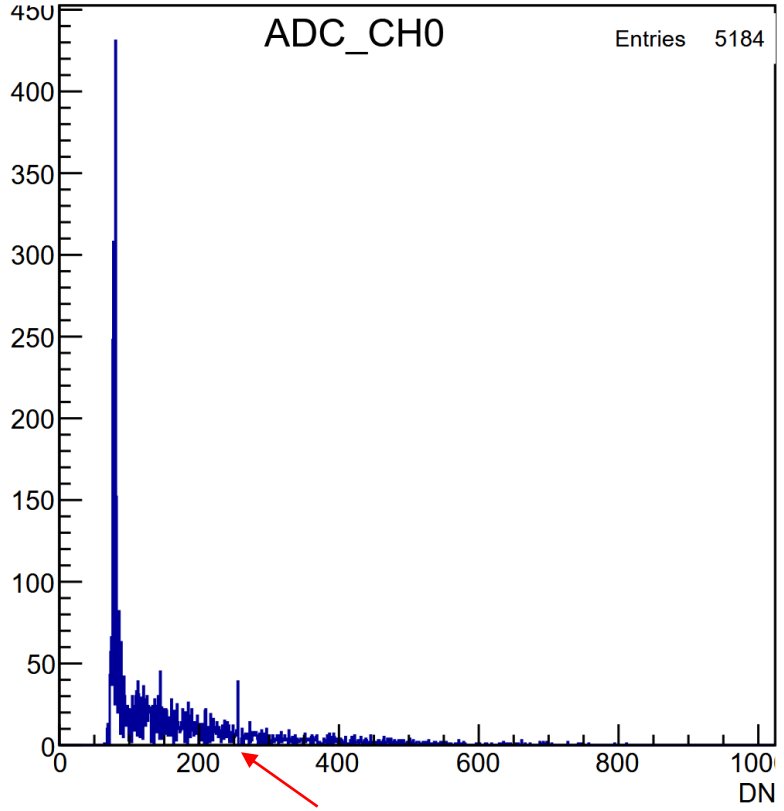

TOA COUNT
Entries 1.414915e+07
COUNT
Channel


TOT COUNT
Entries 1.414915e+07
COUNT
Channel

ProjectionY

- 28V. Event Filter: (TOA > 0)





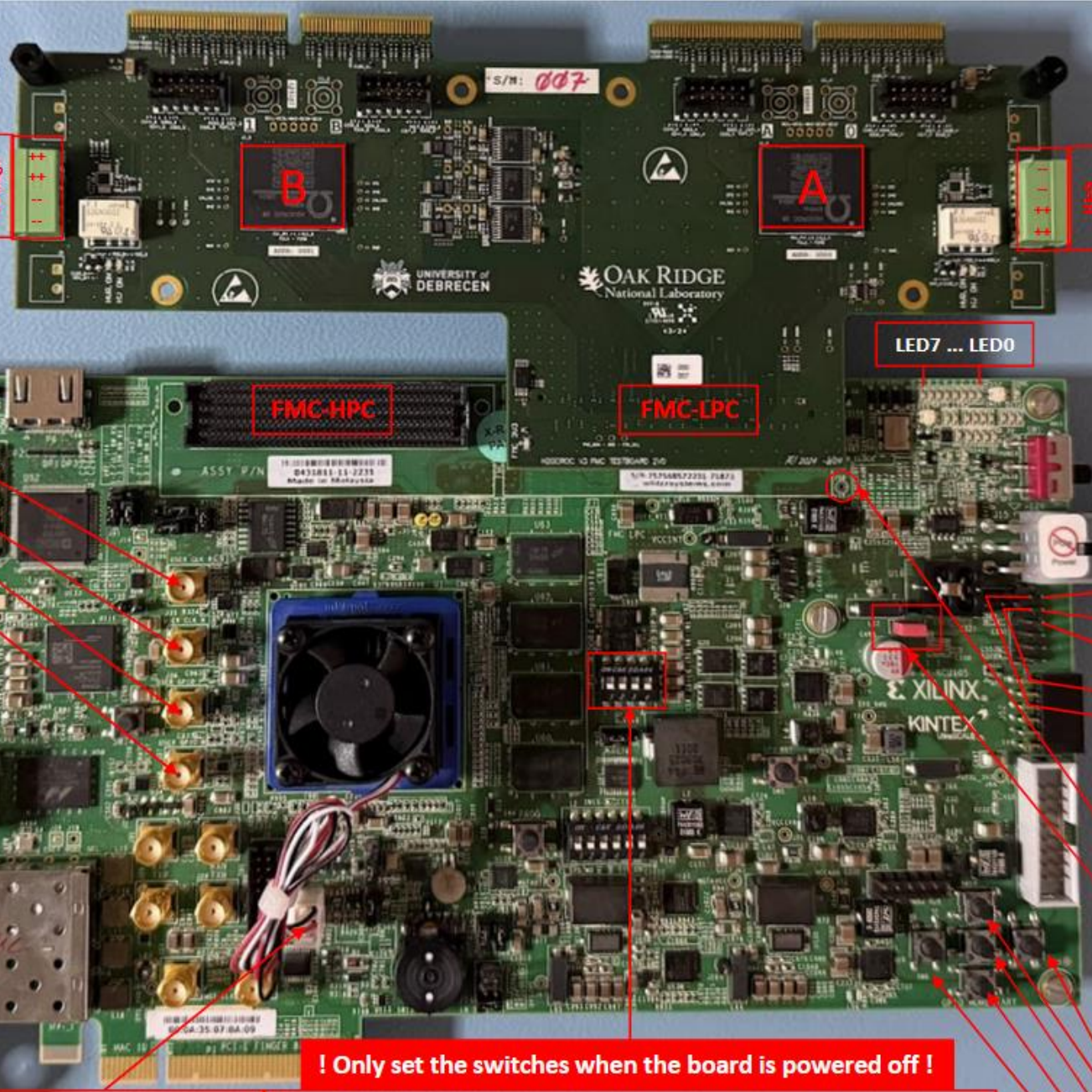
All channel there is a peak.

HGCROC „ProtoBoard” – 5.069
V2 (Blue) V3 (Green) Setup for KCU105

FW version example: 5.64
„5” - Main version / „64” - Sub version
Subversion Start From:
0.. - 10GbE LPC / 64.. - 10GbE LPC+HPC
128.. - 1GbE LPC / 192.. - 1GbE LPC+HPC

FW handles I2C address shifting. Set the ASIC addresses to 0 and 1.
No resistor soldering / changing is necesarry!

In case of KCU105 and ProtoBoard
- 1 ProtoBoard: use the right side (LPC) slot
- 2 ProtoBoard: use both



High Voltage A and High Voltage B is not connected to each other

LED7 - PM BUS / Board Config Ready (Left)
LED6 - External Clock
LED5 - 160MHz Clock ok, ≈1Hz Blinking
LED4 - COMM_ETH_STATUS(0) and
COMM_ETH_READY and
COMM_MAC_RECEIVED
LED3 - COMM TX ACTIVE
LED2 - COMM RX ACTIVE
LED1 - DAQ Started
LED0 - Internal generator is running (Right)

Ext. CLK (40MHz) In P - LVDS
Ext. CLK (40MHz) In N - LVDS
Ext. Trigger In - CMOS 1.8V
Ext. Sync In - CMOS 1.8V

Ext. Sync Out - CMOS 3.3V - Not Used
PM I2C SDA - DBG - CMOS 3.3V - Not Used
PM I2C SCL - DBG - CMOS 3.3V - Not Used

SiPM Calib1 - CMOS 3.3V - Not Used
SiPM Calib0 - CMOS 3.3V - Not Used

Not Used
For Board Programming
1GbE Connection

Packet Size:
- **Control - 46 Byte (Complete: 74+14)**
- **Data Long - 1358 Byte (Complete: 1386+14)**
- **Data Jumbo - 8846 Byte (Complete: 8874+14)**

1.2V after boot

! Don’t forget to install this Jumper !
All others: Factory Default!

10GbE - **! Use the upper slot !**
Tested with: TP-Link TL-SM5310-T
No any special requirement

! Only set the switches when the board is powered off !

Fan is operating when the temperature is over 50°C (SW can overwrite)

Pos. 1 (Left) - On: External Clock / Off: OnBoard Clock
Pos. 2 - On: Dataport = 11001 / Off: Dataport = 11000
Pos. 3-4: Board Address: P3-Bit1, P4-Bit0 >> 4 different address

Manual HV On/Off for Right/A side (E)
Manual External Sync (N) - Not Used
Board RESET (C)
Manual External Trigger (S) - Not Used
Manual HV On/Off for Left/B side (W)

10G HW

- FMC Cable: HDR-169468-01 or HDR-169468-02 (4 rows)
  - https://www.digikey.tw/zh/products/detail/samtec-inc/HDR-169468-01/6678214
- 10GbE, SFP to RJ45: TP-Link TL-SM5310-T
- For PC with Thunderbolt 4: QNAP qna-t310g1t
  - https://www.qnap.com/zh-tw/product/qna-t310g1t
- 10Gbps Switch – Netgear XS512EMv2 NT$39000
- CAT6a/CAT.7 Cable, SSD
  - todo

