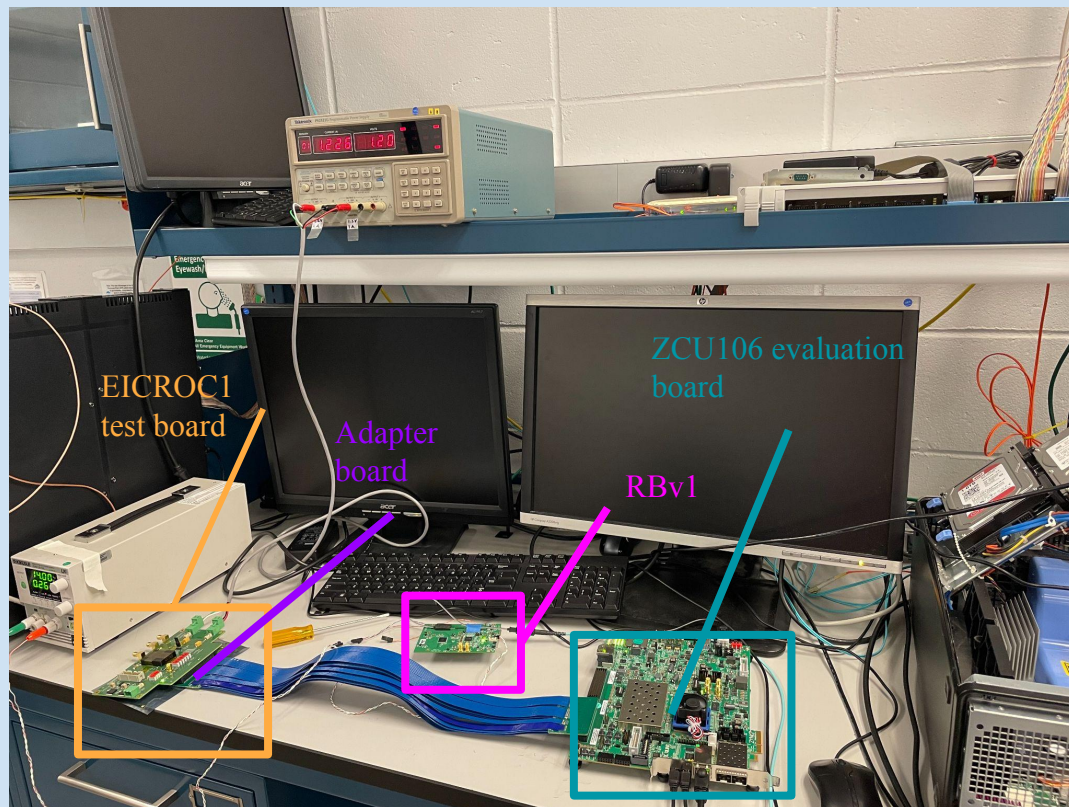
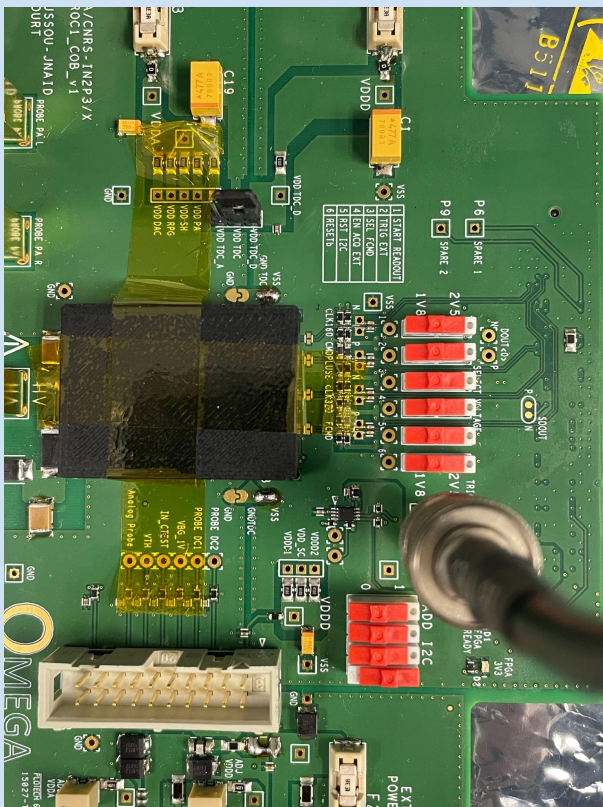


