

# ProtoDUNE FW-Hit-Finding Data Captures

---

3.9.20

Joel Greer

# Data Catalogue

Folder	Date	SSR	Pedestal Words	HF links active per super-logic region (max 5)	Board-reader version
N/A	4/6/20	NO	NO	1	fdaq?
APA6-wHits	24/6/20	YES	NO	1	fdaq/standalone?
APA6-newhits	6/7/20	YES	NO	1	standalone
APA6-newhits2	6/7/20	YES	NO	1	standalone
APA6-10072020 - unknown problems due to readout software settings	10/7/20	YES	NO	?	standalone
APA6-shortruns (3 captures) - unknown problems due to readout software settings	10/7/20	YES	NO	?	standalone
APA6-newFW-Nx (5 captures)	10/7/20	YES	NO	1,2,3,4,5	standalone
APA6-071620 (4 different channel masks)	16/7/20	YES	YES	1	standalone
APA6-071720 (8 different hf-link/threshold combinations)	17/7/20	YES	YES	1 or 5	standalone

# Data Catalogue

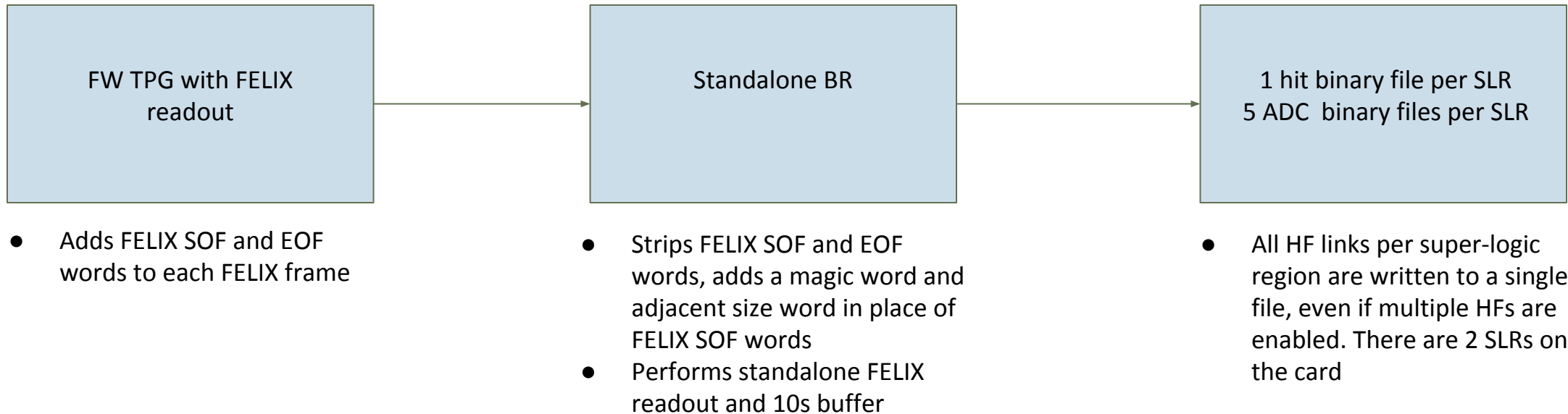
Kept track of in dataflow-firmware issue 82:

<https://gitlab.cern.ch/DUNE-SP-TDR-DAQ/dataflow-firmware/-/issues/82>

Glossary:

- SSR - firmware has state save/restore implemented
- Pedestal words - containing extra words for accumulator and median
- HF links active per super-logic-region (SLR) - max of 5 per SLR. All hit links in 1 SLR are written to a single binary file. 5 adc links are written to 5 binary files per SLR. 1 HF link active means only have to look at 1 adc link.
- Board-reader version - all useful captures use standalone BR

# Data Format



ADC binary files contain an array of raw WIB frames (described in DUNE docdb 1701).

TPG firmware versions which include pedestal words have:

- An additional 'pedestal frame' at the end of each hit packet
- Multiple hit packets per magic/size word

# Data Format - no pedestal words

Word																	Data Frame
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	Boardreader Word 0: Magic Word																BR words
1	Boardreader Word 1: Size (FELIX Frame size in bytes)																
2	crate No.				fibre No.				wireIndex No.								HeaderFrame
3	slot No.		Header Frame : Word 1 : Reserved for flags														
4	Header Frame : Word 2 : Time Stamp																
5	Header Frame : Word 3 : Time Stamp																
6	Header Frame : Word 4 : Time Stamp																
7	Header Frame : Word 5 : Time Stamp																
8	Hit Frame : Word 0 : Hit Start Time ( relative to 64 sample packet )																HitFrame
9	Hit Frame : Word 1 : Hit End time ( relative to 64 sample packet )																
10	Hit Frame : Word 2 : Hit Peak or Amplitude																
11	Hit Frame : Word 3 : Hit Peak time ( relative to 64 sample packet )																
12	Hit Frame : Word 4 : Hit Sum																
13	Hit Frame : Word 5 : Hit Continue and other reserved flags																
14	Boardreader Word 0: Magic Word																BR words
15	Boardreader Word 1: Size (FELIX Frame size in bytes)																
16	crate No.				fibre No.				wireIndex No.								HeaderFrame
17	slot No.		Header Frame : Word 1 : Reserved for flags														
18	Header Frame : Word 2 : Time Stamp																
19	Header Frame : Word 3 : Time Stamp																
20	Header Frame : Word 4 : Time Stamp																
21	Header Frame : Word 5 : Time Stamp																

1 hit packet contains:

- 1 header frame
- N hit frames

There is 1 hit packet per set of BR words

# Data Format - pedestal words

Word	Bit																Data Frame
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	Boardreader Word 0: Magic Word (0xABCD)																BR words
1	Boardreader Word 1: Size (FELIX Frame size in bytes)																
2	crate No.				fibre No.				wireIndex No.								HeaderFrame
3	slot No.		Header Frame : Word 1 : Reserved for flags														
4	Header Frame : Word 2 : Time Stamp																
5	Header Frame : Word 3 : Time Stamp																
6	Header Frame : Word 4 : Time Stamp																
7	Header Frame : Word 5 : Time Stamp																
8	Hit Frame : Word 0 : Hit Start Time ( relative to 64 sample packet )																
9	Hit Frame : Word 1 : Hit End time ( relative to 64 sample packet )																
10	Hit Frame : Word 2 : Hit Peak or Amplitude																
11	Hit Frame : Word 3 : Hit Peak time ( relative to 64 sample packet )																
12	Hit Frame : Word 4 : Hit Sum																
13	Hit Frame : Word 5 : Hit Continue and other reserved flags																
14	0xF		Pedestal Frame : Word 0 : median														Pedestal Frame
15	0xB		Pedestal Frame : Word 1 : accumulator (twos complement)														
16	Pedestal Frame : Word 2 : Magic Word (0xFEED)																
17	Pedestal Frame : Word 3 : Magic Word (0xF00D)																
18	Pedestal Frame : Word 4 : Magic Word (0xDEAD)																
19	Pedestal Frame : Word 5 : Magic Word (0xBEEF)																
20	Boardreader Word 0: Magic Word (0xABCD)																BR words
21	Boardreader Word 1: Size (FELIX Frame size in bytes)																

1 hit packet contains:

- 1 header frame
- N hit frames
- 1 pedestal frame

There are multiple hit packets per set of BR words

# Investigation of the Captures

I have developed scripts to interpret and convert the binary captures. Still need to add functionality to deal with pedestal captures to some.

- To histogram hits in a capture I use a crawler to parse and interpret the binary files: `rawCrawlerHitBinaryCheck.py`
- To extract adc data need to use a C++ tool: `extract_adcs_from_timestamp.cpp`
- To convert hit binary captures to 33b: `hitBinaryMangler.py`
- To convert 33b to hit packets: `hitFormatter.py`
- To run TPG on extracted adc data and compare to extracted hits: `hitComparison.py`
- To convert hit binaries to input format for event display: `evtdisplayhits.py`

Python tools are in dataflow-software:

[https://gitlab.cern.ch/DUNE-SP-TDR-DAQ/dataflow-software/-/tree/jgreer/tpg\\_pedchanfix](https://gitlab.cern.ch/DUNE-SP-TDR-DAQ/dataflow-software/-/tree/jgreer/tpg_pedchanfix)

C++ tools are here: <https://github.com/JatGreer/felix-long-readout-tools>

Today will mainly be looking at the histogrammed, interpreted hits from `rawCrawlerHitBinaryCheck.py`.

# Investigation of the Captures

On the following slides plots on the left are from APA6-newhits2 and plots on the right are from APA6-071720. Looking only at hits from FELIX link 5.

In short, this script:

- Crawls through binary and interprets the BR words
- Uses the size word to read in that amount of data as a (pseudo-FELIX?) frame whose contents depend on the firmware used for the capture
- Decodes the entire frame using some of the same code as is used to convert from binary to 33b format
- Pedestal captures have multiple hit packets per set of BR words, so use pedestal frame magic words to identify where hit packets end

## APA6-newhits2

- no pedestal words
- 1 HF link
- HF threshold=20

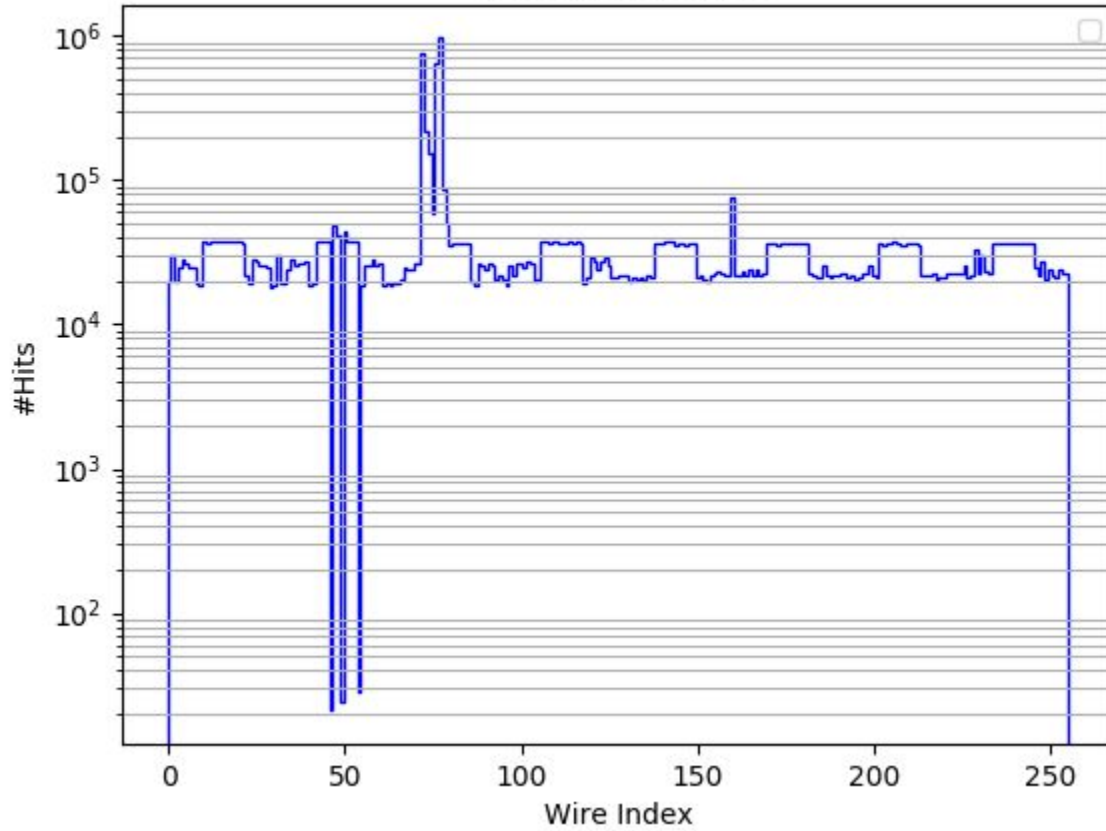
## APA6-071720

- capture 8 from this dataset
- Includes pedestal words
- 5 HF links (1 SLR)
- HF threshold=70