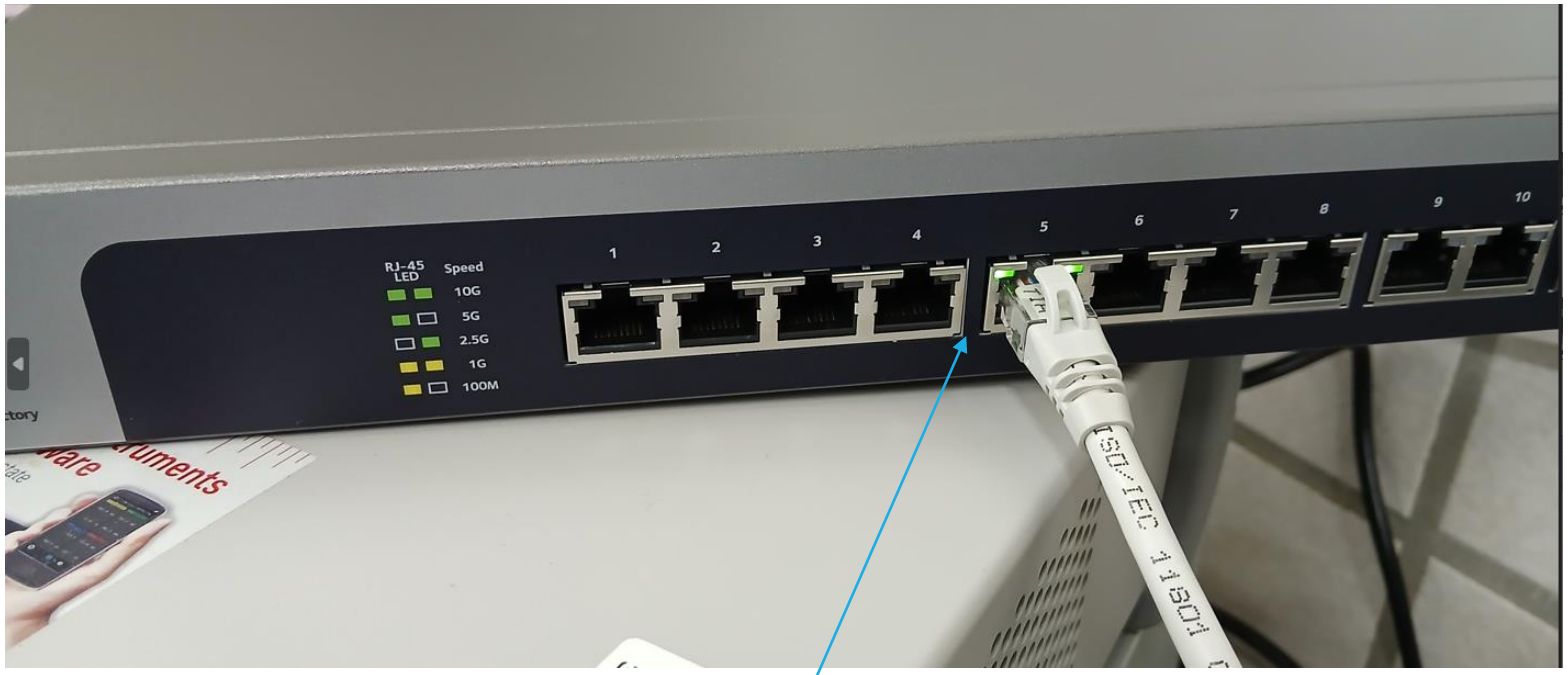
NT$ 1900 x4

NT$ 5084 x4

NT$ 6990 x1

10G Hardware is ready

- 2026/3/4



10G is work

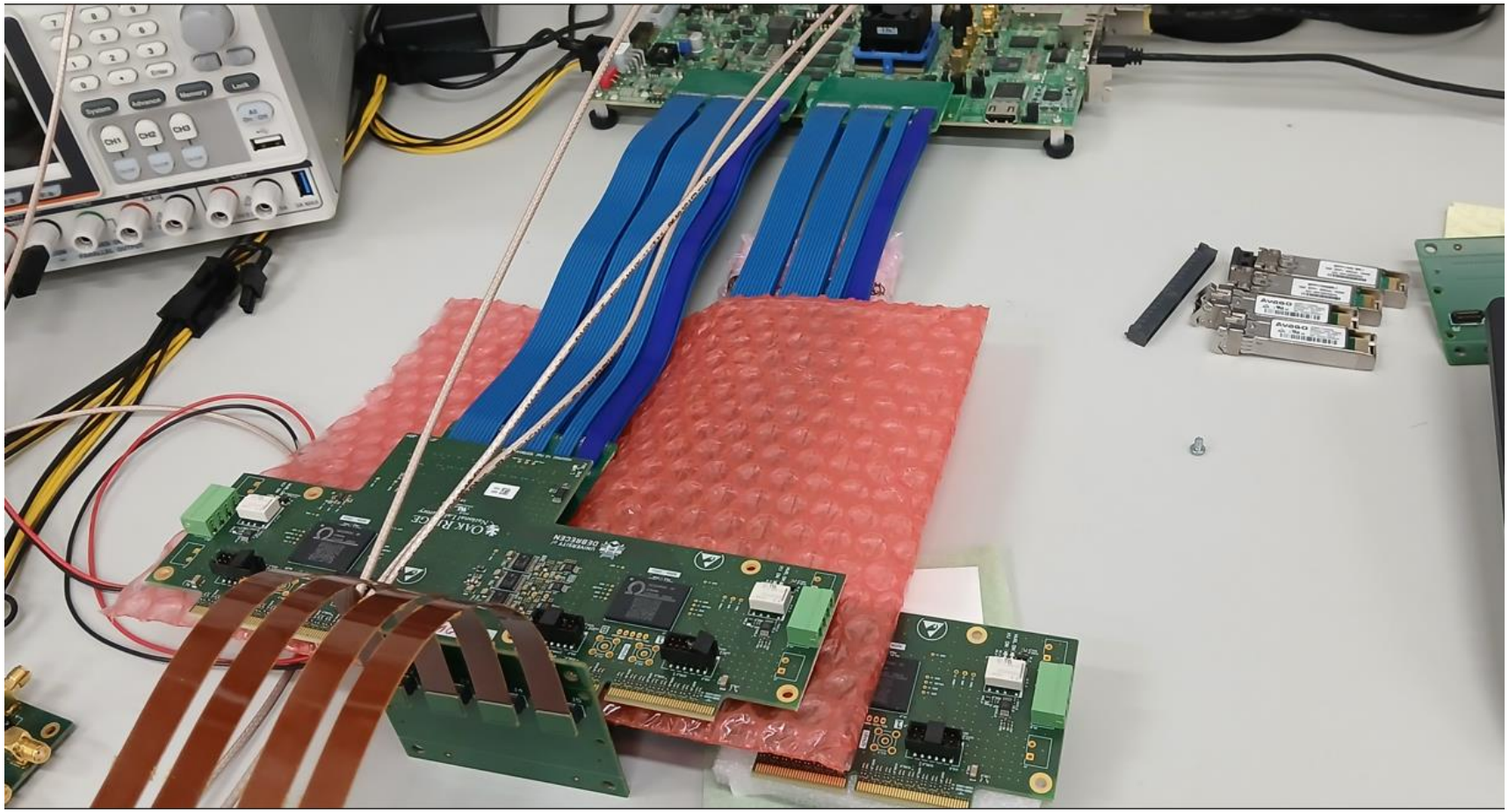
Test FMC Cable

- The cable is very stiff, unable to insert

The LPC connector damaged and frequent failures in I/O delay adjustment.

Test FMC Cable

- Currently



Test IO delay with FMC cable

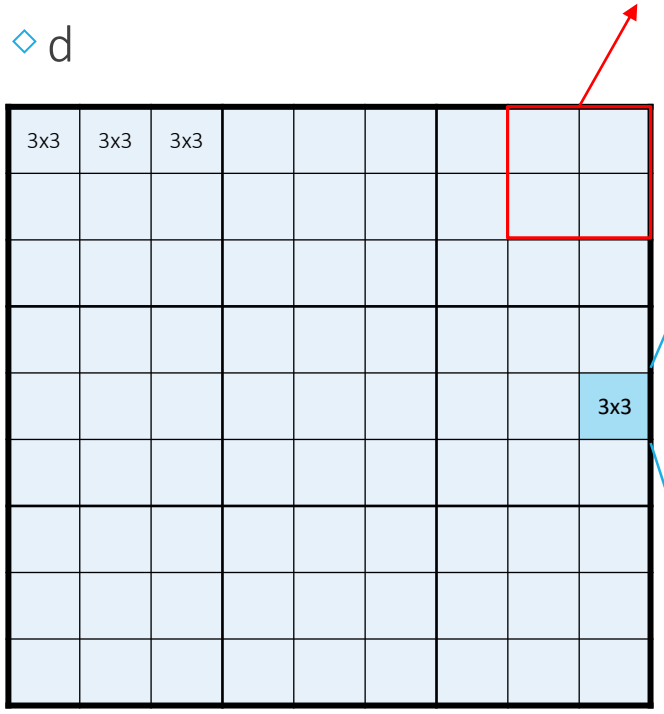
- The I/O alignment script provided by the new firmware seems to work correctly.
- The LPC connector damaged and frequent failures in I/O delay adjustment. This is an old problem, but the advent of FMC cables has made it even more serious.

```
$ python 003_10G_Test_Align.py
Test ASIC 0...
Delay: 46
                 ASIC CMD  DLY10 AR   Data1    Data0    Trig3    Trig2    Trig1    Trig0
Iteration #000:  0xa0 0x0c 2e/2e 0x1d 06666666 acccccc 2cc444c4 acccccc 06666666 acccccc
Iteration #001:  0xa0 0x0c 2e/2e 0x1d 06666666 acccccc acccccc acccccc 06466666 acccccc
Iteration #002:  0xa0 0x0c 2e/2e 0x1d 06666666 acccccc 24444444 acccccc 06666666 2ec44646
Iteration #003:  0xa0 0x0c 2e/2e 0x1d 06666666 acccccc acccc4c4 acccccc 06466666 acccccc
Iteration #004:  0xa0 0x0c 2e/2e 0x1d 06666666 acccccc acccccc acccccc 06464666 acccccc
Iteration #005:  0xa0 0x0c 2e/2e 0x1d 06666666 acccccc 2ccc4444 acccccc 06666666 accccc4
Iteration #006:  0xa0 0x0c 2e/2e 0x1d 06666666 acccccc acccccc acccccc 06444446 acccccc
Iteration #007:  0xa0 0x0c 2e/2e 0x1d 06666666 acccccc acccccc acccccc 06464666 acccccc
Iteration #008:  0xa0 0x0c 2e/2e 0x1d 06666666 acccccc acccccc acccccc 06466666 acccccc
Iteration #009:  0xa0 0x0c 2e/2e 0x1d 06666666 acccccc acccccc acccccc 04444444 acccccc
Iteration #010:  0xa0 0x0c 2e/2e 0x1d 06464666 acccccc acccccc acccccc 04444444 acccccc
Iteration #011:  0xa0 0x0c 2e/2e 0x1d 06666666 acccccc acccccc acccccc 04444444 acccccc
Iteration #012:  0xa0 0x0c 2e/2e 0x1d 06666666 acccccc 2cc44444 acccccc 06666666 accccc4
Iteration #013:  0xa0 0x0c 2e/2e 0x1d 06666666 acccccc acccccc acccccc 04444444 acccccc
Iteration #014:  0xa0 0x0c 2e/2e 0x1d 06666666 acccccc acc4c444 acccccc 06666666 accccc4
Iteration #015:  0xa0 0x0c 2e/2e 0x1d 06666666 acccccc accccc4 acccccc 06466666 acccccc
Iteration #016:  0xa0 0x0c 2e/2e 0x1d 06666666 acccccc acccccc acccccc 06466666 acccccc
Iteration #017:  0xa0 0x0c 2e/2e 0x1d 06666666 acccccc accccc4 acccccc 06466666 acccccc
Iteration #018:  0xa0 0x0c 2e/2e 0x1d 04464666 acccccc acccccc acccccc 24444444 acccccc
Iteration #019:  0xa0 0x0c 2e/2e 0x1d 06666666 acccccc 2cc44444 acccccc 06666666 acccc4c4
```

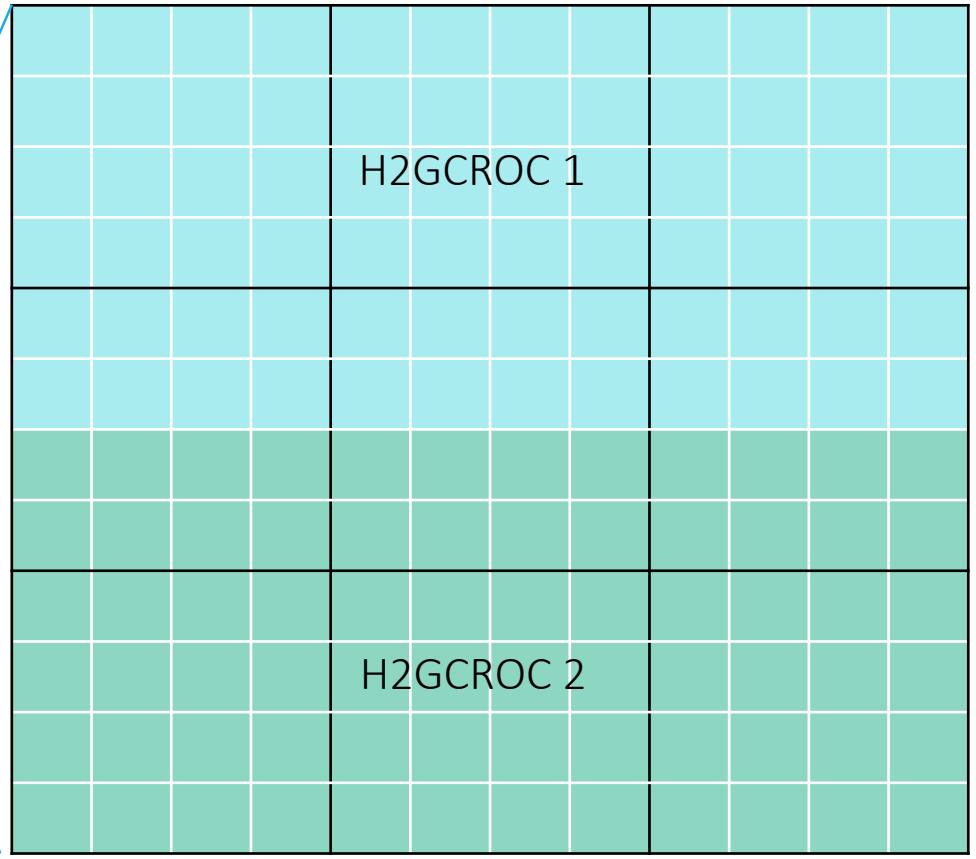
Channel

2026/8 beam test size, need 8 H2GCROC

- d



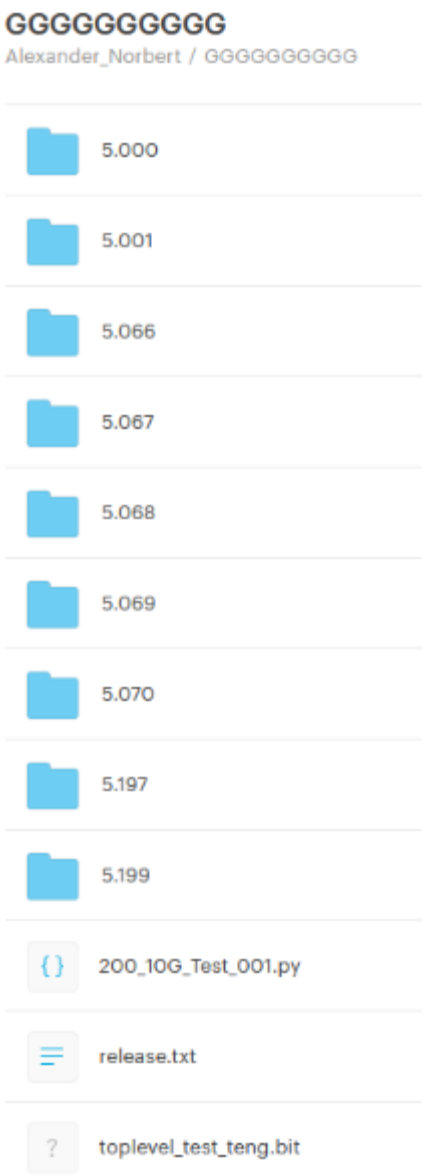
1 EMCAL = 9 Super module
1 Super module = 9x 3x3 Crystal
3x3Crystal = 2 H2GCROC
=>
1 EMCAL = 162 H2GCROC = 20.16 FPGA