EICROC1 Summer Testing - 06/15/26-08/14/26

Rachel Kovach-Fuentes
Rice University

Setup at BNL



Pipeline

zcu106.C: program which talks to ASIC,
performs desired reads/writes to register

ana_zcu.C: program records “reply” from
ASIC in human-readable format

output_converter.py and eicroc1_plotter.cxx:
process output and plots the result

Automated scanners: [v_ref_scan.sh](#),
[pulse_scan.sh](#), [s_curve_scan.sh](#)

I am documenting all code for future use in
[EICROC_BNL_Rice](#) repository

25ns
timestep

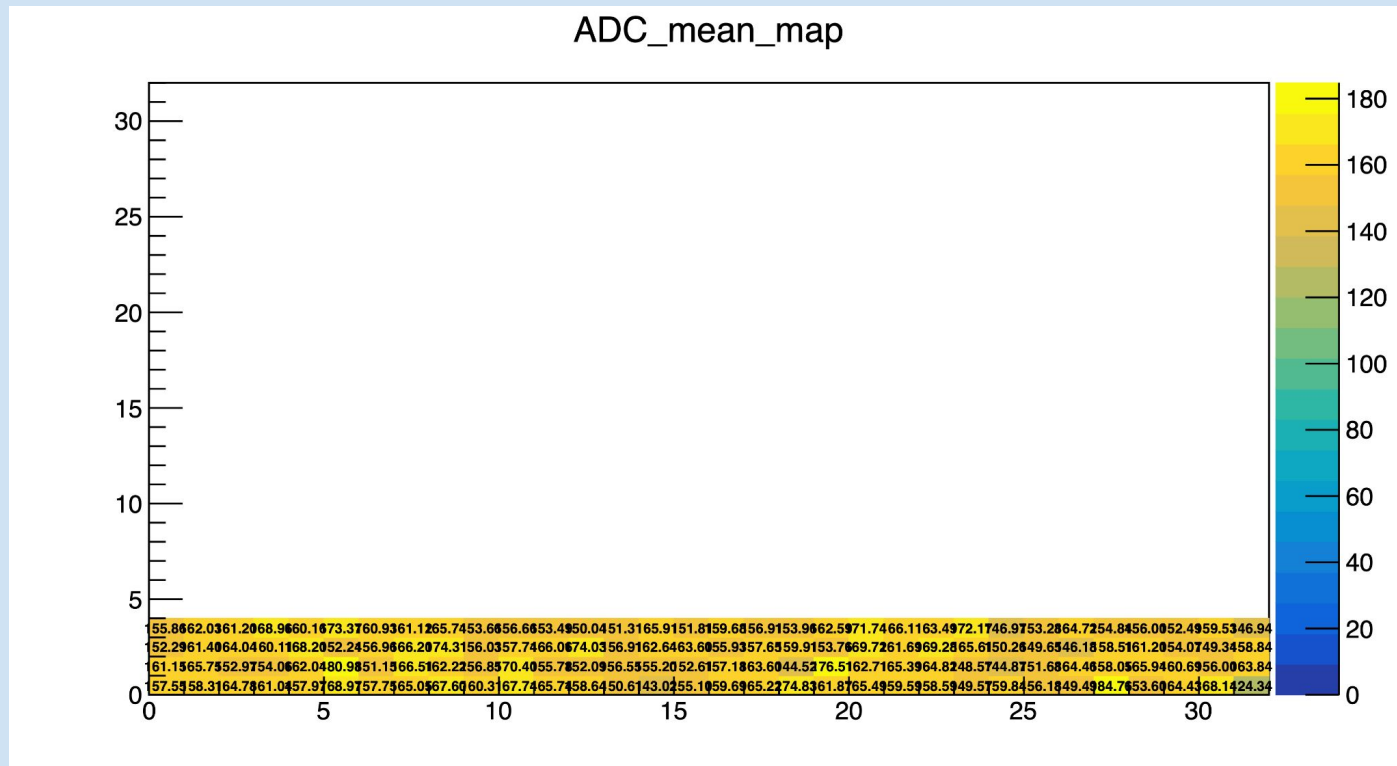
```
I2C read 0x4000 = 0xE3
I2C read 0x4001 = 0x80
I2C read 0x4002 = 0x3C
I2C read 0x4003 = 0x00
I2C read 0x4004 = 0x01
I2C read 0x4005 = 0x20
I2C read 0x4006 = 0xA0
I2C read 0x4007 = 0x40
```