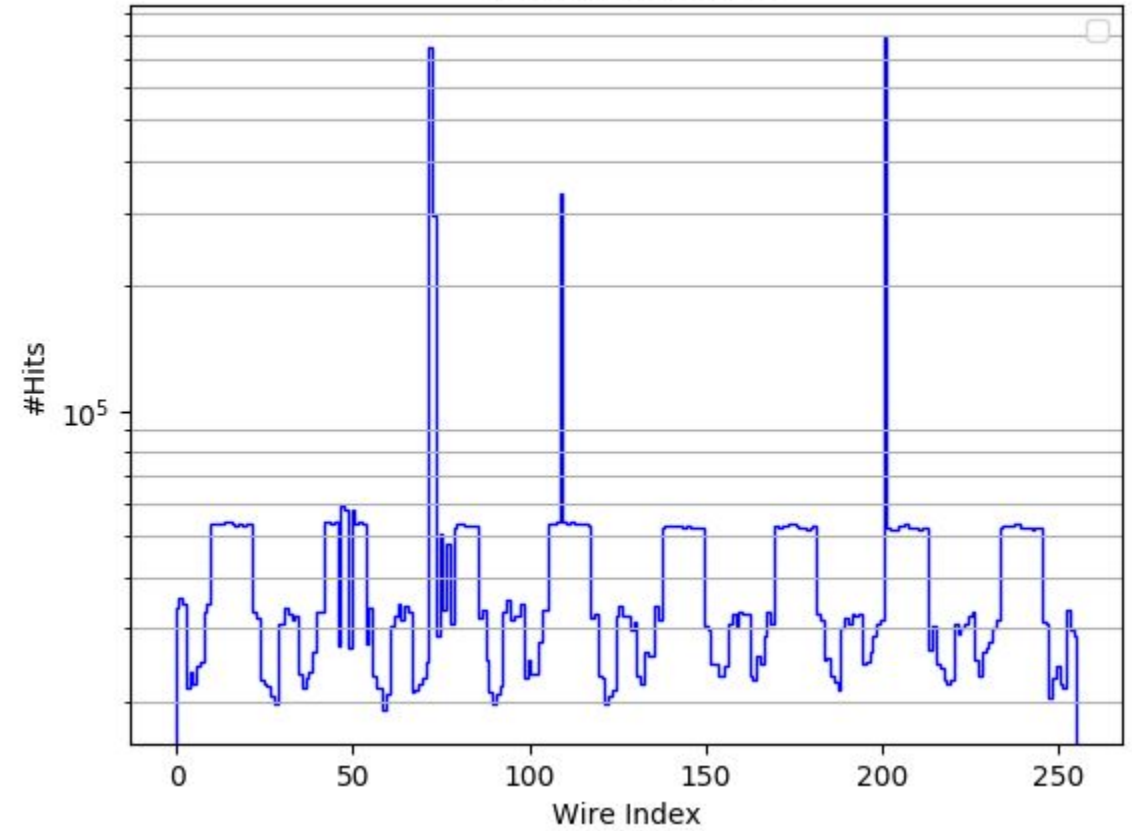


# WireIndex

Wire Index per hit

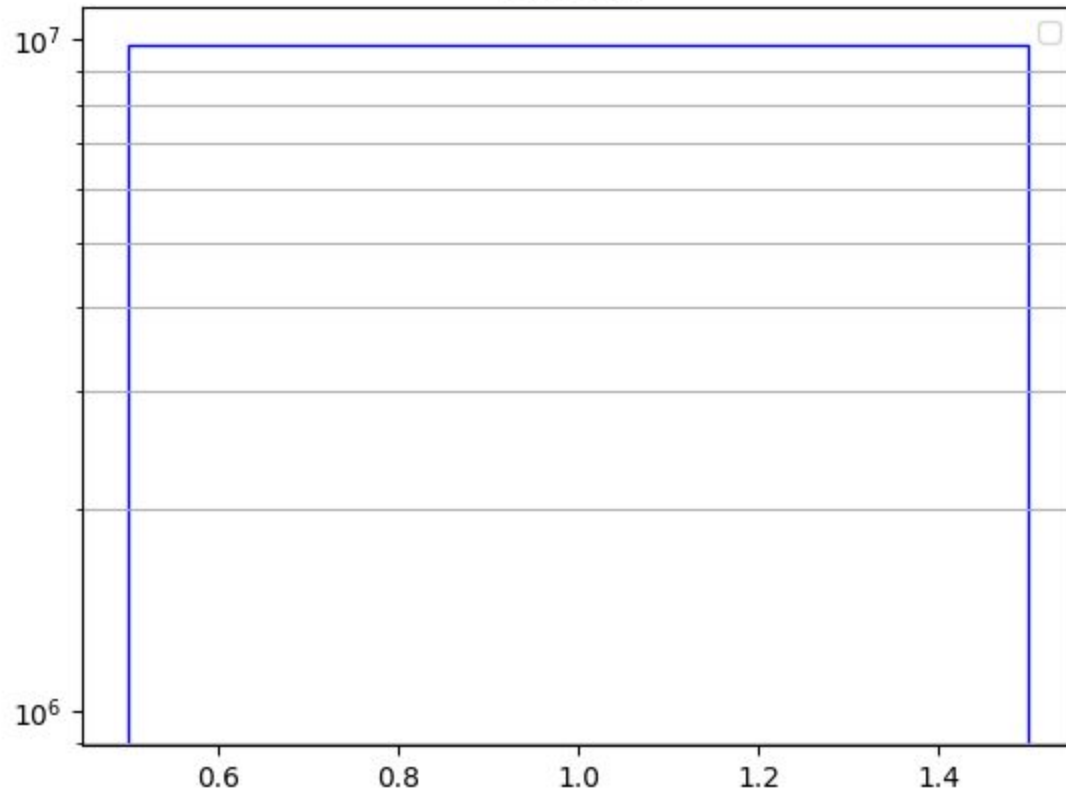


Wire Index per hit

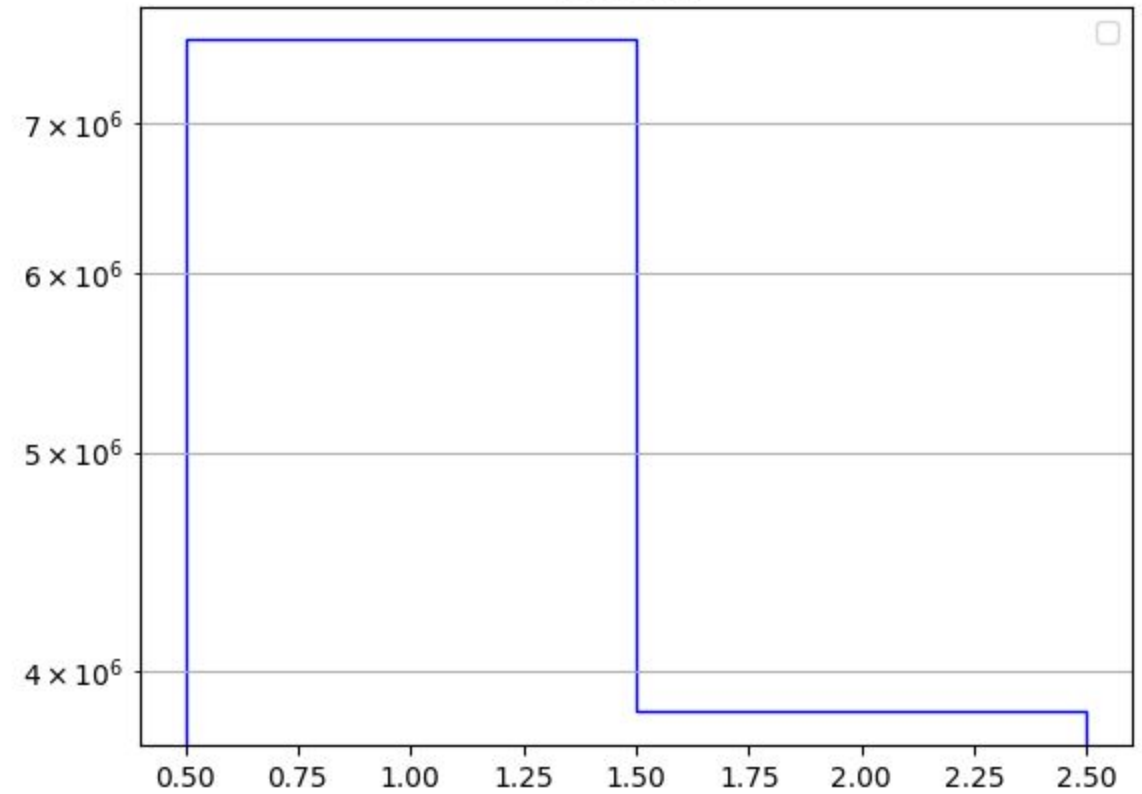


# Fibre

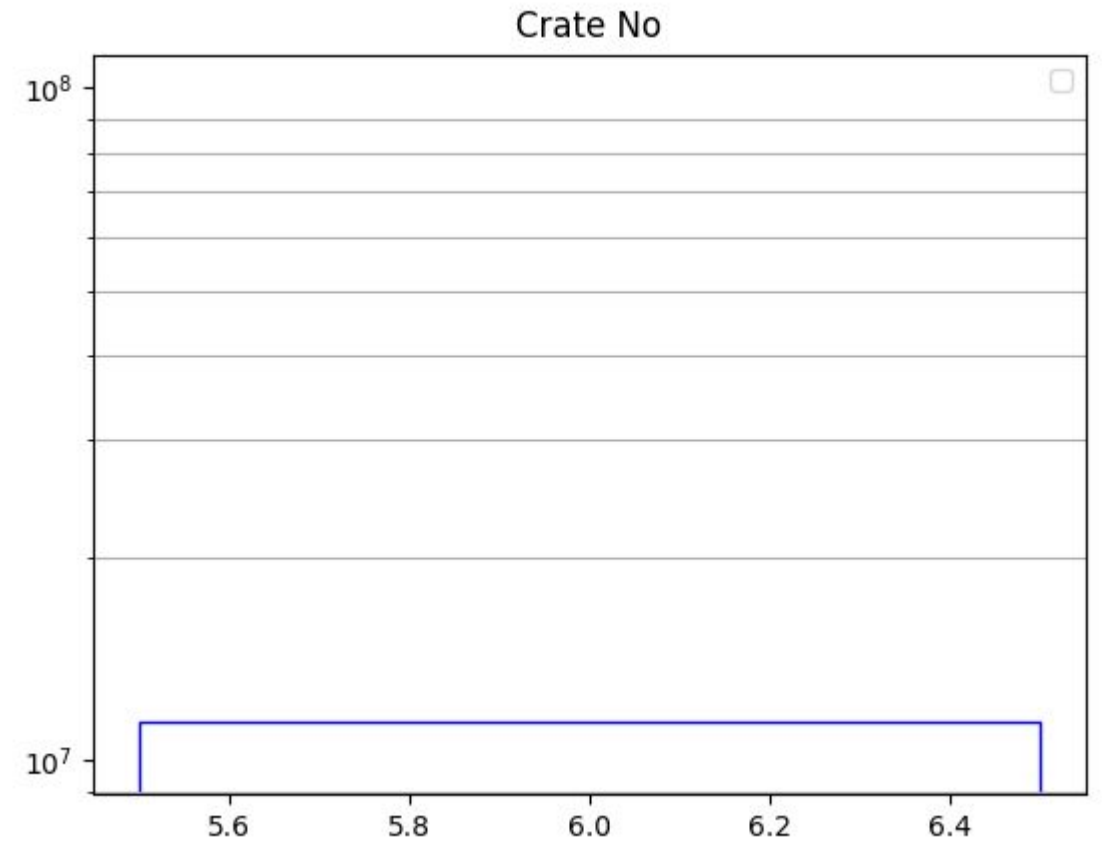
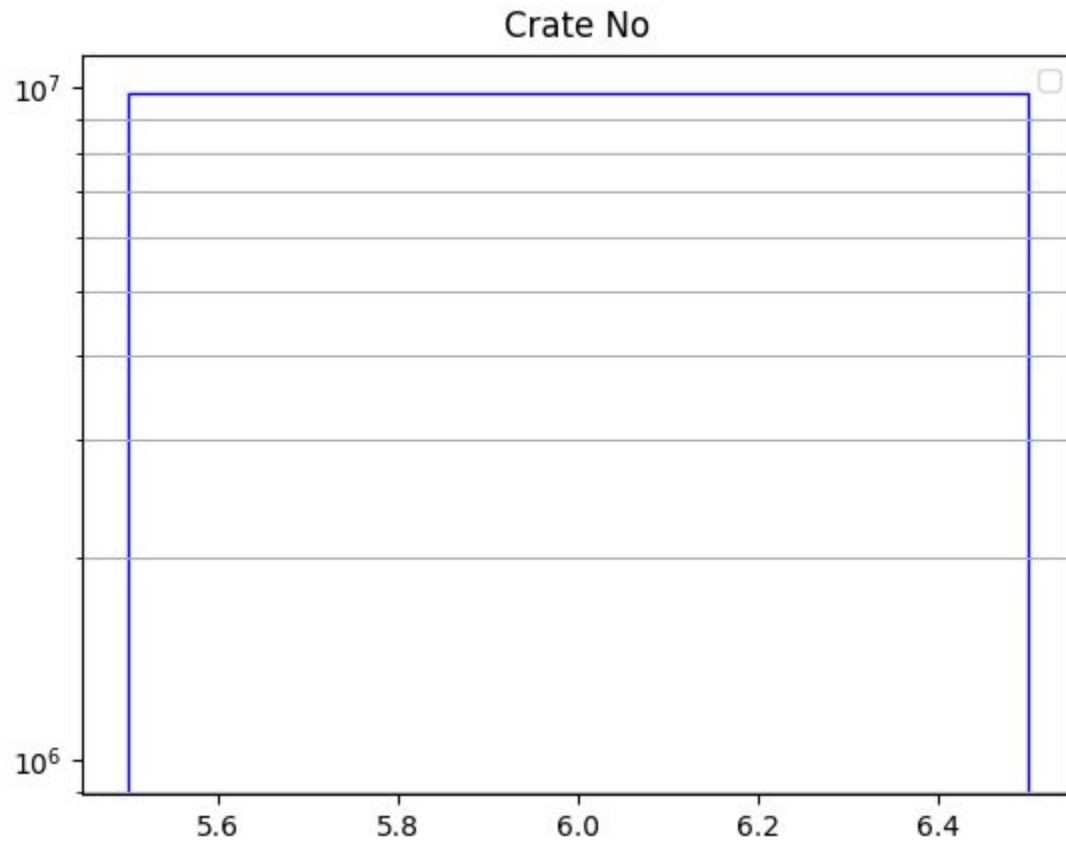
Fibre No



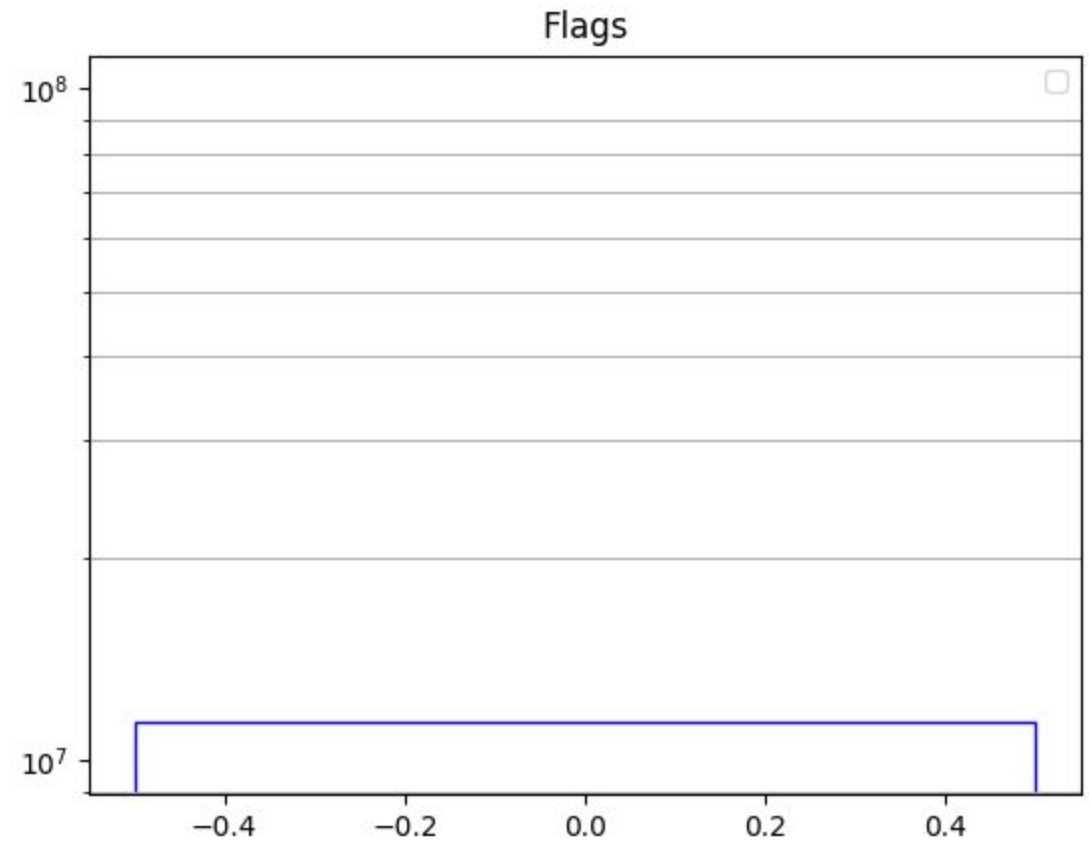
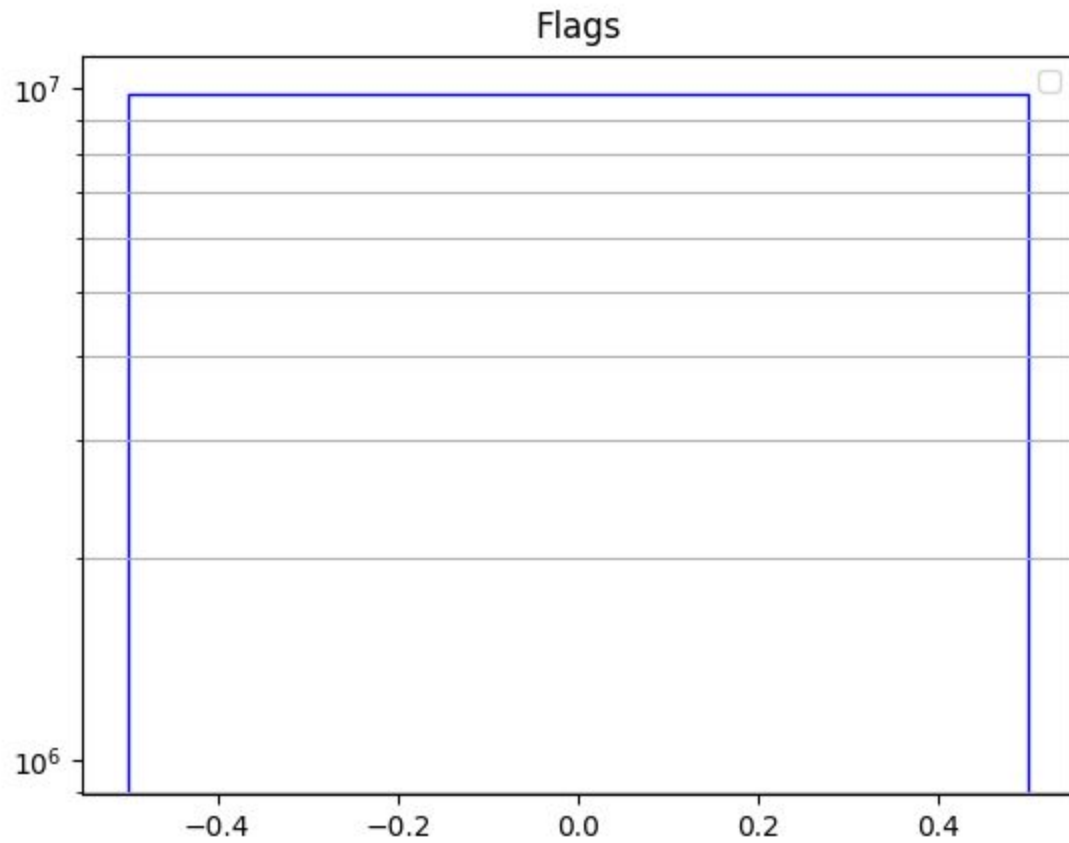
Fibre No