3x3 x 16 = 144 SiPM
144 = 2x H2GCROC
How to connect

Newer FPGA firmware

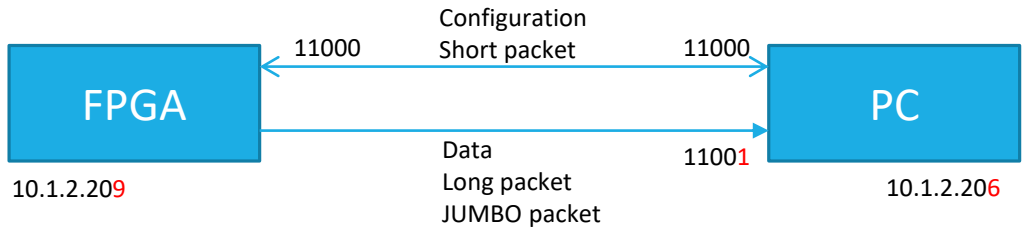
- Firmware 5.069 => main version 5, subversion 69
  - subversion 0 - 63: 10GbE + LPC
  - subversion 64 - 127: 10GbE + LPC + HPC
  - subversion 128 - 191: 1GbE + LPC
  - subversion 192 - ???: 1GbE + LPC + HPC
- Since main version 5, the communication format has changed, need newer software.
- The newer calibration software develop by Shihai Jia, it combine H2GConfig & H2GCalib
  - https://gitlab.cern.ch/sjia/h2gcalibx
- Need a new DAQ software to replace H2GDAQ, online monitoring and decode
  - ... ask them



Multiple FPGA Setting

- DIP1 Select Clock: **OFF = on board clock,** ON = External clock.
- DIP2 Data Port: **OFF = 11000,** ON = 11001
- DIP3 & DIP4: Select IP address

DIP3, DIP4	FPGA IP	Config. Port	PC IP	PC Config Port	PC Data port
OFF,OFF	**10.1.2.208**	**11000**	**10.1.2.207**	**11000**	**11000 + DIP2**
OFF,ON	10.1.2.209	11000	10.1.2.206	11000	11000 + DIP2
ON,OFF	10.1.2.210	11000	10.1.2.205	11000	11000 + DIP2
ON,ON	10.1.2.211	11000	10.1.2.204	11000	11000 + DIP2



Ethernet firewall

- Use external and internal firewall zone
- Add 10G interface enp46s0

```
$ firewall-cmd --zone=public --remove-interface=enp46s0
$ firewall-cmd --zone=internal --add-interface=enp46s0
$ firewall-cmd --zone=internal --add-port=11000-11100/udp
$ firewall-cmd --runtime-to-permanent
$ firewall-cmd --zone=internal --list-all
internal (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp46s0
  sources:
  services: cockpit dhcpv6-client mdns mountd nfs nfs3 rpc-bind samba samba-client ssh
  ports: 11000-11100/udp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

AlmaLinux 9.5

multiple IP

- Setting PC IP address for 10G device

```
$ nmcli con down "enp46s0"
$ nmcli con modify "enp46s0" ipv4.addresses "10.1.2.207/24" ethernet.mtu 9000 ipv4.method manual
$ nmcli con up "enp46s0"
```

- For testing, assign multiple IP addresses to a single interface

```
$ nmcli con down "enp46s0"
$ nmcli con modify "enp46s0" ipv4.addresses
"10.1.2.207/24,10.1.2.206/24,10.1.2.205/24,10.1.2.204/24" ethernet.mtu 9000 ipv4.method manual
$ nmcli con up "enp46s0"
```