1	event	column	row	timebin	adc	tdc	hit
2	0,0,0,0	161	0,0				
3	0,0,0,1	162	0,0				
4	0,0,0,2	162	0,0				
5	0,0,0,3	161	0,0				
6	0,0,0,4	160	0,0				
7	0,0,0,5	159	0,0				
8	0,0,0,6	158	0,0				
9	0,0,0,7	158	0,0				
10	0,0,1,0	161	0,0				
11	0,0,1,1	162	0,0				

A look at ADC mean for noise for all active pixels (currently, only 4 rows are active)

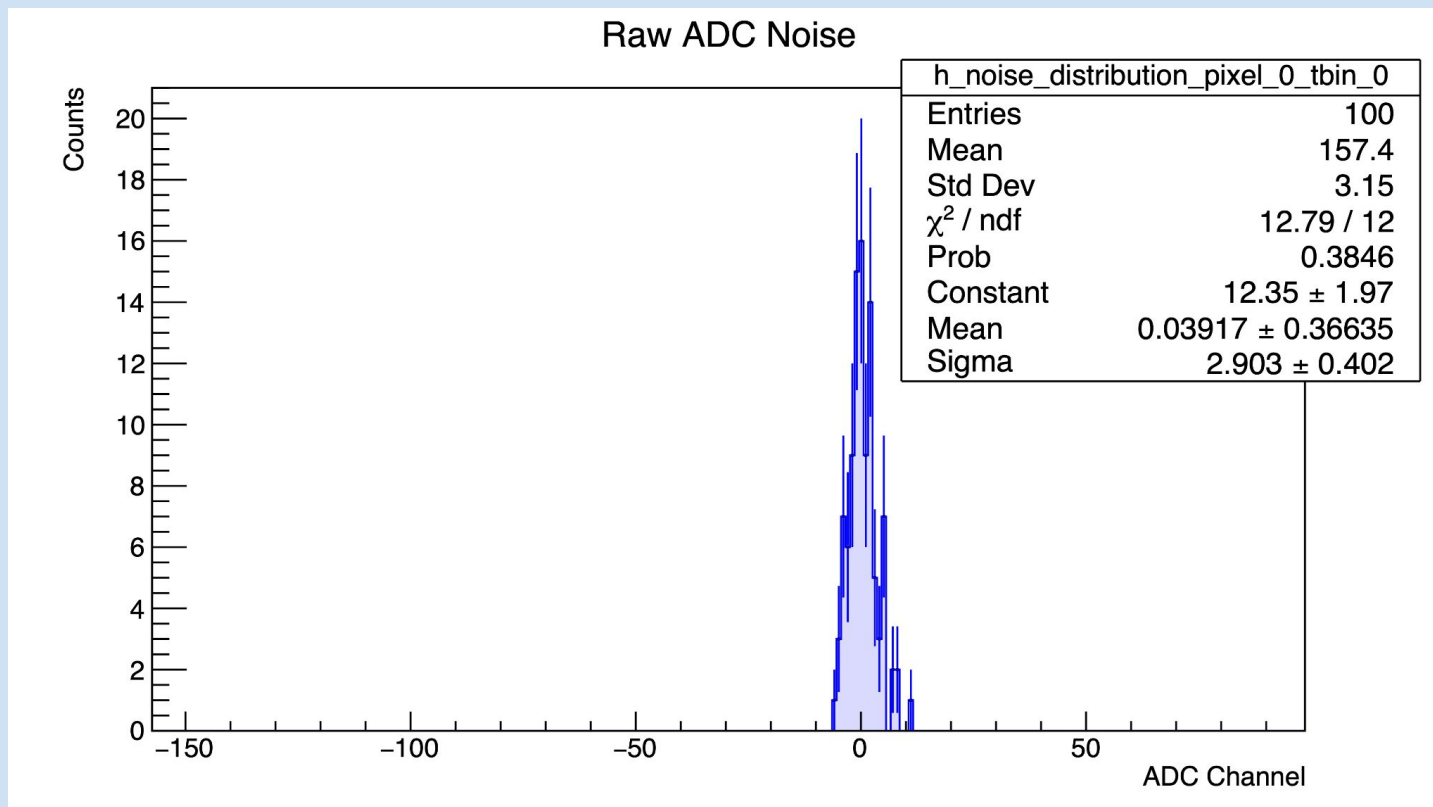
Shows the mean taken across 100 events, all 8 timebins – only one outlier appears