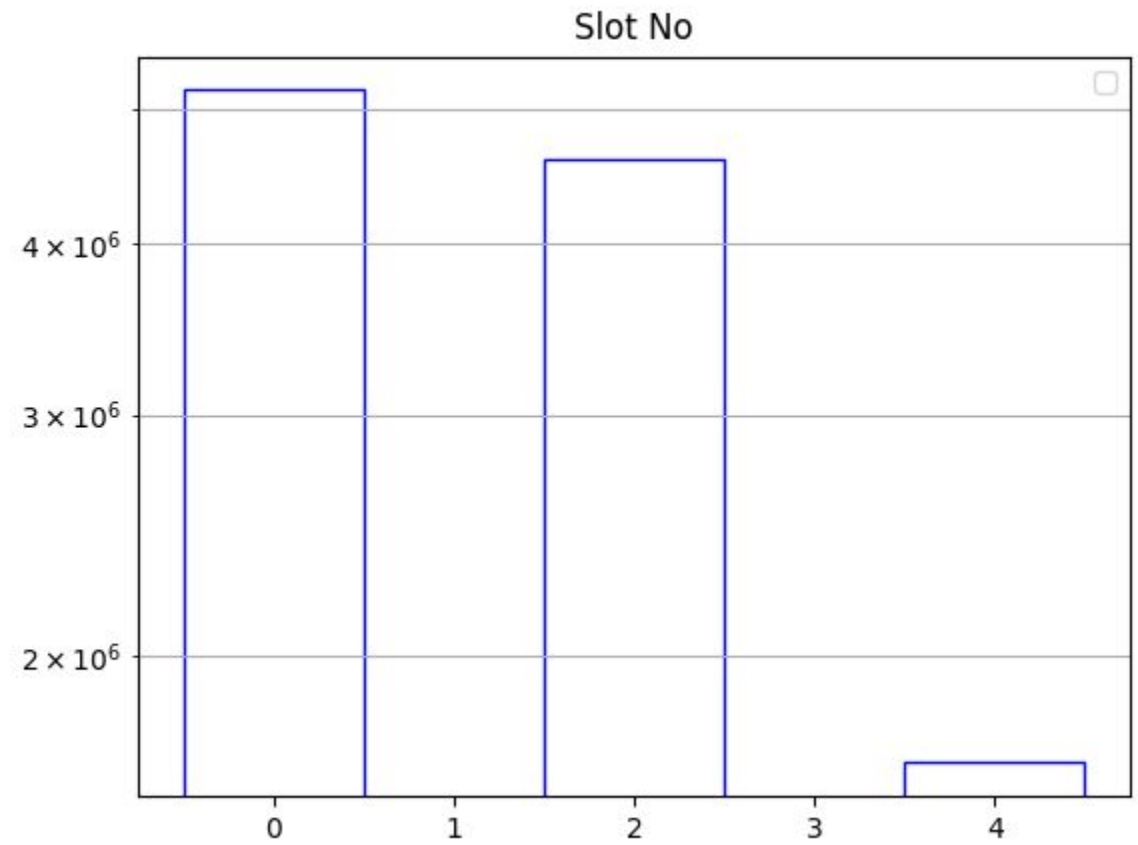
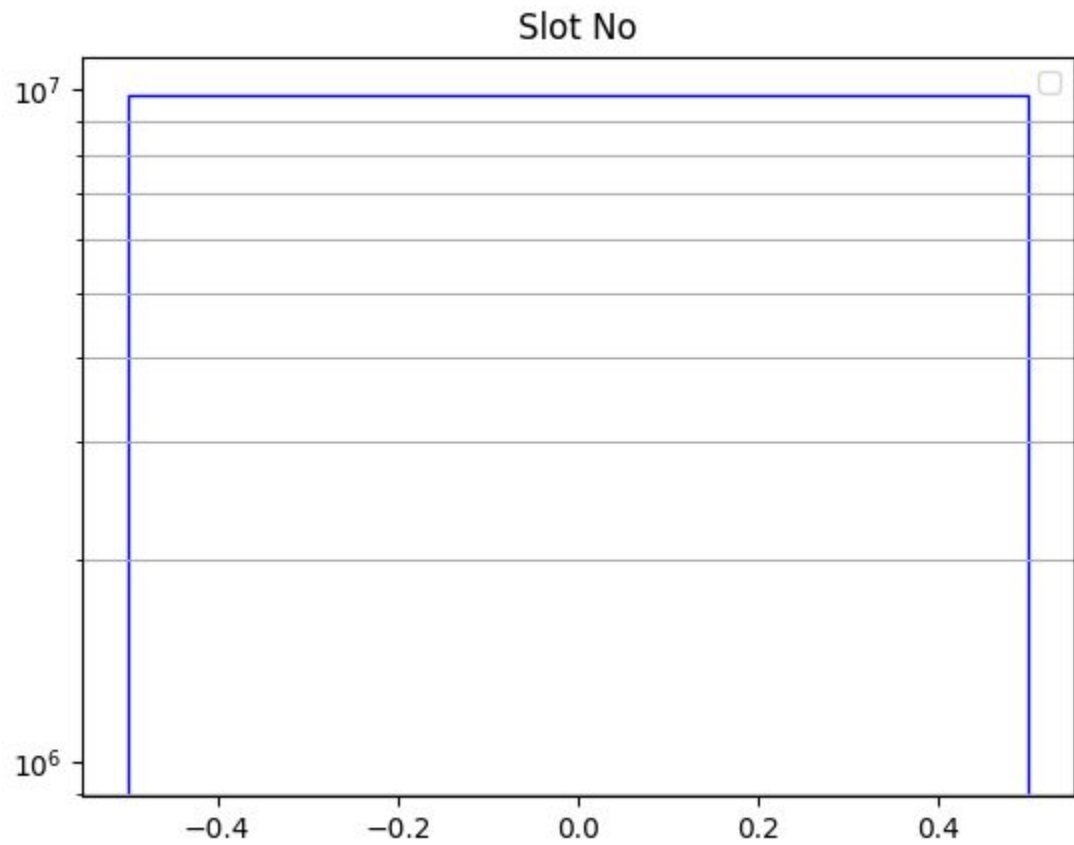
# Crate



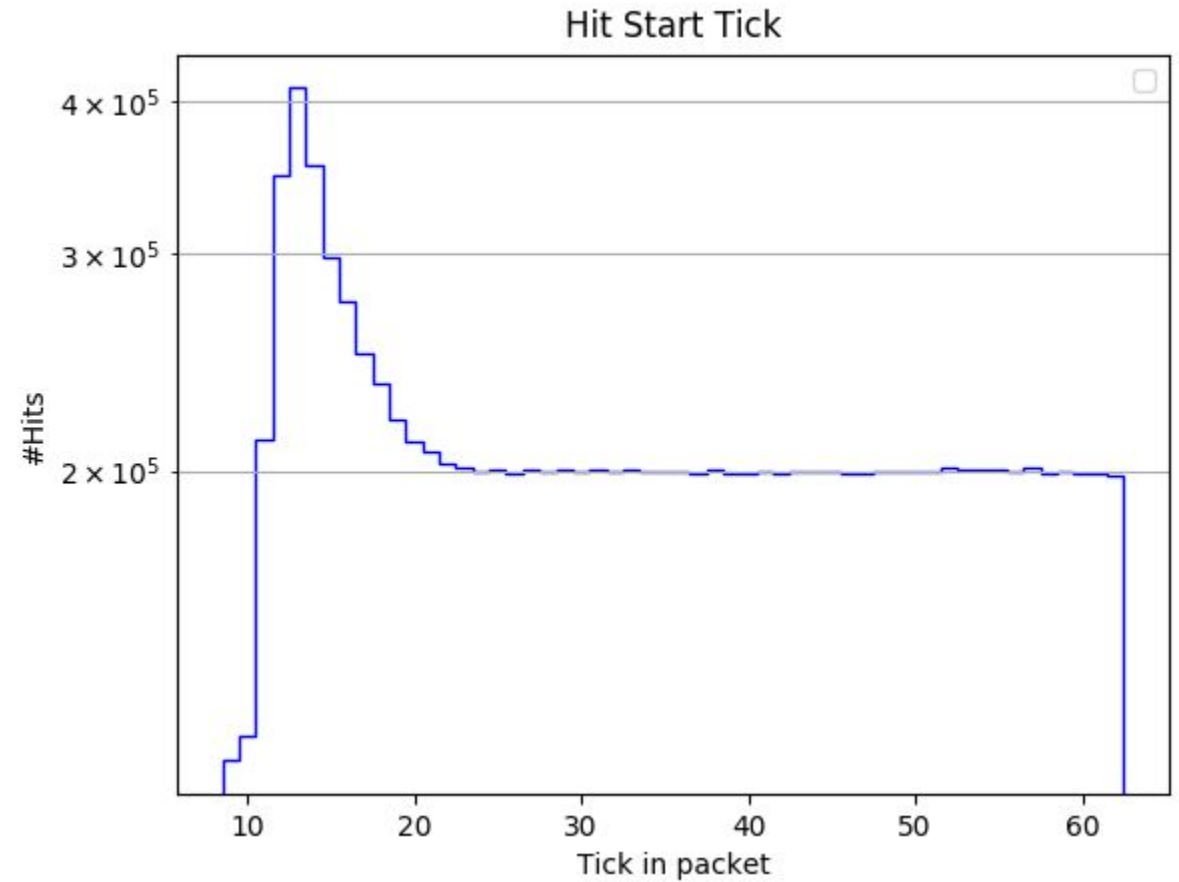
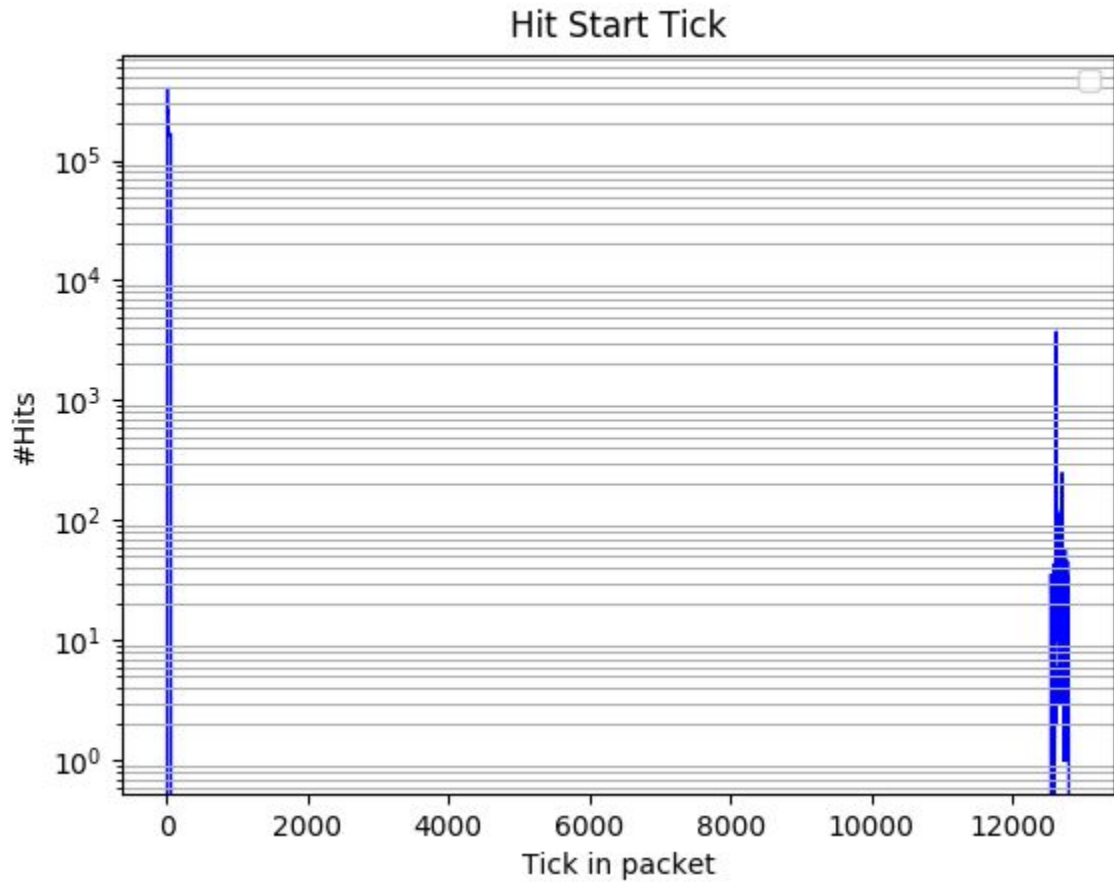
# Flags



# Slots



# Hit Start Tick



# Header Frame Bug

Some of you may remember that when looking at some of the early captures that Tony and I identified a bug which looked like either:

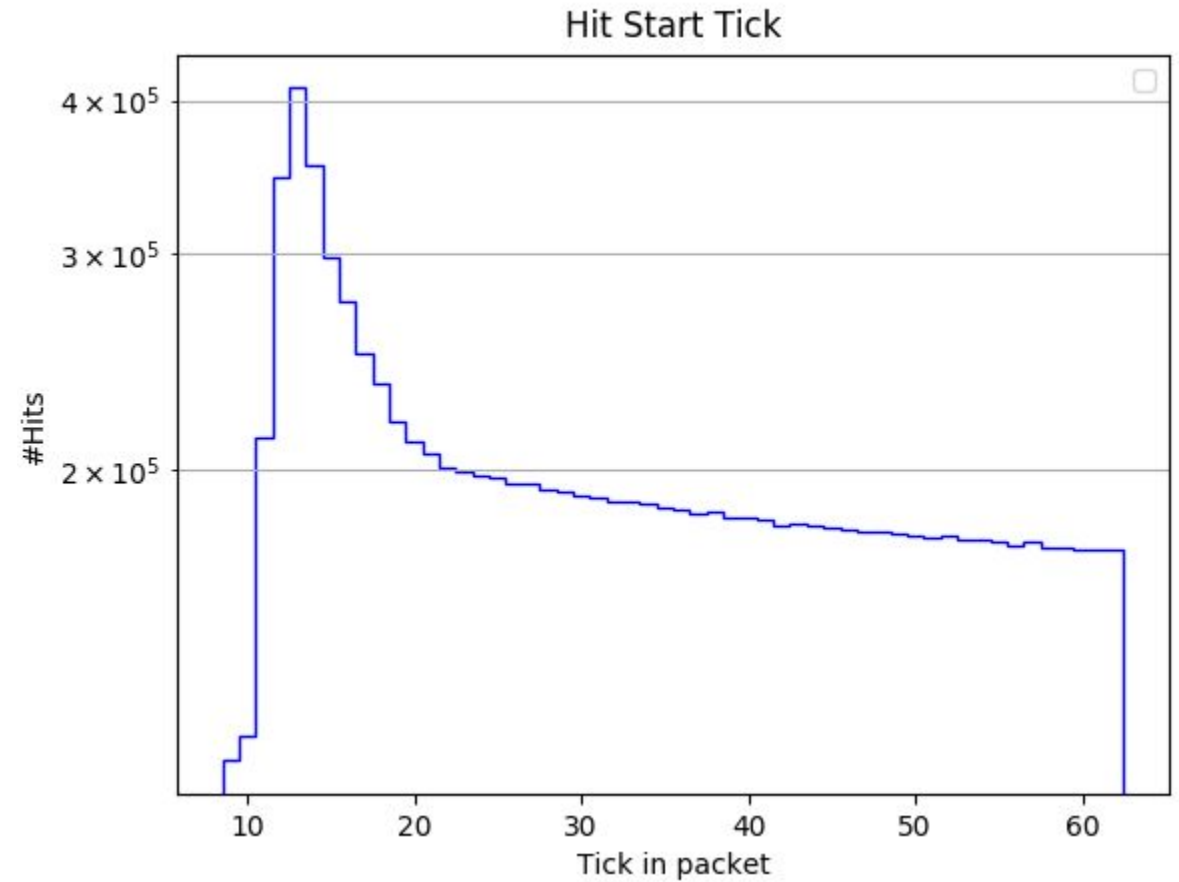
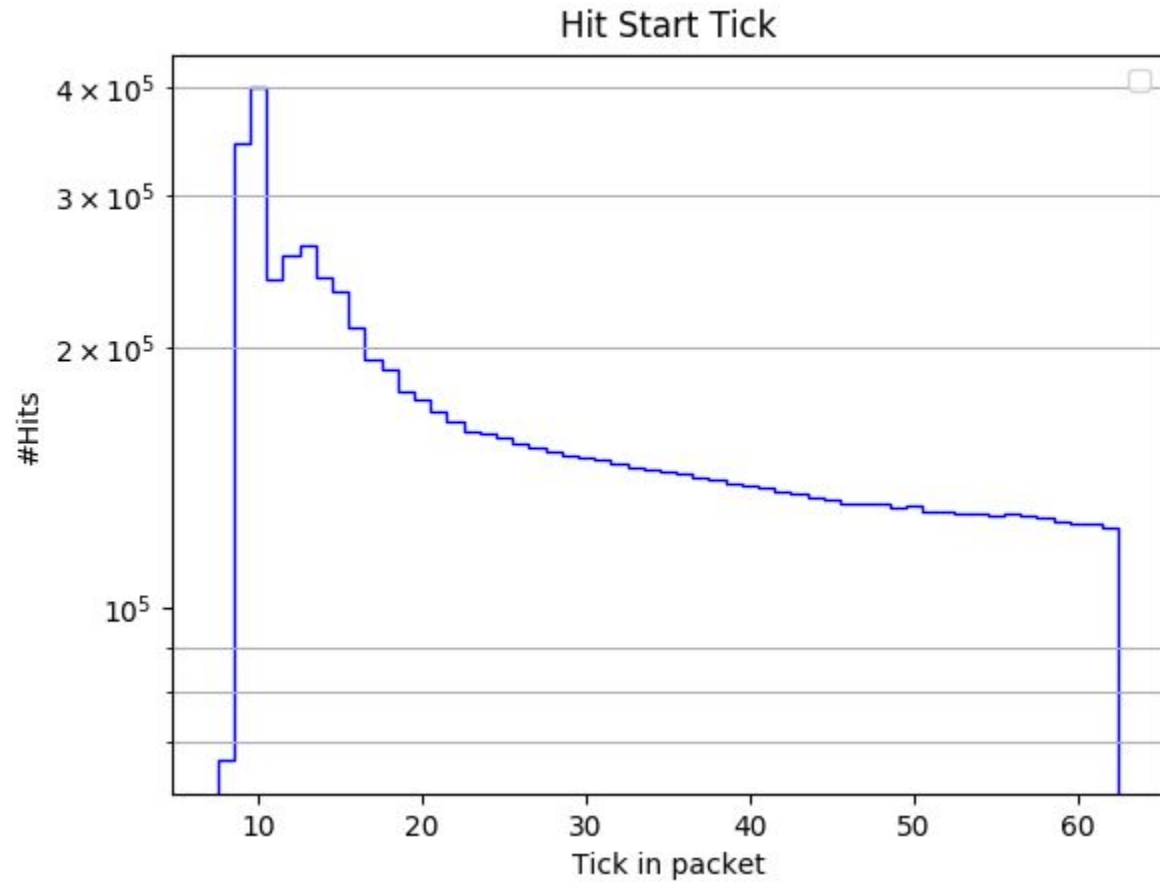
- A header frame was flagged as a hit frame
- A header frame and a hit frame were OR'ed on top of each other

Might be an arbitrator bug

From initial visual inspections we found that it only occurred in packets where there appeared to be more than 1 hit and would only manifest in the 2nd or later hit. So instead if we look at plots where we only use the first hit from each hit packet.

```
, Hit Header:  
crate: 6, fibre: 1, wireIndex: 77, slot: 0, flags: 0, tstamp: 79701165942110043  
  
Hit Payload:  
0 : hitStartTime: 25, hitEndTime: 42, hitPeakADC: 46, hitPeakTime: 33, hitSumADC: 649, hitFlags: 0, hitContinues: 0  
1 : hitStartTime: 12620, hitEndTime: 0, hitPeakADC: 63323, hitPeakTime: 50217, hitSumADC: 10185, hitFlags: 141, hitContinues: 1  
2 : hitStartTime: 17, hitEndTime: 26, hitPeakADC: 33, hitPeakTime: 21, hitSumADC: 287, hitFlags: 0, hitContinues: 0
```

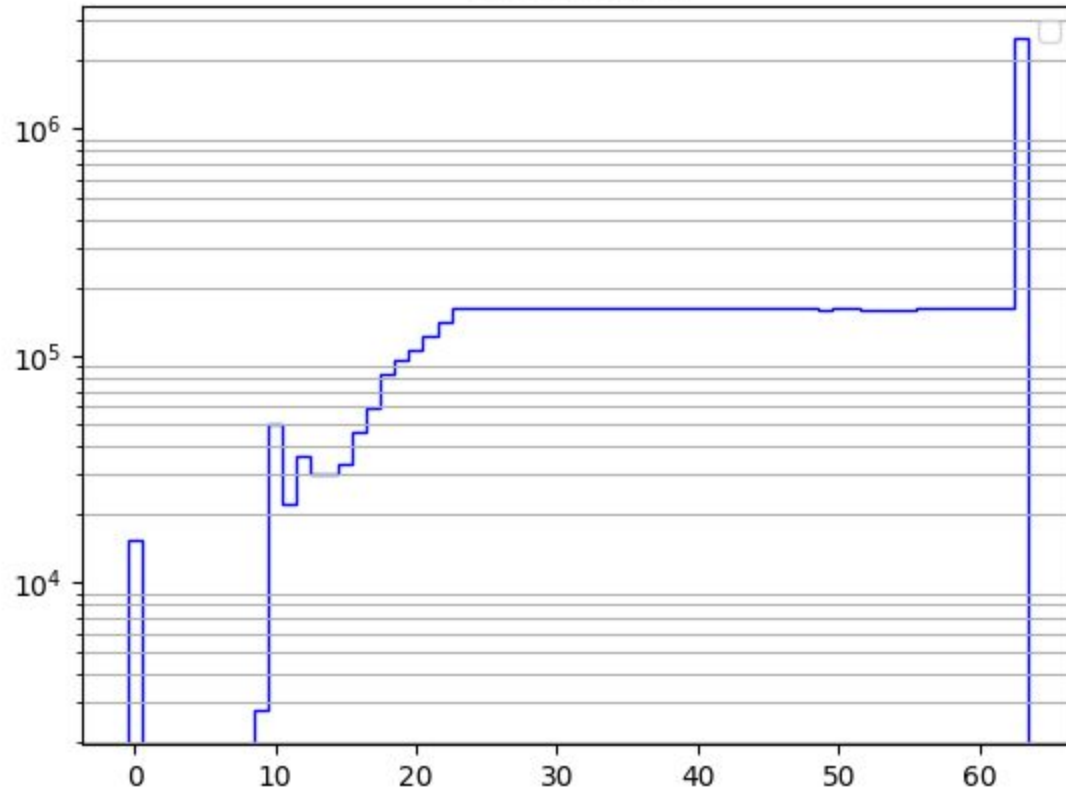
# Hit Start Tick (1st hit in packet)