```
$ nmcli dev status
$ nmcli con show
```

AlmaLinux 9.5

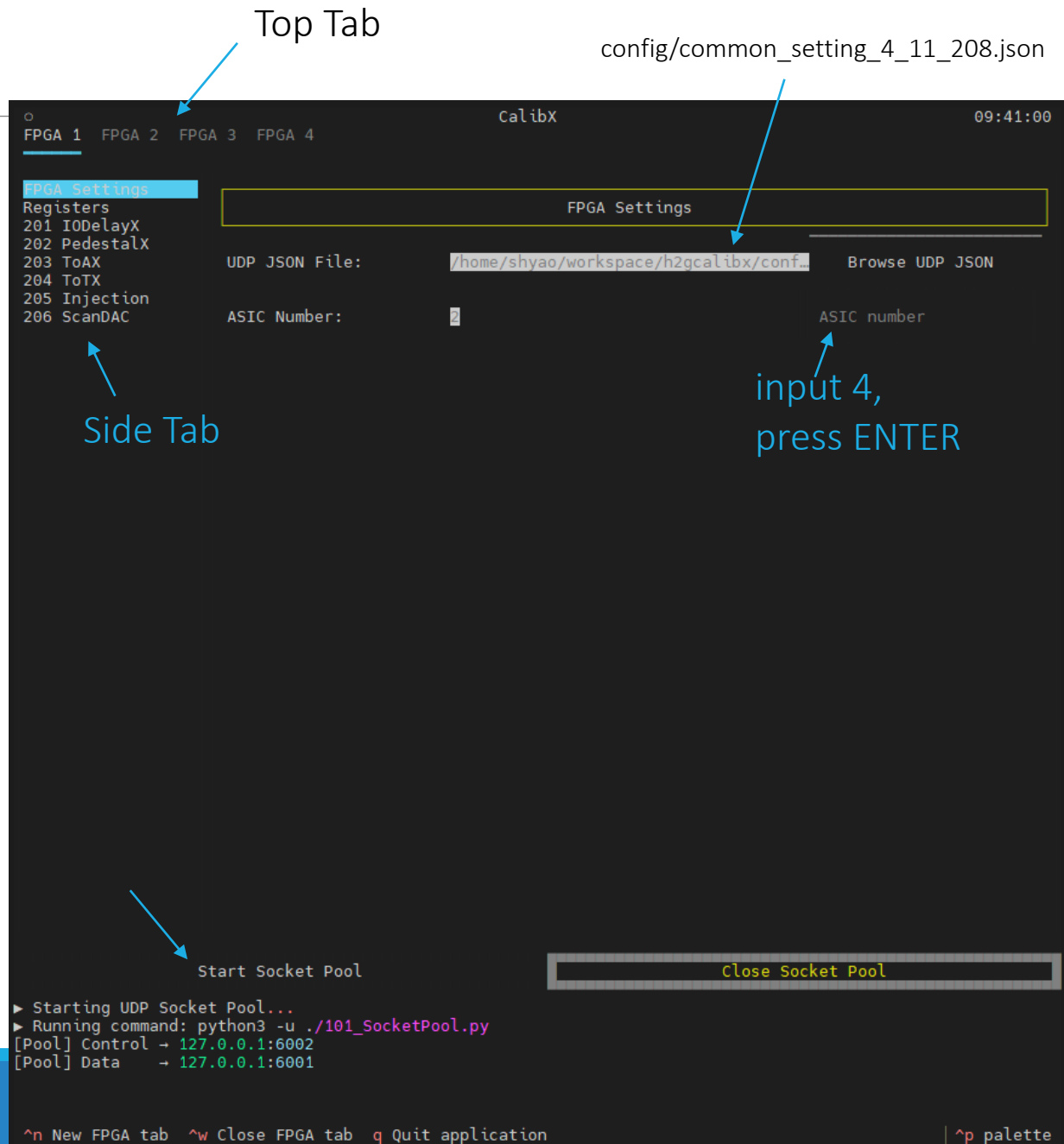
Software h2gcalibx

- git clone https://gitlab.cern.ch/sjia/h2gcalibx

58e4fd28c9f44 2026/2/23

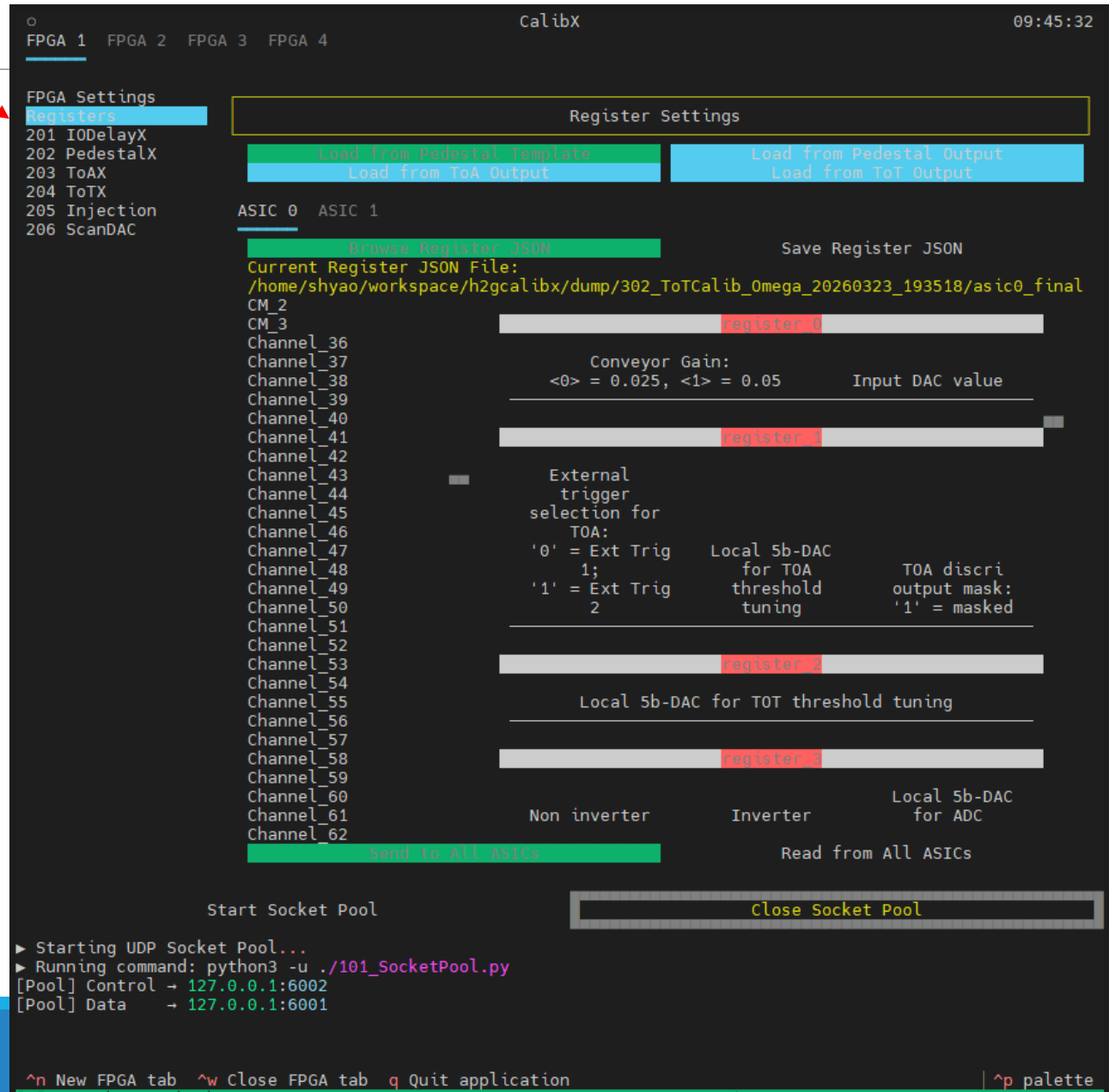
- conda create -n h2gc -c conda-forge "python=3.12.3" loguru matplotlib numpy pandas textual tqdm pytest textual-image pymupdf pillow gnuplot
- conda activate h2gc
- python 200_UI.py

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



Side Tab Registers

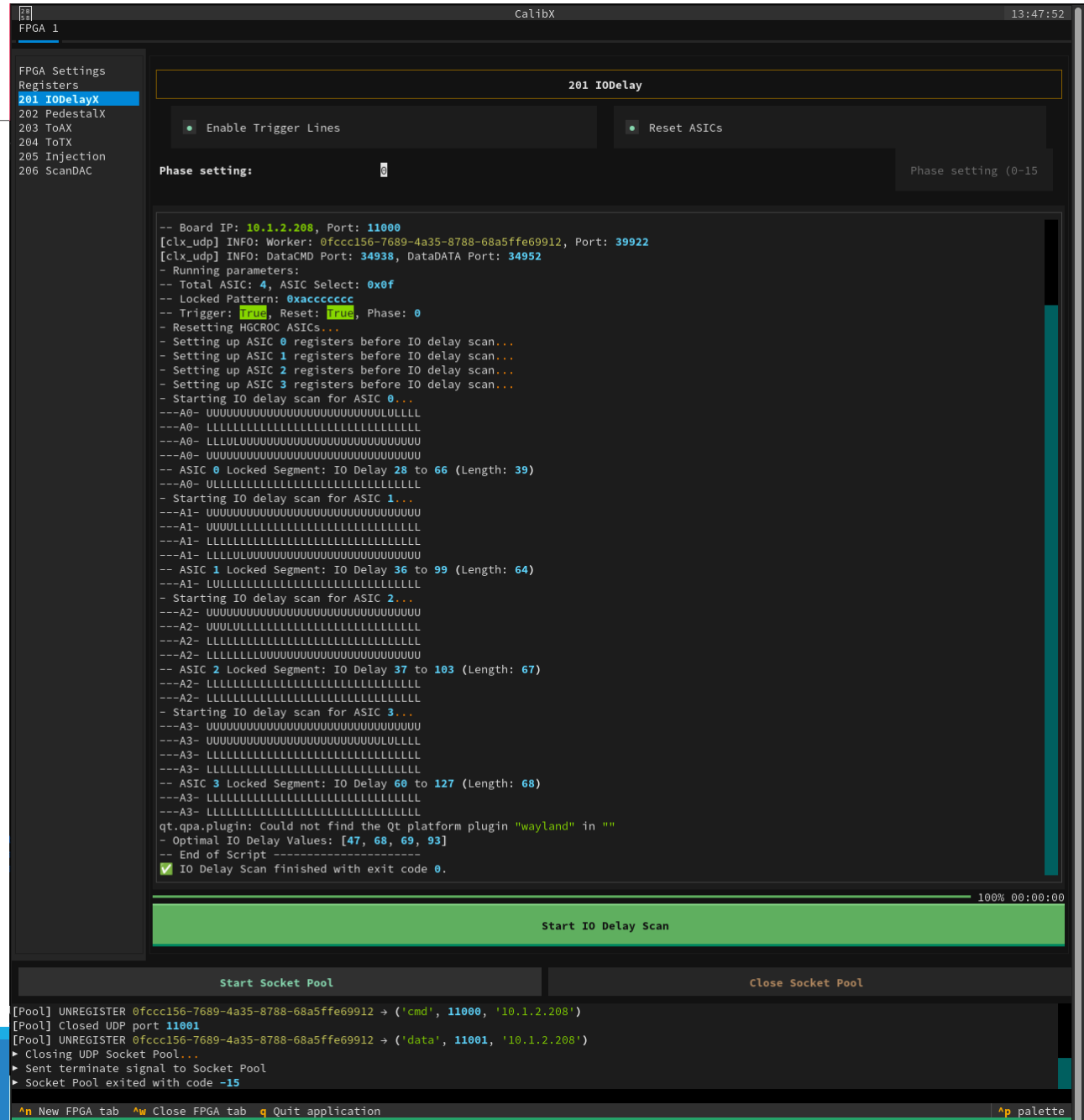
- After calibration, you can use this UI to upload parameter to ASIC



201 IODelayX

- Phase setting = 12

- FMC Cable problem (or connector problem)
  - The user interface always displays success, but the data is actually being lost silently.
  - After removing the cable, I can continue working, but there are only two ASIC chips.



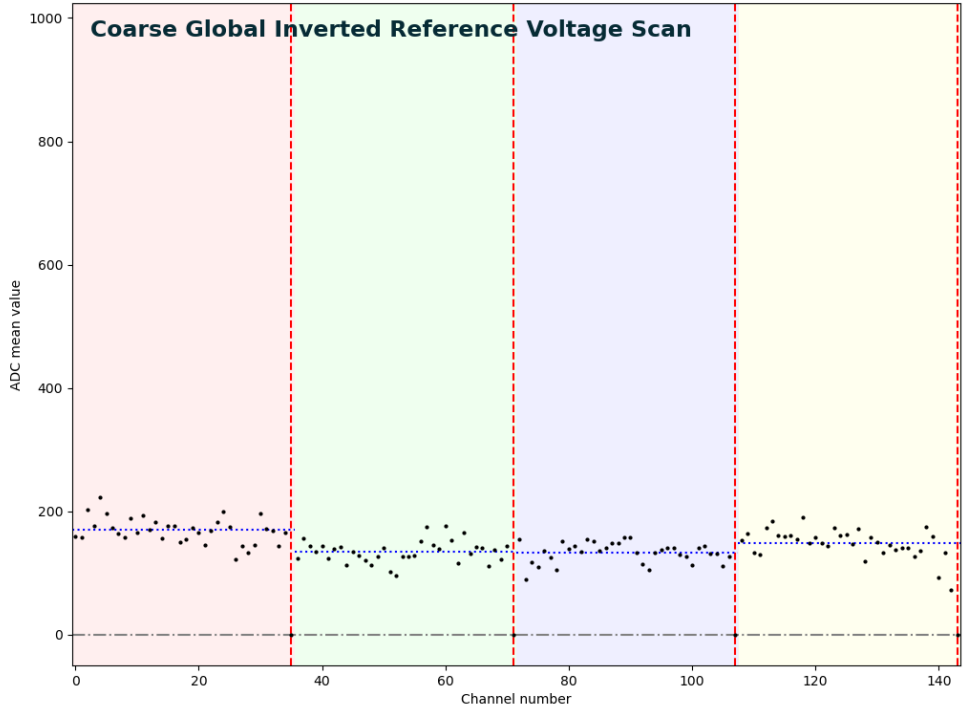
202 PedestalX

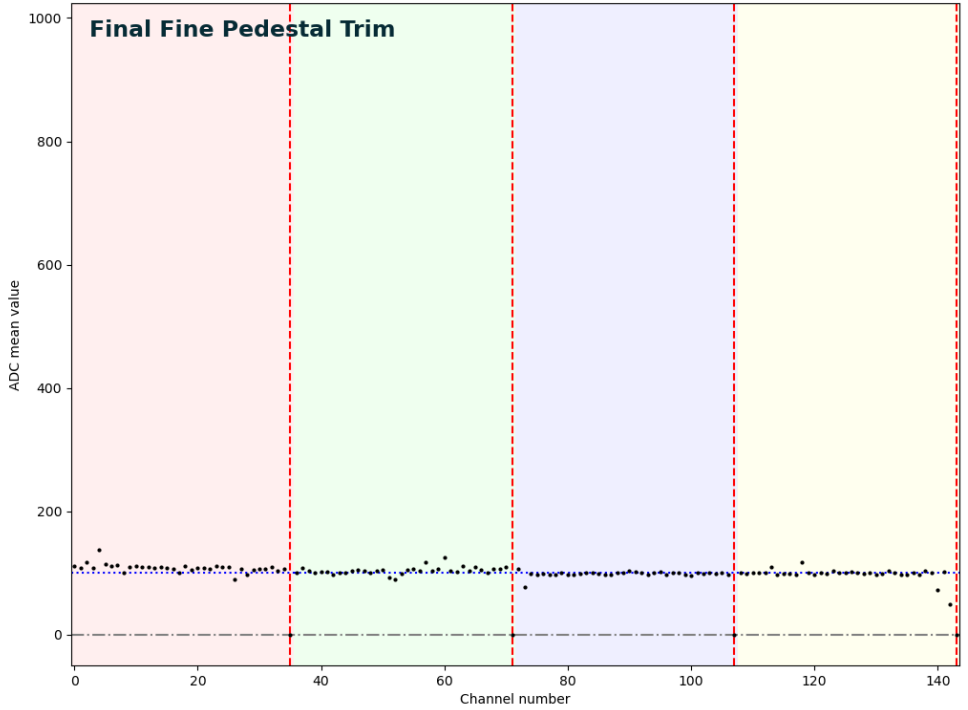
- Pre-amp reset default setting
  - Rf = 8 => 80K Ohm
  - Cf = 10 => 500 fF
  - Cf_Comp = 10 => 500 fF
  - $\tau = Rf * (Cf + Cf_{Comp})$ = 80ns = 3 samples
- Conveyor Gain default setting
  - CC = 8 => Gain = 0.2

CC	0.025, 0.05, 0.1, 0.2	
Cd (pF)	5, 10, 20	At the conveyor output and at the preamplifier input. To ensure the preamp stability
Rf (Ohm)	10K, 20K, 40K, 80K	In parallel, these resistors provide 15 values to be adjusted with the Cf and Cf_comp values
Cf (fF)	50, 100, 200, 400	Added with the Cf_comp capacitors, provide the preamplifer constant time with Rf
Cf_comp (fF)	50, 100, 200, 400	Same purpose than Cf capacitors but connected differently to improve the preamplifier stability.

202 PedestalX

- The same as before

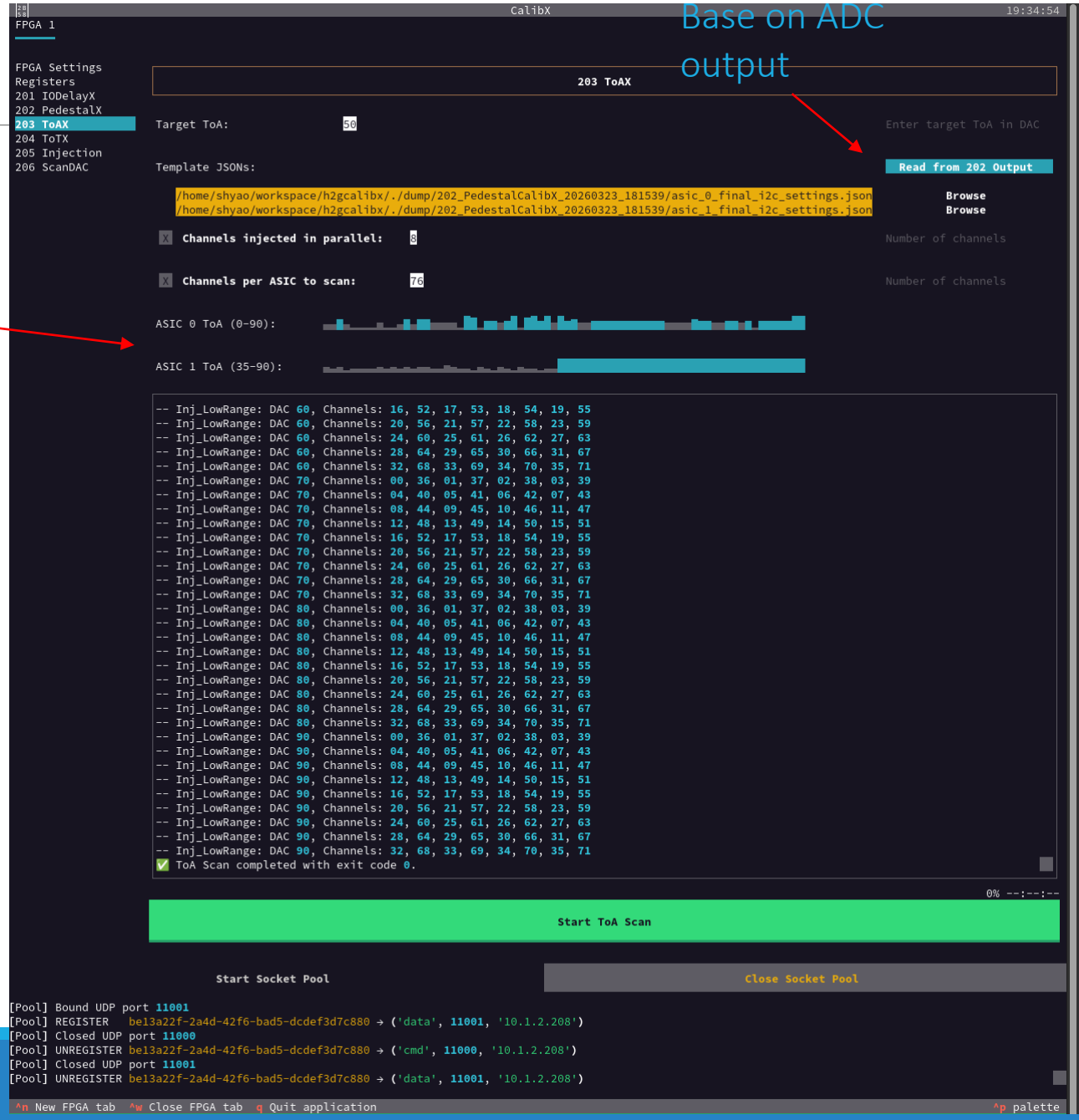




203 ToAX

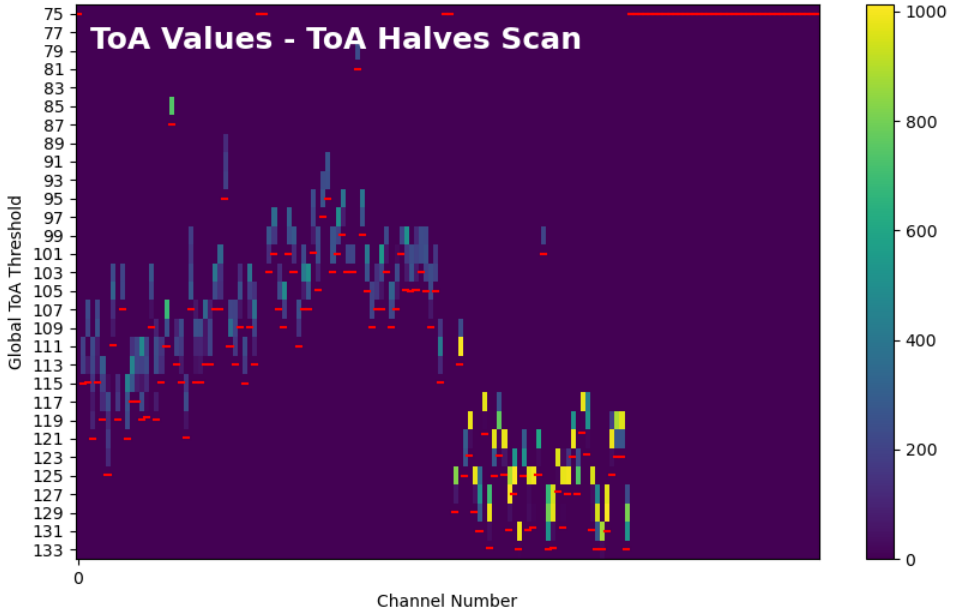
- By inject known signal, calibration ToA threshold
- Target TOA = 50

meaning?

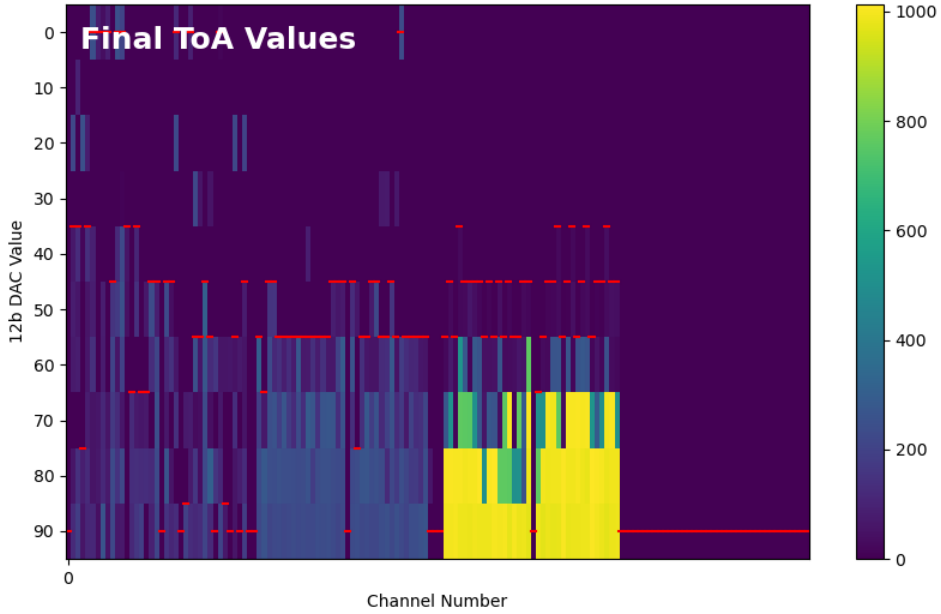


203 ToAX

- .



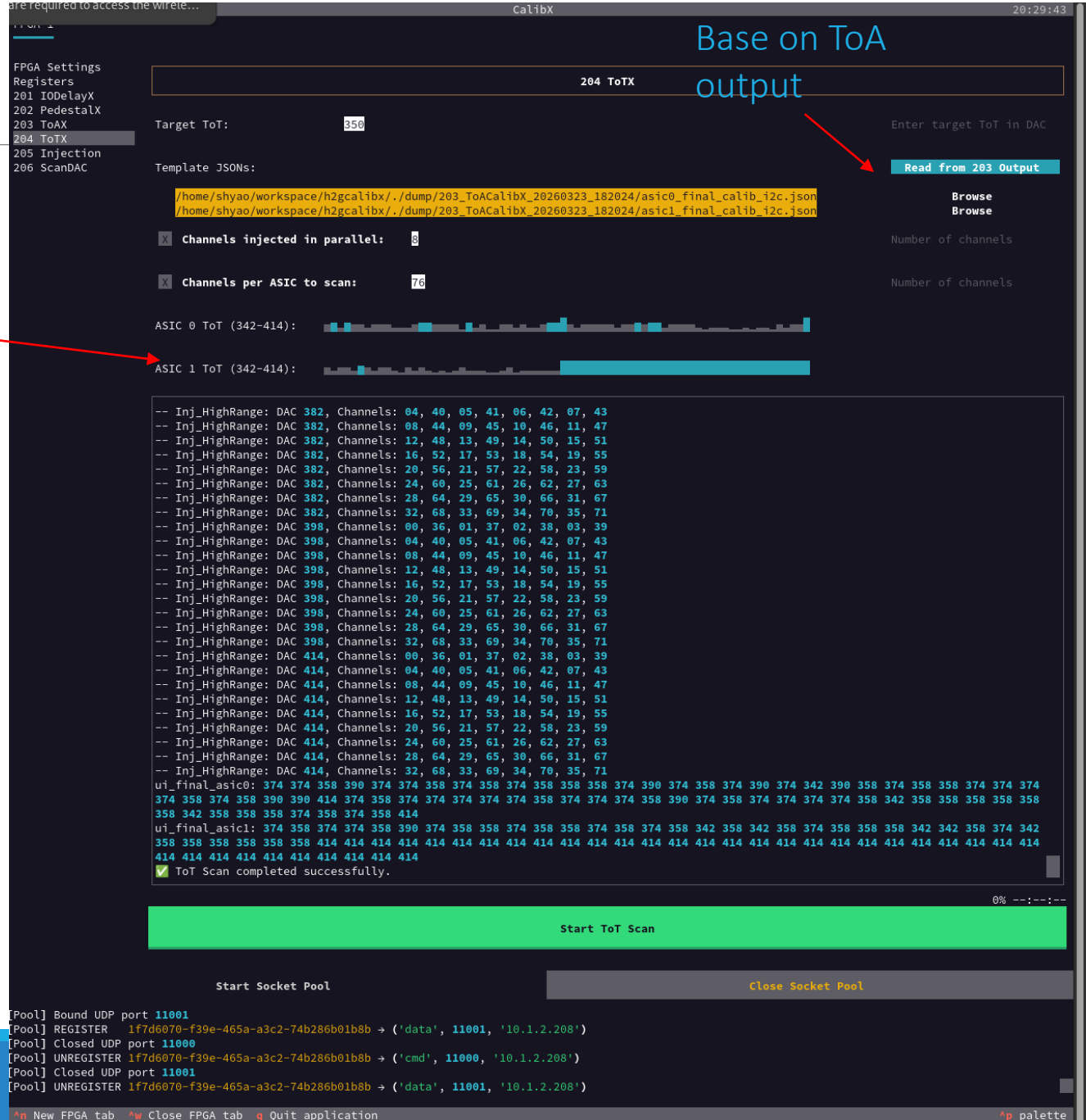
seems broken



204 TOTX (302 TOT Omega?)