Plotting macro allows further slicing by tbin, region of chip, etc.

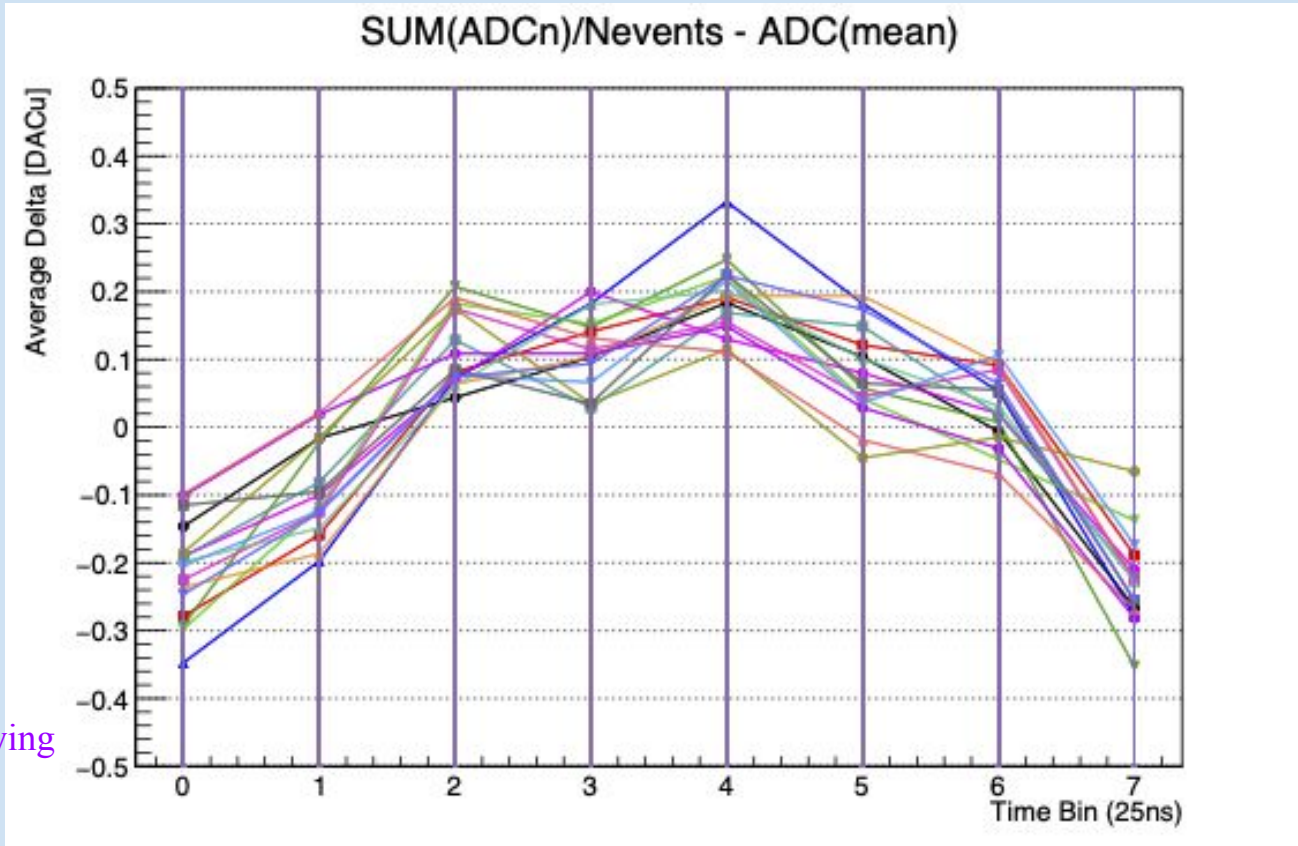


Approach for a single pixel and tbin (pixel 0, tbin 0)

1. Plot raw ADC in time bin
2. Extract mean for each time bin
3. Plot of mean ADC vs. time bin
4. Shift mean to 0 to “normalize” all of the pixels and measure their relative drift



Result 1: Baseline behavior of 16 pixels of interest, mean shifted to 0



There is an underlying parabolic structure across all channels

Tuning V_ref_corr to shift ADC mean downwards

This picture presents the different slow control parameters that allow to enable a certain signal on the probes of the test PCB.