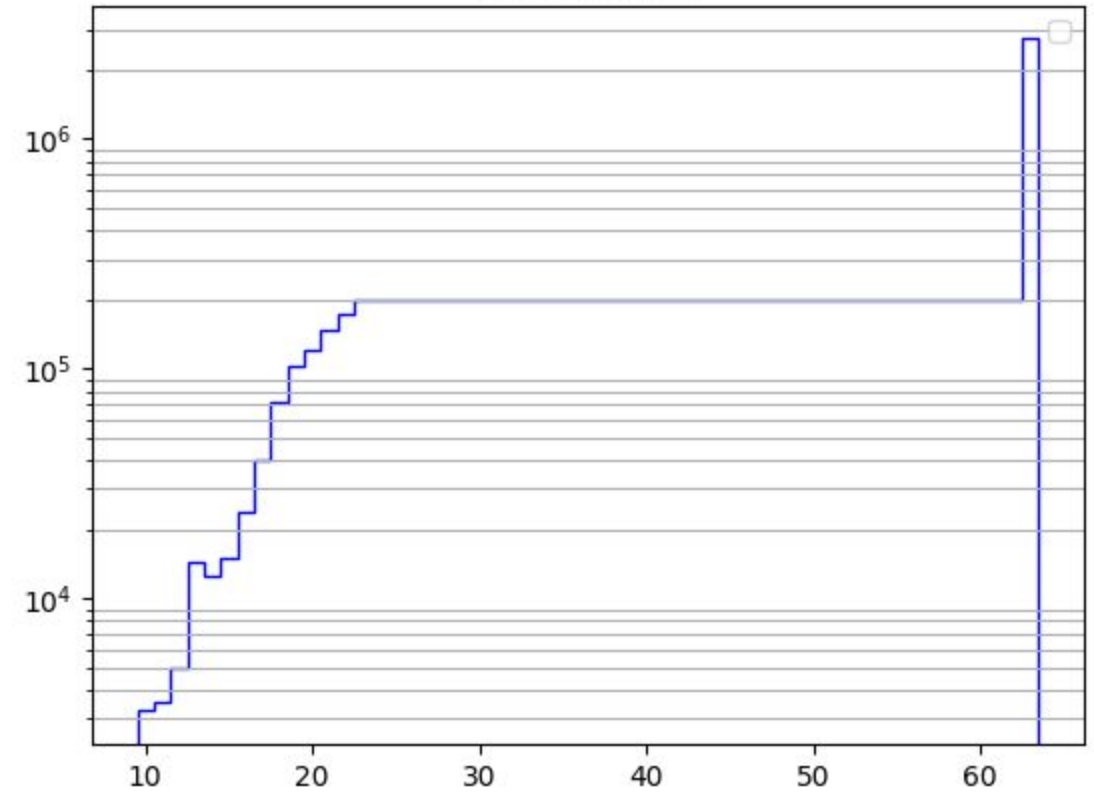


# Hit End Tick

Hit End Tick

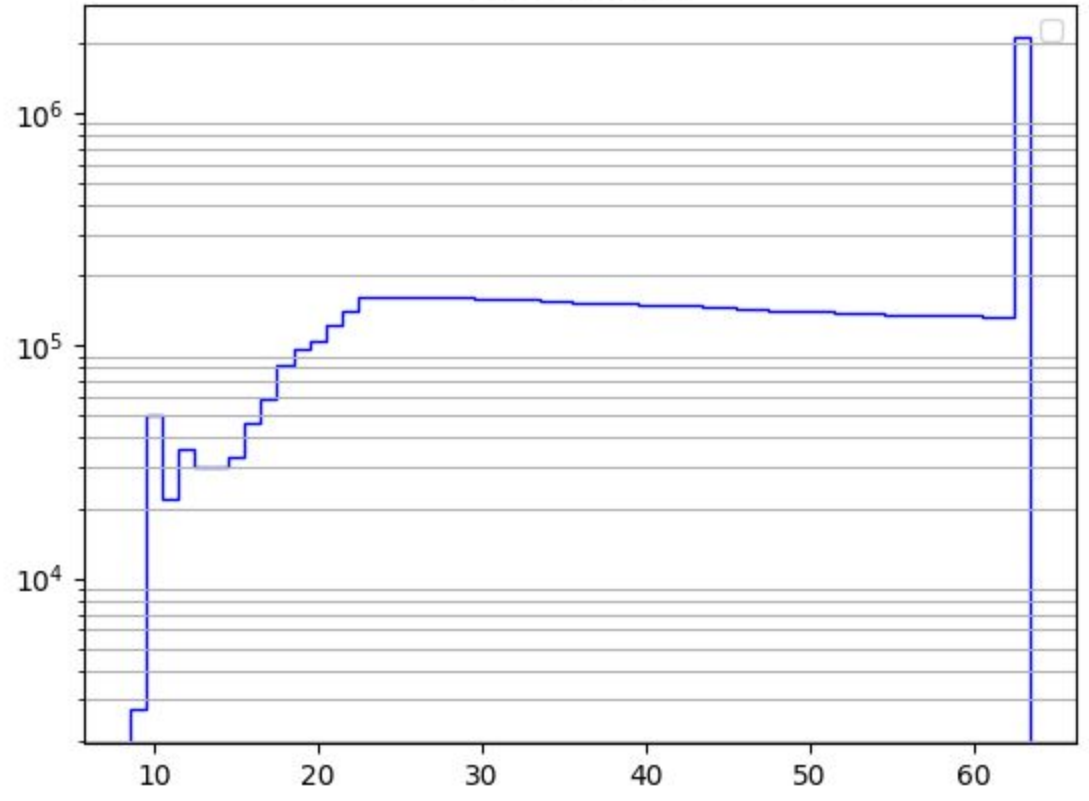


Hit End Tick

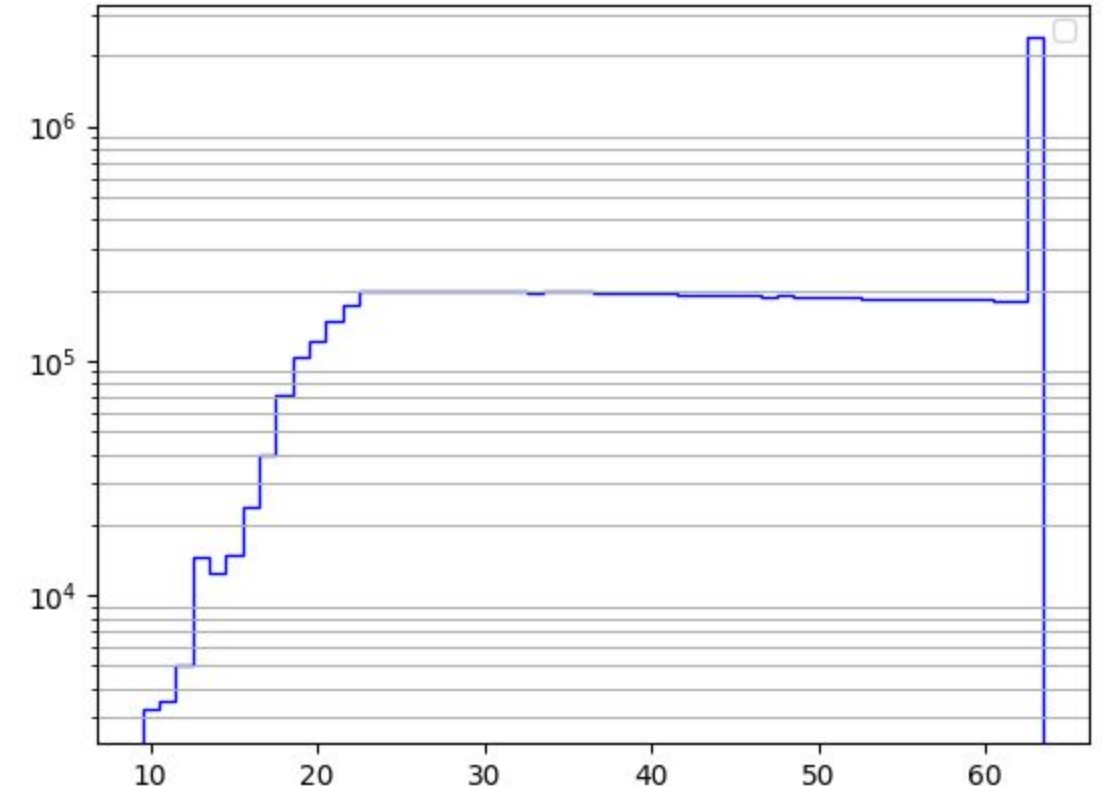


# Hit End Tick (1st hit in packet)

Hit End Tick

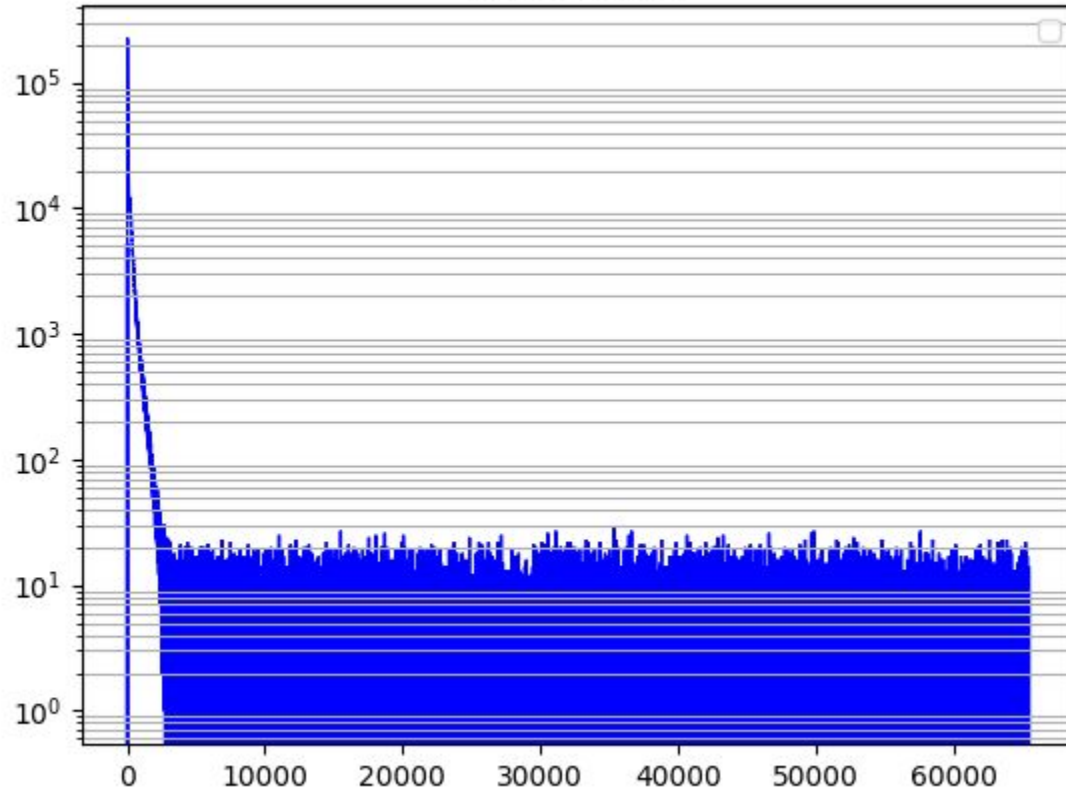


Hit End Tick

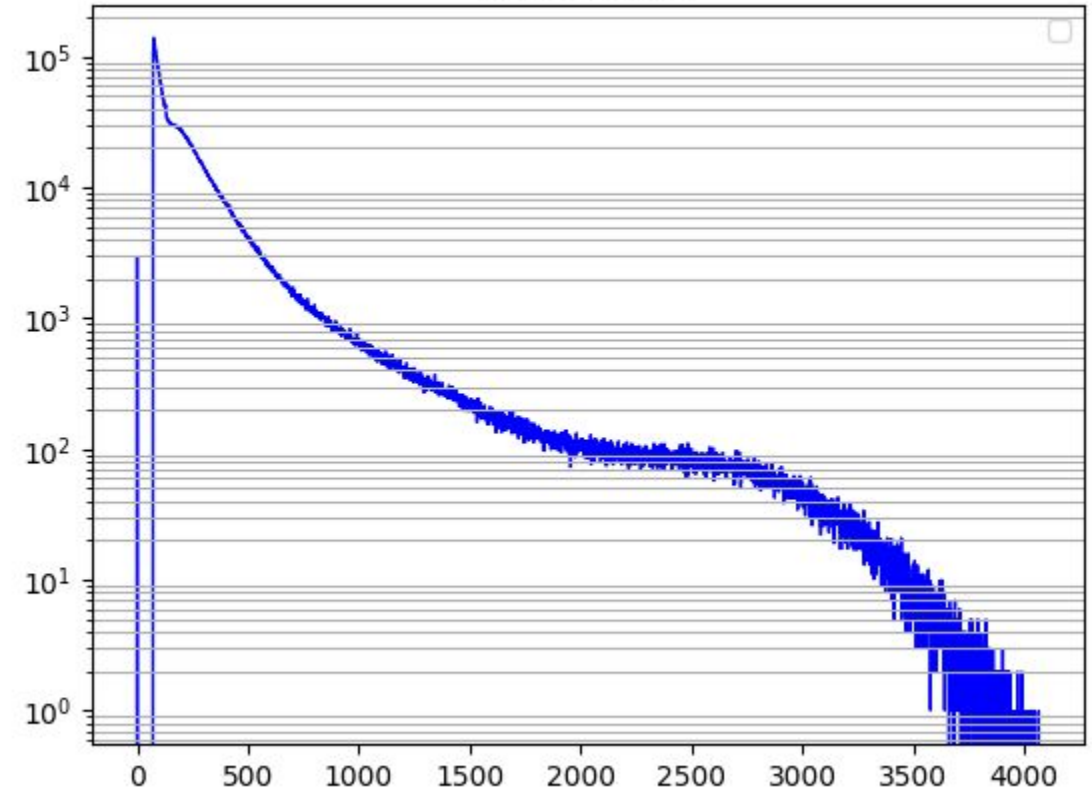


# Hit Peak

Hit Peak ADC

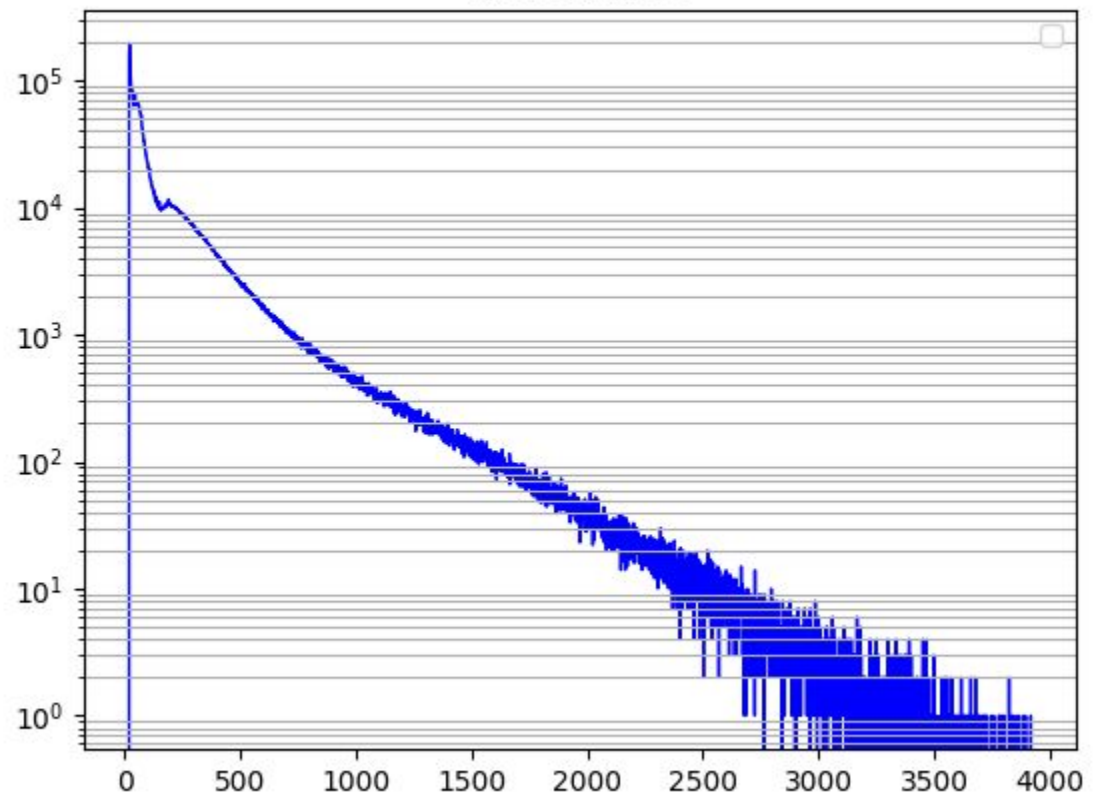


Hit Peak ADC

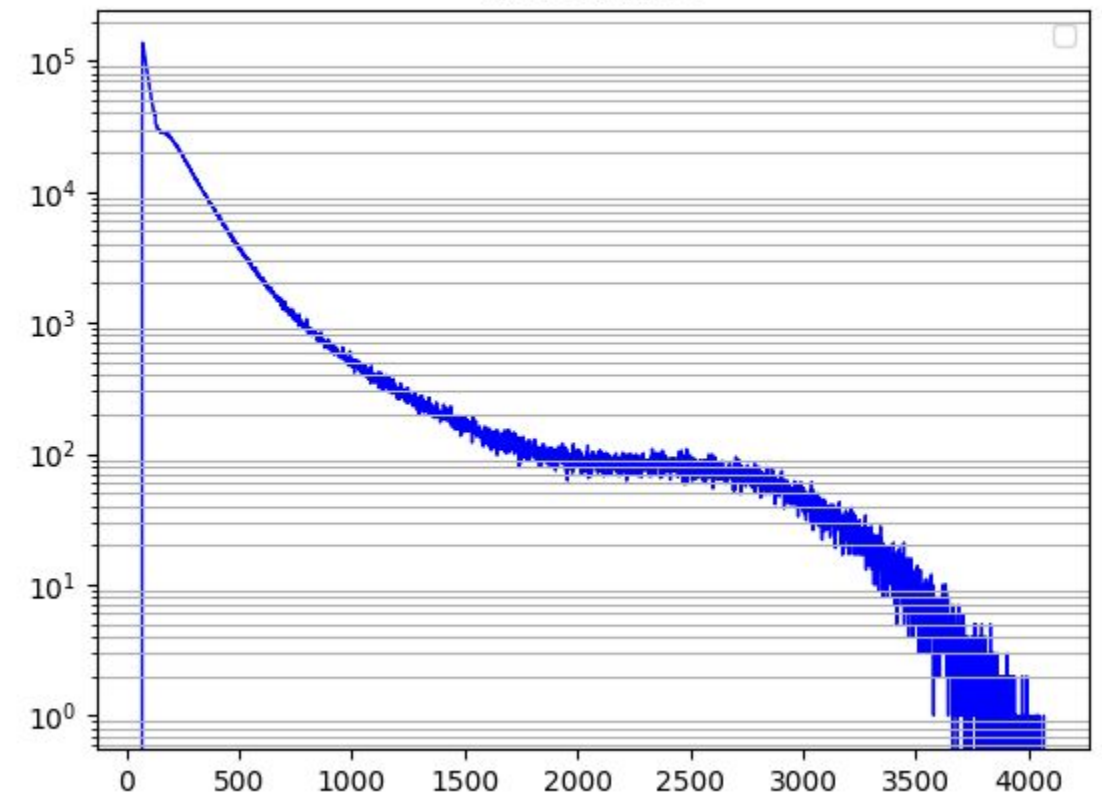


# Hit Peak (1st hit in packet)

Hit Peak ADC

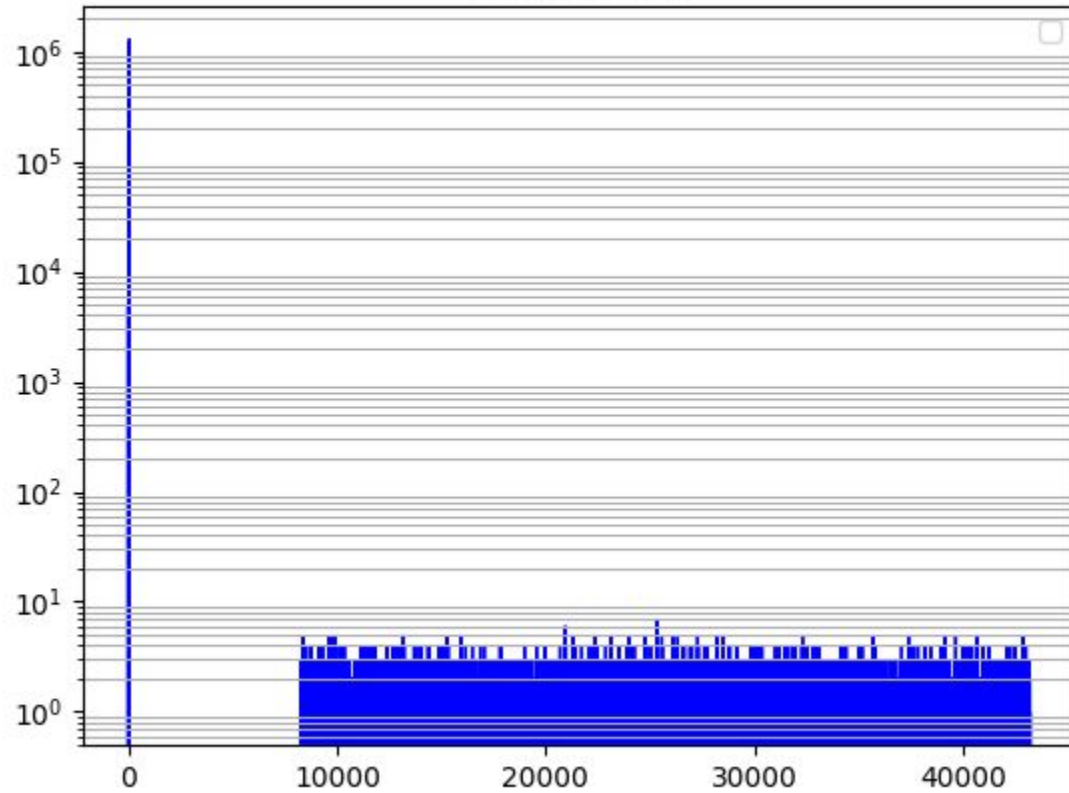


Hit Peak ADC

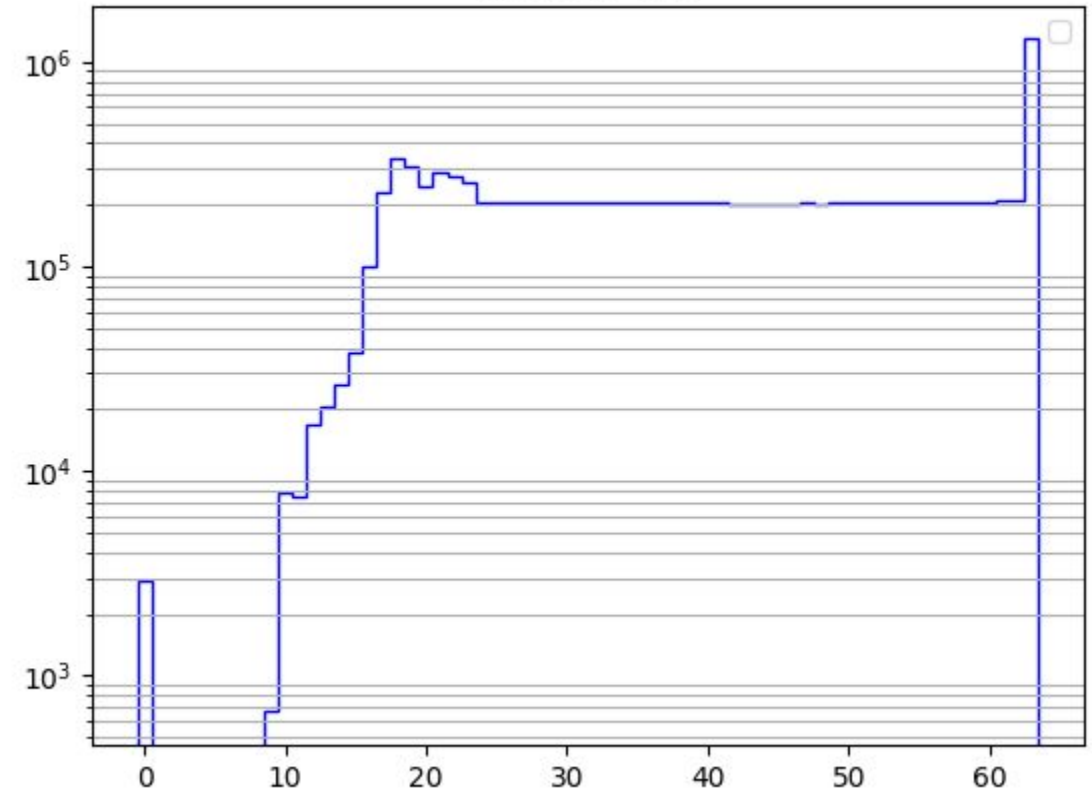


# Hit Peak Tick

Hit Peak Tick

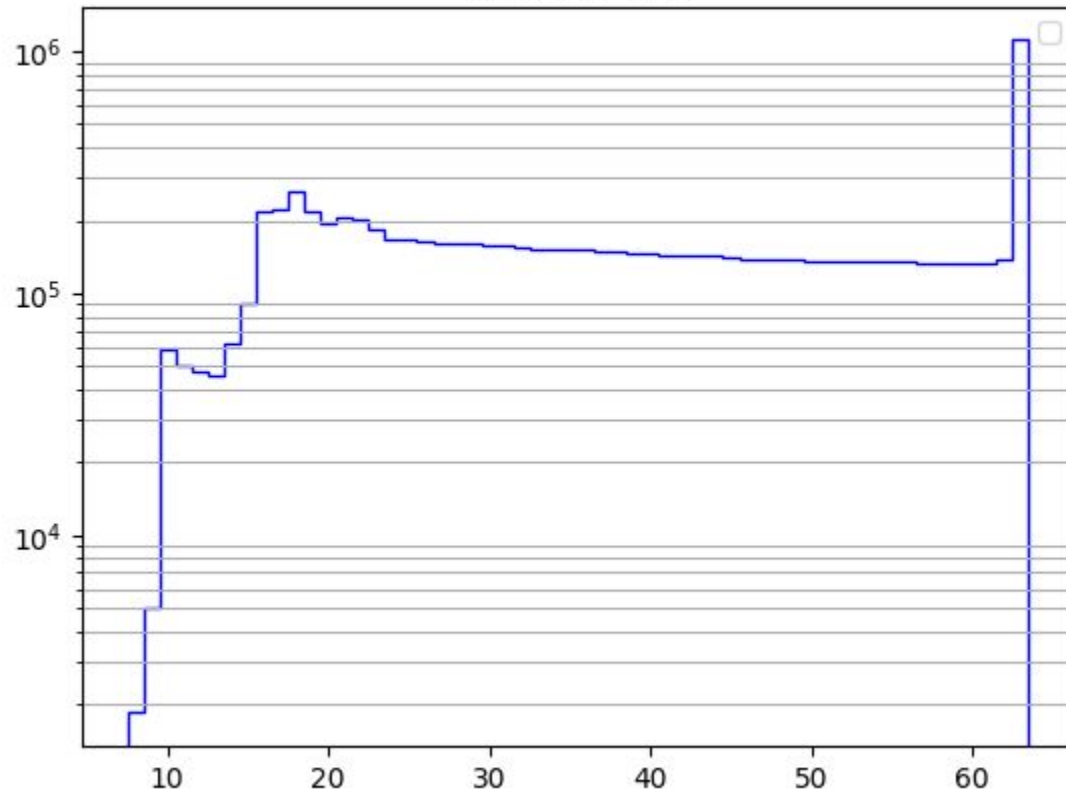


Hit Peak Tick

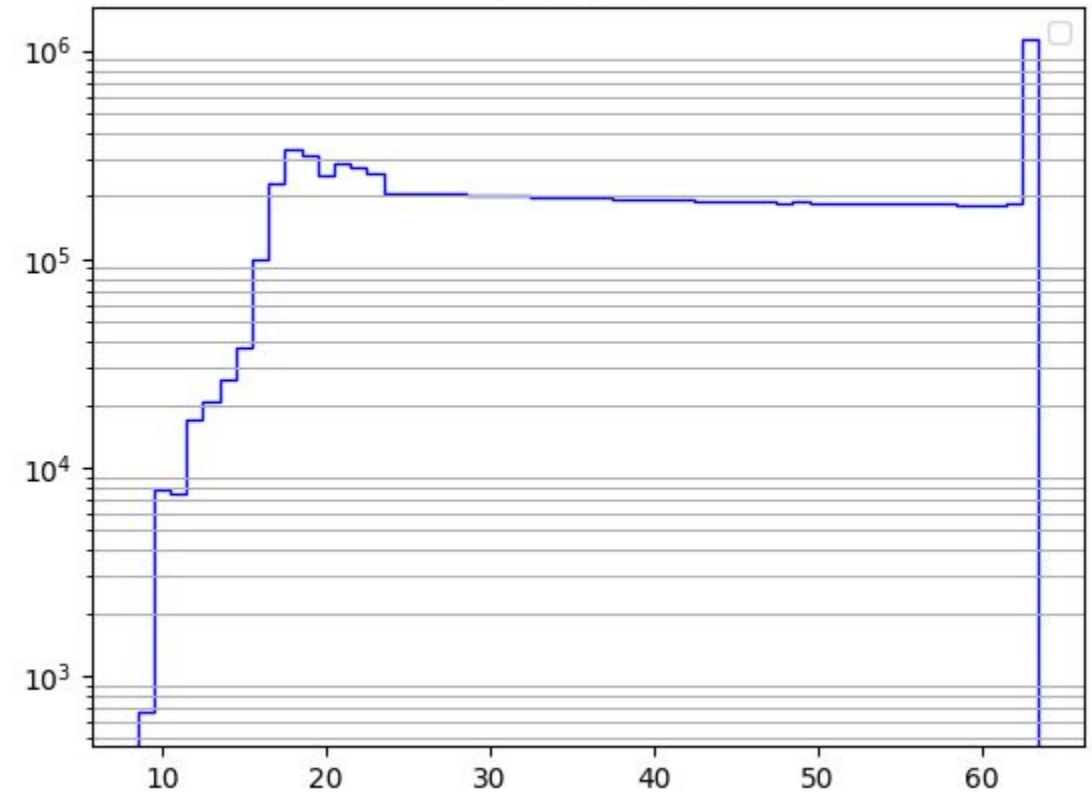


# Hit Peak Tick (1st hit in packet)

Hit Peak Tick

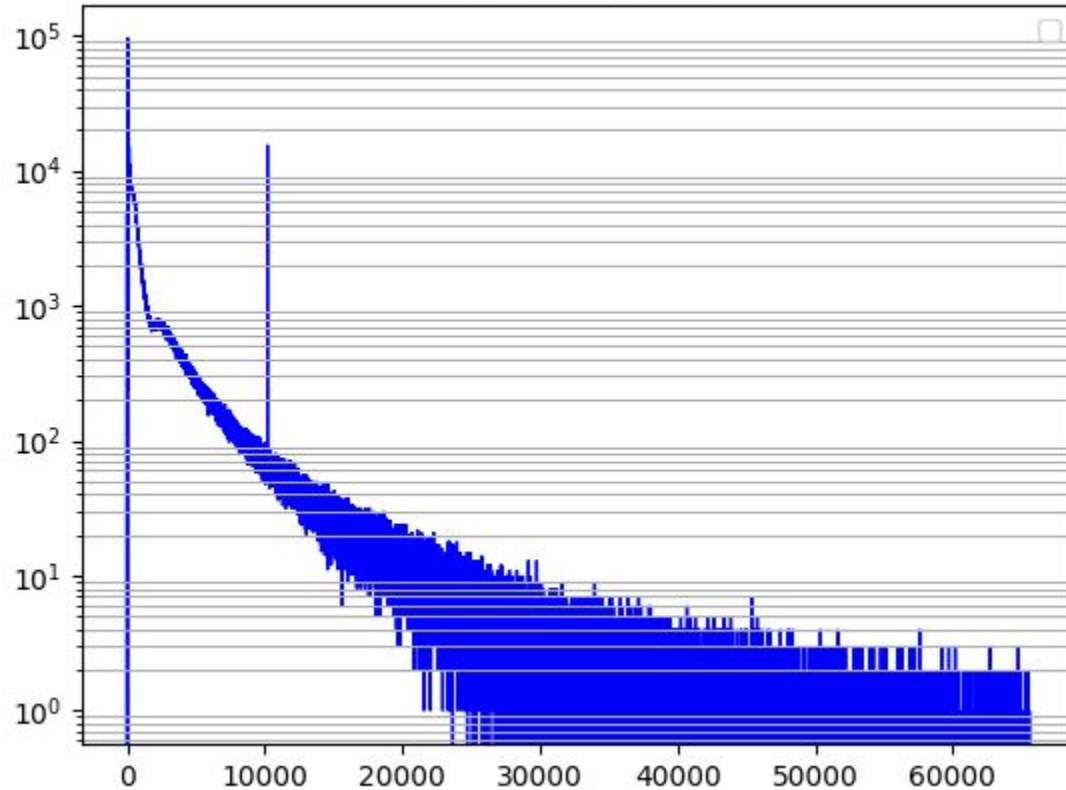


Hit Peak Tick

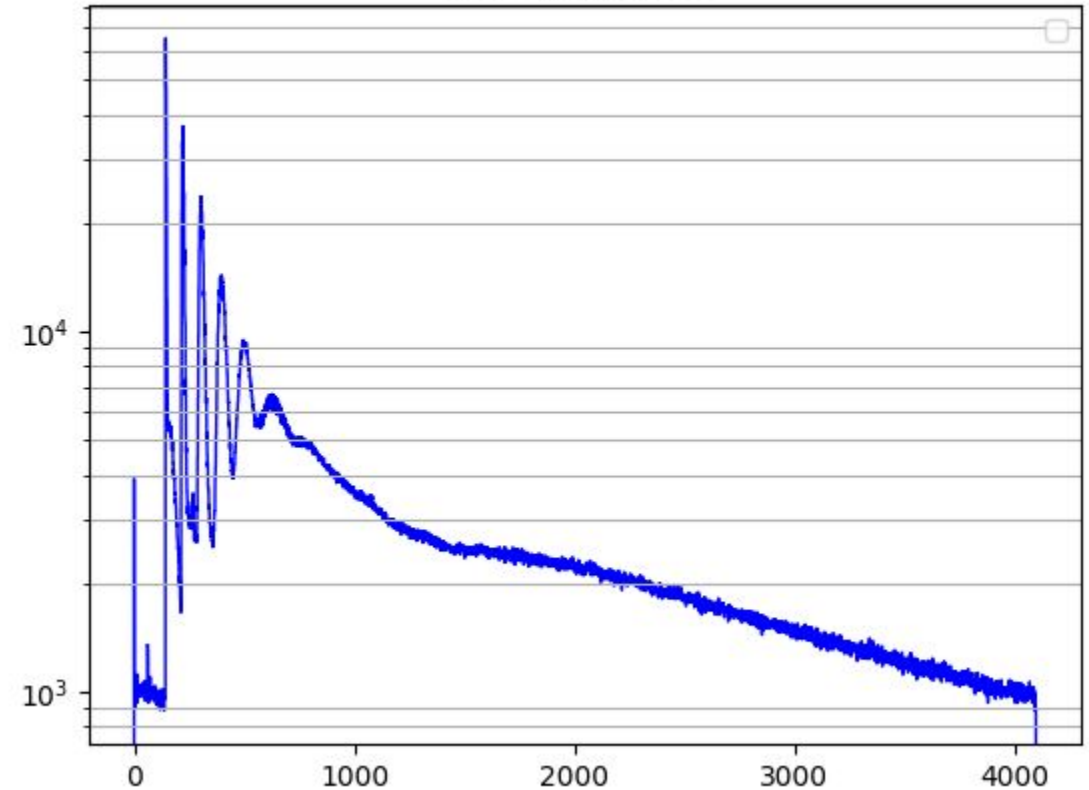


# Hit SummedADC

Hit SummedADC

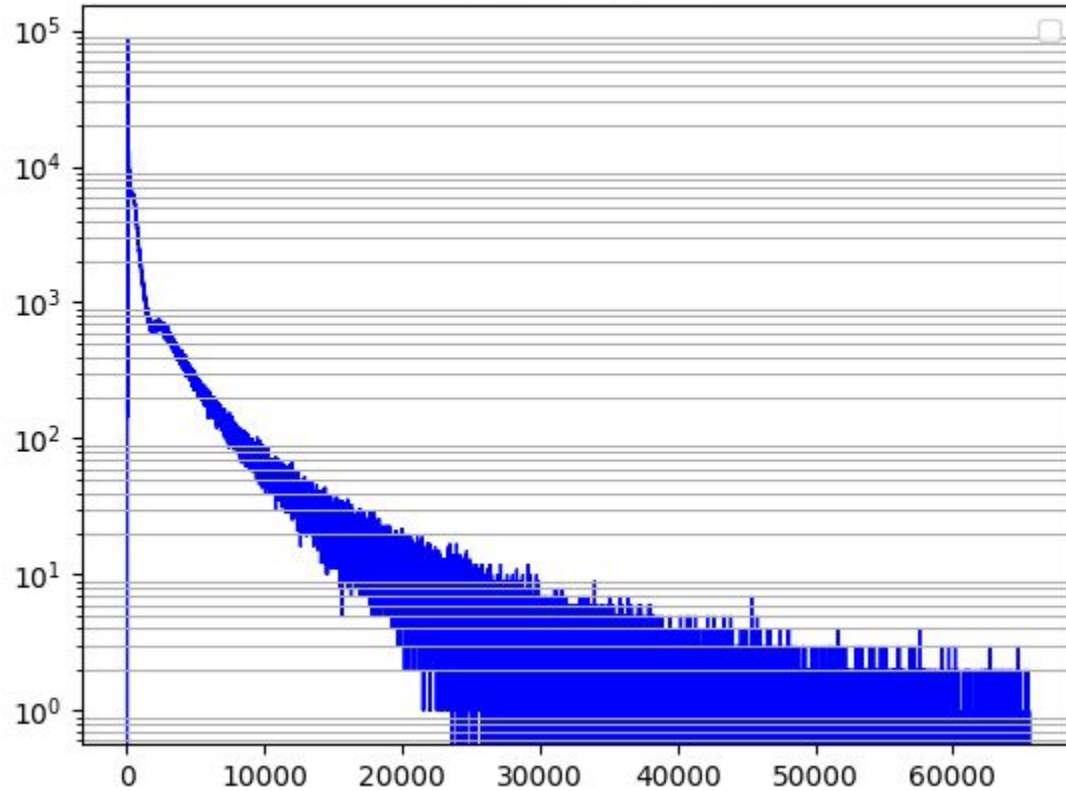


Hit SummedADC

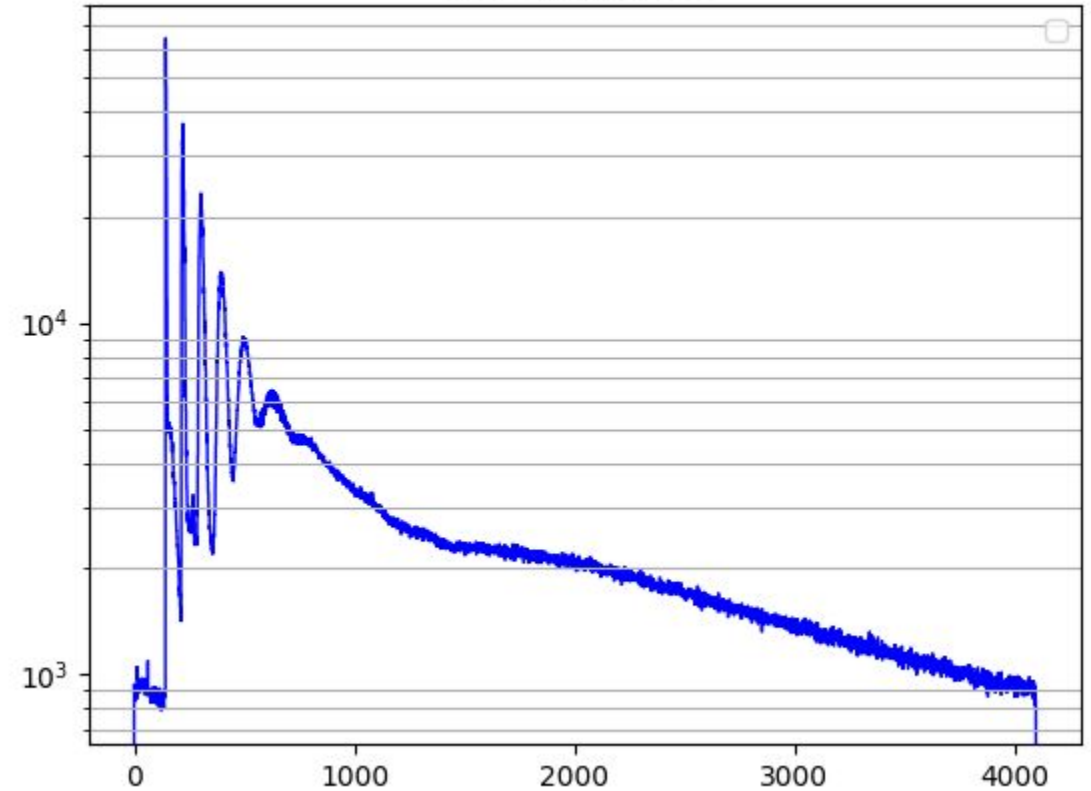