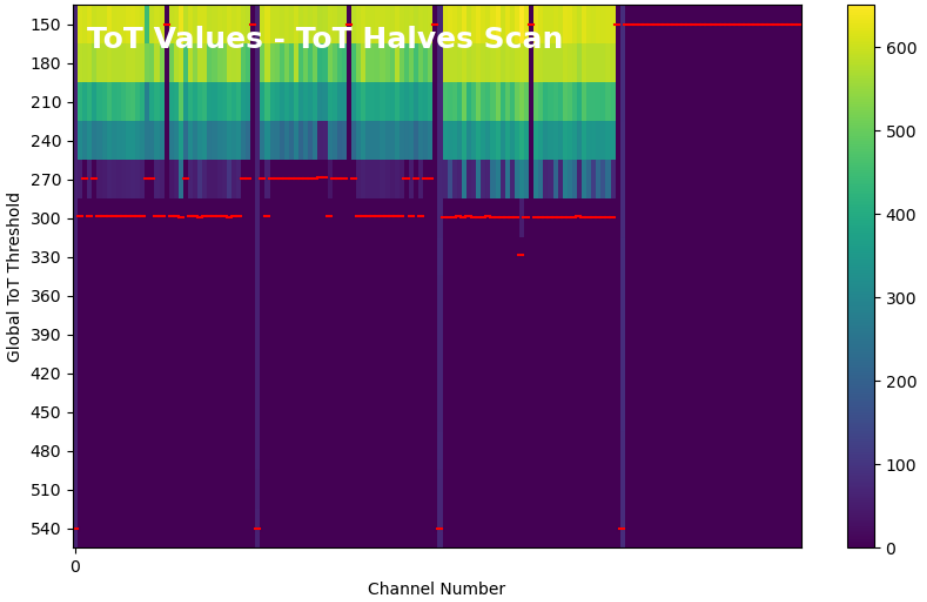
- By inject known large signal, calibration ToT threshold
- Target TOT = 350

meaning?

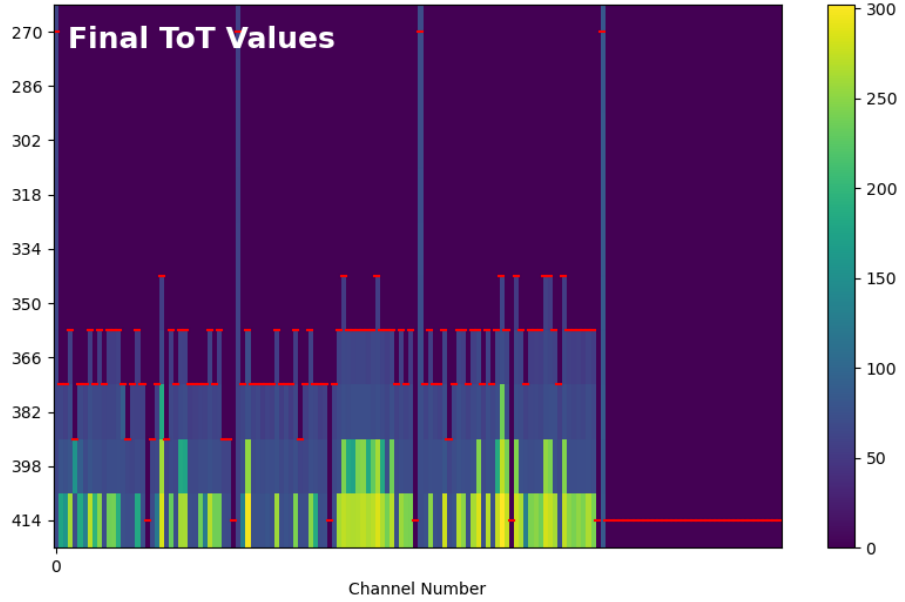


204 TOTX (302 TOT Omega)

- oops



something wrong ?



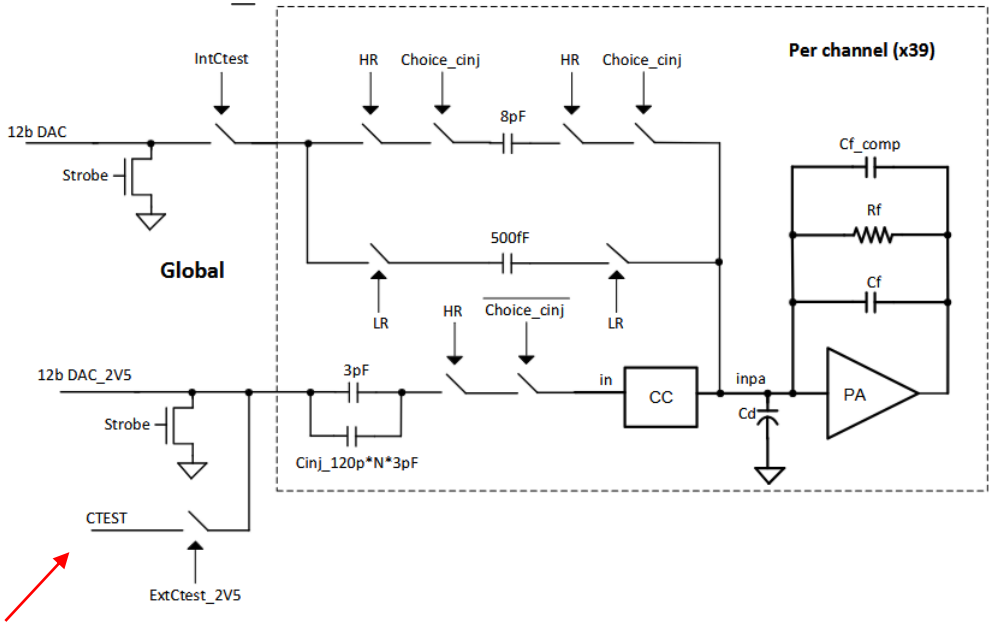
?

205 Injection

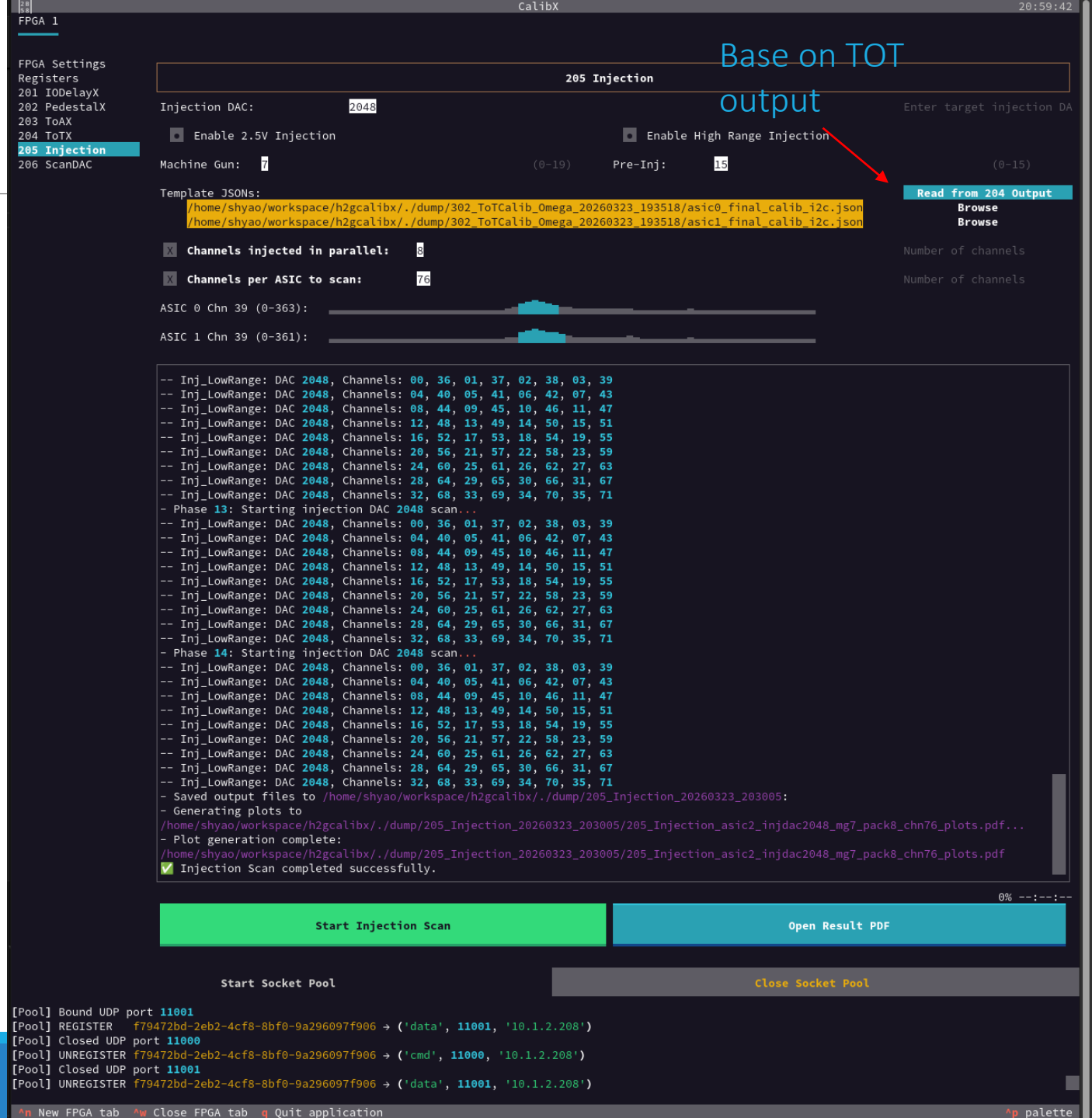
- Two 12bit DAC for calibration
  - 12b DAC: 0 - around 1 V
  - DAC_2V5: 0 - 2.5V

per ½ H2GC

- There is also a 12bit DAC output to PCB
  - DAC_ALDO: 0 - 2V

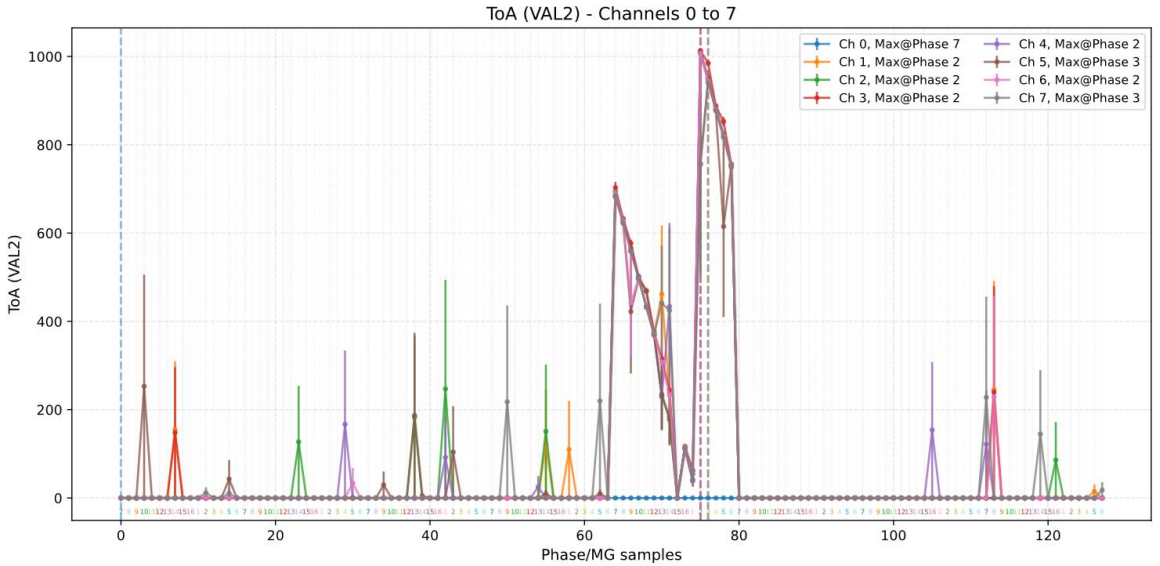


Use external signal to calibration all ASICs?

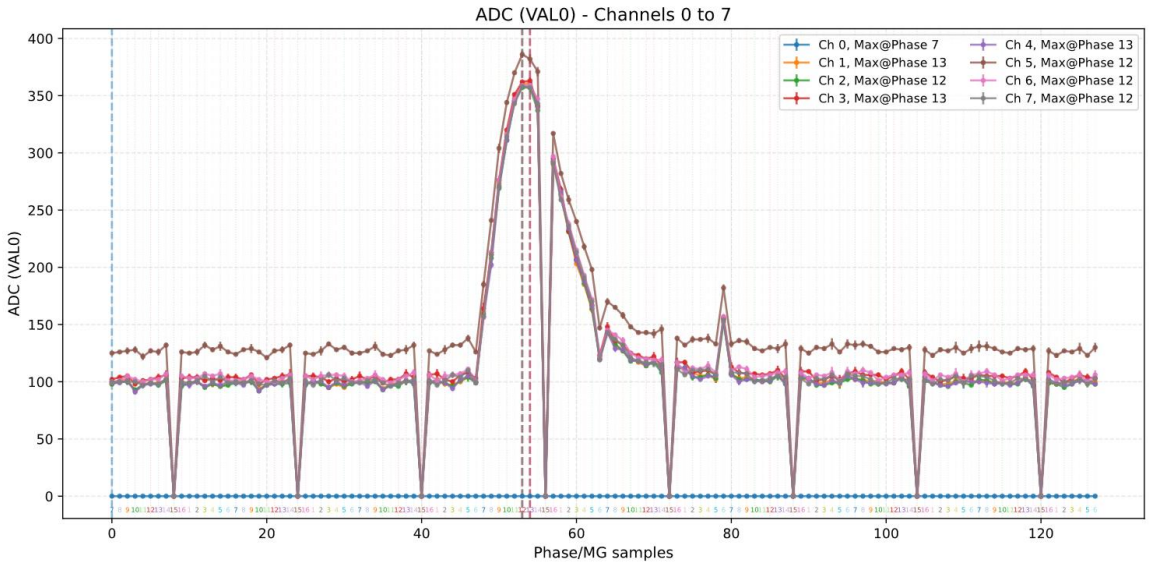


205 Injection

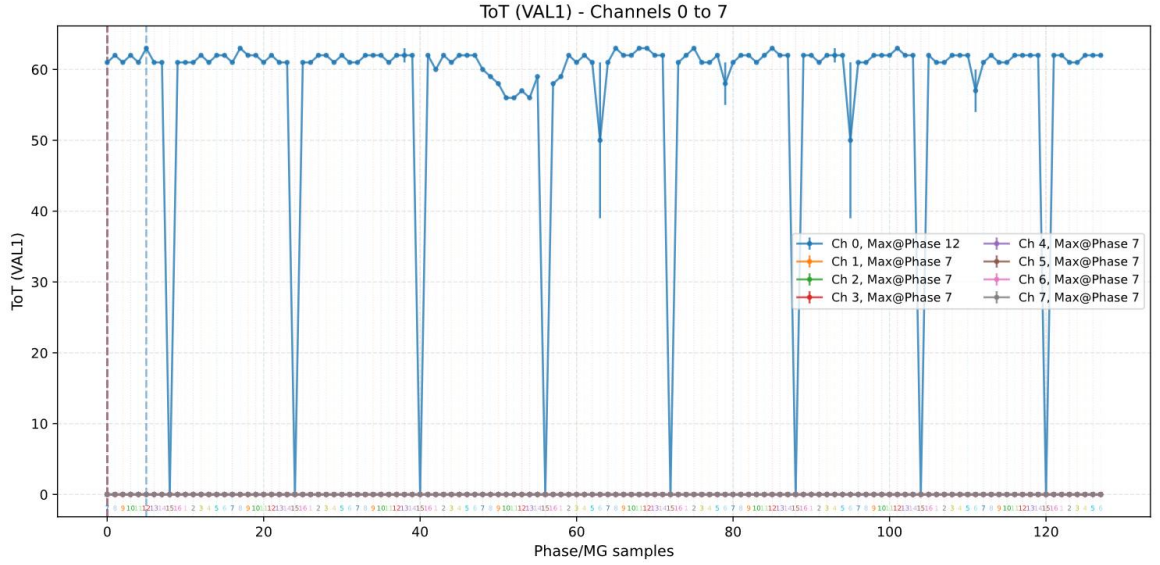
- Change phase
- Seem to broken



TOA



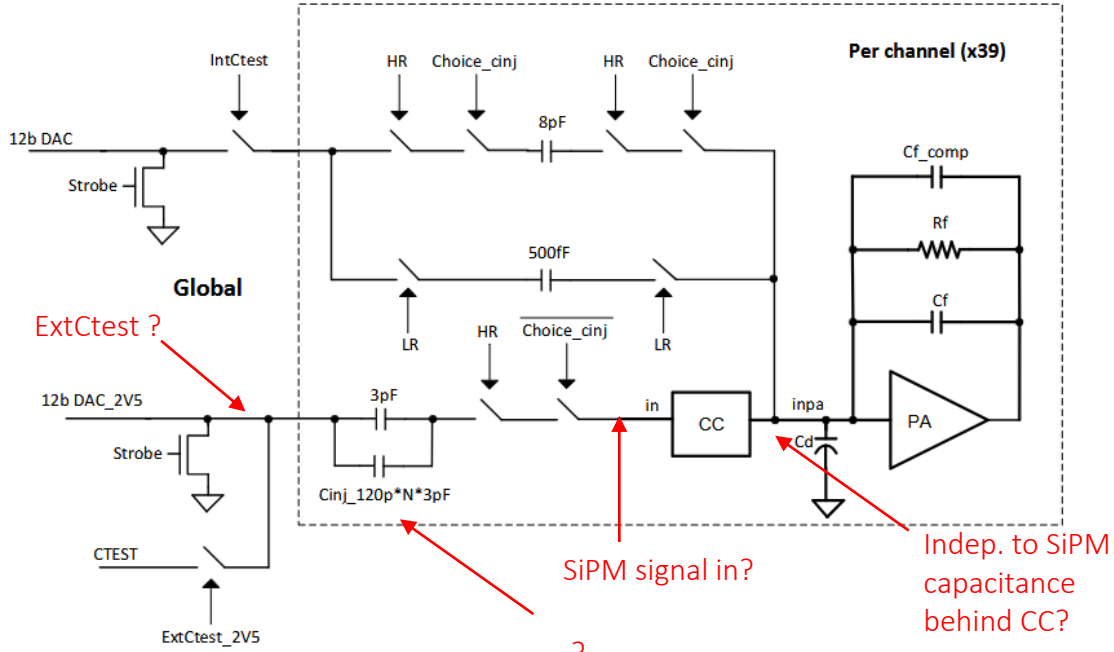
ADC



TOT

Register

- Channel wised register:
  - HR, LR, Cf, Cd, Rf, Cf_comp
- ½ H2GC register:
  - IntCtest, ExtCtest, ExtCtest_2V5, choice_cinj

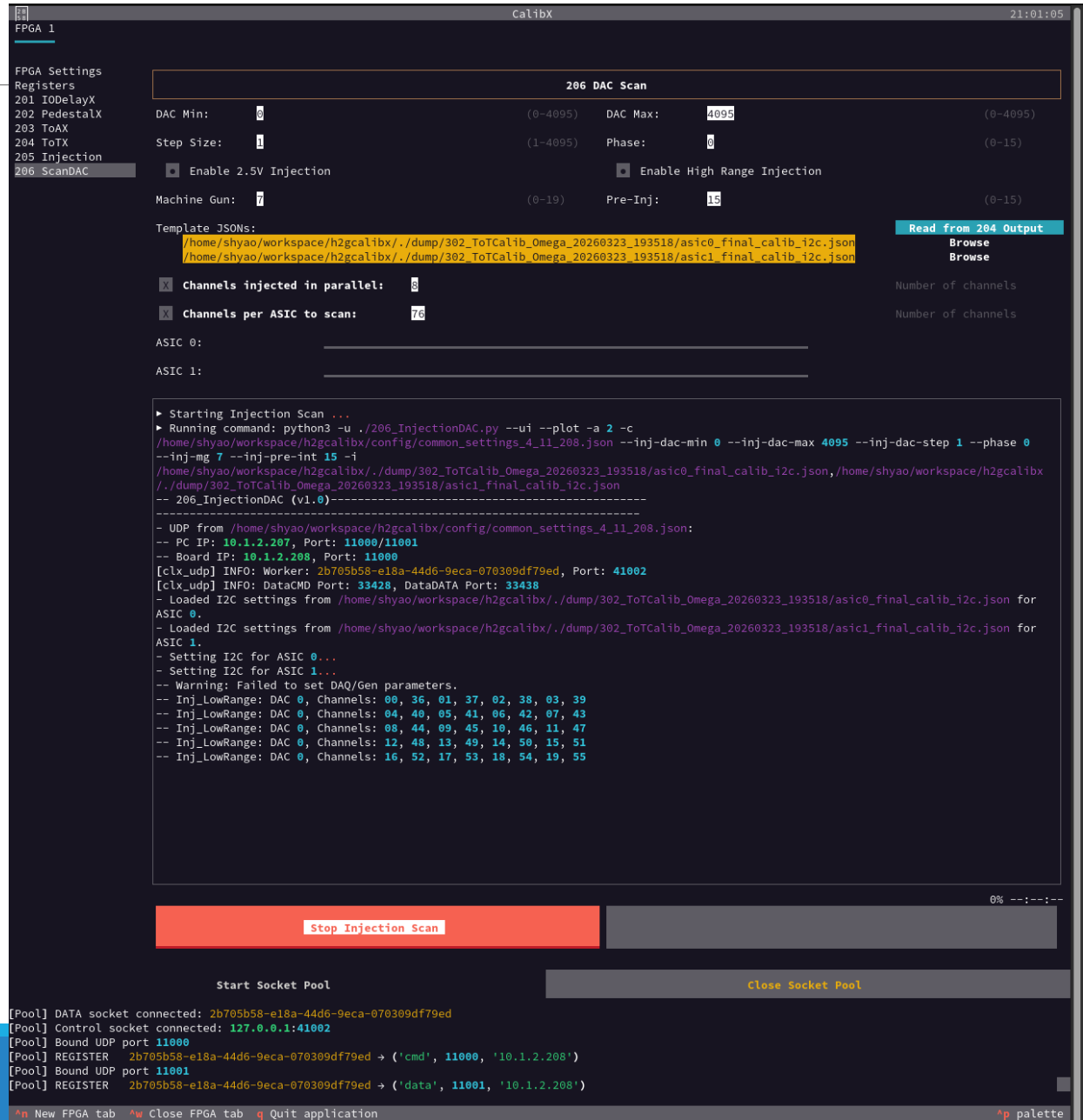


The following tables show how to select the different configurations for internal/external calibration:

	DAC LR	DAC HR	CINJ 3pF	CINJ 3pF*N	CTEST 3pF	CTEST 3pF*N
HighRange	0	1	1	1	0	0
LowRange	1	0	0	0	0	0
Choice_cinj	X	1	0	0	0	0
Cinj_120p	0	0	0	1	0	1
IntCtest	1	1	0	0	0	0
ExtCtest_2V5	0	0	0	0	1	1
Calib INPA	0 - 500fC	0 - 8pC	-	-	-	-
Calib IN	-	-	0 – 3pC	0 – 3pC*N	-	-
Calib Ext Inj	-	-	-	-	CTEST * (0 – 3pC)	CTEST * (0 – 3pC*N)
Normal ?	0	0	0	0	0	0

206 ScanDAC

- Take long time, many hours
  - DAC 0 – 4095, Step = 1, phase = 0 -15
- not complete



software update?

- Asking the update of software about H2GDAQ, online monitor and decode

UDP packet

- Firmware V4.12
  - Short packet = 40B = 0B (UDP header) + 40B (DATA)
  - Long packet = 1452B = 12B (UDP header) + 40B (DATA) * 36
  - Jumbo packet = 8972B = 12B (UDP header) + 40B (DATA) * 224

- Firmware V5.069
  - Short packet = 46B = 6B (UDP header) + 40B (DATA)
  - Long packet = 1358B = 6B (UDP header) + 8B (RFU) + 192B (DATA) * 7
  - Jumbo packet = 8846B = 6B (UDP header) + 8B (RFU) + 192B (DATA) * 46

UDP Long packet format V5 (FPGA -> PC)