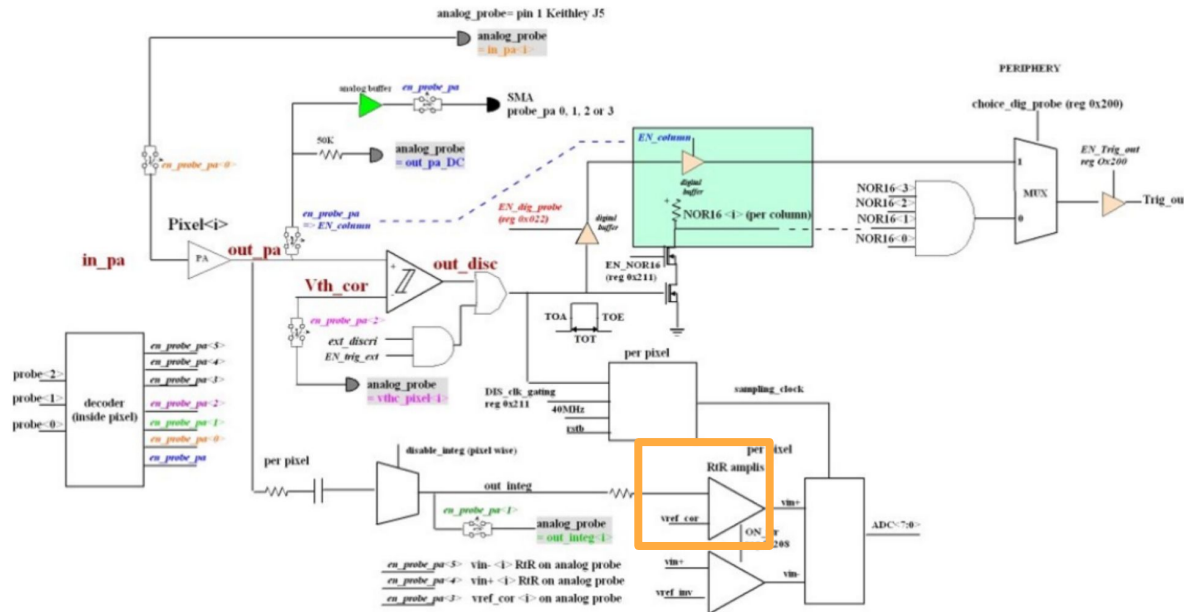