# Hit SummedADC (1st hit in packet)

Hit SummedADC



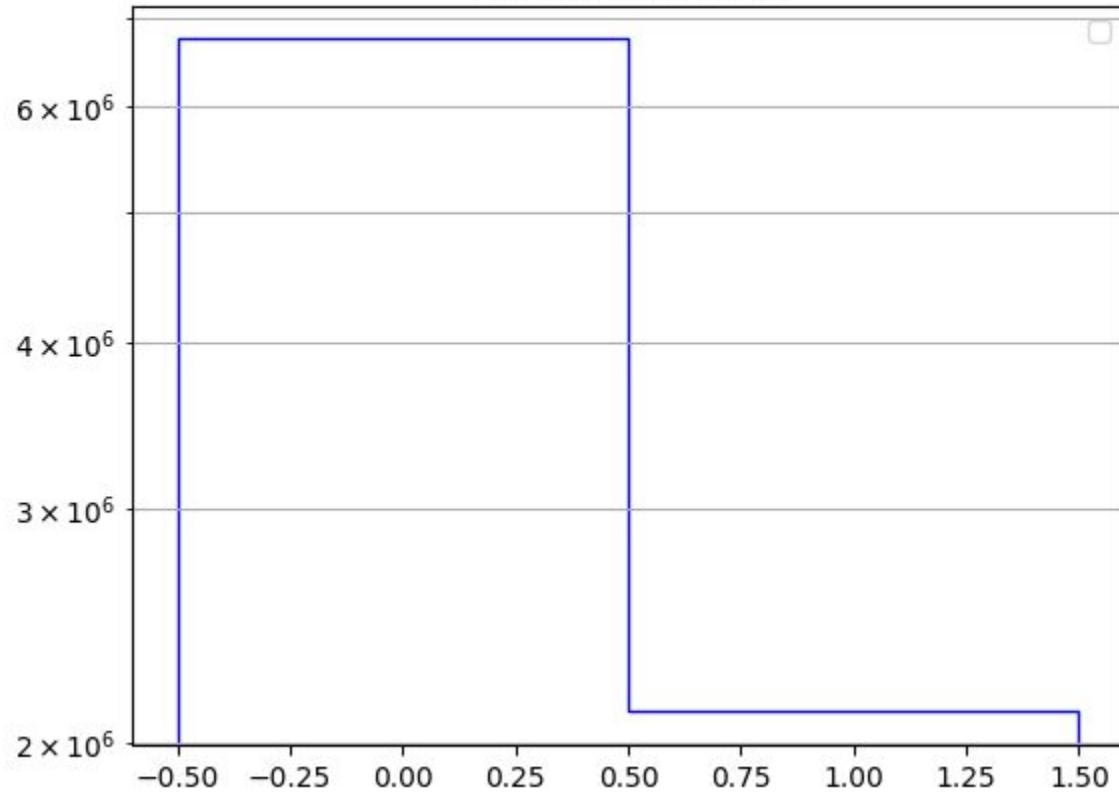
Hit SummedADC



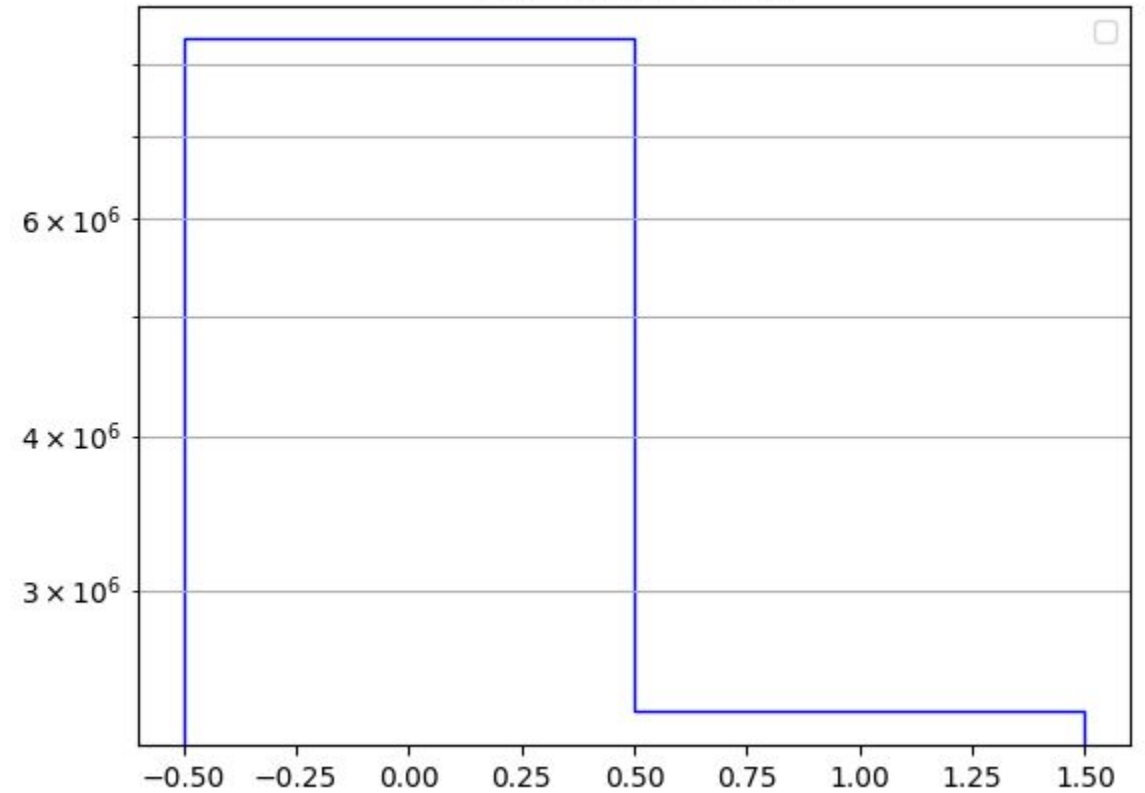


# Hit Continues Flag (1st hit in packet)

Hit Continues Flag

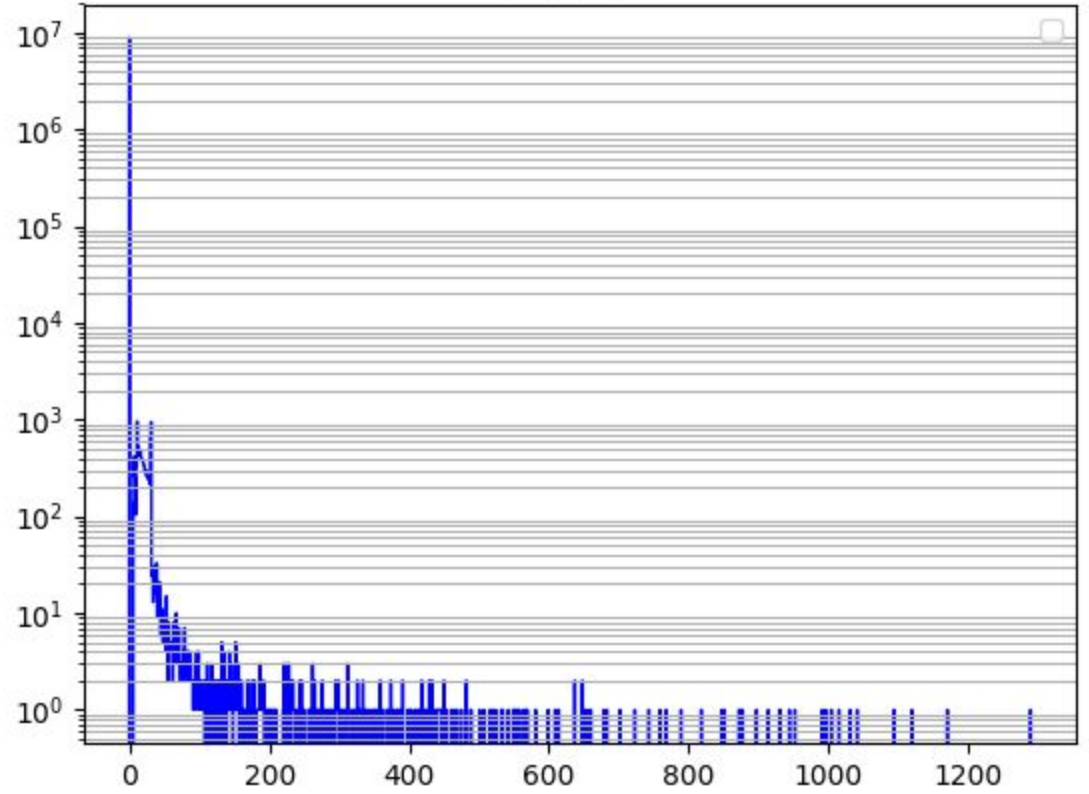


Hit Continues Flag

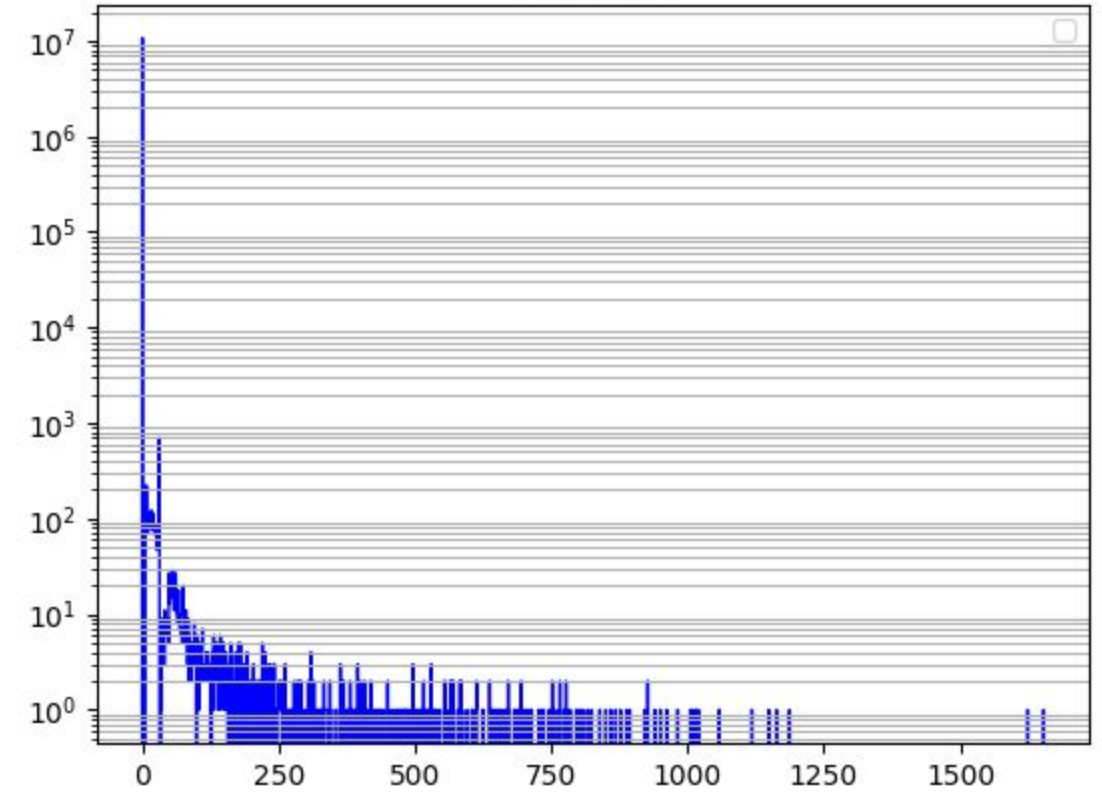


# HitFlags(1st hit in packet)

Hit Flags

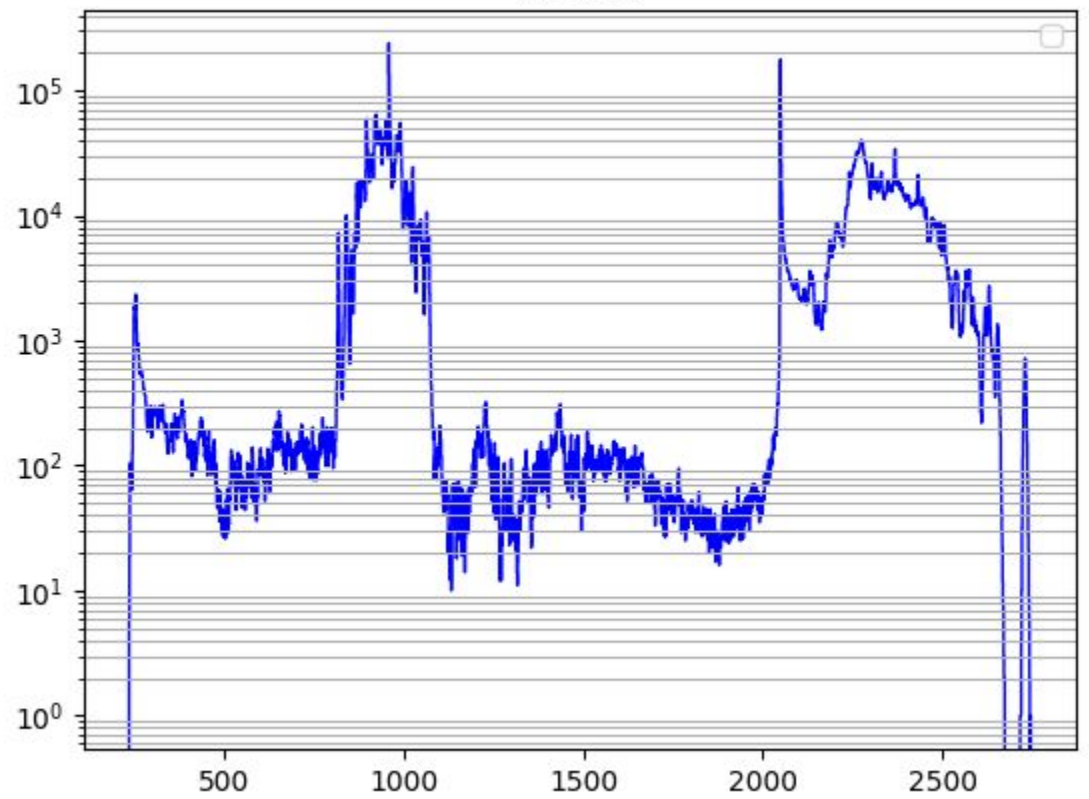


Hit Flags

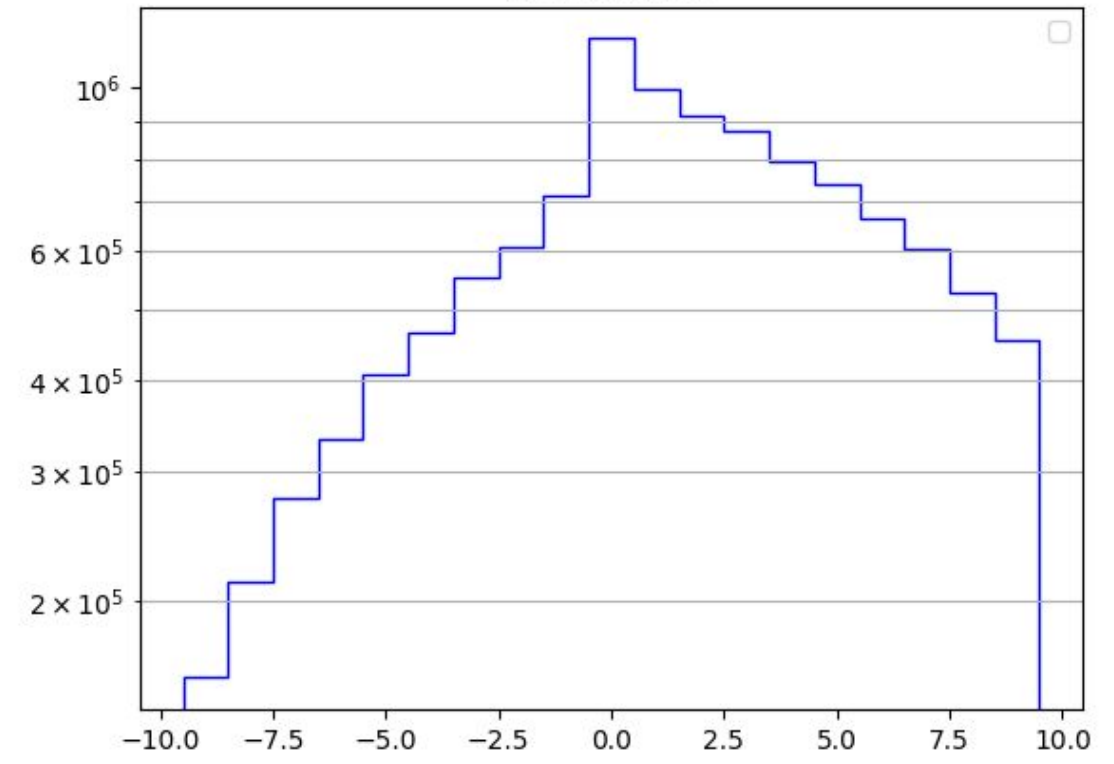


# Median and Accumulator

Median



Accumulator



# Summary

---

- We see some ‘impossible’ values for some of the hit quantities, indicating one or more bugs:
  - We know there is an issue in the firmware where it looks like a header frame is flagged as a hit frame - removing all hits where this could occur seems to improve the plots for APA6-newhits2
  - There are some impossible values on some of the APA6-071720 hit plots as well - need to investigate the hit packets for these timestamps using the 33b conversion of the captures