- One read command generate two ½ H2GC Data per ASIC, each H2GC Data is 160 Bytes
- A UDP Long packet contain:

UDP header

UDP TX Counter	Addr, Port	RFU

PACK1

Header, type, addr	Trigger In Counter	Trigger Out Counter	Event Counter	Time Stamp		Spare	
DaqH	CM	CH0	CH1	CH2	CH3	CH4	CH5
CH6	CH7	CH8	CH9	CH10	CH11	CH12	CH13
CH14	CH15	CH16	CH17	Calib	CH18	CH19	CH20
CH21	CH22	CH23	CH24	CH25	CH26	CH27	CH28
CH29	CH30	CH31	CH32	CH33	CH34	CH35	CRC-32

½ H2GC DATA
A DAQ frame
(Next page)

....

PACK7

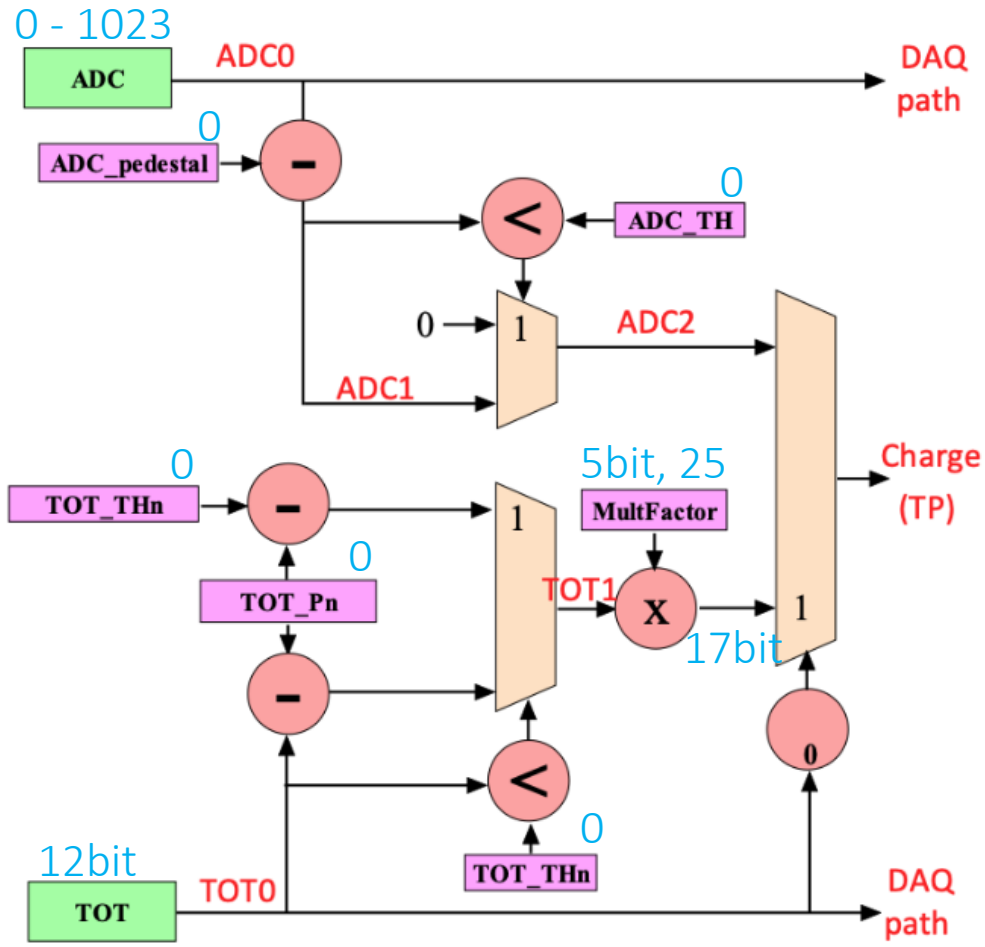
Header, type, addr	Trigger In Counter	Trigger Out Counter	Event Counter	Time Stamp		Spare	
DaqH	CM	CH0	CH1	CH2	CH3	CH4	CH5
CH6	CH7	CH8	CH9	CH10	CH11	CH12	CH13
CH14	CH15	CH16	CH17	Calib	CH18	CH19	CH20
CH21	CH22	CH23	CH24	CH25	CH26	CH27	CH28
CH29	CH30	CH31	CH32	CH33	CH34	CH35	CRC-32

½ H2GC DATA
A DAQ frame
(Next page)

....

Internal trigger V5

- No way was found to enable this feature from the user interface. Try using custom code.
- ADC
  - ADC_pedestal: One per channel, 8bit, default = 0
  - ADC_TH: One per ½ H2GC, 5bit, default = 0
    - ADC2 = (ADC0 – ADC_pedestal > ADC_TH) ? ADC0-ADC_pedestal : 0;
- TOT
  - TOT_THn: One per Group, 8bit, default = 0
  - TOT_Pn: One per Group, 8bit, default = 0
  - MultiFactor: One per ½ H2GC, 5bit, default = 25
- Two kind of Group
  - TC4: 8 Groups each 4 channels, use TR0, TR1, TR2, TR3
  - TC9: 4 Groups each 9 channels, use TR0, TR2



Internal trigger V5

- Huh

Bits	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC0												0	0	0	0	0	0	0	0	0	0
ADC Pedestal														0	0	0	0	0	0	0	0
ADC Threshold																	0	0	0	0	0

TOT0	0	0	0	0	0	0	0	0	0	0	0
TOT Pedestal					0	0	0	0	0	0	0	0
MultiFactor									1	1	0	0	1

TC4:

ADC2	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOT2	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TC9:

TOT2	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Raw Data V5

- A custom program can send start_daq and store UDP packet
- red boxed fields require byte swapping, i.e. Big-endian