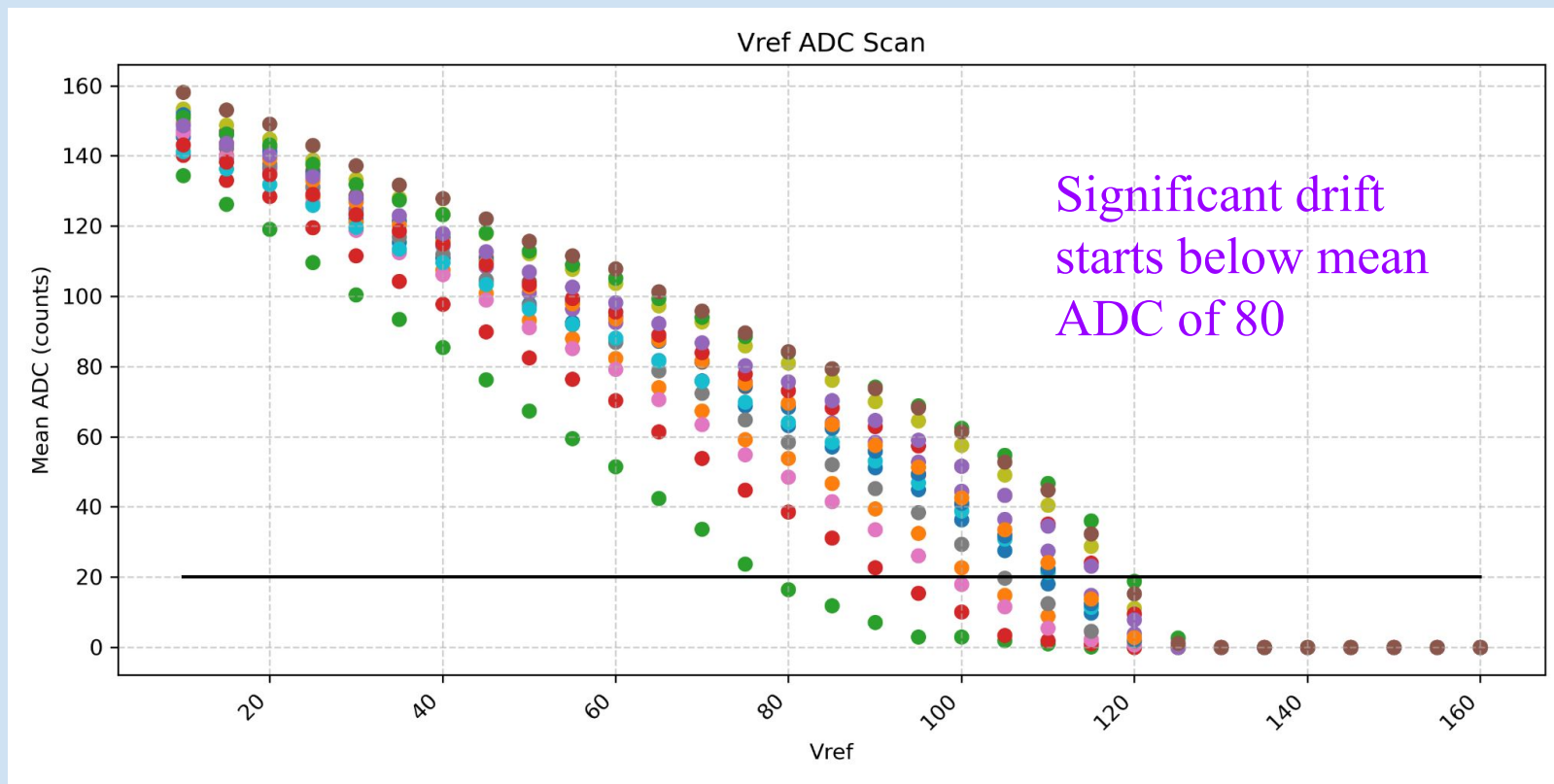


