VERSO DAQ & VMM Calibration

Daniel Joseph Antrim
daniel.joseph.antrim@cern.ch

NSW Electronix Weekly
Friday October 6th 2017

UNIVERSITY of CALIFORNIA
IRVINE

Outline

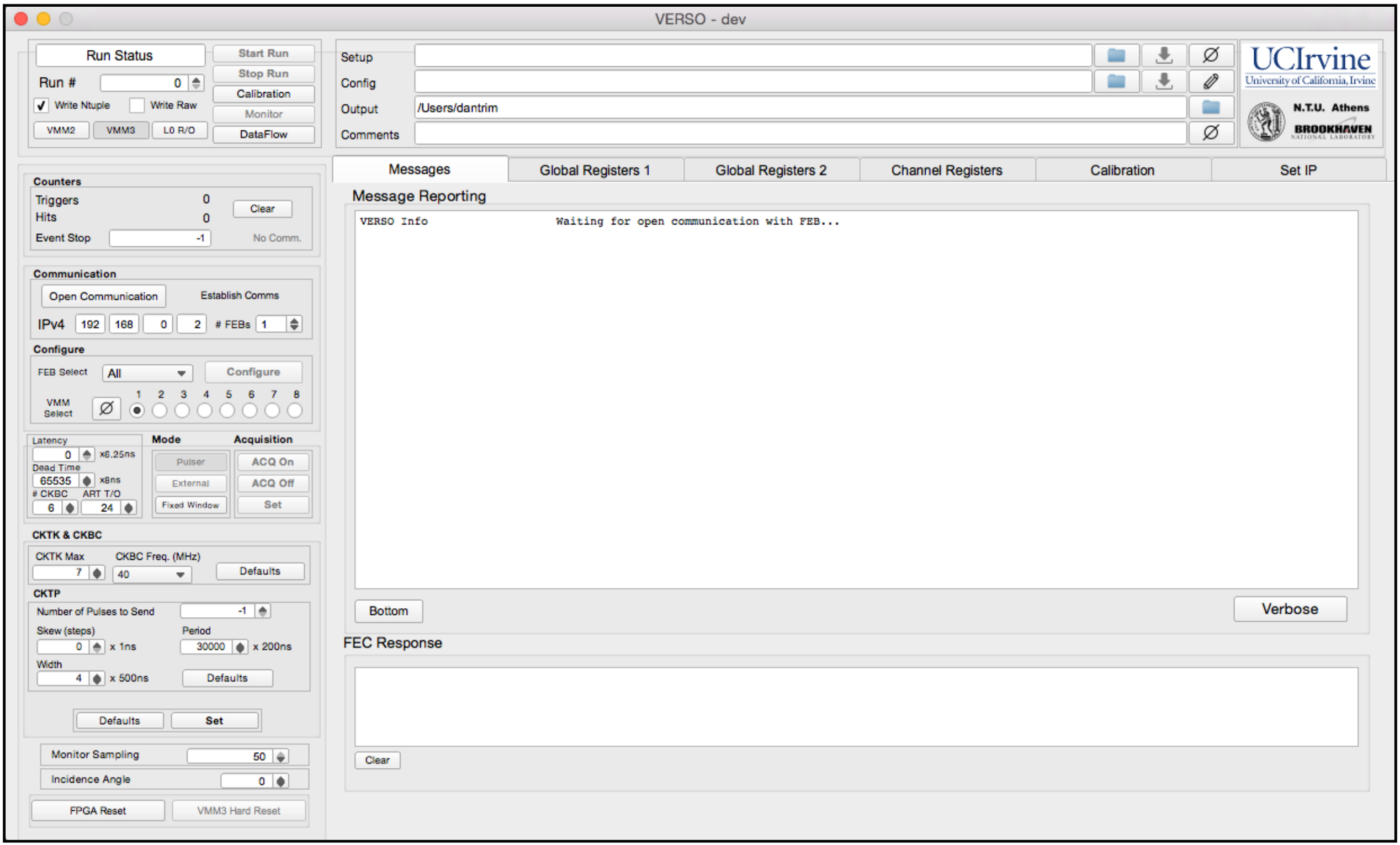
- VERSO Overview
- Calibration Overview
  - Procedures

VERSO Overview

- **VERSO**
  - **V**MM **E**thernet **R**eadout **SO**ftware
  - Central DAQ software for:
    - Configuring VMM-based front-end boards
      - *VMM2* and *VMM3* in continuos or L0-trigger mode
      - Arbitrary # of front-end boards
    - (Fast) buffered, UDP based readout and event building
    - Calibration
      - Calibration runs implementing xADC-based and pulser-based calibration loops
    - Monitoring capabilities
- It is a *graphical* interface
- Developed alongside the baseline NSW electronics NTUA/BNL firmware
- Hosted on NSWElectronics GitLab under vmm_readout_software/*

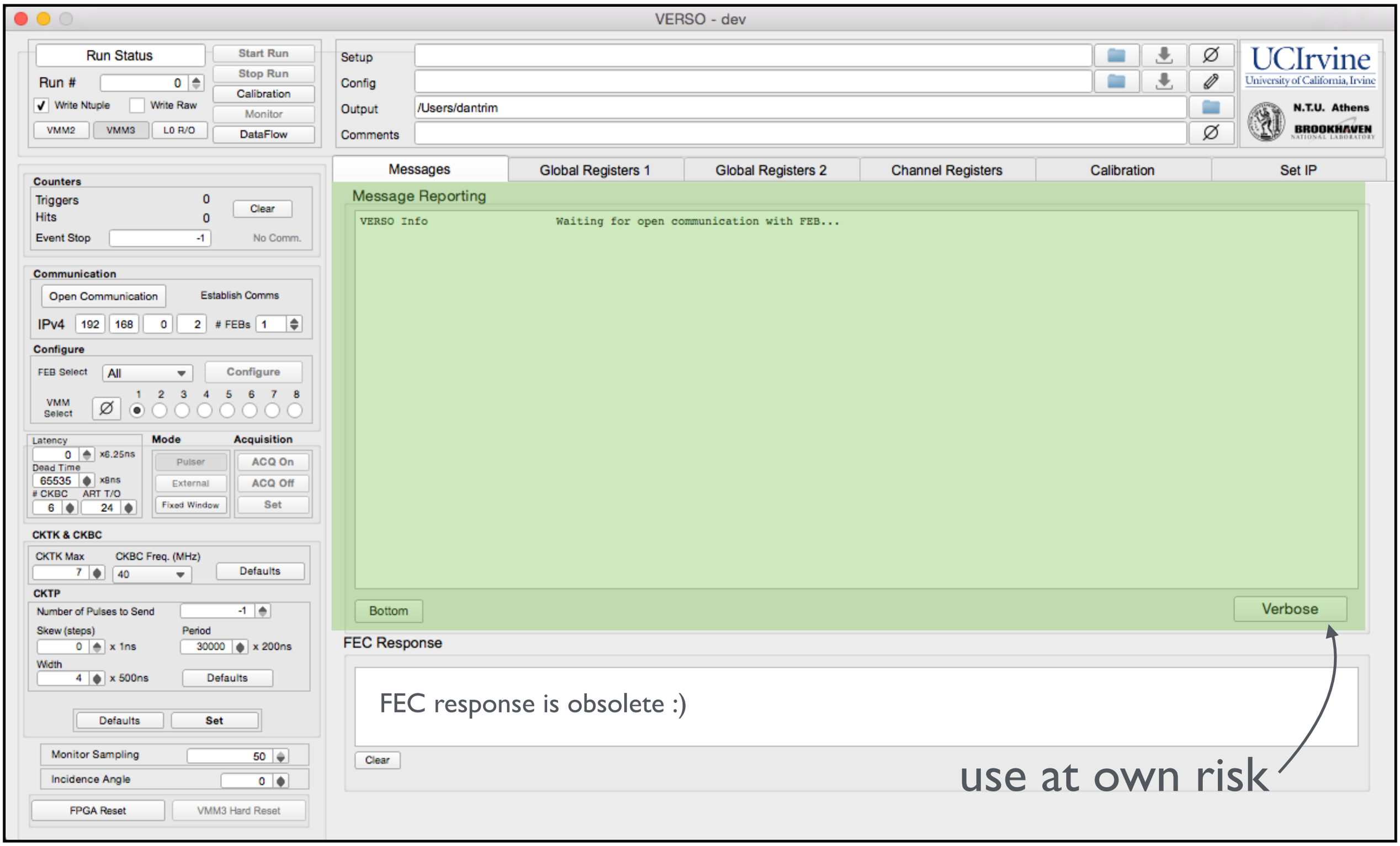
* will soon have the verso/ repo up to date

VERSO Overview



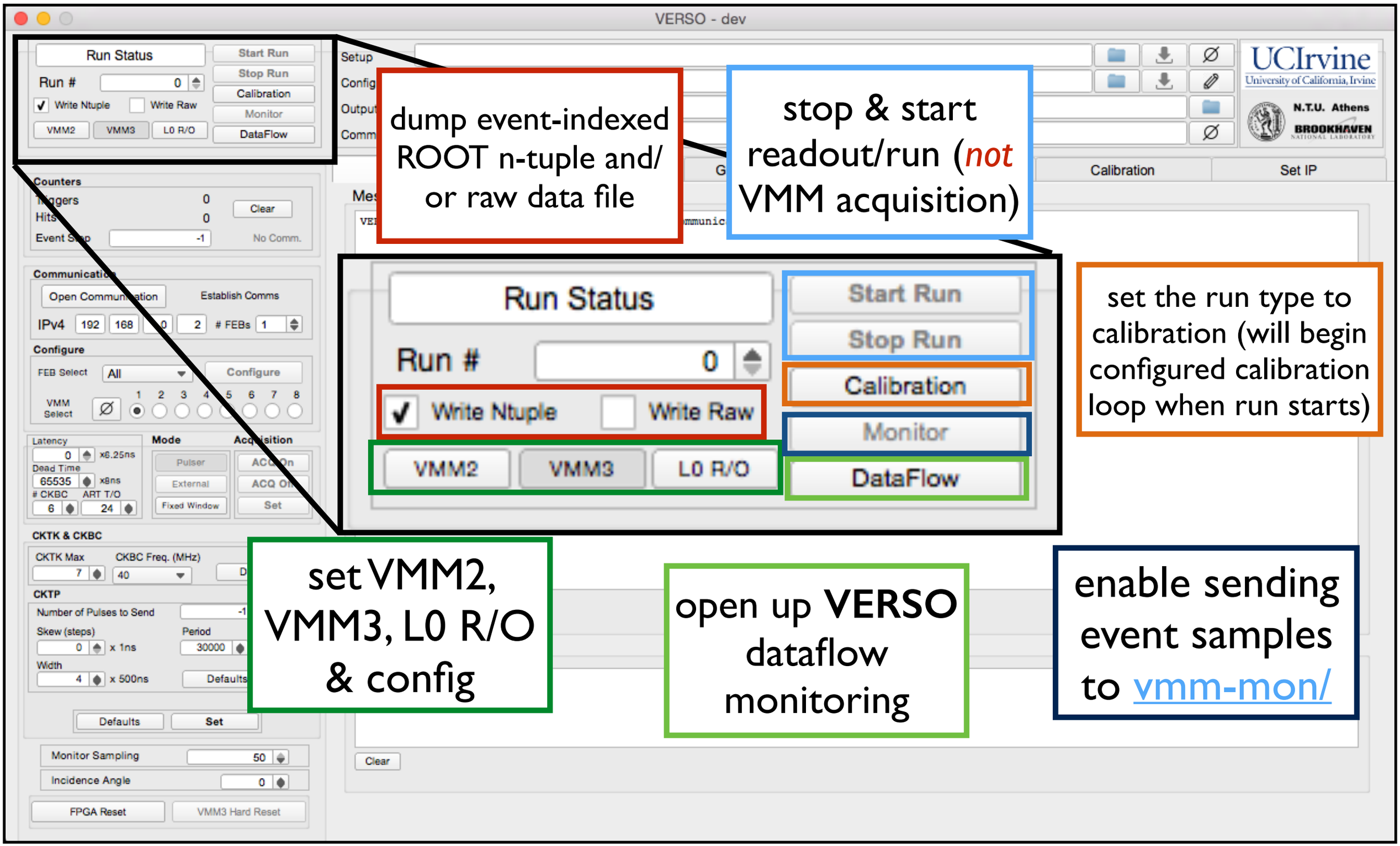
this is what it looks like when you start it up