```
[shyao@cats myprj]$ od -Ax -t x1 -j 0 -N14 data/run1.bin
000000 00 1c 49 af d0 f9 ee 00 00 00 00 00 00 00
```

UDP Header

```
[shyao@cats myprj]$ od -Ax -t x4 -j 14 -w32 -N1358 data/run1.bin
00000e 24015aaa 01000000 04000000 00000000 00000000 40000000 00000000 00000000
00002e 050681f1 00c40100 00004009 0000d003 00004007 0000b005 00001008 0000a005
00004e 00009006 00001008 0000e008 00005008 00000008 00005008 0000c007 00004008
00006e 00005009 0000f007 00001009 00006008 00008005 0000f008 00005007 0000b007
00008e 00001006 00002008 00004008 00009008 0000a006 00009007 00003007 00009007
0000ae 00009008 00002007 0000d007 00001008 00001007 00003007 00003006 d4c6bef7
0000ce 24005aaa 01000000 04000000 00000000 00000000 40000000 00000000 00000000
0000ee 8505a9f9 00300100 00001006 00003006 0000e008 0000d007 0000400a 0000c005
00010e 00001006 00005006 00009005 00008007 0000c003 0000e007 00006004 00009007
00012e 00009004 00004006 00004005 00006005 00003005 00003003 00008004 00002003
00014e 00001003 00005004 00002005 00001006 00005005 0000d003 0000e004 0000b005
00016e 00002006 00004007 00003006 0000a006 0000c005 00009005 00004004 3f196ed5
00018e 25015aaa 01000000 04000000 00000000 00000000 40000000 00000000 00000000
0001ae 050681f1 00740100 0000c006 00005005 00008003 0000e005 0000c005 00000007
0001ce 0000a005 00006005 0000c006 00002005 0000a006 0000f005 00005005 00004004
0001ee 0000c004 00001008 00006006 00002006 0000e006 00004004 00004007 00009005
00020e 00009007 00006006 00002004 0000d005 0000f005 0000e005 00004004 0000c003
00022e 0000e005 00001009 0000e007 0000d001 0000d006 00002003 00001006 81bd3bf4
00024e 25005aaa 01000000 04000000 00000000 00000000 40000000 00000000 00000000
00026e 8505a9f9 00700100 0000b006 0000c009 00001009 00008006 00007007 0000a007
00028e 00008006 00002007 0000d005 00009006 0000b007 00001006 00001006 0000a007
0002ae 00001008 0000d005 00009004 00009008 00007005 00005007 0000e007 0000c007
0002ce 0000000a 0000f006 00008006 00000009 0000f006 00006005 0000a009 00004007
0002ee 00006007 00008008 00001006 0000f007 00004007 0000f008 00005005 0984db82
00030e 24005aaa 01000000 04000000 01000000 00000000 e4000000 00000000 00000000
00032e 8509aaf9 002c0100 00001006 00001006 00000009 0000a007 0000500a 0000c005
00034e 00001006 00006006 0000a005 00007007 0000c003 0000d007 00008004 0000a007
00036e 0000d004 00004006 00006005 00004005 00005005 00003003 00007004 00002003
00038e 00002003 00005004 00001005 00002006 00005005 0000d003 0000f004 0000a005
0003ae 00002006 00005007 00004006 0000c006 0000d005 0000a005 00005004 15627d3d
0003ce 24015aaa 01000000 04000000 01000000 00000000 e4000000 00000000 00000000
0003ee 050a82f1 00c40100 00004009 00000004 00002007 00009005 00002008 0000c005
00040e 0000a006 00004008 0000e008 00005008 0000f007 00005008 0000d007 00002008
00042e 00005009 00000008 00002009 00008008 00008005 0000e008 00004007 0000c007
00044e 00000006 00001008 00005008 00009008 0000a006 0000a007 00001007 00009007
00046e 00009008 00001007 0000c007 00002008 00002007 00002007 00005006 9f4976e0
00048e 25005aaa 01000000 04000000 01000000 00000000 e4000000 00000000 00000000
0004ae 8509aaf9 006c0100 00009006 0000b009 00000009 00009006 0000c007 00009007
0004ce 00008006 00002007 0000d005 00009006 0000e007 00001006 00001006 0000c007
0004ee 00005008 0000d005 00009004 00009008 00009005 00003007 0000f007 0000c007
00050e 0000e009 00000007 00008006 00001009 00002007 00006005 00008009 00002007
```

Decode V5