Figure B.1: EICROC0 pixel probe configuration presentation

Result 2: Vref scan results over 16 pixels of interest



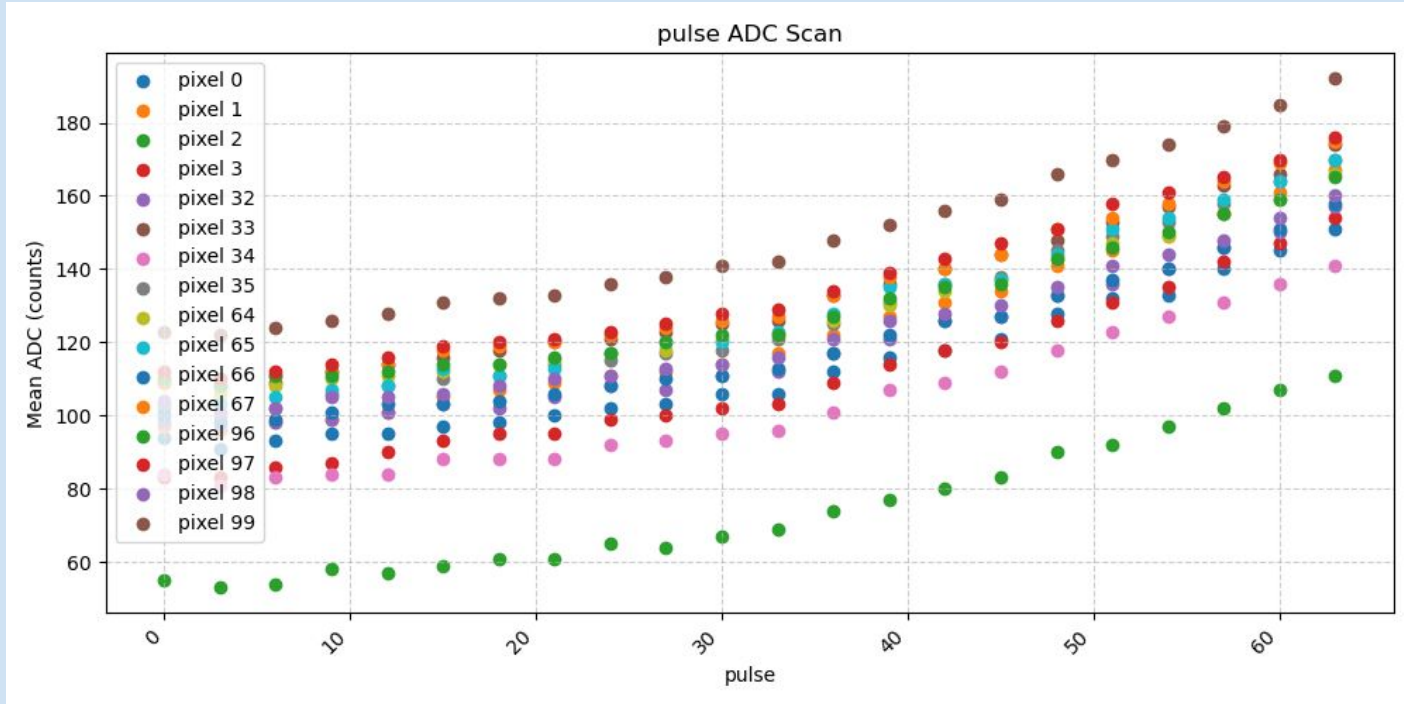
Result 3: Pulse scan results over 16 pixels of interest - 6b internal pulser

EICROC0:

- 100 fF capacitor
- ~0.5fC to 25fC injected

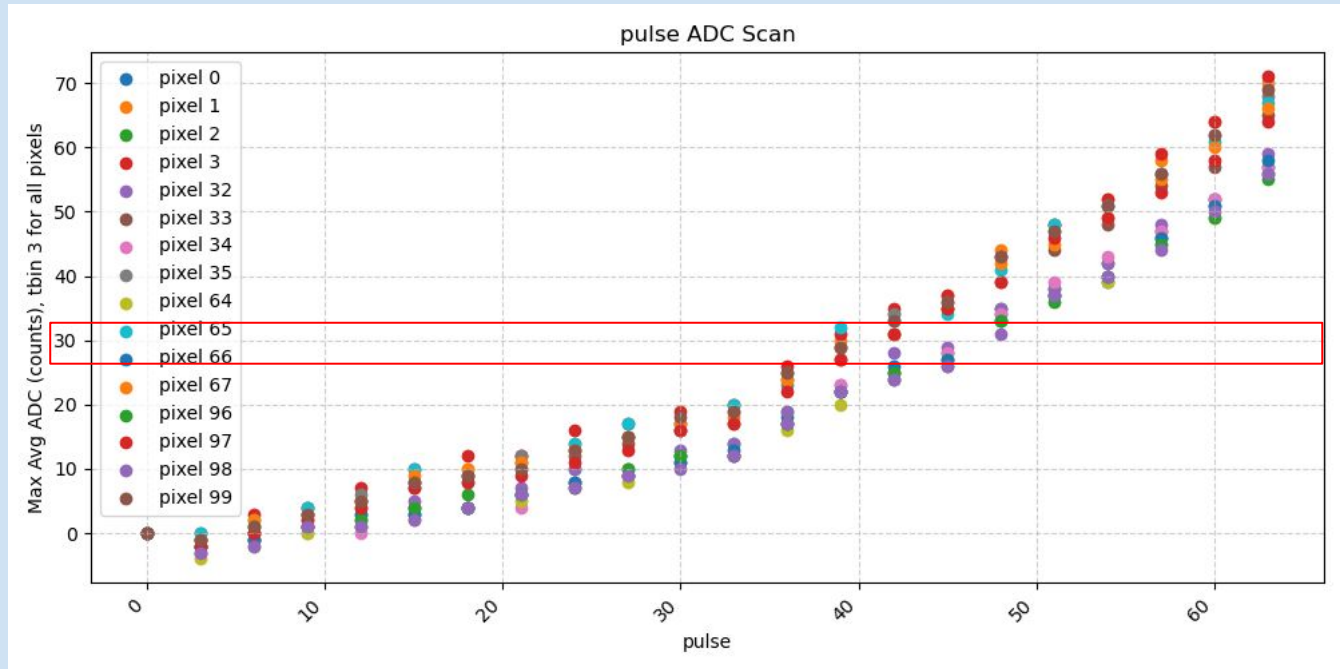
EICROC1:

- 202fF
- ~1fC to 56 fC injected



Due to addressing issues, I can't write to the individual v_{ref} registers, so instead I artificially shift the first point to 0 when plotting to see how closely the curves overlap

~25 fC: limit of
range EICROC0
could sweep



Currently trying to generate S-curves to characterize the TDC

Characterize discriminator efficiency ($\# \text{ good events} / \# \text{ events total}$) vs threshold for all pixels on chip

→ Evaluate noise floor per channel

→ Ideally, v_{thresh} can be tuned so that all S-curves overlap as closely as possible

Still unsure why the CMD and ACQ seem to not align with preamp signal

CMD

ACQ window

Preamp output 2

Is this the real preamp output or just noise?



We are having problems getting the v_thresh to write to the EICROC1 register

```
Reg 8: 0x4007 = [0b01000000)] -- 0x40
Reg 9: 0x4008 = [0b11110000)] -- 0xF0
Reg 10: 0x4009 = [0b10000000)] -- 0x80
Reg 11: 0x400A = [0b00000100)] -- 0x04
Reg 12: 0x400B = [0b01100011)] -- 0x63
Reg 13: 0x400C = [0b00000000)] -- 0x00
Reg 14: 0x400D = [0b01010011)] -- 0x53
Reg 15: 0x400E = [0b10100111)] -- 0xA7
Reg 16: 0x400F = [0b00110011)] -- 0x33
Reg 17: 0x4010 = [0b01001001)] -- 0x49
```

Initially writing to the registers seems to work

But when the values are read back after, they do not match what was written

```
global10 read check - 0x14
global11 read check - 0x8
```

Still debugging this

Summary

- Noise behavior of EICROC1 analyzed - underlying parabola/oscillation found
- Scan over Vref values completed for selected pixels and pedestal values determined
- Pulse scan over 6b register completed, linearity seen above EICROC0's limit
- S-curves in progress, still debugging register 10 and 11 read/write issue

Backup

Split by timebin for 1 pixel (pixel 0), raw values

Distributions shift slightly over time