VERSO Overview



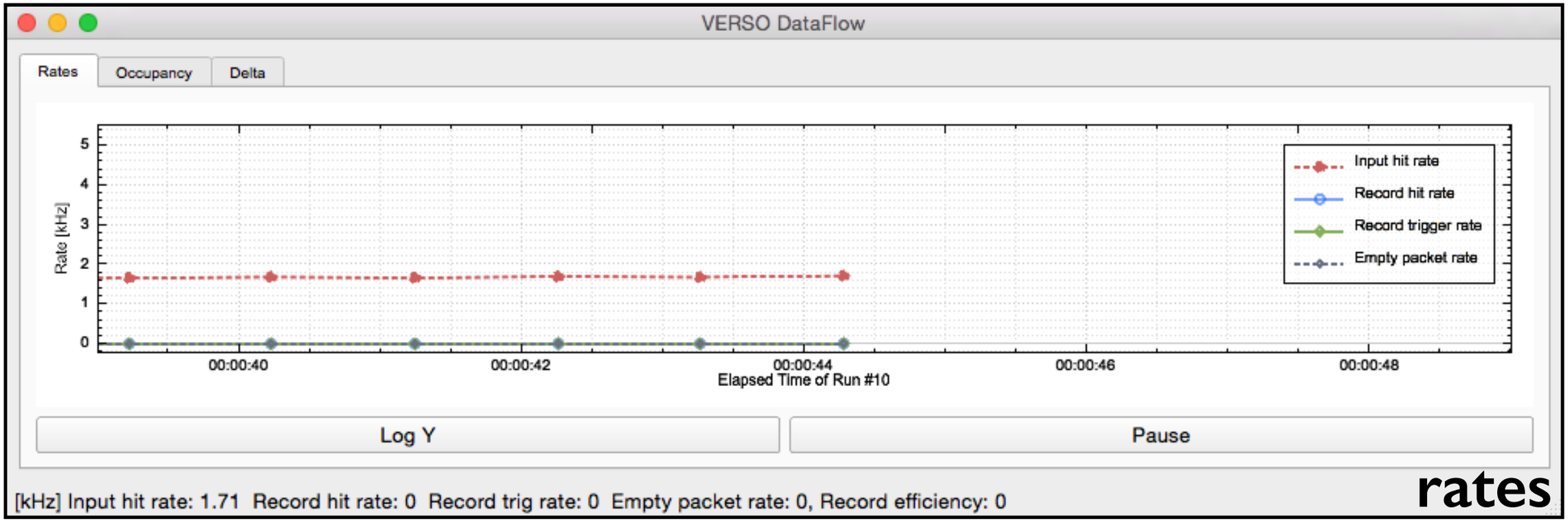
useful messages appear here

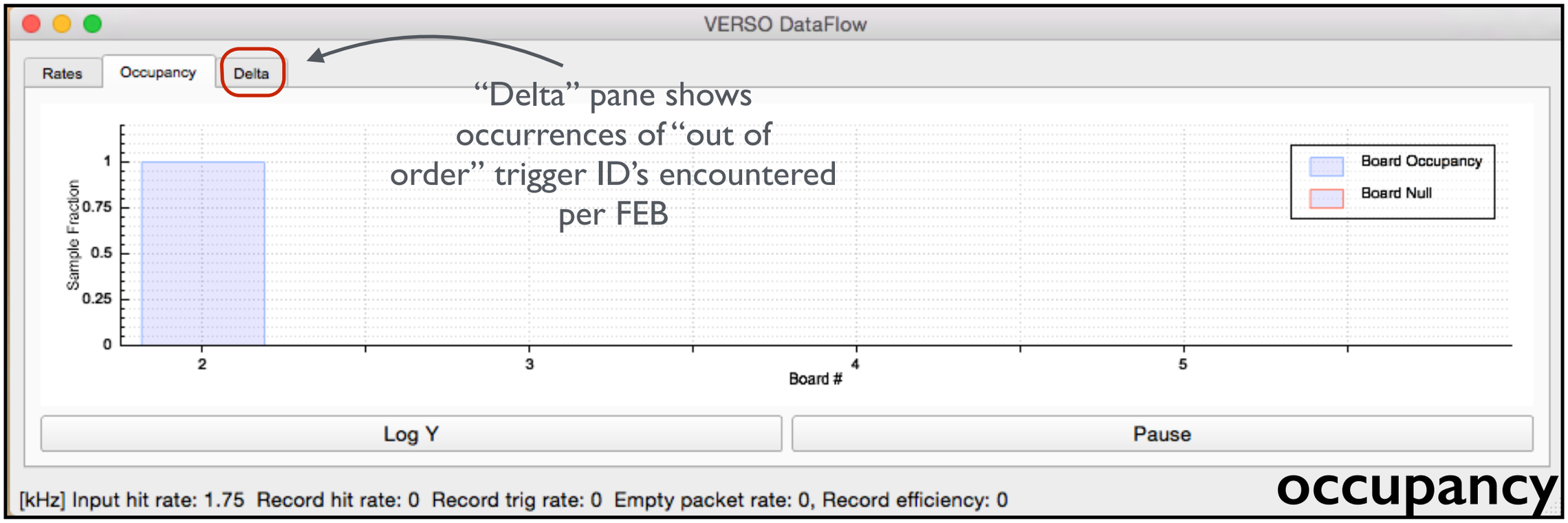
VERSO Overview



run control

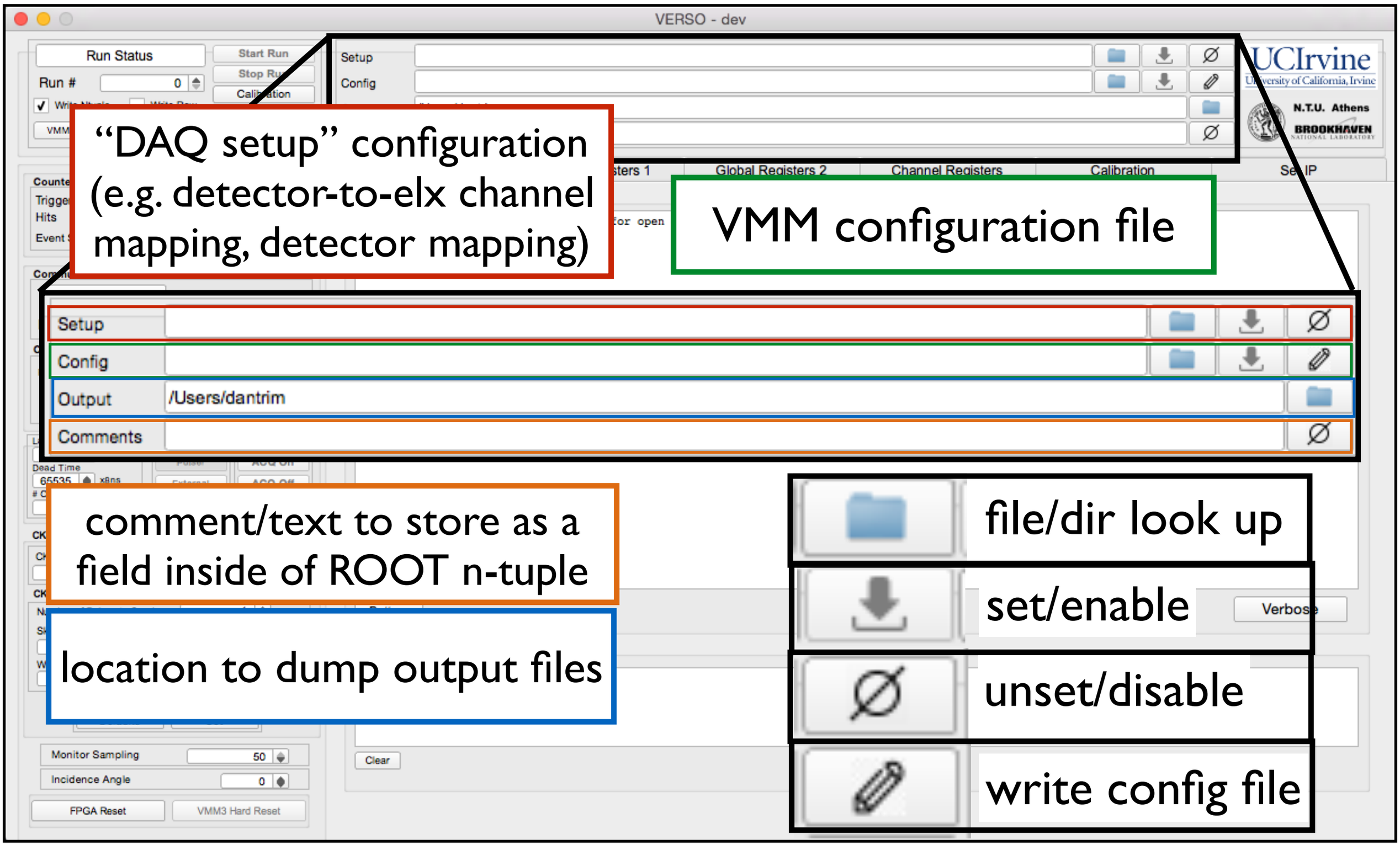
VERSO Overview





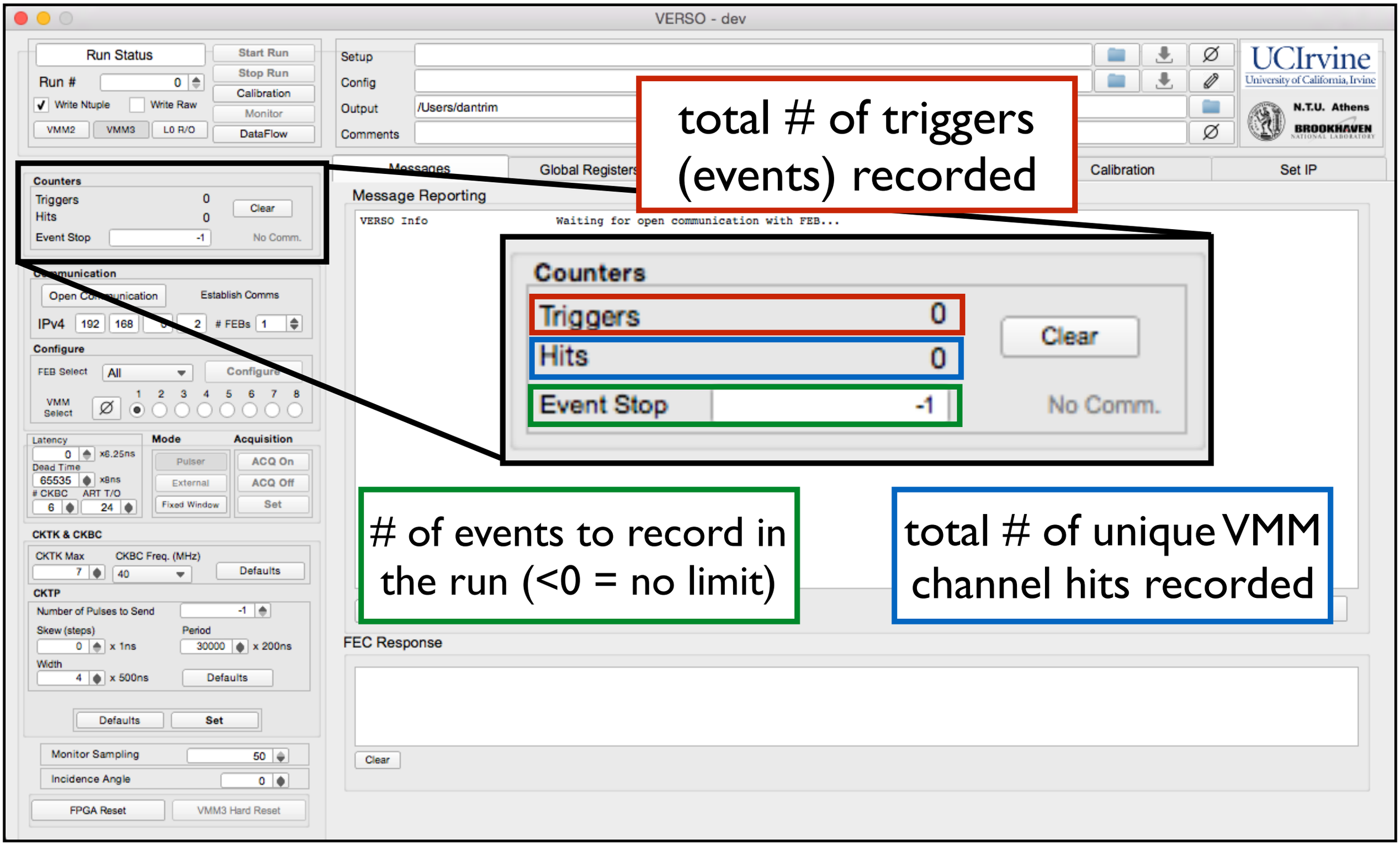
dataflow window

VERSO Overview



various top-level configuration parameters

VERSO Overview



total # of triggers (events) recorded

# of events to record in the run (<0 = no limit)

total # of unique VMM channel hits recorded

sum totals

VERSO Overview

setup UDP paths and test (ping) FEB connection

base IP address (address of *first* FEB)

# of FEBs to talk to

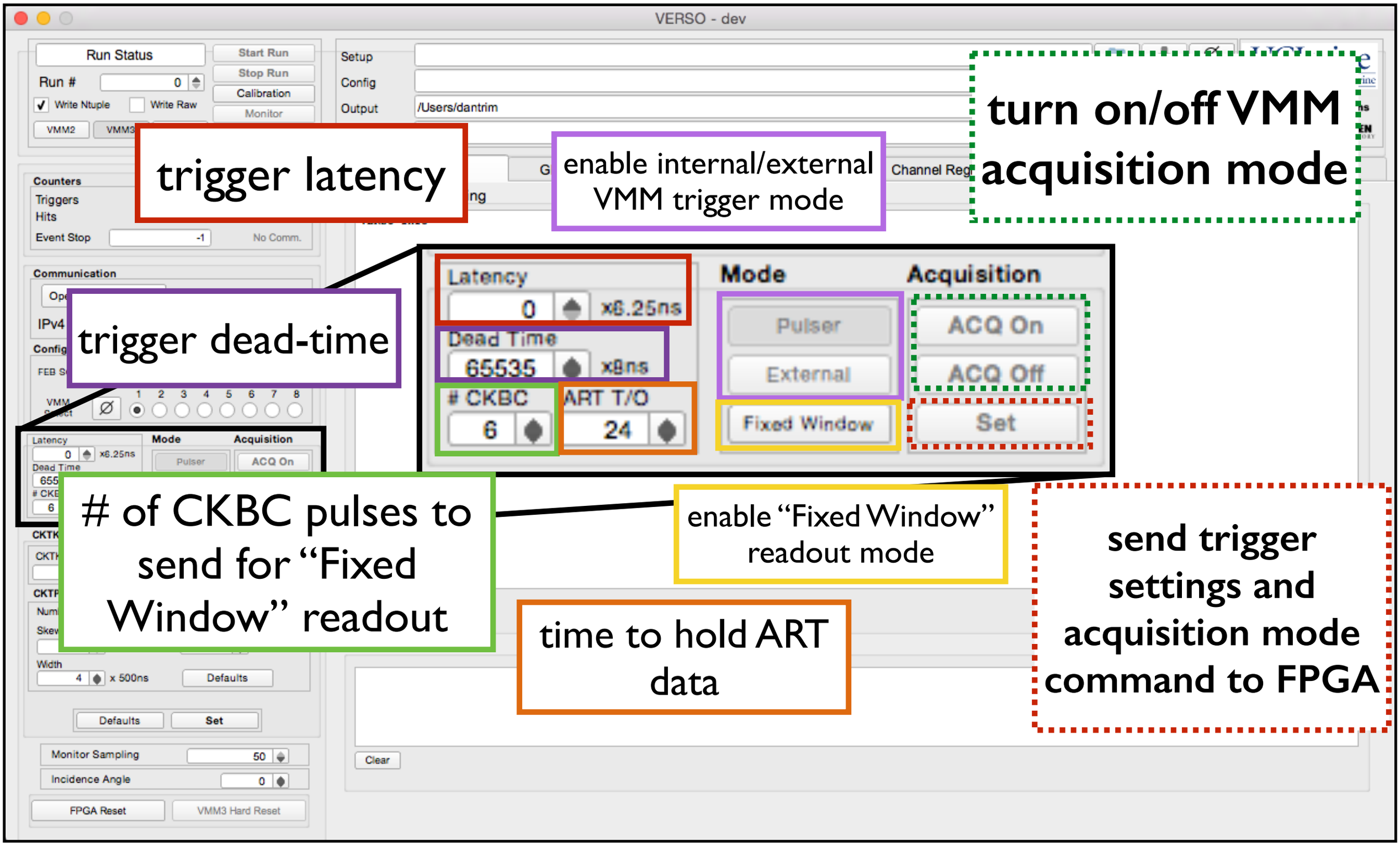
FEB # to send VMM SPI to (also sets to which FEB we send all other commands)

VMM id on selected FEB to configure

send VMM SPI to selected FEB + VMMs

board and VMM selection + configuration

VERSO Overview



trigger settings and acquisition mode

VERSO Overview

CKTK and CKBC configuration (40, 20,&10 MHz CKBC possible)

CKTP (test pulse) configuration

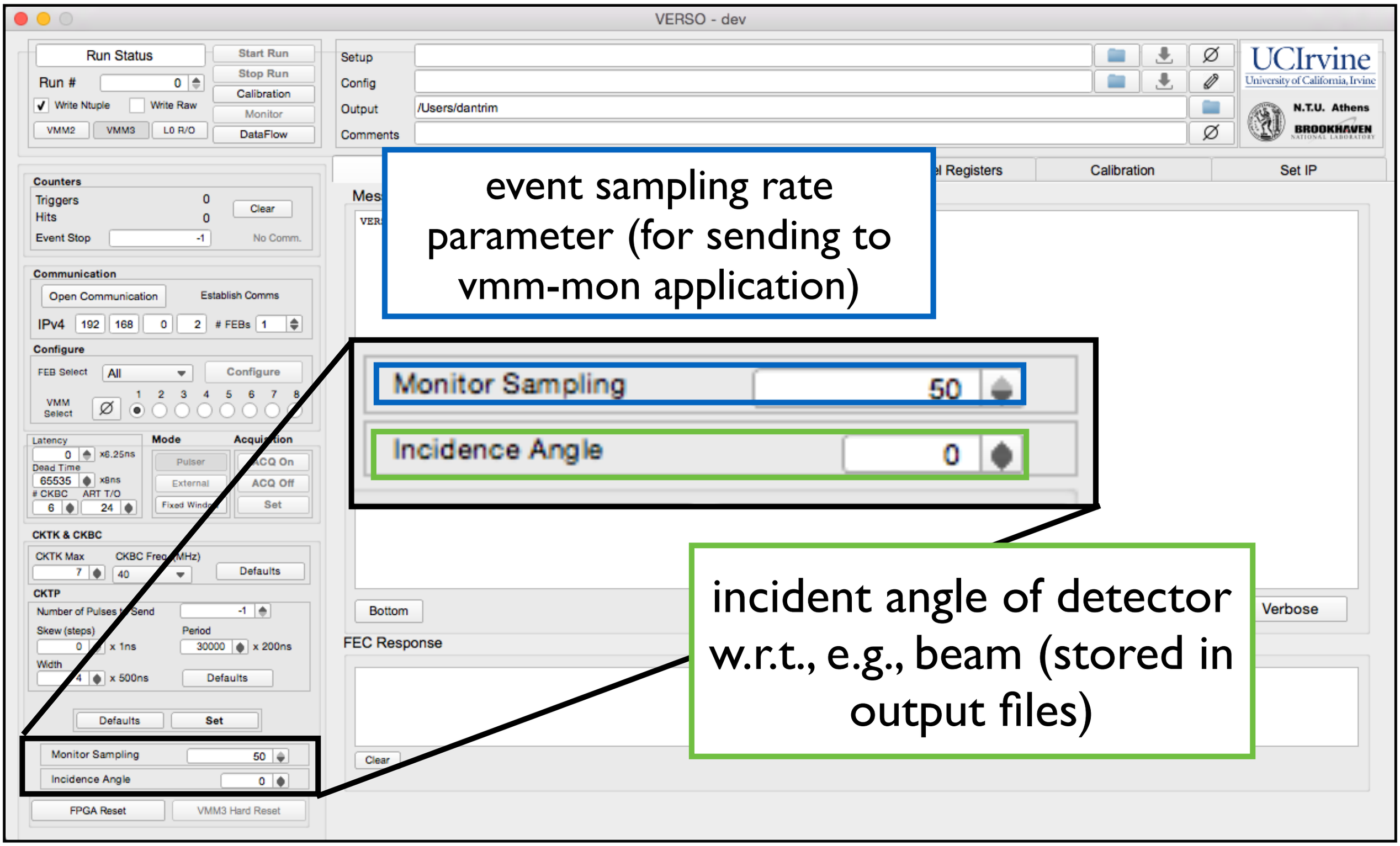
send fixed # of pulses (<0 for no-limit) (value stored in output file)

send CKTK, CKBC, and CKTP configuration to the FEB

CKTP skew w.r.t. CKBC (1ns steps for 40 ns clock, 6.25 ns steps otherwise)

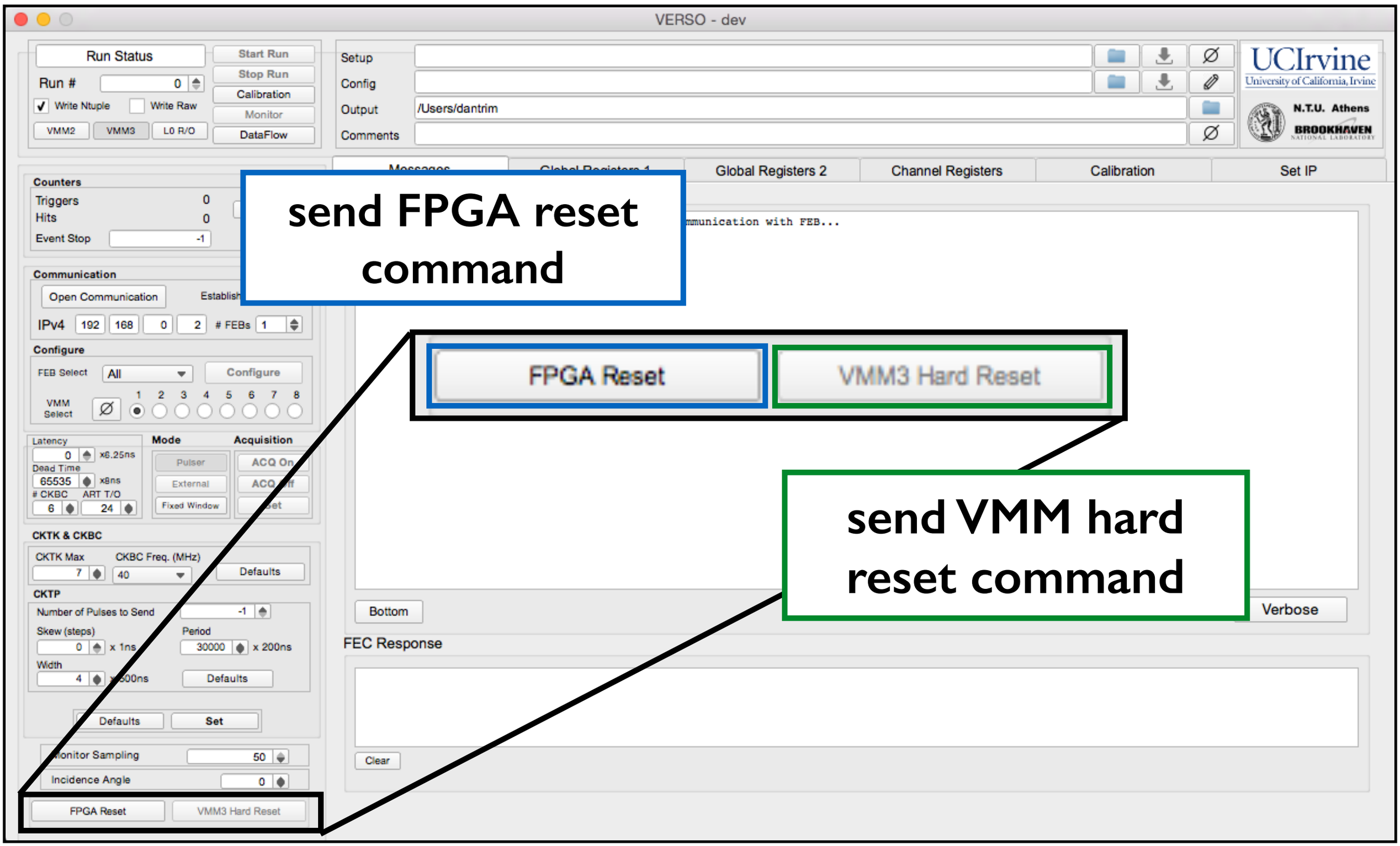
FPGA clocks configuration

VERSO Overview



miscellany

VERSO Overview



resets

VERSO Overview

associated bit names in-line

VERSO - dev

VMM2 | VMM3 | L0 R/O | DataFlow

trim

Comments

UCIrvine University of California, Irvine

N.T.U. Athens

BROOKHAVEN NATIONAL LABORATORY

Counters

Triggers 0 Clear

Hits 0

Event Stop -1 No Comm.