- Beyond that there are some distributions which need more investigation:
  - hitFlags should all be 0
  - SummedADC plot shape - due to threshold?
  - Ideally we would know what slot/fibre number corresponds to each HF in each SLR to check the header values look correct
- We should still be able to use the first hit in each hit packet and the pedestal information to verify that we know what the TPG firmware was doing using python emulation

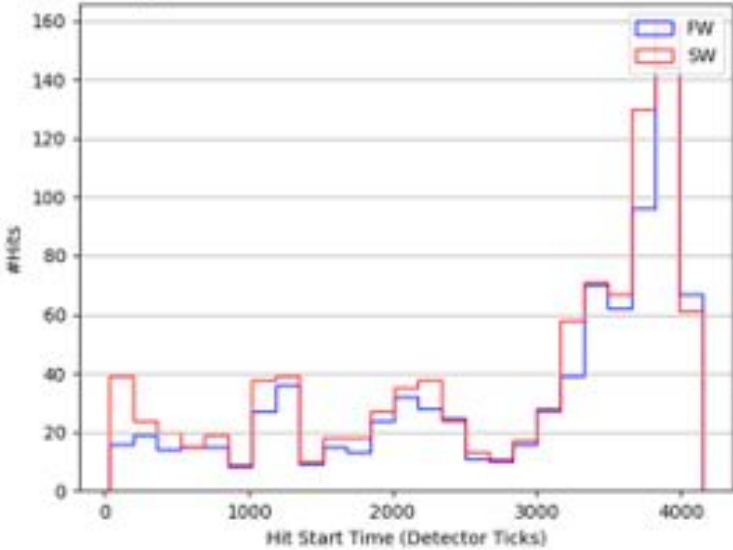
# Python TPG emulation

For the udaq tech review I extracted the hits and adcs which overlapped for the APA6-newhits capture.

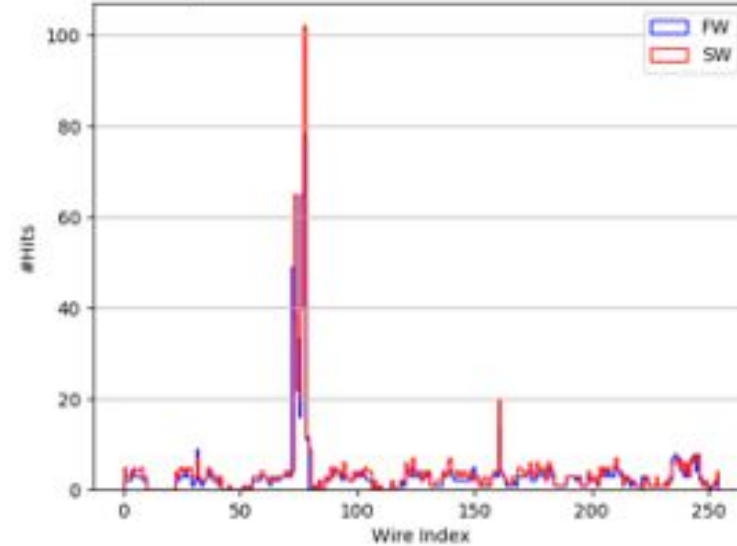
- Scripts required:
  - To extract adc data: `extract_adcs_from_timestamp.cpp`
  - To convert hit binary captures to 33b: `hitBinaryMangler.py`
  - To run TPG on extracted adc data and compare to extracted hits: `hitComparison.py`
- This capture had no pedestal words, so set the initial median to the 1st adc sample on each wire and the accumulator to 0