```
[shyao@cats myprj]$ ./build/debug/ana/ana decode data/run1.bin
Open rawdata file data/run1.bin
udp_tx_counter: 1853871 fpga_ip_address: 208 fpga_send_port: 249 RFU: ee 00 00 00 00 00 00 00
Address: 01 header: aa5a packet_type: 24 Trigger_IN_Counter: 1 Trigger_OUT_Counter: 4 Event_Counter: 0 timestamp: 64
hd: 15  H1: 0  H2: 0  H3: 0  tr: 5  BxCounter: 385  Event: 1  Orbit: 4  BX_NEW  timestamp diff: 64
f1810605 0001c400 09400000 03d00000 07400000 05b00000 08100000 05a00000
06900000 08100000 08e00000 08500000 08000000 08500000 07c00000 08400000
09500000 07f00000 09100000 08600000 05800000 08f00000 07500000 07b00000
06100000 08200000 08400000 08900000 06a00000 07900000 07300000 07900000
08900000 07200000 07d00000 08100000 07100000 07300000 06300000 f7bec6d4
Address: 00 header: aa5a packet_type: 24 Trigger_IN_Counter: 1 Trigger_OUT_Counter: 4 Event_Counter: 0 timestamp: 64
hd: 15  H1: 0  H2: 0  H3: 0  tr: 5  BxCounter: 2473  Event: 1  Orbit: 3  BX_NEW  timestamp diff: 64
f9a90585 00013000 06100000 06300000 08e00000 07d00000 0a400000 05c00000
06100000 06500000 05900000 07800000 03c00000 07e00000 04600000 07900000
04900000 06400000 05400000 05600000 05300000 03300000 04800000 03200000
03100000 04500000 05200000 06100000 05500000 03d00000 04e00000 05b00000
06200000 07400000 06300000 06a00000 05c00000 05900000 04400000 d56e193f
Address: 01 header: aa5a packet_type: 25 Trigger_IN_Counter: 1 Trigger_OUT_Counter: 4 Event_Counter: 0 timestamp: 64
hd: 15  H1: 0  H2: 0  H3: 0  tr: 5  BxCounter: 385  Event: 1  Orbit: 4  BX_NEW
f1810605 00017400 06c00000 05500000 03800000 05e00000 05c00000 07000000
05a00000 05600000 06c00000 05200000 06a00000 05f00000 05500000 04400000
04c00000 08100000 06600000 06200000 06e00000 04400000 07400000 05900000
07900000 06600000 04200000 05d00000 05f00000 05e00000 04400000 03c00000
05e00000 09100000 07e00000 01d00000 06d00000 03200000 06100000 f43bbd81
Address: 00 header: aa5a packet_type: 25 Trigger_IN_Counter: 1 Trigger_OUT_Counter: 4 Event_Counter: 0 timestamp: 64
hd: 15  H1: 0  H2: 0  H3: 0  tr: 5  BxCounter: 2473  Event: 1  Orbit: 3  BX_NEW
f9a90585 00017000 06b00000 09c00000 09100000 06800000 07700000 07a00000
06800000 07200000 05d00000 06900000 07b00000 06100000 06100000 07a00000
08100000 05d00000 04900000 08900000 05700000 07500000 07e00000 07c00000
0a000000 06f00000 06800000 09000000 06f00000 05600000 09a00000 07400000
07600000 08800000 06100000 07f00000 07400000 08f00000 05500000 82db8409
```

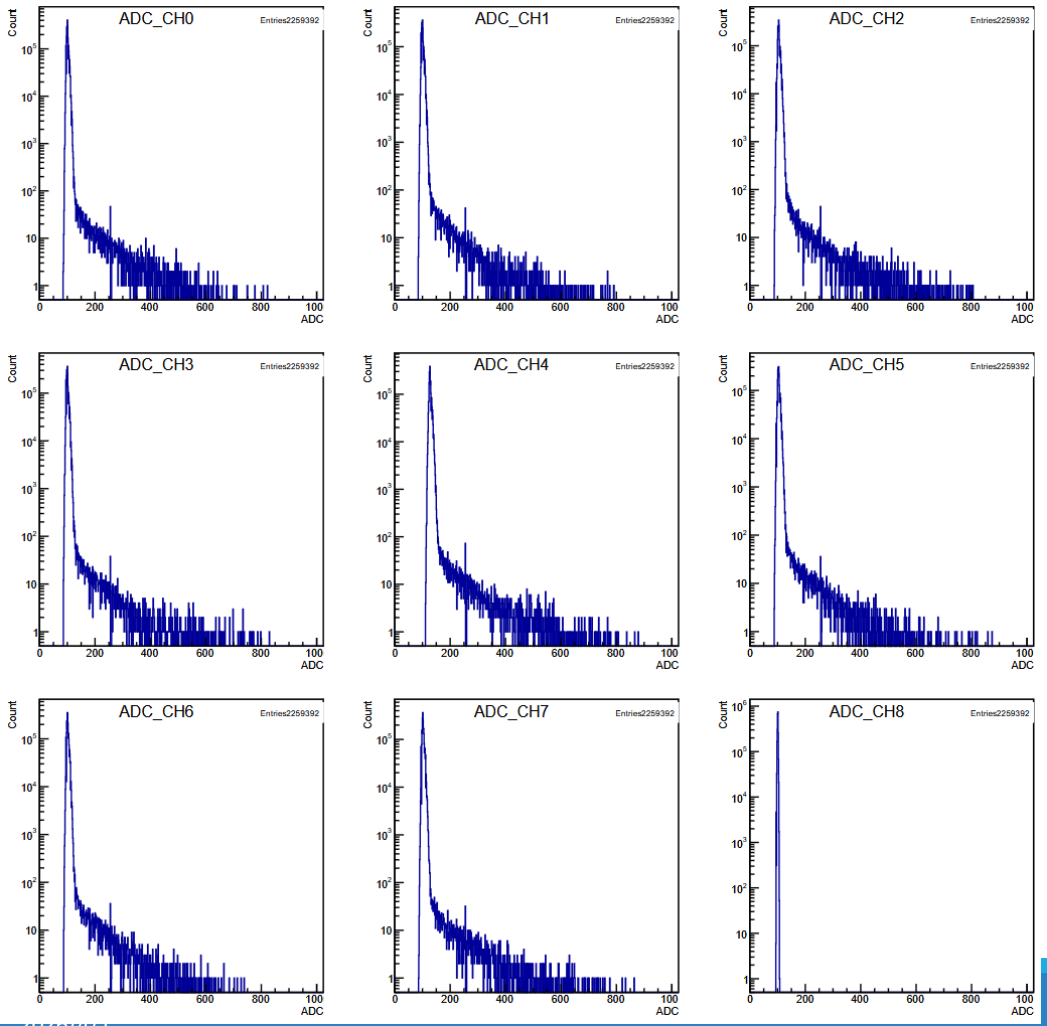
DAQ

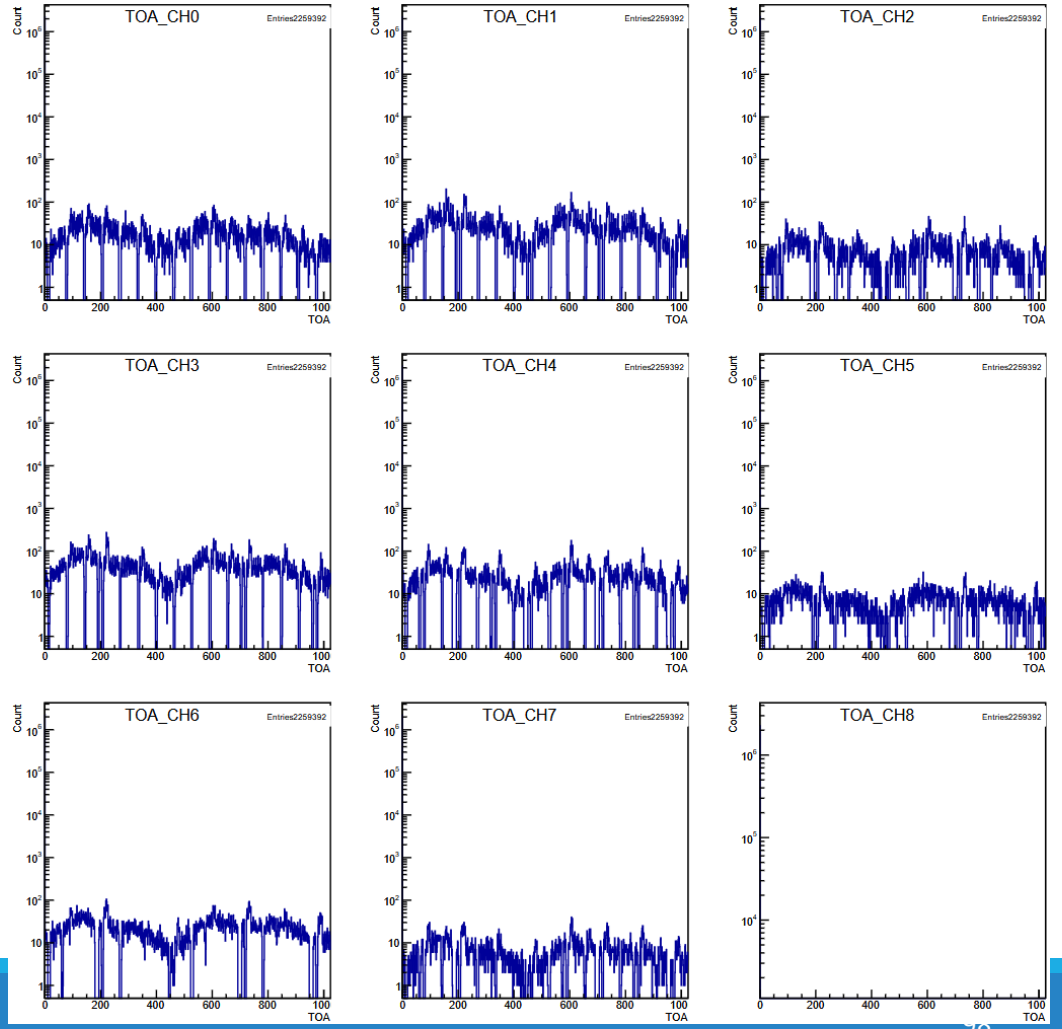
- Custom...
- Generator = 100000, period = 40000, Machine Gun = 31, Voltage = 28V

```
******************************************************************************
*Tree    :events    : A Tree with Events                                      *
*Entries :    90432 : Total =      2501195088 bytes  File  Size =  273825839 *
*        :          : Tree compression factor =   9.13                       *
******************************************************************************
*Br    0 :toa       : toa[144][32]/s                                          *
*Entries :    90432 : Total  Size=  833731575 bytes  File Size  =   14359615 *
*Baskets :     3227 : Basket Size=    3200512 bytes  Compression=  58.05     *
*.............................................................................*
*Br    1 :tot       : tot[144][32]/s                                          *
*Entries :    90432 : Total  Size=  833731575 bytes  File Size  =    3972140 *
*Baskets :     3227 : Basket Size=    3200512 bytes  Compression= 209.86     *
*.............................................................................*
*Br    2 :adc       : adc[144][32]/s                                          *
*Entries :    90432 : Total  Size=  833731575 bytes  File Size  =  255423704 *
*Baskets :     3227 : Basket Size=    3200512 bytes  Compression=   3.26     *
*.............................................................................*
total 90432 entries
```

Analysis

- After convert to root file, analysis is the same





To be continue…