Communication

Open Communication | Establish Comms

IPv4 192 168 0 2 # FEBs 1

Configure

FEB Select All | Configure

VMM Select 1 2 3 4 5 6 7 8

Latency 0 x6.25ns

Dead Time 65535 x8ns

CKBC 6 | ART T/O 24

Mode: Pulser | External | Fixed Window

Acquisition: ACQ On | ACQ Off | Set

CKTK & CKBC

CKTK Max 7 | CKBC Freq. (MHz) 10 | Defaults

CKTP

Number of Pulses to Send -1

Skew (steps) 0 x 6.25ns | Period 30000 x 200ns

Width 4 x 500ns | Defaults

Defaults | Set

Monitor Sampling 50

Incidence Angle 0

FPGA Reset | VMM3 Hard Reset

Messages | Global Registers 1 | Global Registers 2 | Channel Registers | Calibration | Set IP

Global Configuration Registers

Test Pulse DAC [sdp10]	300	Ch. Polarity [sp]	Negative	Ch. Gain [sg]	3.0 mV/fC	Peak Time [st]	200 ns
Threshold DAC [sdt10]	300	Neighbor Logic [sng]	Disabled	TAC Slope Adj. [stc]	60 ns	Sub-hysteresis [ssh]	Disabled
Leakage Curr. [slg]	Enabled	An. Tristates [sdrv]	Disabled	Dyn. Discharge [sfm]	Disabled	Dis. at Peak [sdp]	Disabled

Monitor Multiplexing

Ch. Monitor [sm5-sm0]	0	Monitor Mode [scmx]	Enabled (Channel)	Route to PDO Output [sbmx]	Disabled

Analog Buffer

TDO analog output [sbft]	Disabled	PDO analog output [sbfp]	Disabled	MO analog output [sbfm]	Enabled

ADC Enable

6-bit mode [s6b]	Disabled	8-bit mode [s8b]	Enabled	Hi. Res. (10-bit/8-bit) [s10b]	Enabled

Direct Timing Output

Enable [sttt]	Disabled	Mode	TtP (Thresh-to-Pk)	[stpp]	0	[stot]	0

Address in Real Time (ART)

Enable ART [sfa]	Disabled	Detect Mode [sfam]	Timing at Threshold

ADC Conversion

10-bit time [sc10b]	200 ns	8-bit time [sc8b]	100 ns	6-bit time [sc6b]	Low

Dual Clocks Enable

Data [sdcks]	Disabled	ART [sdcka]	Disabled	6-bit [sdck6b]	Disabled

VMM SPI configuration — global registers I panel

VERSO Overview

associated bit names in-line

VERSO - dev

VMM2 | VMM3 | L0 R/O | DataFlow

Comments

UCIrvine University of California, Irvine — N.T.U. Athens — BROOKHAVEN NATIONAL LABORATORY

Messages | Global Registers 1 | Global Registers 2 | Channel Registers | Calibration | Set IP

Counters
Triggers 0 Clear
Hits 0
Event Stop -1 No Comm.

Communication
Open Communication — Establish Comms
IPv4 192 168 0 2 # FEBs 1

Configure
FEB Select: All — Configure
VMM Select: 1 2 3 4 5 6 7 8

Latency 0 x6.25ns
Dead Time 65535 x8ns
CKBC 6 ART T/O 24
Mode: Pulser / External / Fixed Window
Acquisition: ACQ On / ACQ Off / Set

CKTK & CKBC
CKTK Max 7 — CKBC Freq. (MHz) 10 — Defaults

CKTP
Number of Pulses to Send -1
Skew (steps) 0 x 6.25ns — Period 30000 x 200ns
Width 4 x 500ns — Defaults

Defaults | Set

Monitor Sampling 50
Incidence Angle 0

FPGA Reset | VMM3 Hard Reset

Global Configuration Registers

Direct Out I/Os **[slvs]**	Disabled	ART flag sync. **[ssart]**	Disabled	Skip Channels 16-47 **[s32]**	Disabled
Tail cancellation **[stlc]**	Disabled	Fast recovery **[srec]**	Disabled	Bipolar shape **[sbip]**	Disabled
Auto-reset **[stcr]**	Disabled	Reset at 6b compl. **[sfrst]**	Disabled	Time ramp at thresh. **[srat]**	Disabled

100 Ohm SLVS Termination

On CKBC **[slvsbc]**	Disabled	On CKTP **[slvstp]**	Disabled	On CKTK **[slvstk]**	Disabled
On CKDT **[slvsdt]**	Disabled	On CKART **[slvsart]**	Disabled	On CKTKI **[slvstki]**	Disabled
On CKENA **[slvsena]**	Disabled	On CK6b **[slvs6b]**	Disabled		

L0

L0 core **[sL0ena]**	Disabled (Reset)	Mixed signals in L0 **[sL0enaV]**	Disabled	L0 BC offset **[l0offset]** (0-4095)	4060
Ch. tagging BC offset **[offset]** (0-4095)	0	Ch. tagging BC rollover **[rollover]** (0-4095)	4095	Trig. window size **[window]** (0-7)	7
Max hits per L0 **[truncate]** (0-63)	63	# L0 to skip on overflow **[nskip]** (0-127)	0	Clocks w/ L0 disabled **[sL0cktest]**	Disabled
Invert BCCLK **[sL0ckinv]**	Disabled	Invert DCK **[sL0dckinv]**	Disabled	BCID skip **[nskipm]**	Disabled

Configuration Reset

Hard Reset **[reset]** Disabled

VMM SPI configuration — global registers II panel

VERSO Overview

associated bit names in-line

VERSO - dev

VMM2 | VMM3 | L0 R/O | DataFlow

Comments

UCIrvine University of California, Irvine — N.T.U. Athens — BROOKHAVEN NATIONAL LABORATORY

Messages | Global Registers 1 | Global Registers 2 | Channel Registers | Calibration | Set IP

Counters
Triggers 0
Hits 0
Event Stop -1
Clear | No Comm.

Communication
Open Communication | Establish Comms
IPv4 192 168 0 2 # FEBs 1

Configure
FEB Select: All | Configure
VMM Select: 1 2 3 4 5 6 7 8

Latency 0 x6.25ns
Dead Time 65535 x8ns
CKBC 6 | ART T/O 24

Mode: Pulser | External | Fixed Window
Acquisition: ACQ On | ACQ Off | Set

CKTK & CKBC
CKTK Max 7 | CKBC Freq. (MHz) 10 | Defaults

CKTP
Number of Pulses to Send -1
Skew (steps) 0 x 6.25ns | Period 30000 x 200ns
Width 4 x 500ns | Defaults

Defaults | Set

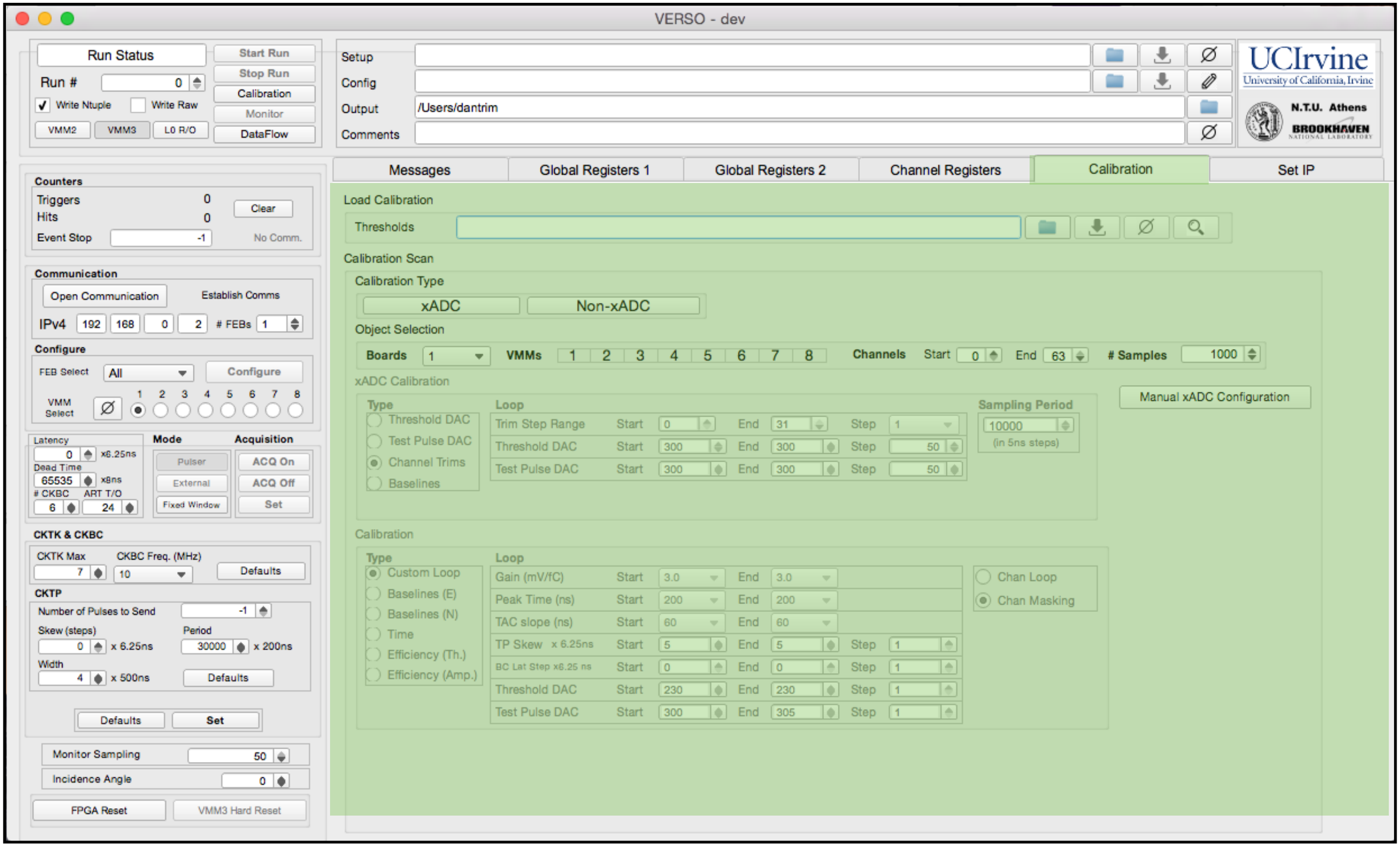
Monitor Sampling 50
Incidence Angle 0

FPGA Reset | VMM3 Hard Reset

CH	SC	SL	STH	ST	SM	0 mV	SMX	0	0	0	CH	SC	SL	STH	ST	SM	0 mV	SMX	0	0	0
0					■	0 mV		0	0	0	32				■		0 mV		0	0	0
1					■	0 mV		0	0	0	33				■		0 mV		0	0	0
2					■	0 mV		0	0	0	34				■		0 mV		0	0	0
3					■	0 mV		0	0	0	35				■		0 mV		0	0	0
4					■	0 mV		0	0	0	36				■		0 mV		0	0	0
5					■	0 mV		0	0	0	37				■		0 mV		0	0	0
6					■	0 mV		0	0	0	38				■		0 mV		0	0	0
7					■	0 mV		0	0	0	39				■		0 mV		0	0	0
8					■	0 mV		0	0	0	40				■		0 mV		0	0	0
9					■	0 mV		0	0	0	41				■		0 mV		0	0	0
10				■		0 mV		0	0	0	42				■		0 mV		0	0	0
11				■		0 mV		0	0	0	43				■		0 mV		0	0	0
12				■		0 mV		0	0	0	44				■		0 mV		0	0	0
13				■		0 mV		0	0	0	45				■		0 mV		0	0	0
14				■		0 mV		0	0	0	46				■		0 mV		0	0	0
15				■		0 mV		0	0	0	47				■		0 mV		0	0	0
16				■		0 mV		0	0	0	48				■		0 mV		0	0	0
17				■		0 mV		0	0	0	49				■		0 mV		0	0	0
18				■		0 mV		0	0	0	50				■		0 mV		0	0	0
19				■		0 mV		0	0	0	51				■		0 mV		0	0	0
20				■		0 mV		0	0	0	52				■		0 mV		0	0	0
21				■		0 mV		0	0	0	53				■		0 mV		0	0	0
22				■		0 mV		0	0	0	54				■		0 mV		0	0	0
23				■		0 mV		0	0	0	55				■		0 mV		0	0	0
24				■		0 mV		0	0	0	56				■		0 mV		0	0	0
25				■		0 mV		0	0	0	57				■		0 mV		0	0	0
26				■		0 mV		0	0	0	58				■		0 mV		0	0	0
27				■		0 mV		0	0	0	59				■		0 mV		0	0	0
28				■		0 mV		0	0	0	60				■		0 mV		0	0	0
29				■		0 mV		0	0	0	61				■		0 mV		0	0	0
30				■		0 mV		0	0	0	62				■		0 mV		0	0	0
31				■		0 mV		0	0	0	63				■		0 mV		0	0	0

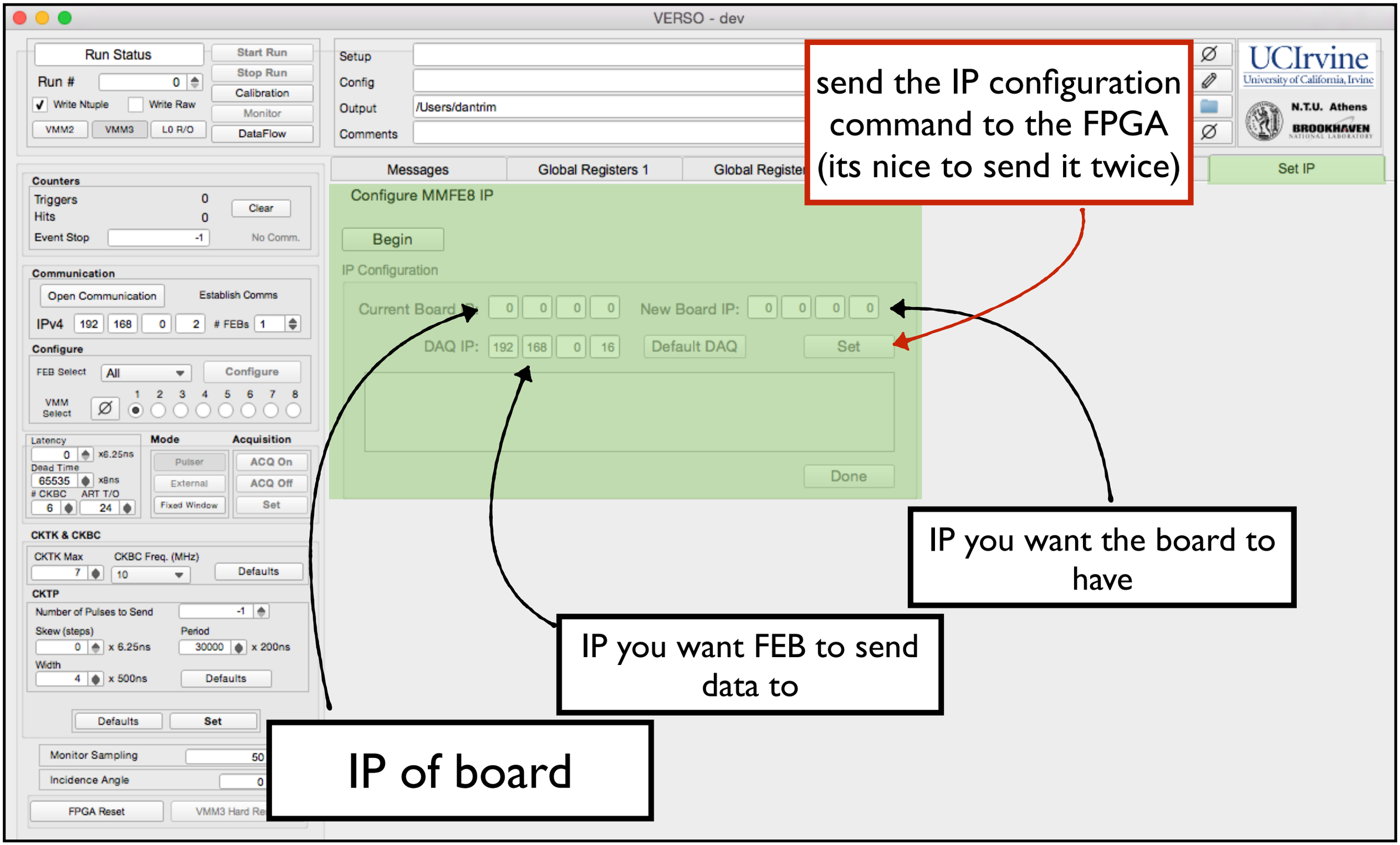
VMM SPI configuration — channel registers panel

VERSO Overview



calibration loop configuration panel

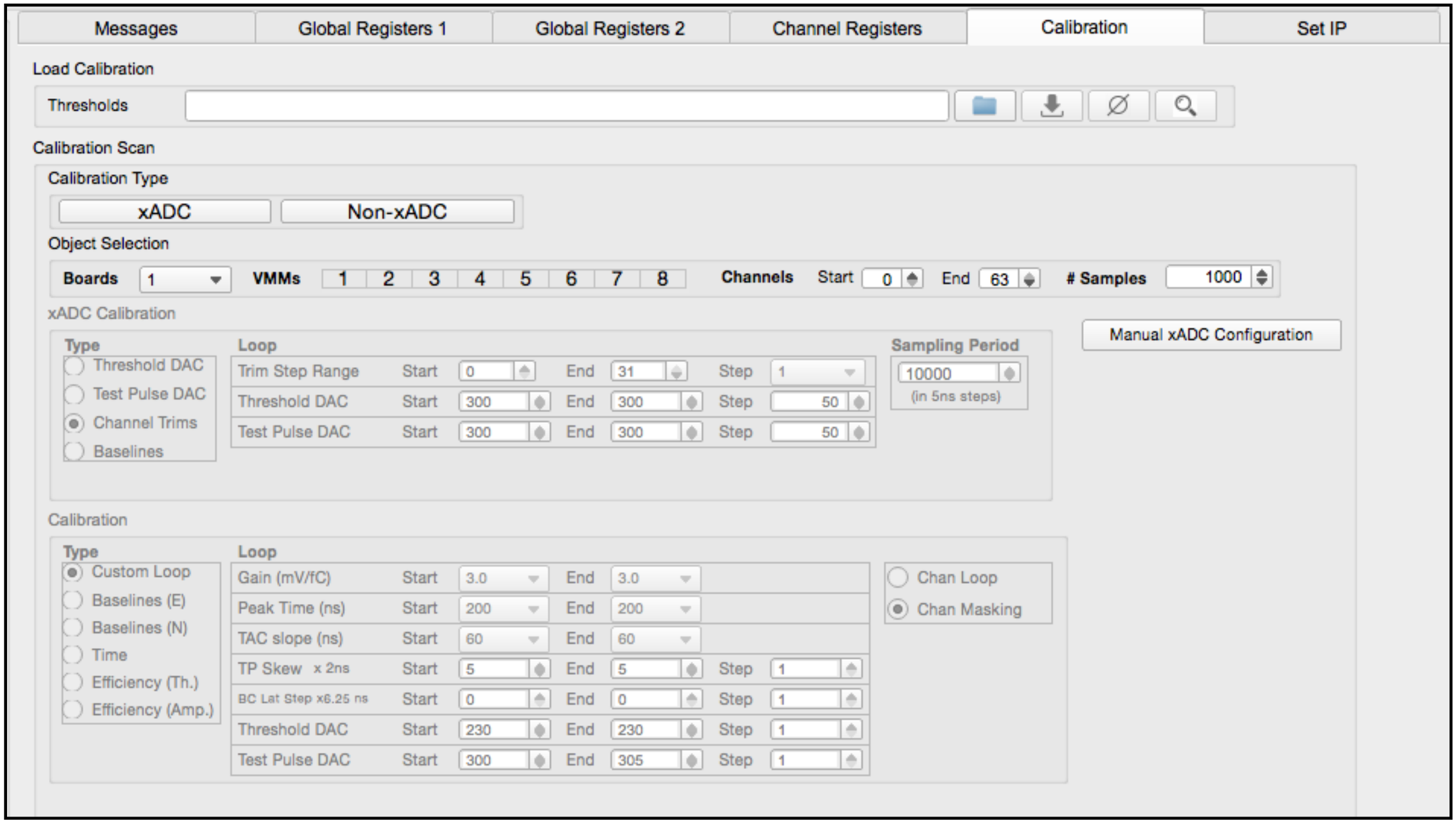
VERSO Overview



FEB IP configuration panel

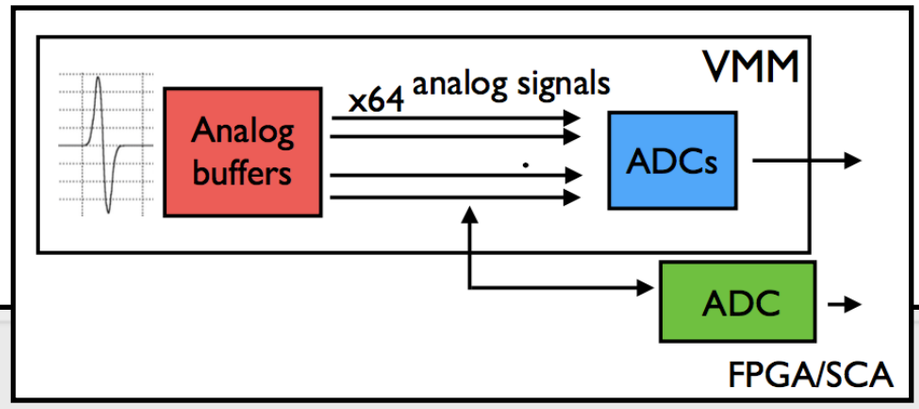
Calibration Overview

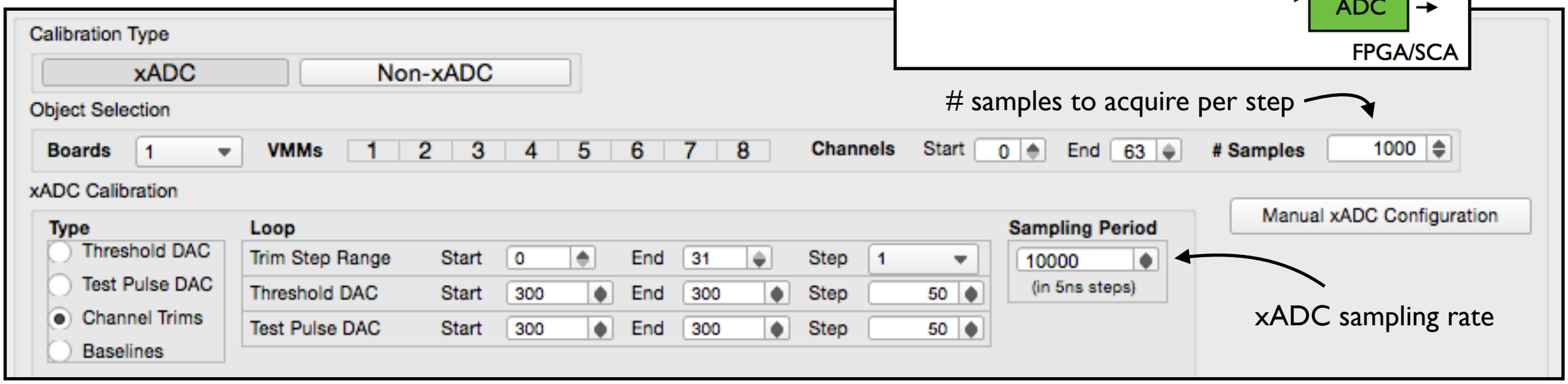
Loading of the calibration loops is done within the VERSO calibration panel. When set, the next run will process these loops and build the calibration n-tuple.



xADC based sampling

For selected (board, chips, channels)…





samples to acquire per step

xADC sampling rate

We can perform calibration routines to sample:

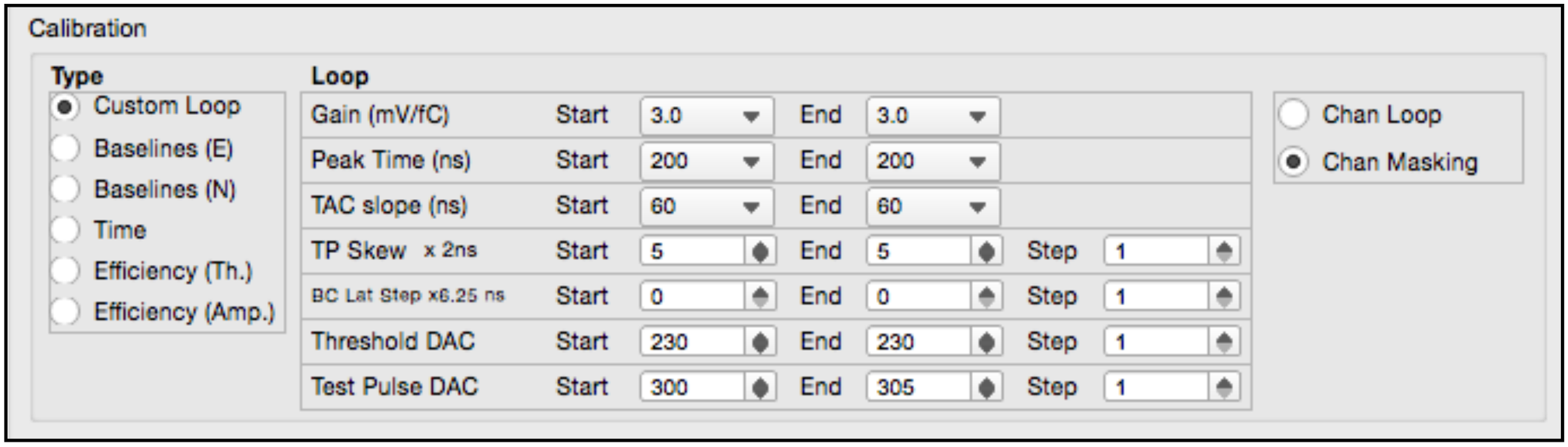
1. Analog DAC levels (threshold and pulser)
2. Channel-by-channel threshold variations, stepping over the VMM threshold trimmer values
3. Channel baseline & noise levels

Output data is not standard VMM events but xADC samples

… + combinations of them

Pulser based sampling

For selected (board, chips, channels)…

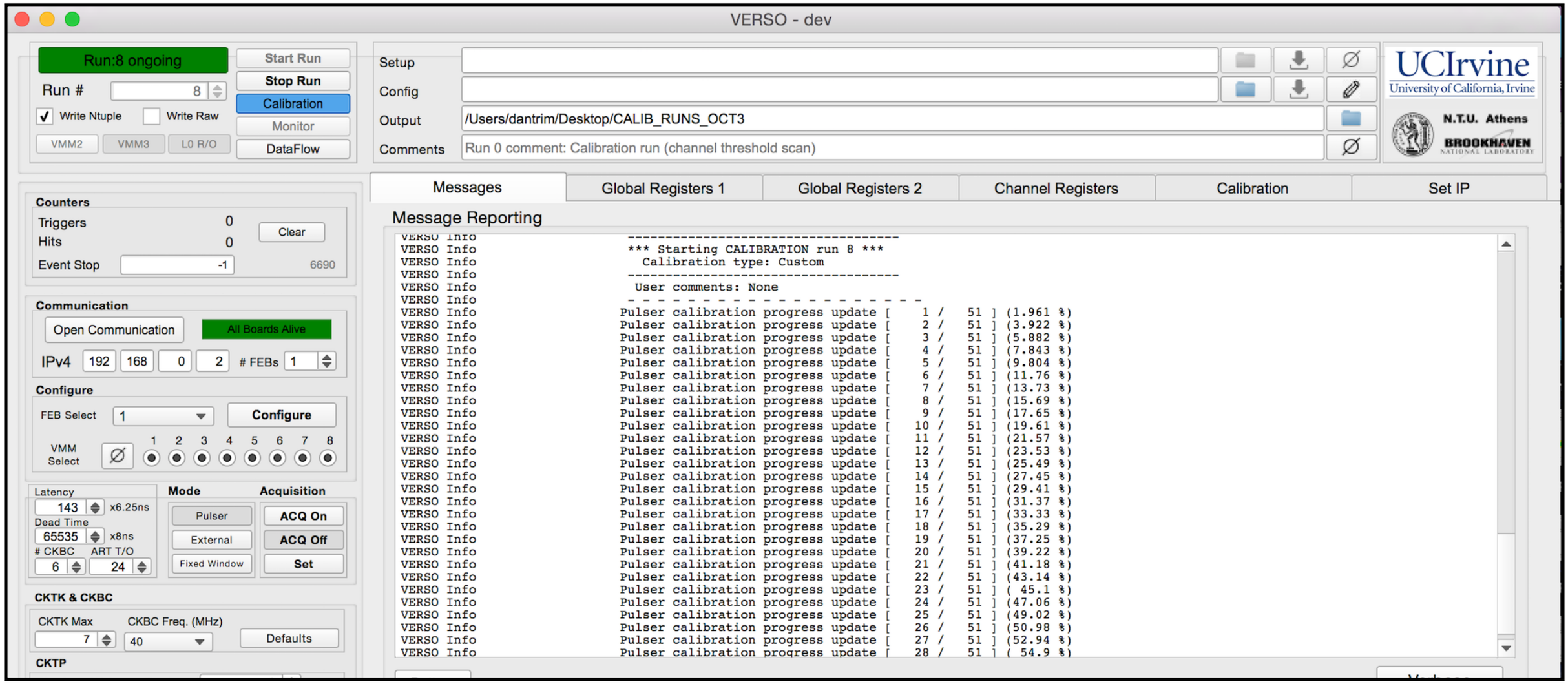


We can perform loops over the shown parameters. The quantity will be looped over if `End != Start`

Output from FEB is same format as VMM event data (we are using the internal pulser) but stored slightly differently for calibration analysis purposes

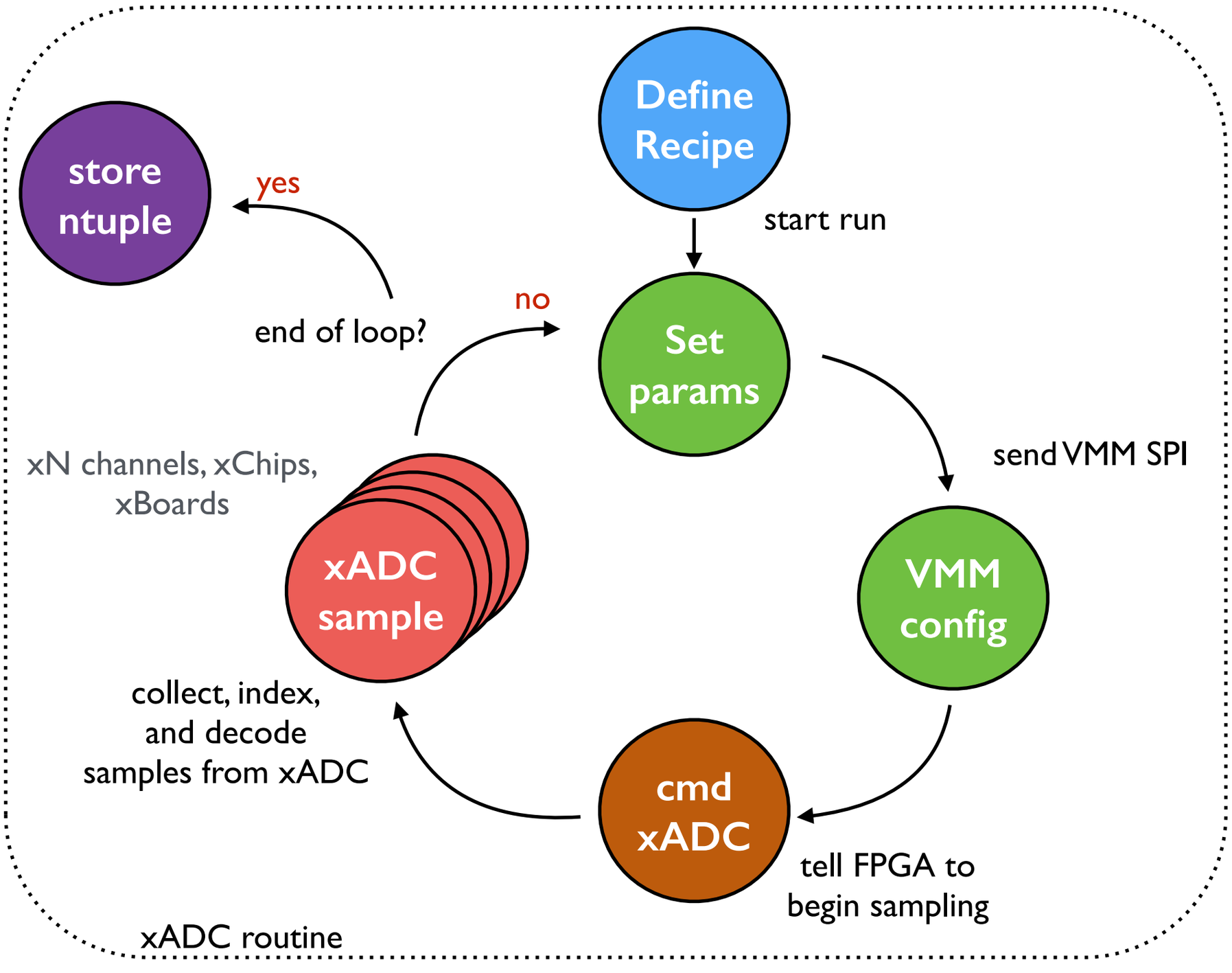
Calibration Overview

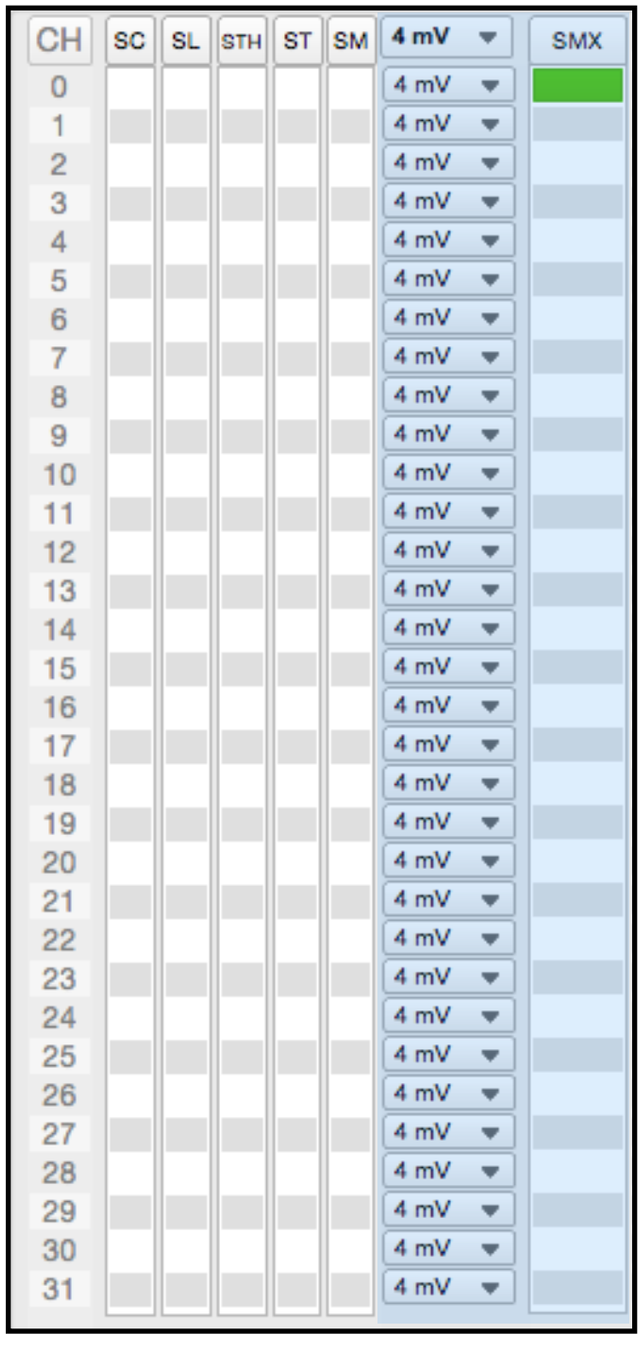
- For the test beam data taking and analysis we exercised much of the calibration
  - In the time leading up to the TB, the pulser-based calibration was totally overhauled
  - Pulser-based calibration module much more robust and efficient (smarter than previous handling)



Channel Threshold Equalization

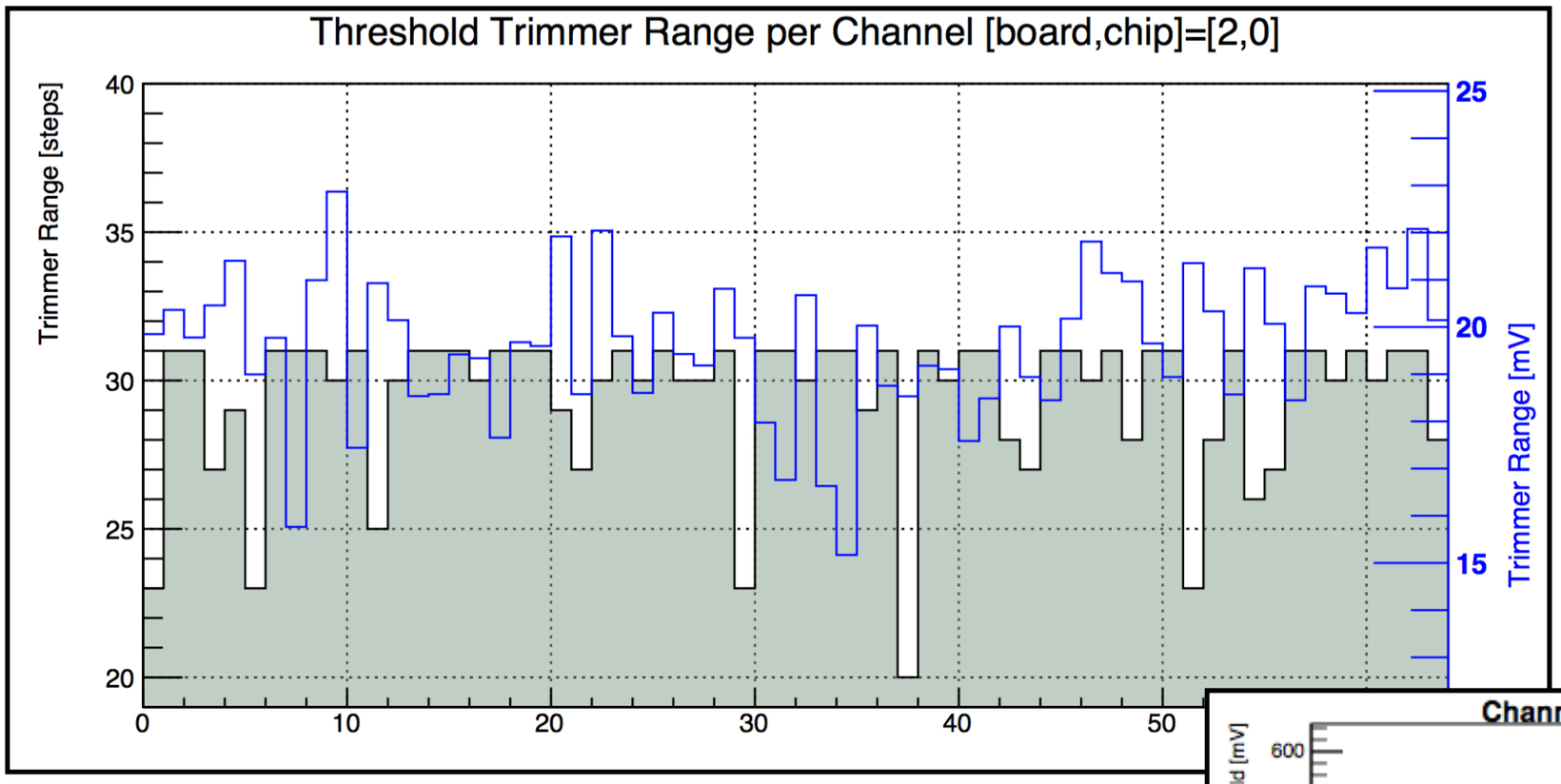
- Before initial data taking, we performed the channel threshold trimmer calibration in order to equalize the channel-by-channel response
- This is an xADC based sampling procedure that steps through the channel trimmer settings for each channel and measures the threshold with N samplings per setting

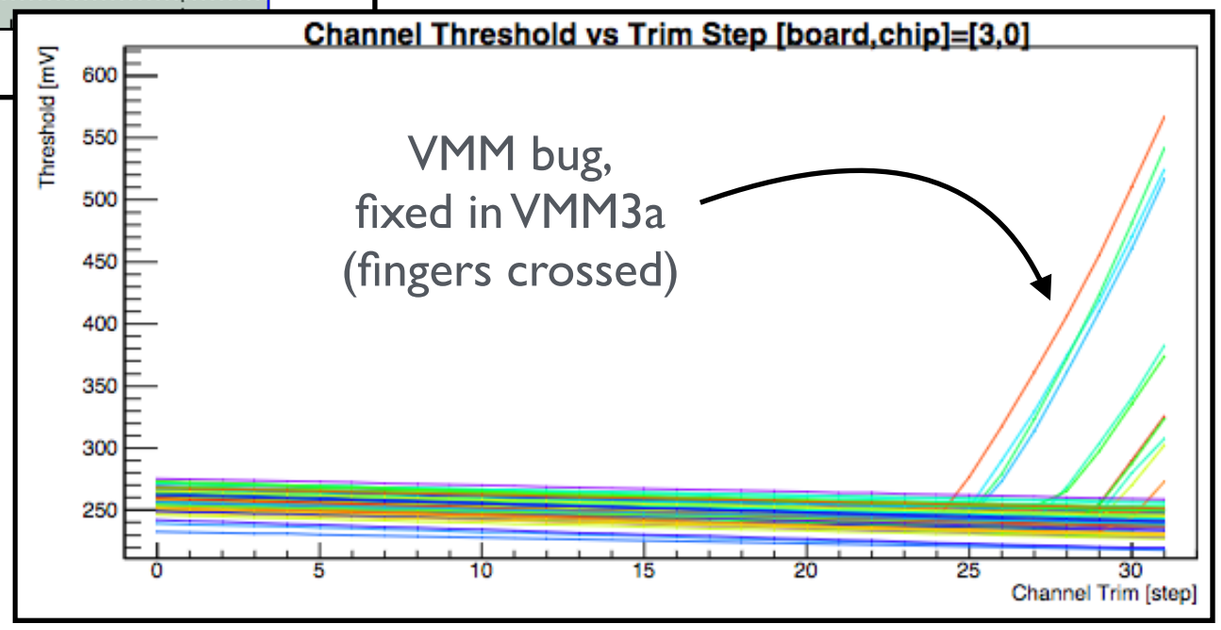




Channel Threshold Equalization

As we step through the trimmers, we vary each channel's threshold and get a measure of the overall channel trim range



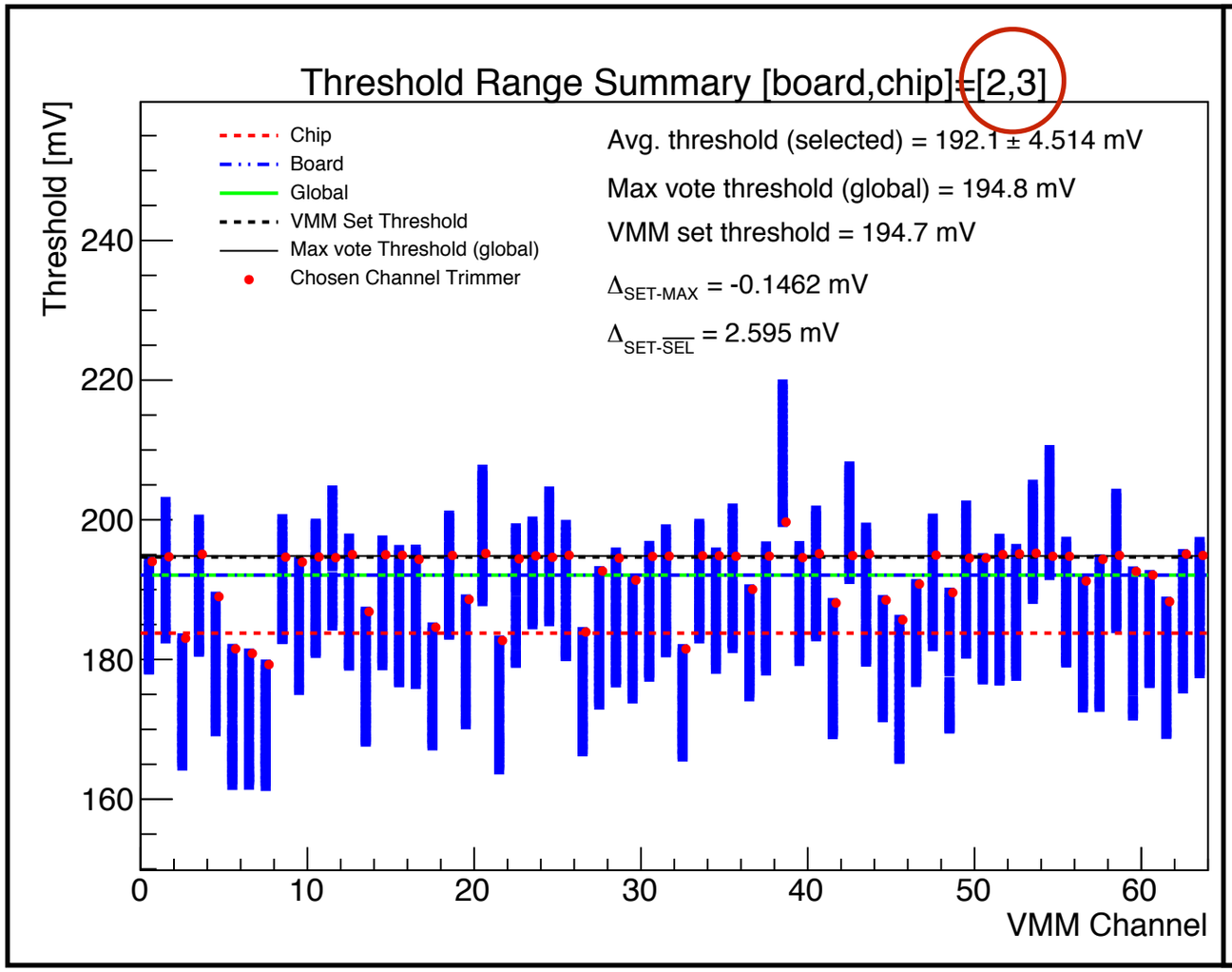


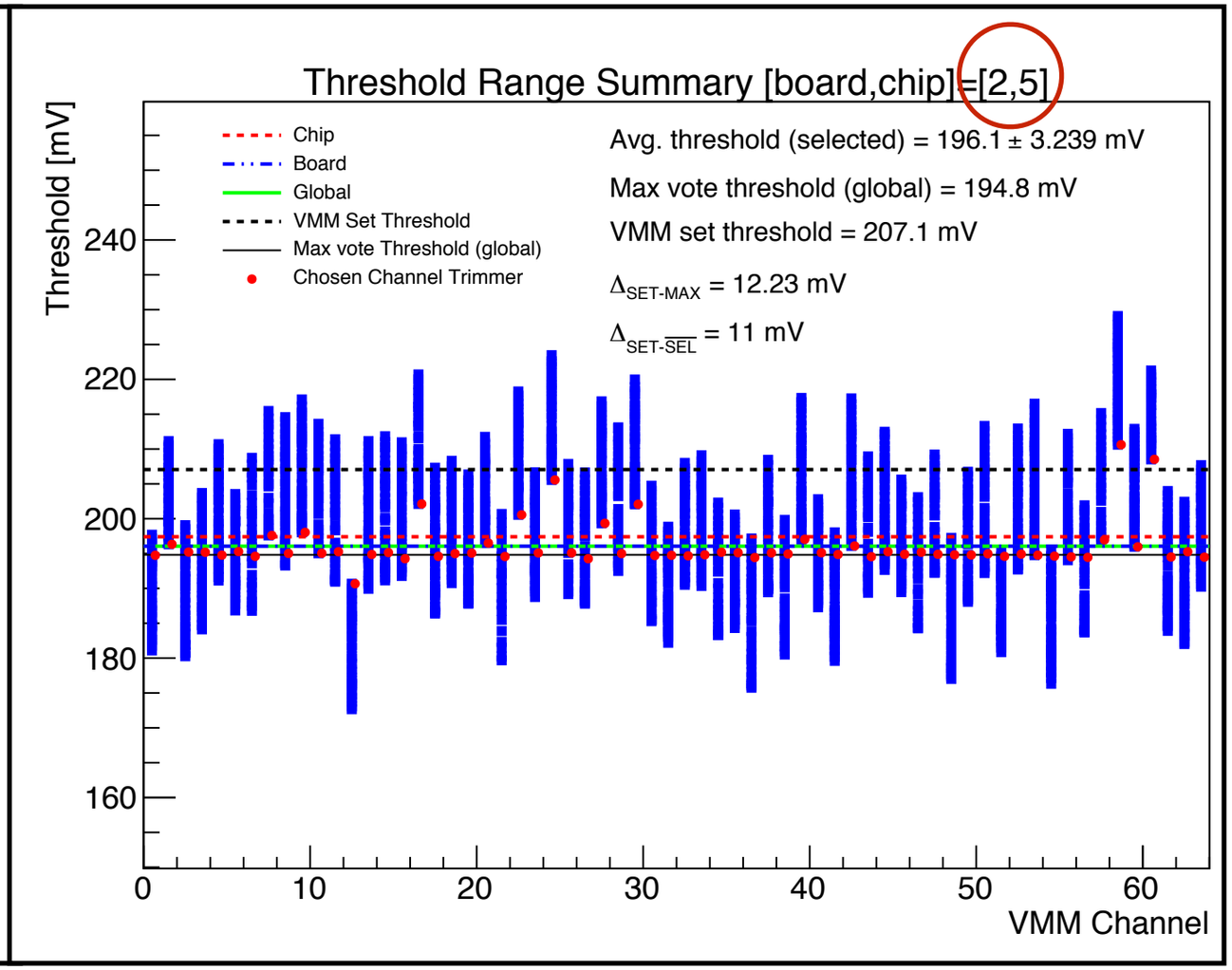
Inaccessible trim steps are detected by the calibration analysis software and not considered during the equalization procedure

Channel Threshold Equalization

The channel-by-channel threshold variation is "equalized" by a max voting procedure:

1. For all loaded boards, chips, channels, determine the threshold (mV) ranges accessible
2. Find the threshold (mV) that is attainable by the maximum number of VMM channels
3. For each VMM channel, find the trimmer setting that gets that channel closest to that "max-voted" threshold
4. Store these trimmer settings (at per chip, per board, and "global" granularity)



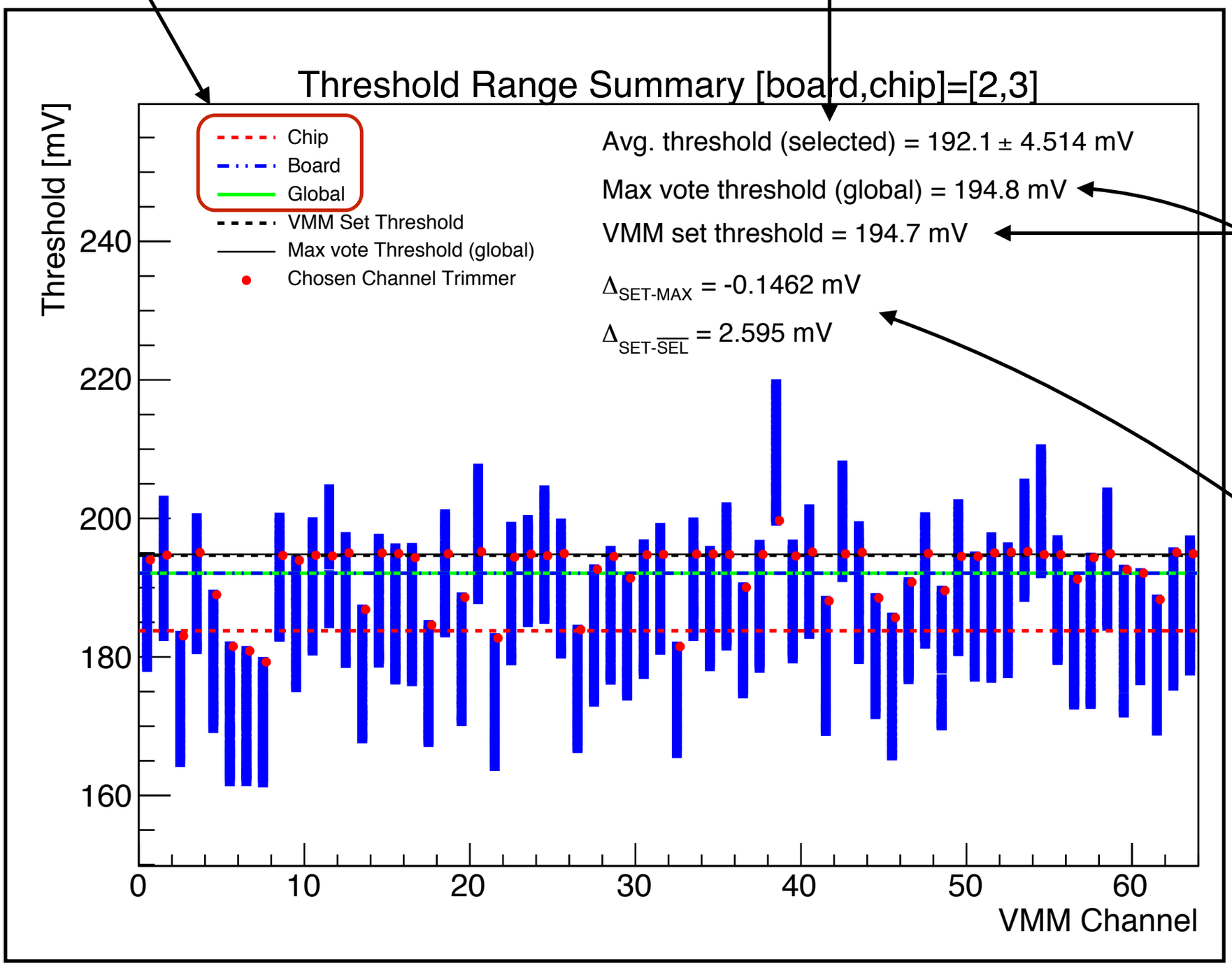


Channel Threshold Equalization

average thresholds at the trimmer settings that equalize at the per chip, per FEB, and "global" level

average of the threshold values at the chosen trimmers

channel trimmer calibration can load in the DAC calibration to determine the VMM threshold in mV — this VMM had threshold DAC of 230 counts



max voted threshold to every channel aspires to

Δ's between VMM configured threshold (DAC) and the max-voted and average-of-selected thresholds

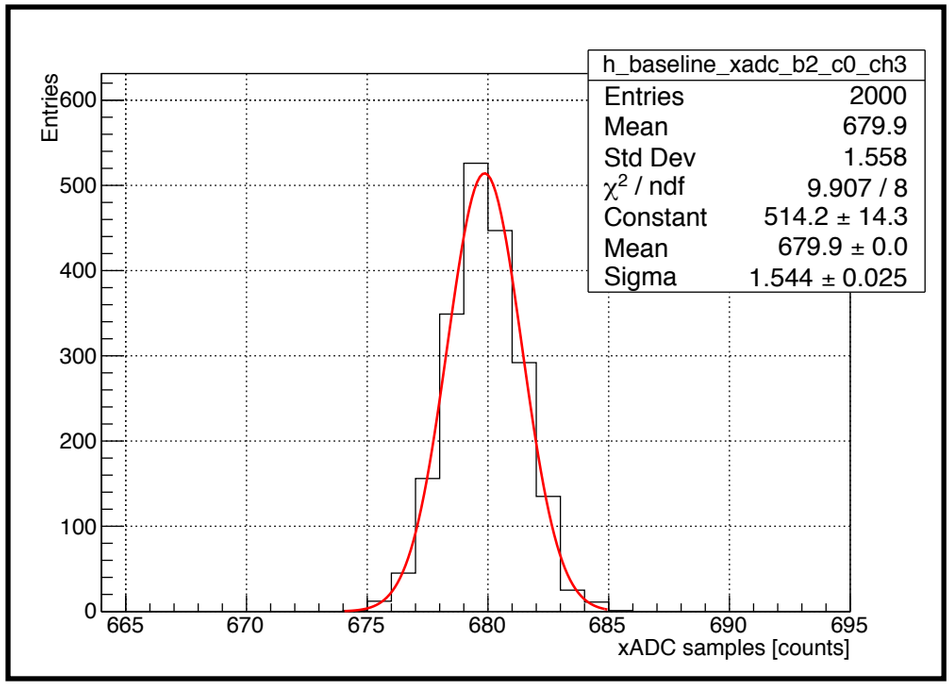
divergent trims removed prior to all equalization

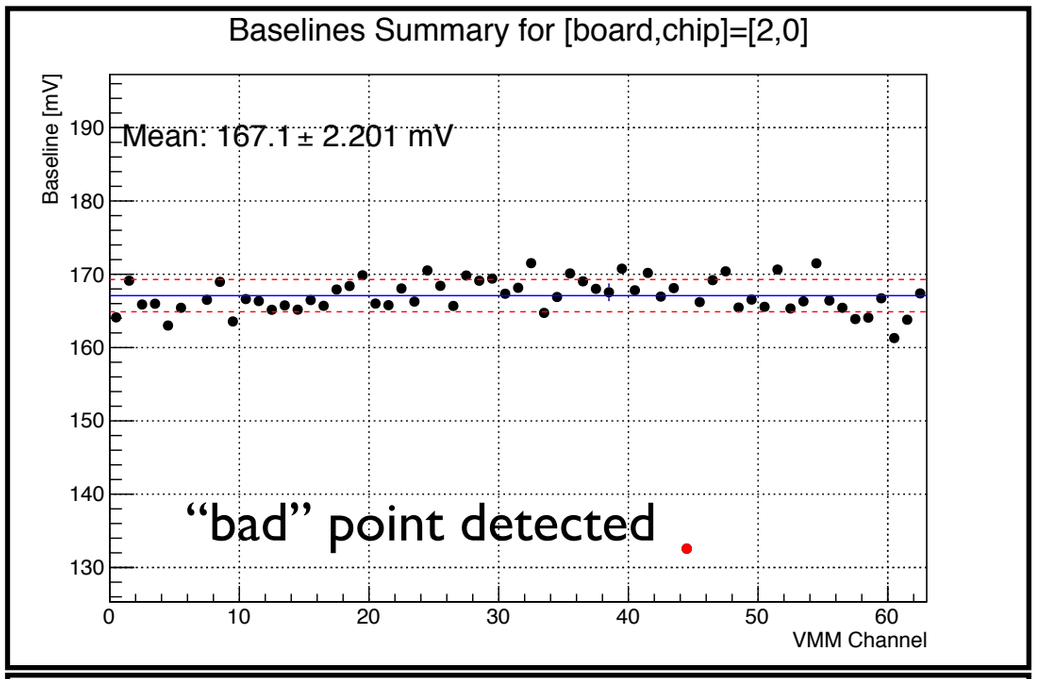
Baselines and Noise Measurement

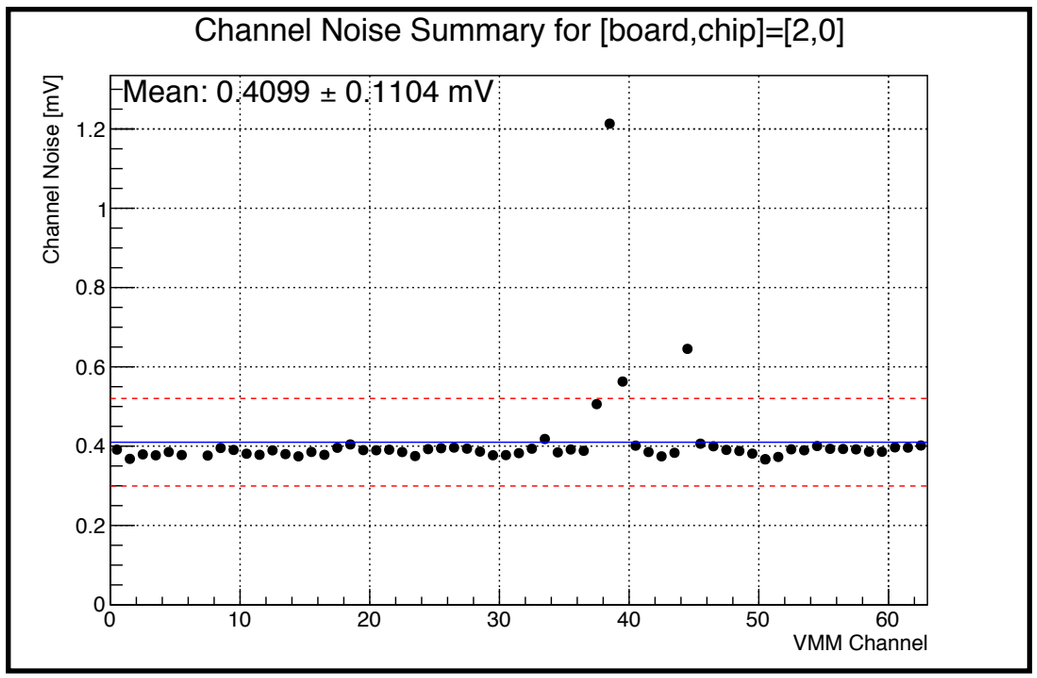
Use the xADC to sample the channel input baselines and noise

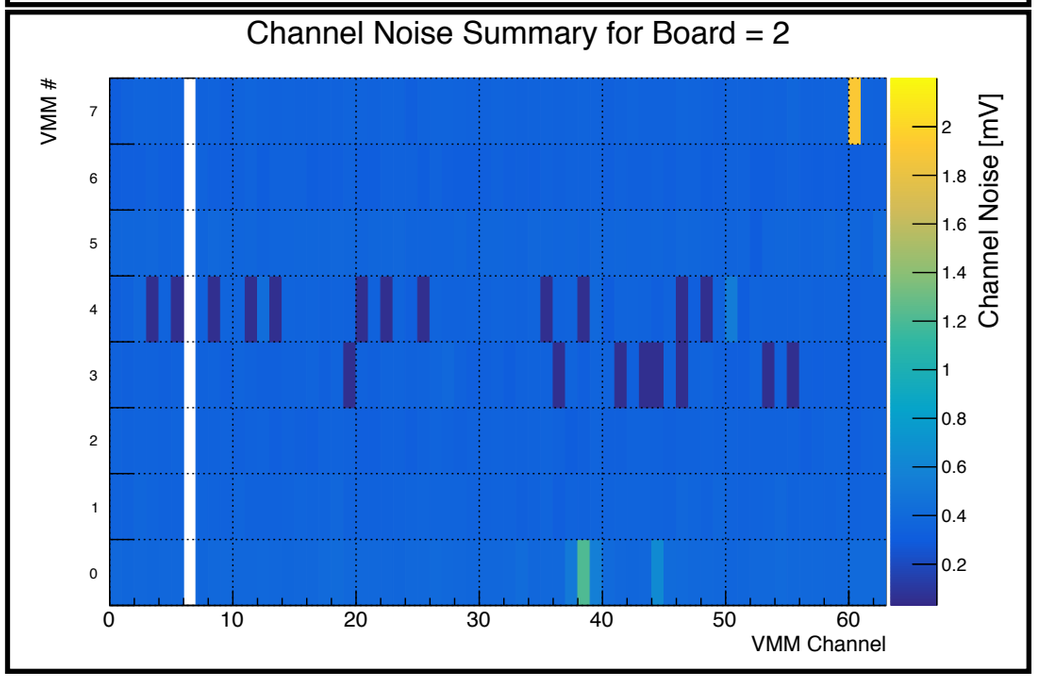
Procedure:

1. For each board and chip, sample each channel's input level with the xADC
2. Perform a gaussian fit of the samples per channel: mean is taken as that channel's baseline, width is taken as the channel noise





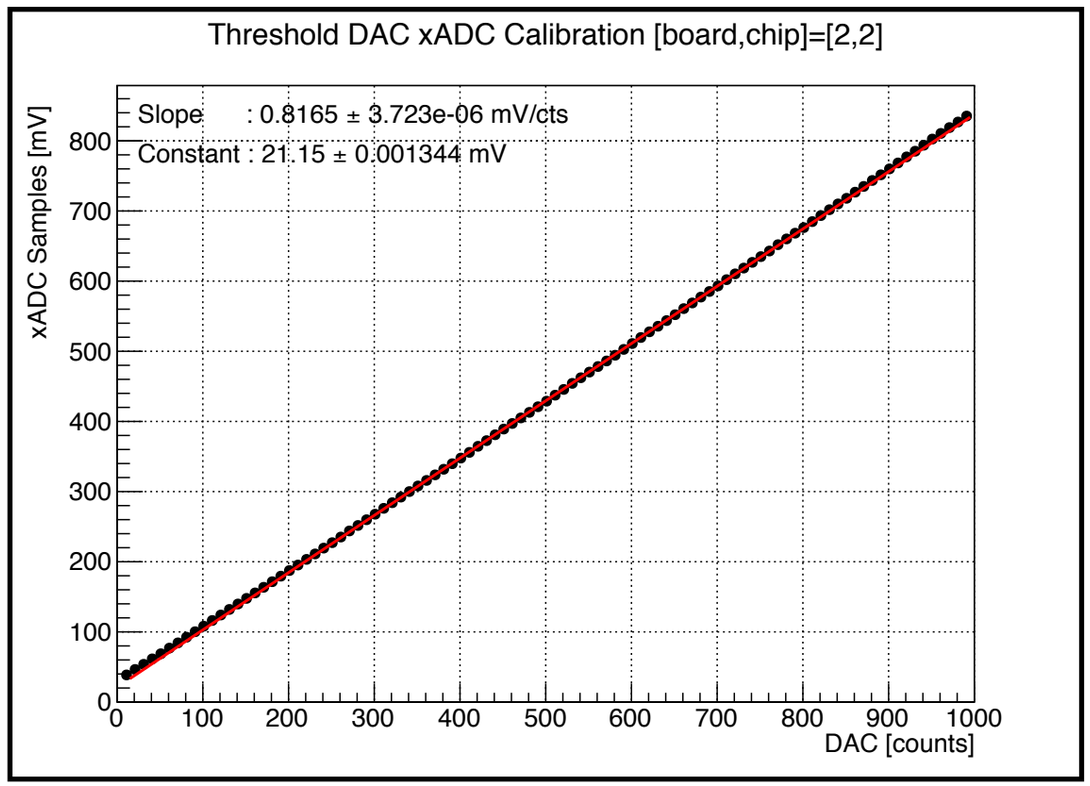




Use the xADC to sample the DAC levels

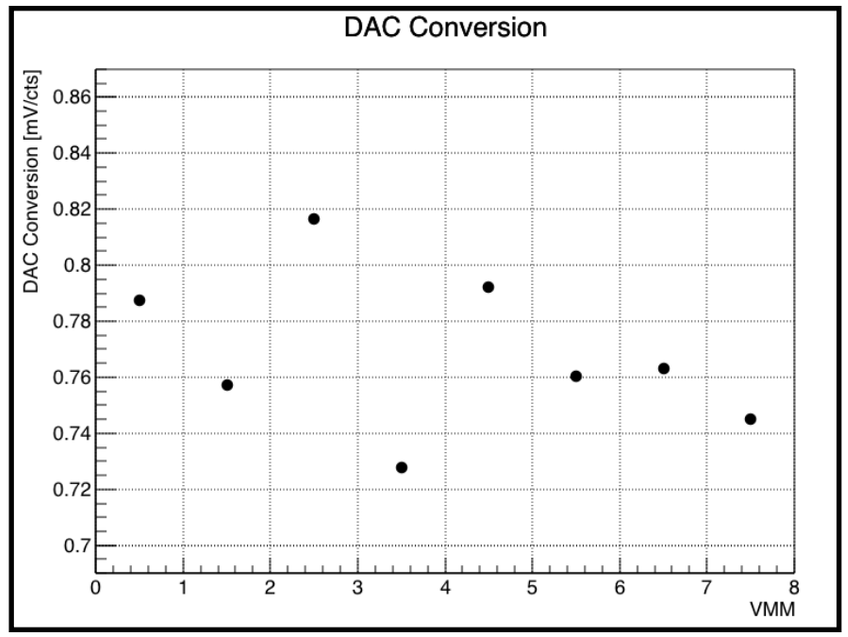
Procedure:

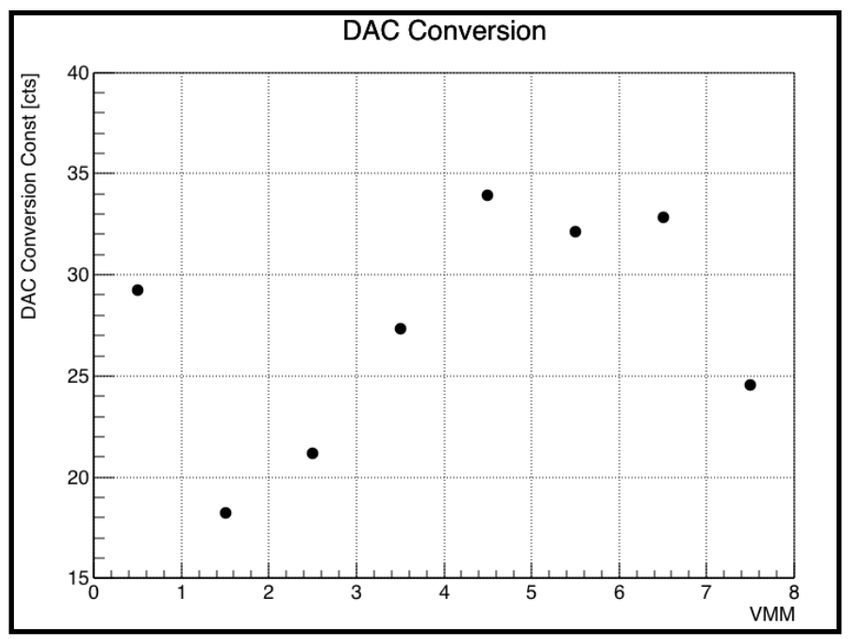
1. Step through the VMM DAC values (pulser and/or threshold)
2. At each DAC, sample the DAC levels with the xADC
3. From fit, determine conversion constants to absolute scale



slopes

intercepts

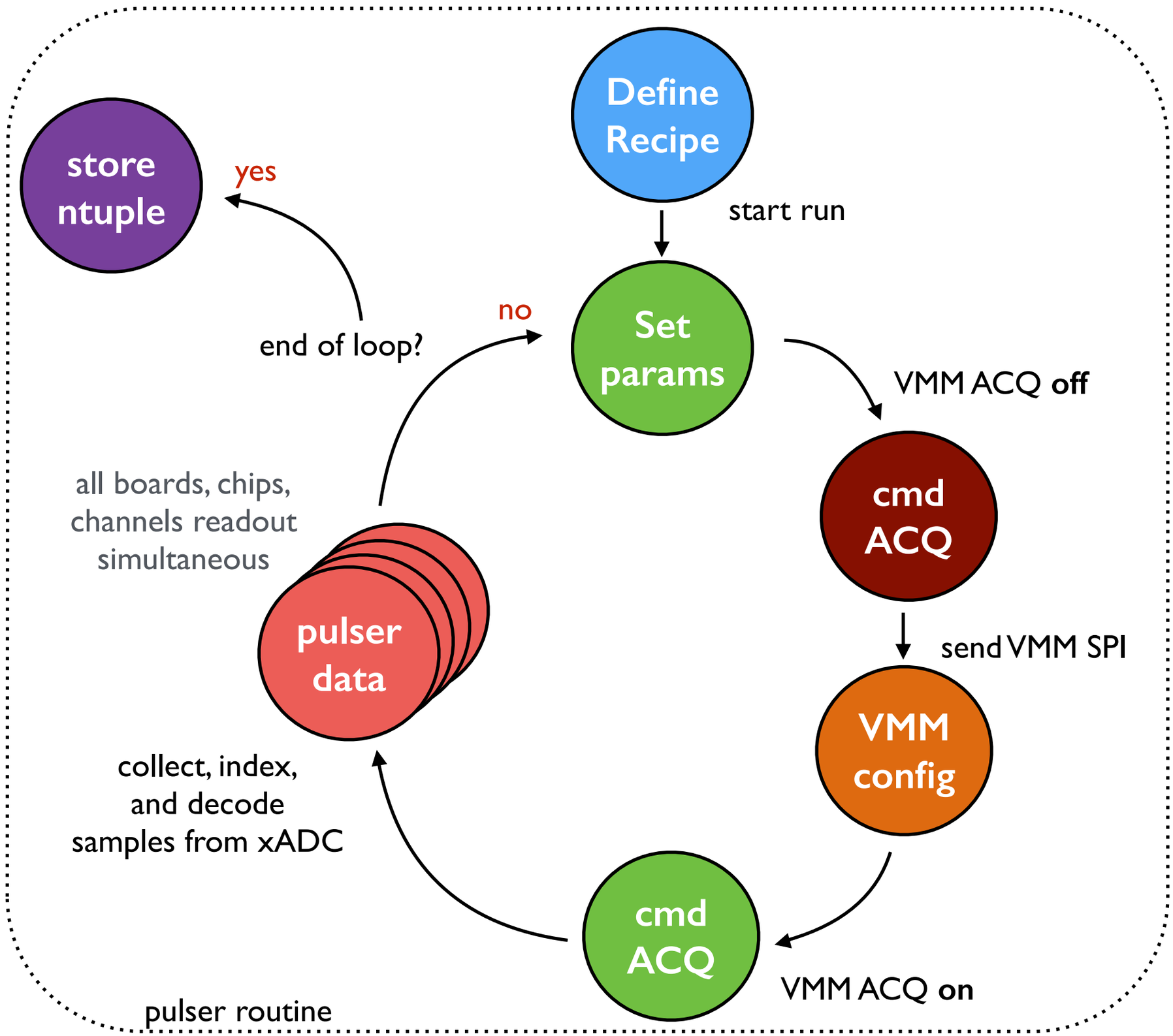




Can load these data into to other calibration procedures and analysis

Channel Gain and Pedestal Measurement

Pulser-based calibration to measure channel-by-channel variation in gain and PDO pedestal

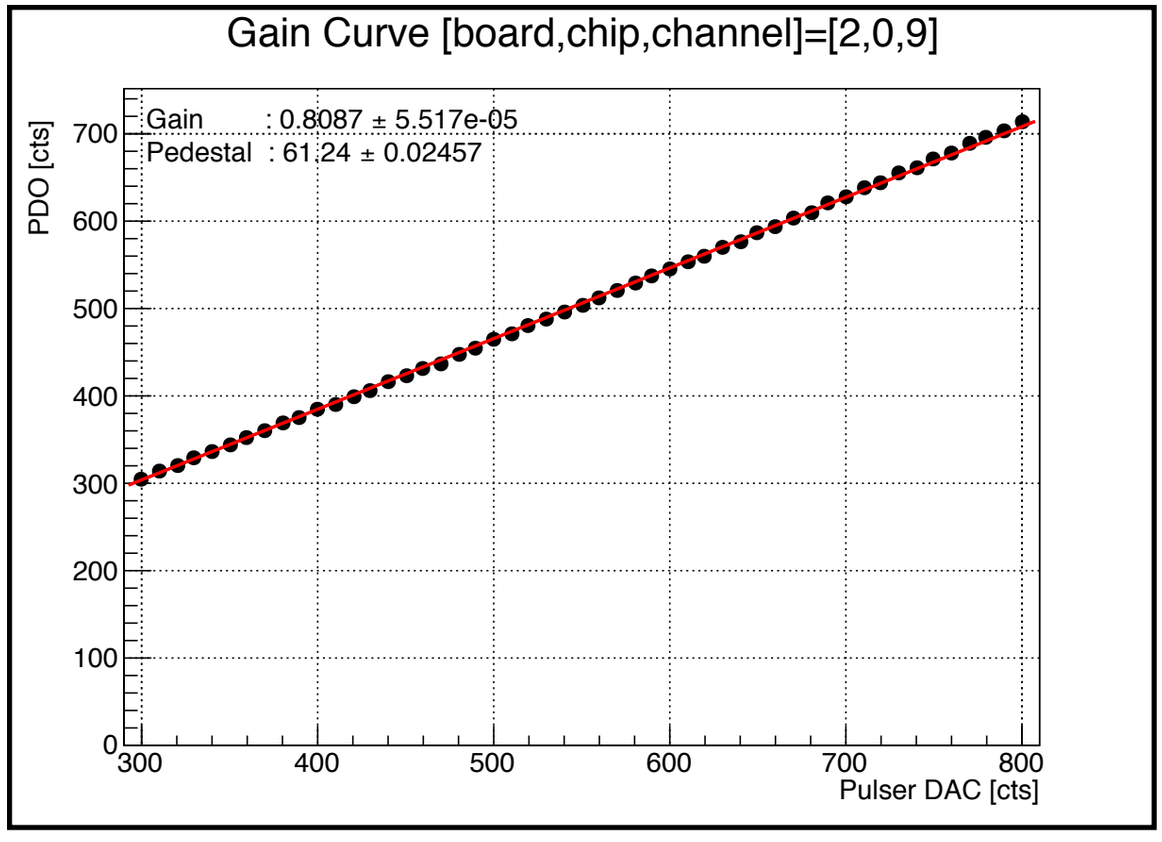


Channel Gain and Pedestal Measurement

Pulser-based calibration to measure channel-by-channel variation in gain and PDO pedestal for offline correction of the data

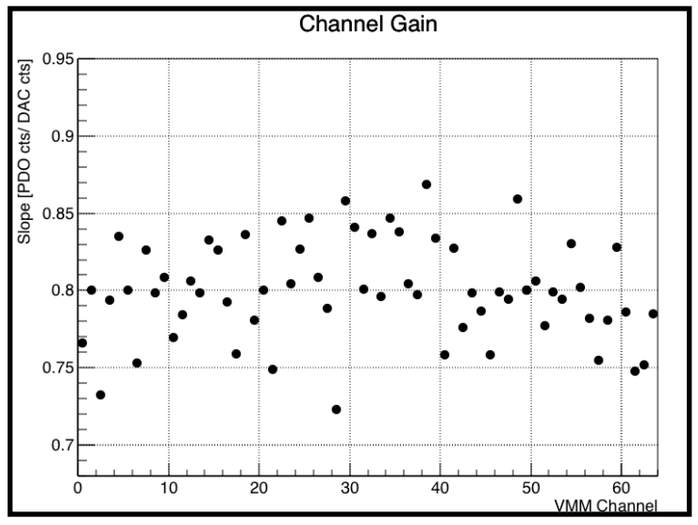
Procedure:

1. Step through pulser DAC values
2. At each DAC collect N samples per channel
3. From linear relation between PDO and DAC, get gain curve and pedestal parameters

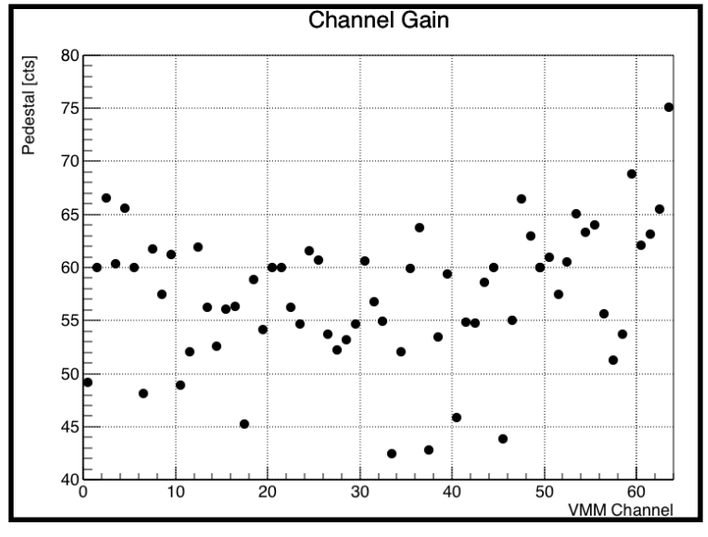


Loading the pulser DAC calibration allows you to get absolute gain in mV/fC

slopes (gains)



pedestals

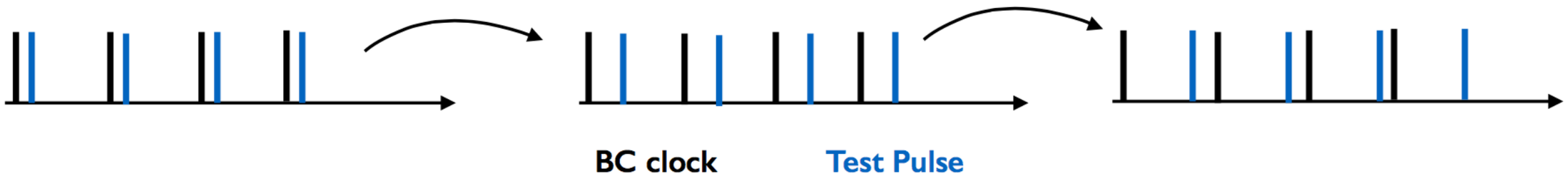


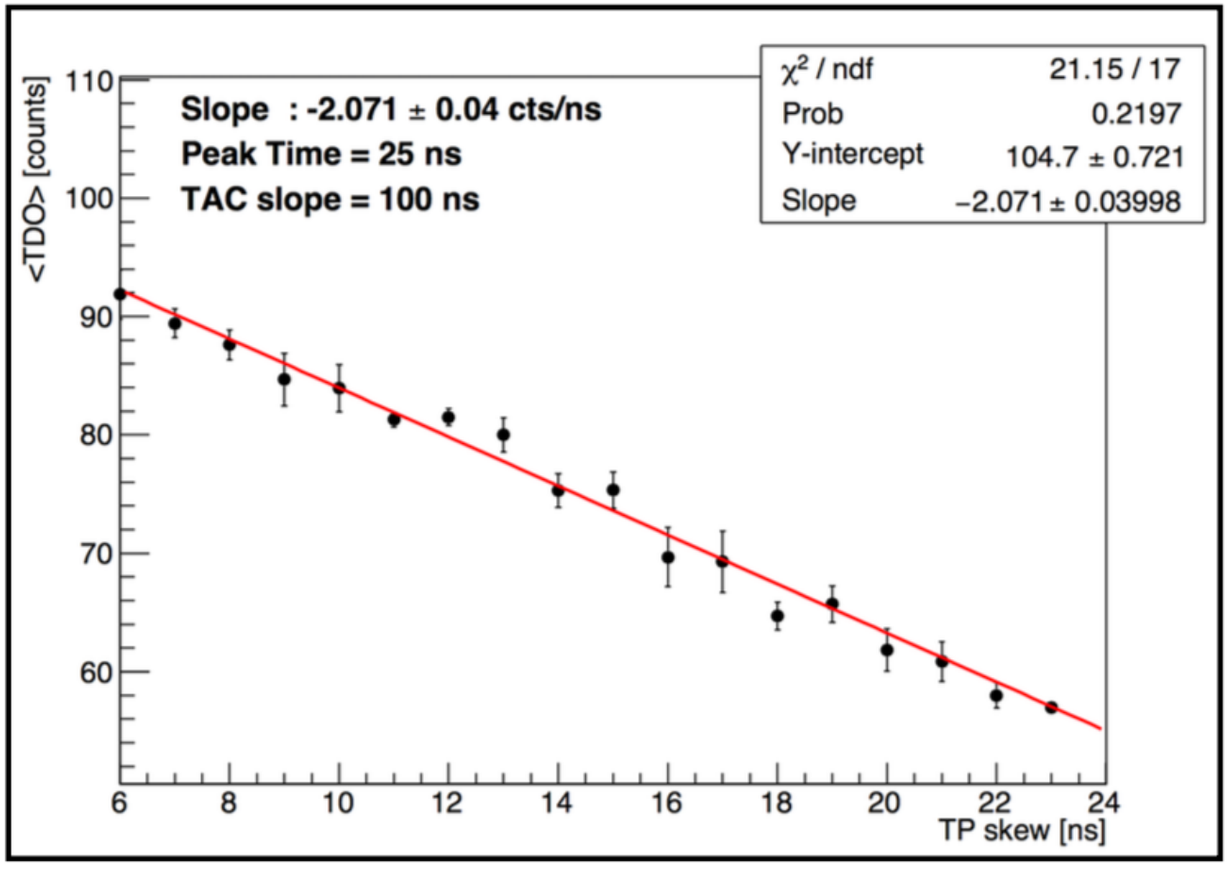
Timing Calibration

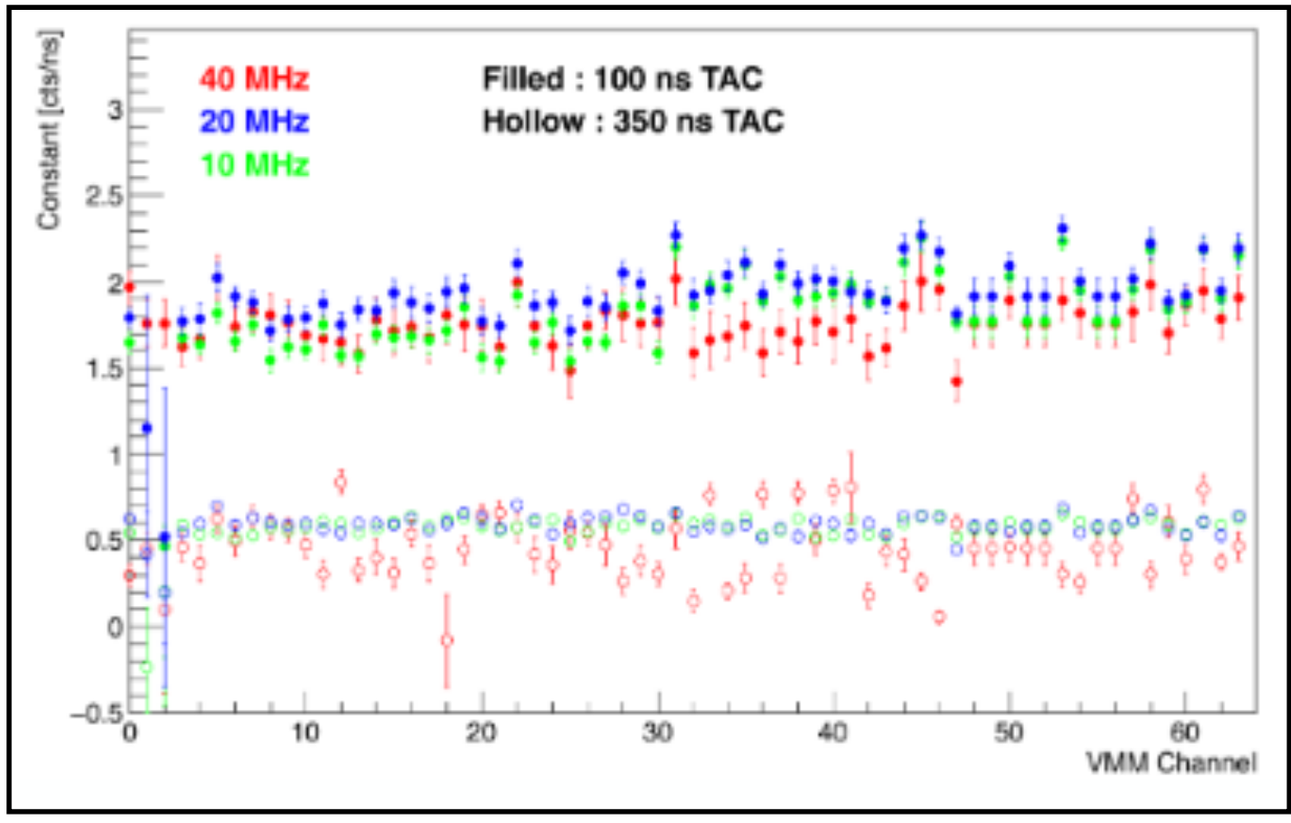
There are two methods for timing calibration

CKTP (test pulse) skewing

1. Skew the CKTP relative to CKBC in known time steps
2. From relation between TDO and known time delay, obtain conversion from TDO ADC counts to ns





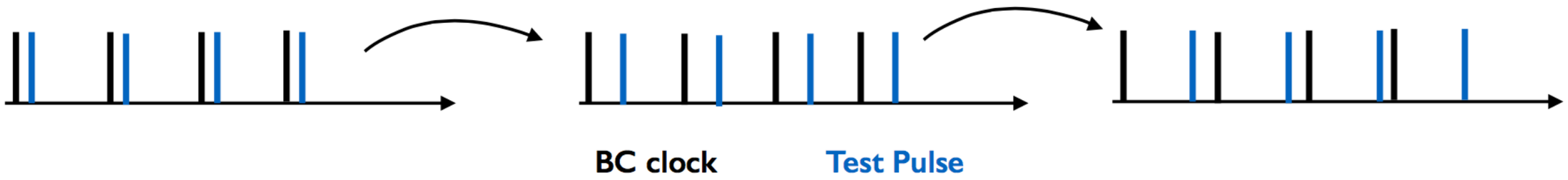


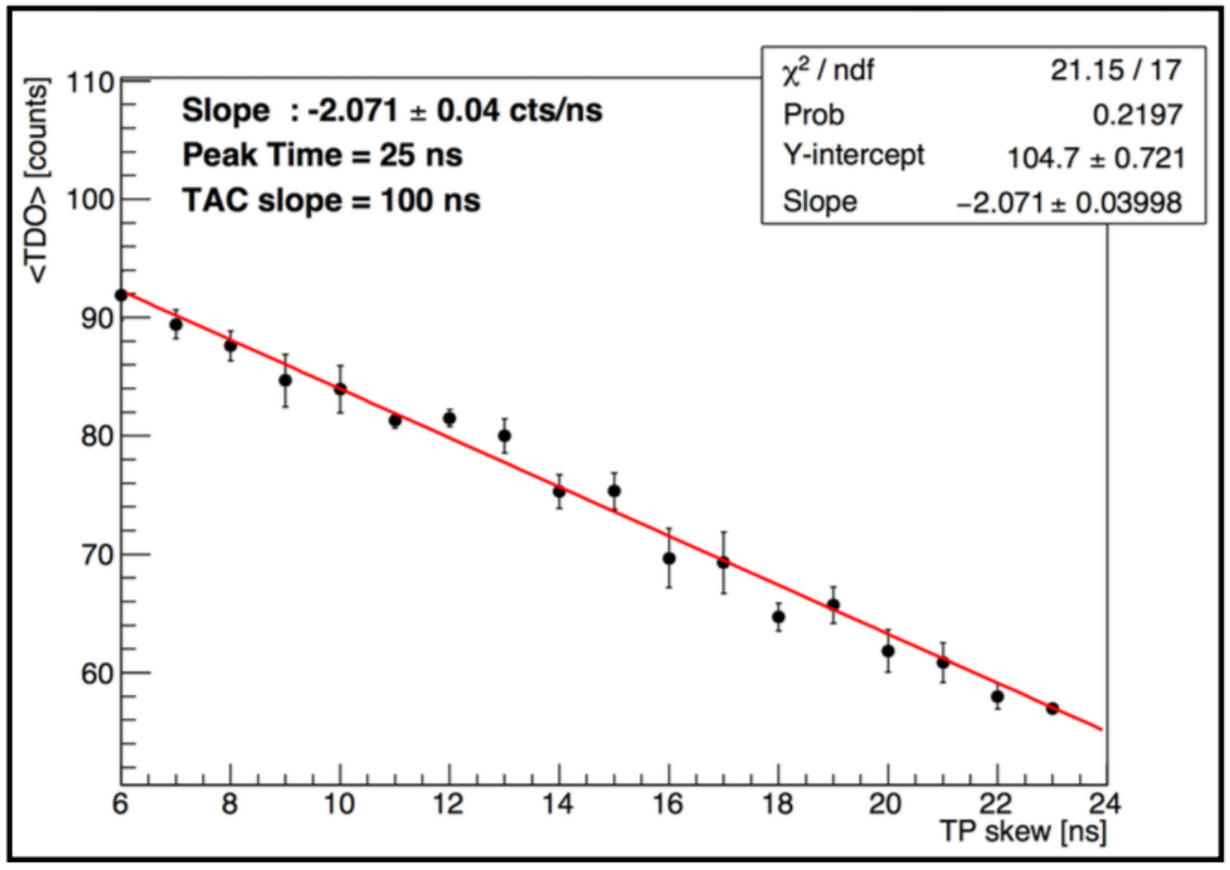
Timing Calibration

There are two methods for timing calibration

CKTP (test pulse) skewing

1. Skew the CKTP relative to CKBC in known time steps
2. From relation between TDO and known time delay, obtain conversion from TDO ADC counts to ns





With this method:

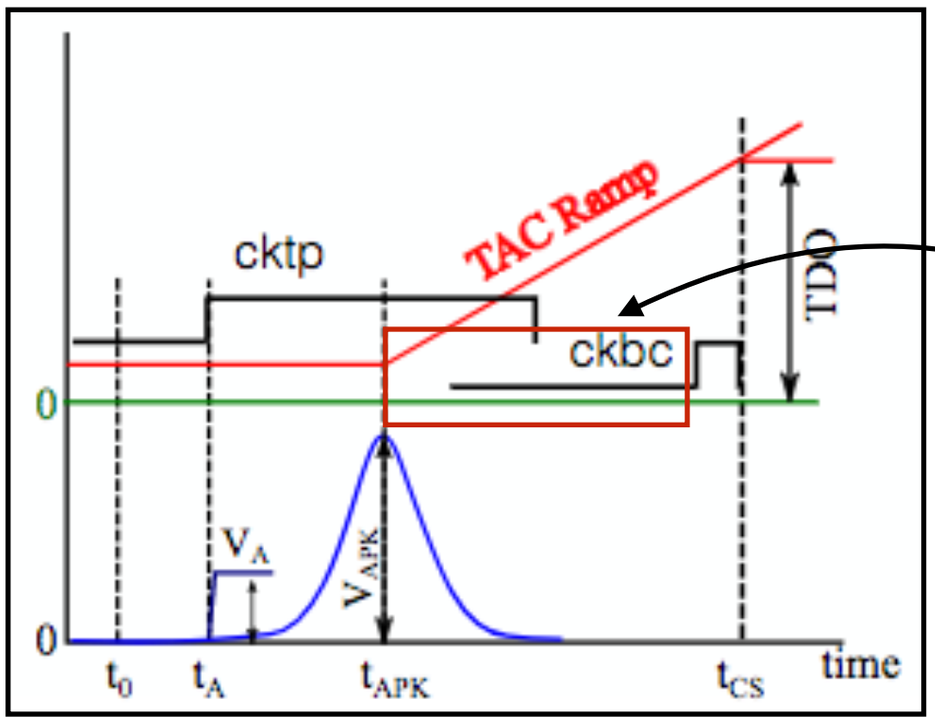
1. we cannot measure the full TDO range
2. we cannot measure any TDO pedestal
3. we have a limited number of sampling steps (CKTP skew constrained to be within time window of one BC — and first 3-5 cannot be used)

There are two methods for timing calibration

CKBC Latency Delays

1. Configure the VMM to be run in "Fixed Window" mode (i.e. TAC ramp has a *fixed latency*)
2. TAC ramp stops at next falling edge of CKBC after peak found — in "Fixed Window" mode we set CKBC low after signal crossing until the fixed latency is reached, and throw (a configurable # of) CKBC signals to stop the TAC
3. From relation between TDO and the known BC latency steps, obtain conversion between TDO and time as well as possible methods for TDO ADC pedestal measurement

we implemented the ability to do this after timing analysis begun on the test beam data



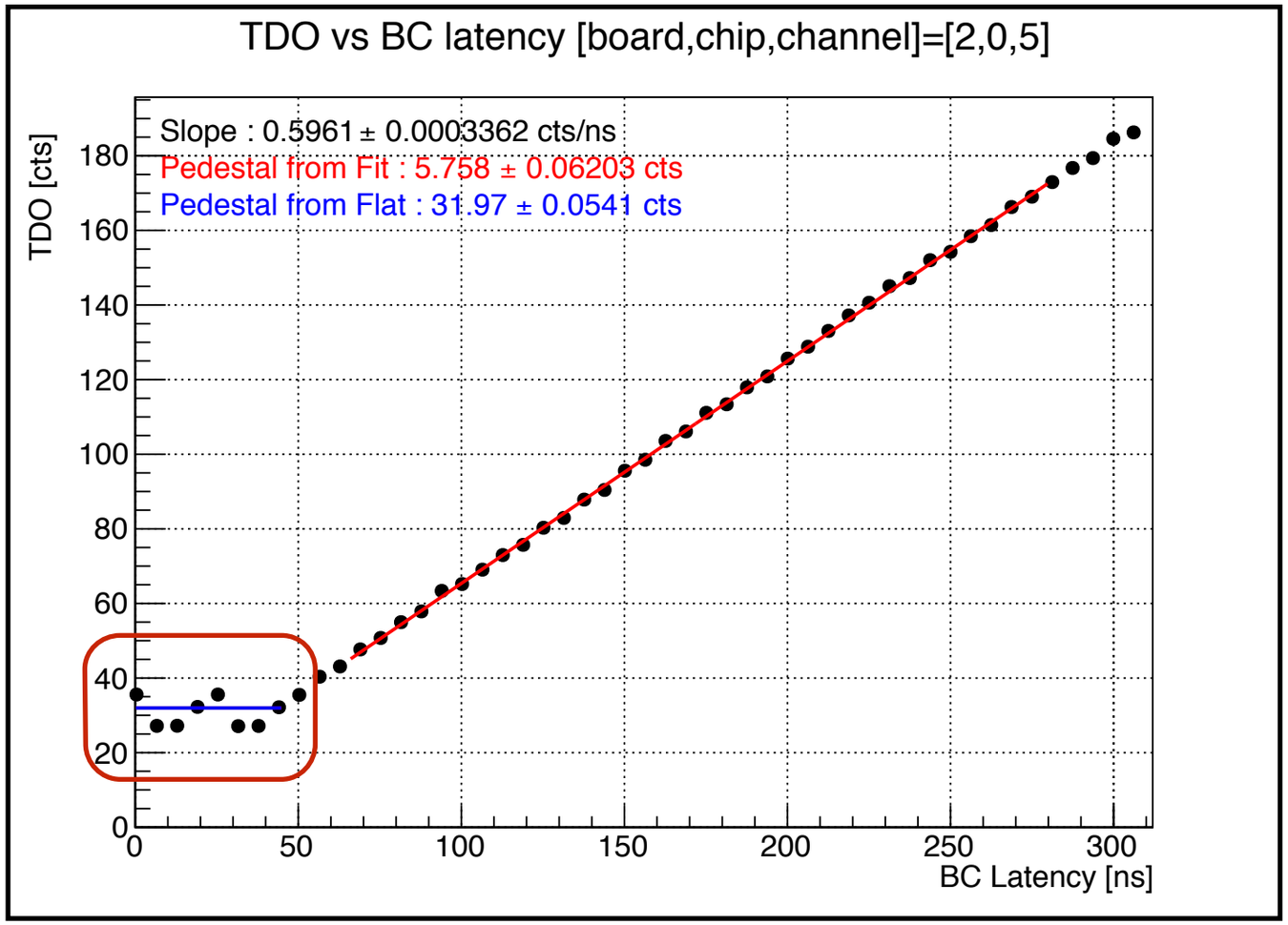
we adjust this window length

There are two methods for timing calibration

CKBC Latency Delays

1. Configure the VMM to be run in "Fixed Window" mode (i.e. TAC ramp has a *fixed latency*)
2. TAC ramp stops at next falling edge of CKBC after peak found — in "Fixed Window" mode we set CKBC low after signal crossing until the fixed latency is reached, and throw (a configurable # of) CKBC signals to stop the TAC
3. From relation between TDO and the known BC latency steps, obtain conversion between TDO and time as well as possible methods for TDO ADC pedestal measurement

"flat" region is prior to peak-found (peak time = 50ns here)



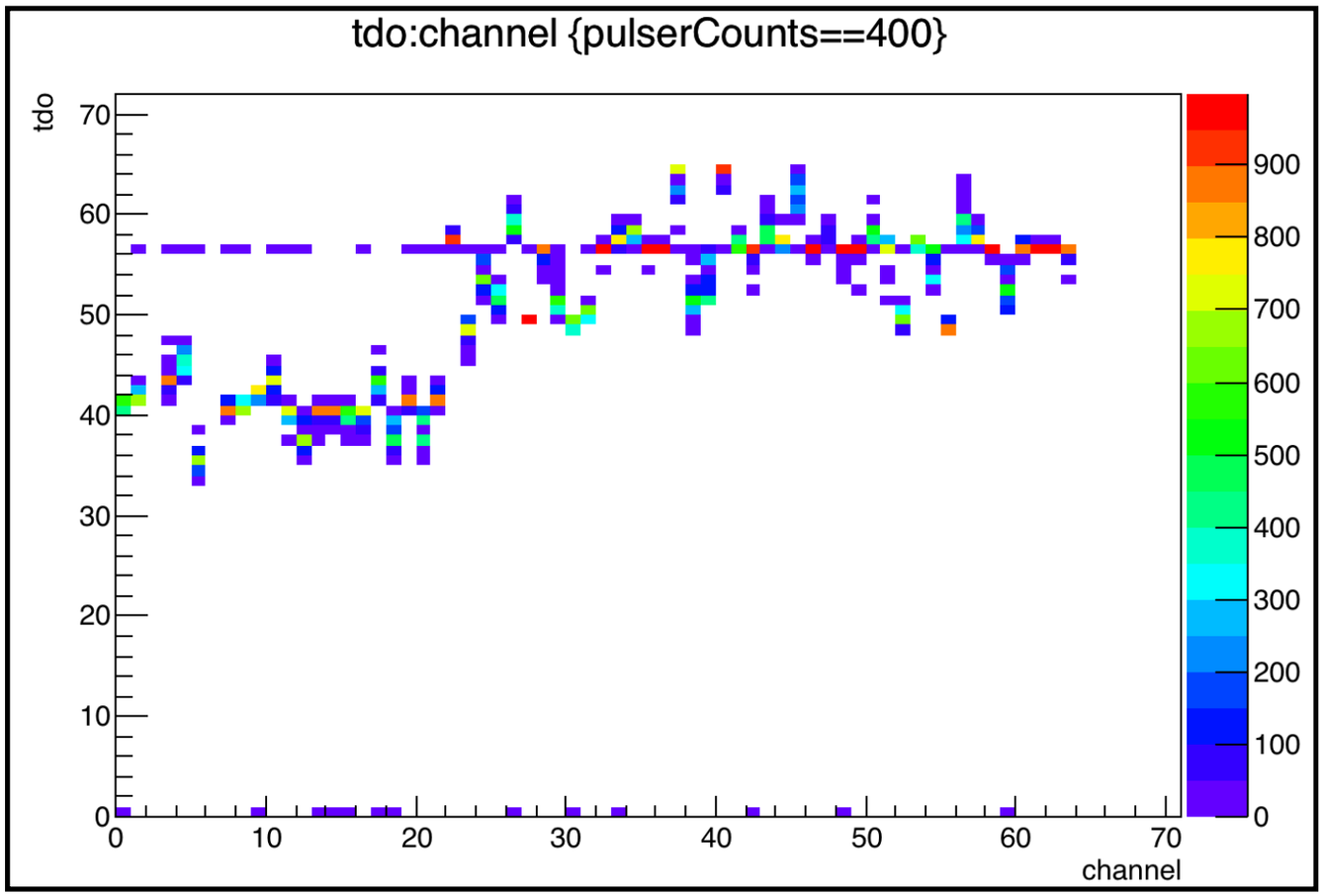
With this method we can achieve many more (higher quality) measurements and can obtain a measure of the pedestal

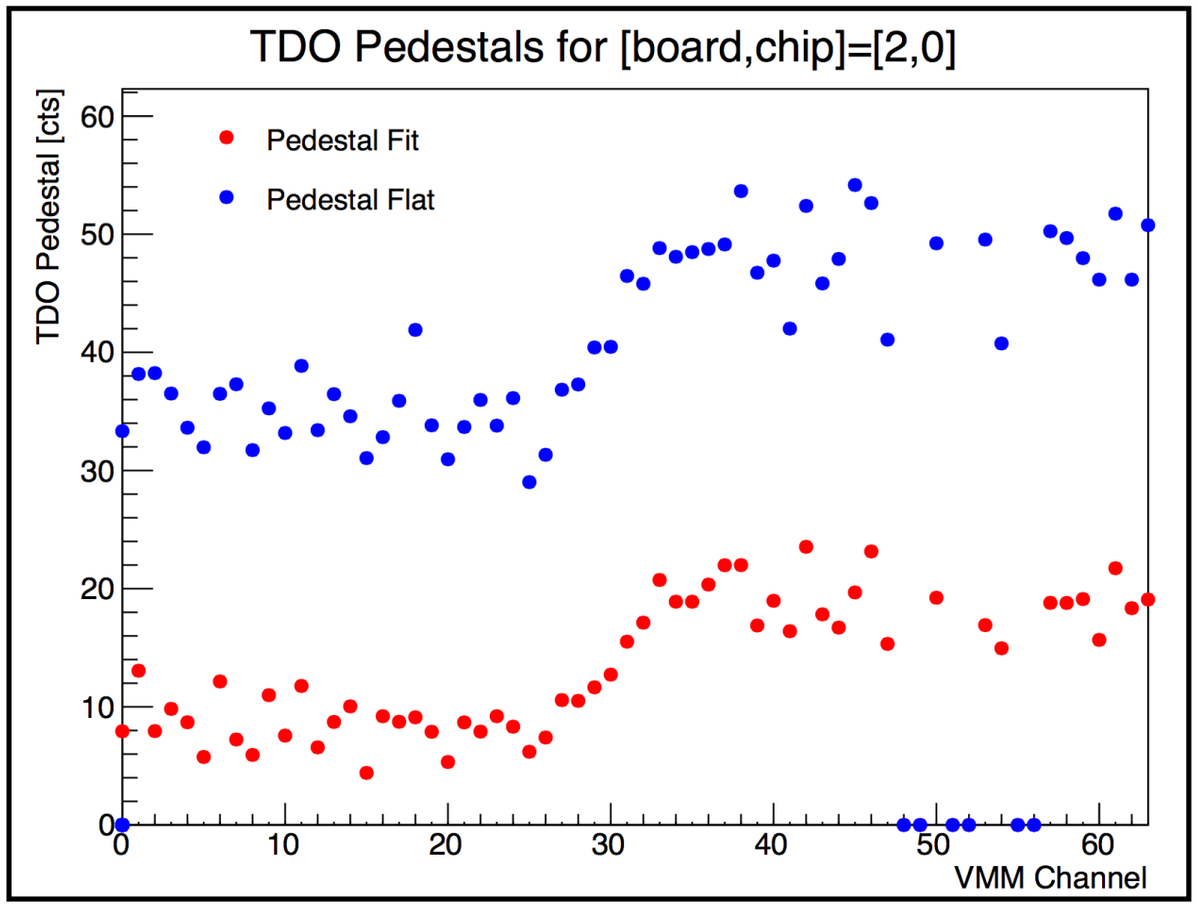
Two pedestal measures: y-intercept of linear fit and/or constant fit in "flat" region

There are two methods for timing calibration

CKBC Latency Delays

Having the pedestal correction proved useful during the test beam analysis, where we saw channel-by-channel pedestal variations in the TDO





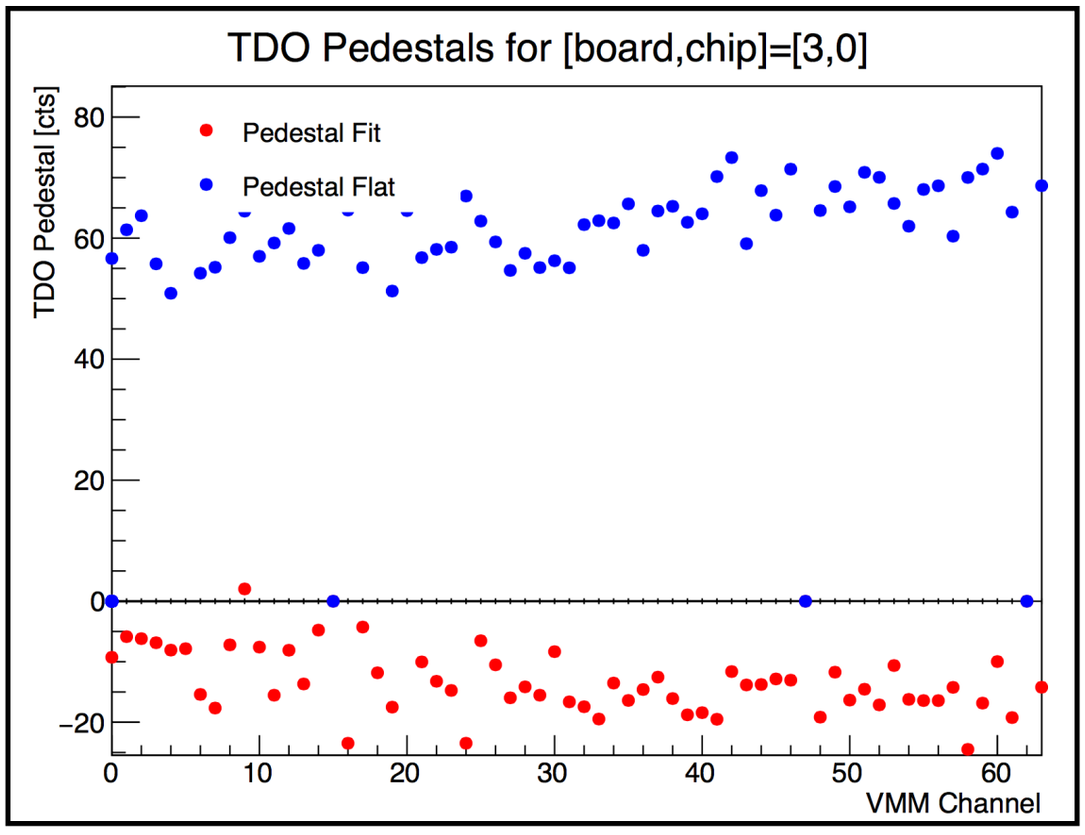
There are two methods for timing calibration

CKBC Latency Delays

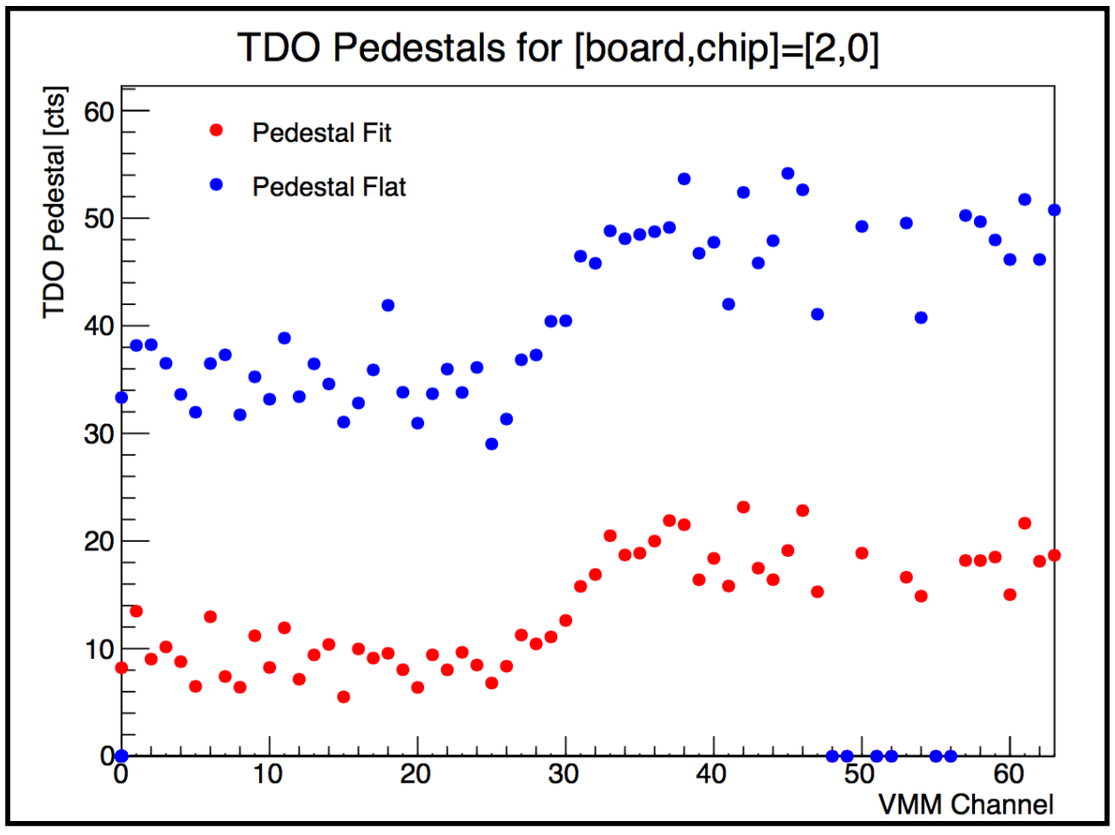
The conversion constants are different for different TAC slopes (of course)

We also saw/see dependence of the TDO pedestals on the TAC

TAC = 100 ns



TAC = 350 ns



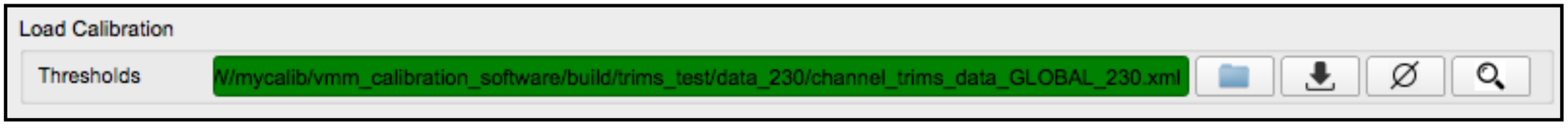
Calibration Overview

- All of these calibration analysis procedures are implemented in the software package vmm_calibration_software/ (except for CKTP skewing timing calibration) and run on the calibration n-tuples produced by VERSO

- All of the plots (and more) shown here are produced in this package, as well as the calibration parameters being stored in text/XML for later use in analysis/data taking

  - e.g. Channel trimmer calibration can be loaded into VERSO and during VMM configuration, each VMM and board will be configured with its chosen & equalized trimmer settings

  - Will work to incorporate the other calibration data (e.g. DAC + baseline calibration) into VERSO, though much is for off-line analysis




```
VERSO Info                    Setting threshold calibration file to:
VERSO Info                        /Users/dantrim/workarea/NSW/mycalib/vmm_calibration_software/build/trims_test/data_230/channel_trims_data_GLOBAL
VERSO Info                    _230.xml
VERSO Info                    Loading threshold calibration
VERSO Info                    Threshold calibration loaded successfully!
======================================== Loaded Threshold Settings ========================================
-----------------------------------------------------------------------------------------------------------
BOARD IP: 192.168.0.2
   CHIP  7 |    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
             23 16 13 27 20 10 31  6 30 30 31 17 13  0 16 12 18 31 31 20 13 19 18 31 27 16 30 26 12  3 16 31
           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
             32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
             30 14 31 24  0 14  5 23 31 31  4 23 31 10 21 25 10 28 31 16  0 31 24 27 31 25 12 31 16 21 29 25
           - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
===========================================================================================================
```