- We got very similar distributions when histogramming the hit packet properties despite the rough comparison
- To verify that we understand the TPG firmware used in protoDUNE we are extending the python tpg to:
  - Be able to set the initial median and accumulator values from the hits
  - Be able to produce the same output hits as the firmware (bugs and all) so that we can verify what the bugs are and that we understand the behaviour of the fw

# Python TPG emulation

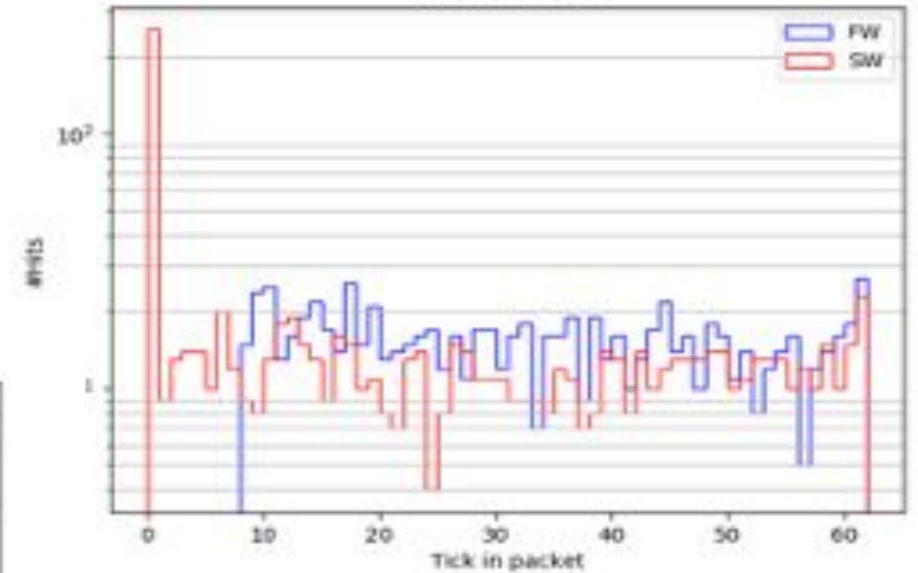
Comparison of Hit Start Times between SW and FW TPG



Comparison of Wire Index between FW and SW TPG

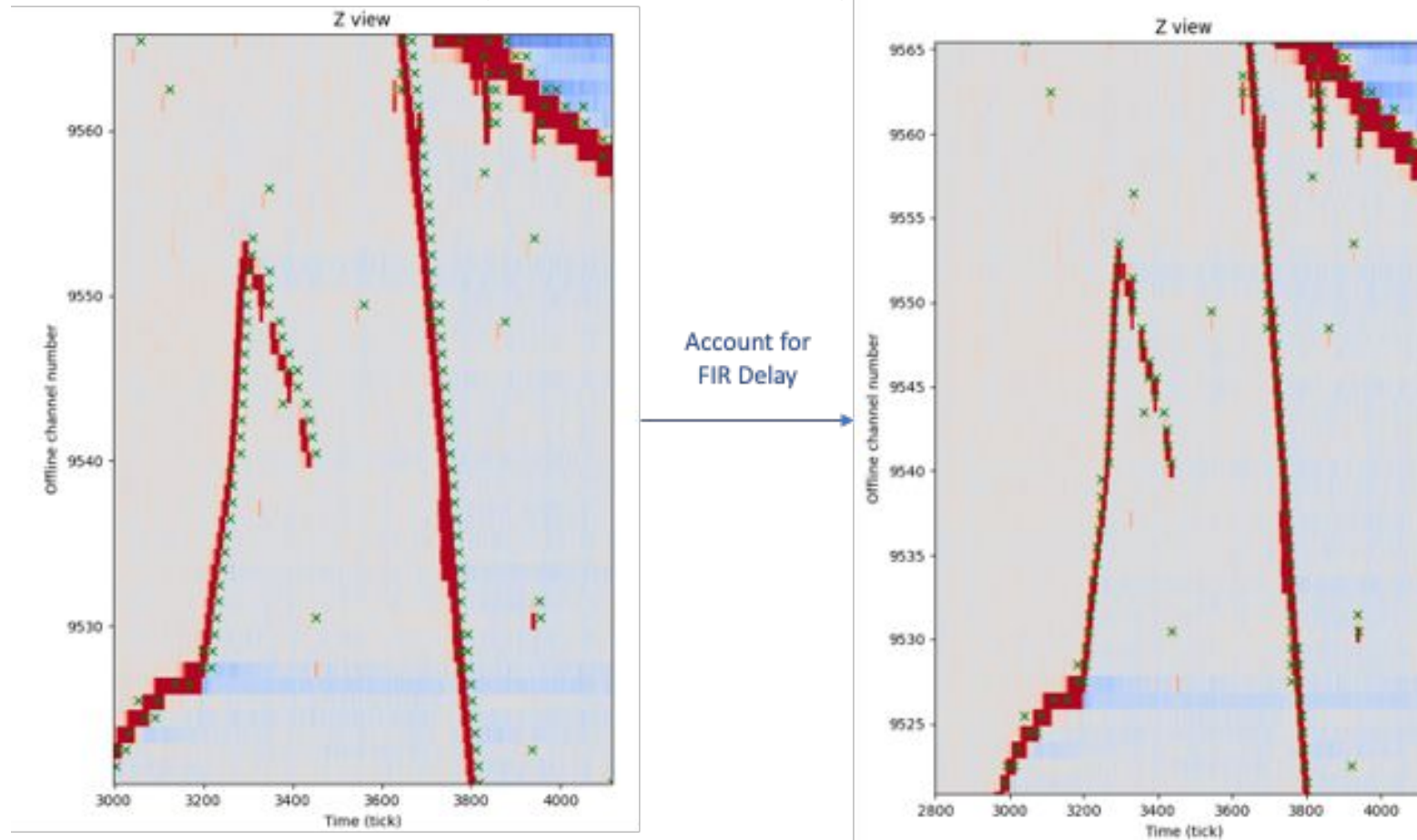


Hit Start Tick



# Event Displays From FW Hits

These event displays of the firmware hits overlaid on the ADCs also made it into Kostas' udaq tech review slides (APA6-newhits capture). Collection plane:



# Event Displays From FW Hits

These event displays of the firmware hits overlaid on the ADCs also made it into Kostas' udaq tech review slides (APA6-newhits capture). Induction plane:

