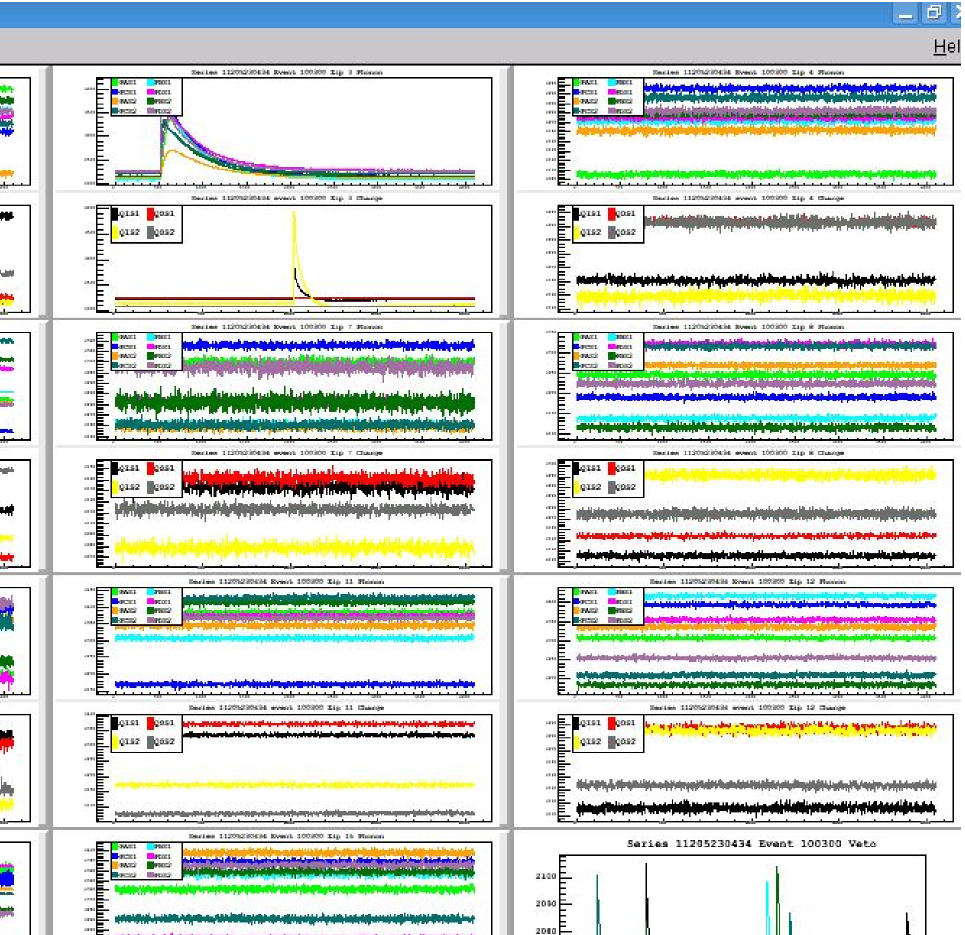
MIDAS Report from SCDMS-SNOLAB

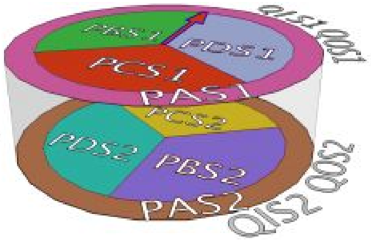
How we use MIDAS => Open Issues

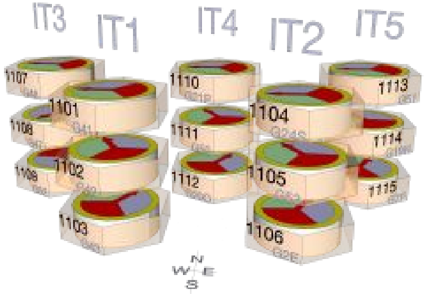
UBC: Belina von Krosigk, Danika MacDonell, Scott Oser, Bill Page, Andrew Scharff, Ben Smith
USD: Sudip Poudel, Amy Roberts, Joel Sander
UMN: Anthony Villano
A&M: Maxx Tepper, Xuji Xiao, Lei Zheng,
Toronto: Matt Wilson
Berkeley: Bruno Serfass

Amy Roberts, MIDAS workshop • 07.26.2017

What does SuperCDMS data look like?



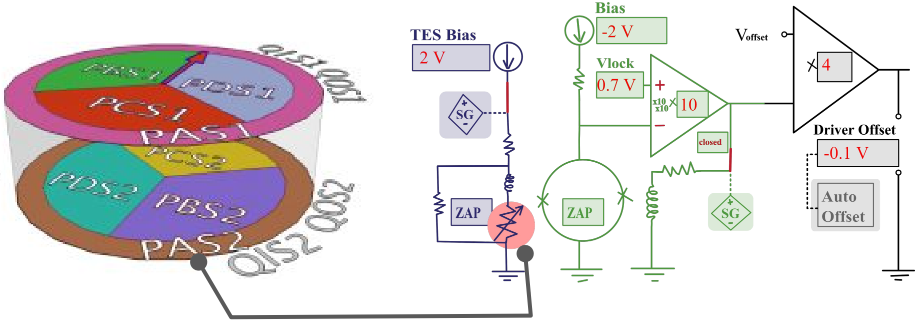




- Digitized pulses from many detectors, each with many channels
- The temperature of the detectors
- Detector settings (bias, current through sensors, etc.)

Take data with SuperCDMS

- User tunes 4+ detector channels
- Shine light on the detector be taking data!
- Take data with bias set to 0V for ~5 minutes
- Shine light on the detector again, keep detector working well
- Take data with bias set to 1V



T1_V0.mid.gz
ODB snapshot
random pulses
triggered pulses
ODB snapshot

T2_V0.mid.gz
ODB snapshot
triggered pulses
ODB snapshot

T3_V0.mid.gz
ODB snapshot
triggered pulses
random pulses
ODB snapshot

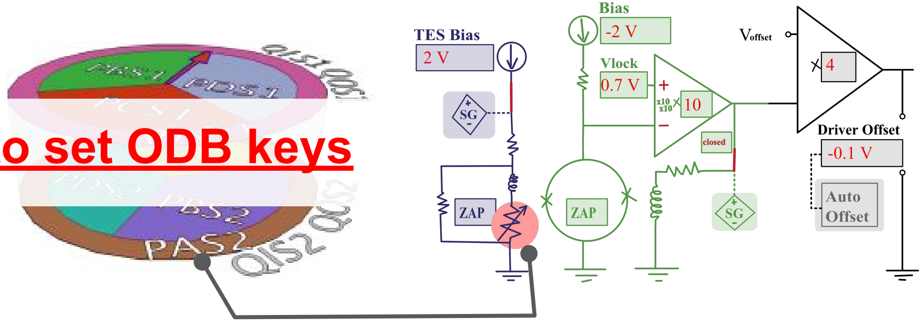
T1_1V.mid.gz

T2_1V.mid.gz

T3_1V.mid.gz

Take data with SuperCDMS

- User tunes 4+ detector channels
- Shine light on the detector be taking data!
- Take data with bias set to 0V for ~5 minutes
- Shine light on the detector again, keep detector working well
- Take data with bias set to 1V



Use mhttpd to set ODB keys

T1_V0.mid.gz
ODB snapshot
random pulses
triggered pulses
ODB snapshot

T2_V0.mid.gz
ODB snapshot
triggered pulses
ODB snapshot

T3_V0.mid.gz
ODB snapshot
triggered pulses
random pulses
ODB snapshot

T1_1V.mid.gz

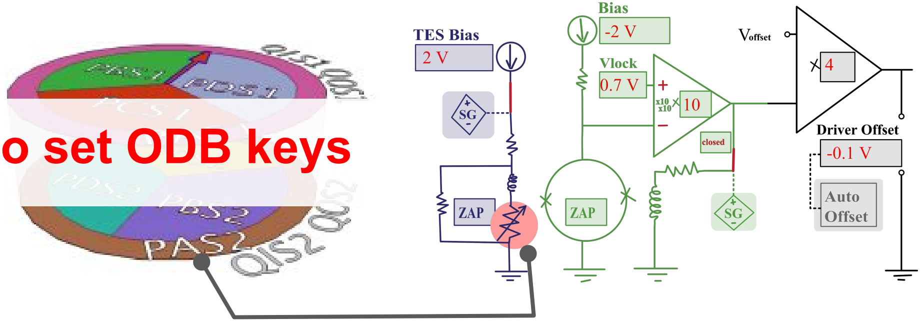
T2_1V.mid.gz

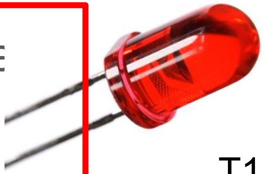
T3_1V.mid.gz

Take data with SuperCDMS

- User tunes 4+ detector channels
- Shine light on the detector be taking data!
- Take data with bias set to 0V for ~5 minutes
- Shine light on the detector again keep detector working well
- Take data with bias set to 1V

Use mhttpd to set ODB keys





T1_V0.mid.gz
ODB snapshot
random pulses
triggered pulses
ODB snapshot

T2_V0.mid.gz
ODB snapshot
triggered pulses
ODB snapshot

T3_V0.mid.gz
ODB snapshot
triggered pulses
random pulses
ODB snapshot

T1_1V.mid.gz

T2_1V.mid.gz

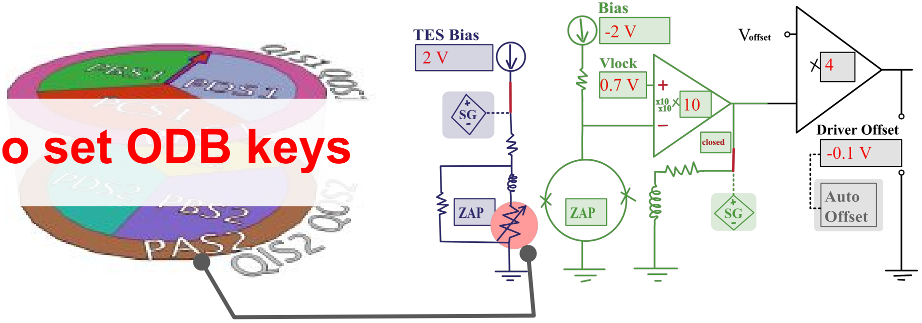
T3_1V.mid.gz

Sequencer

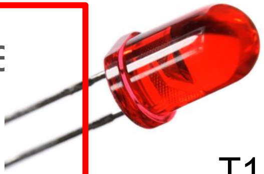
Take data with SuperCDMS

- User tunes 4+ detector channels

Use mhttpd to set ODB keys



- Shine light on the detector be taking data!



- Take data with bias set to 0V for ~5 minutes

MIDAS History to monitor detectors => adjust if needed

T1_V0.mid.gz

ODB snapshot
random pulses
triggered pulses
ODB snapshot

T2_V0.mid.gz

ODB snapshot
triggered pulses
ODB snapshot

T3_V0.mid.gz

ODB snapshot
triggered pulses
random pulses
ODB snapshot

- Shine light on the detector again keep detector working well

- Take data with bias set to 1V

Sequencer

T1_1V.mid.gz

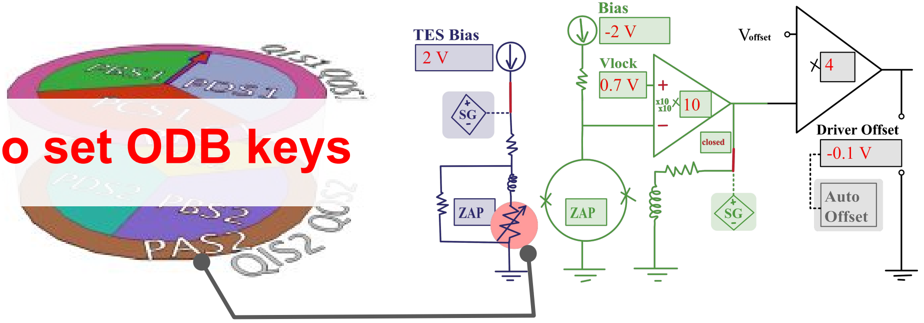
T2_1V.mid.gz

T3_1V.mid.gz

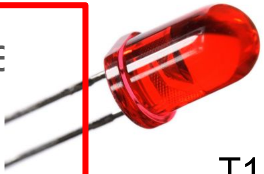
Take data with SuperCDMS

- User tunes 4+ detector channels

Use mhttpd to set ODB keys



- Shine light on the detector be taking data!



- Take data with bias set to 0V for ~5 minutes

MIDAS History to monitor detectors => adjust if needed

T1_V0.mid.gz
ODB snapshot
random pulses
triggered pulses
ODB snapshot

Start/Stop transitions to collect random pulses

triggered pulses
ODB snapshot

d.gz
ODB snapshot
triggered pulses
random pulses
ODB snapshot

- Shine light on the detector again keep detector working well

- Take data with bias set to 1V

Sequencer

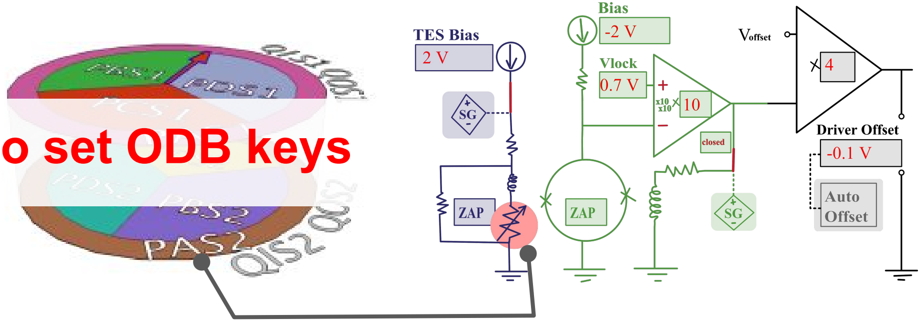
T1_1V.mid.gz

T2_1V.mid.gz

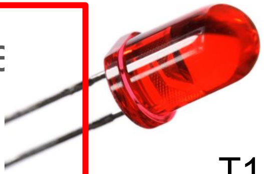
T3_1V.mid.gz

Take data with SuperCDMS

- User tunes 4+ detector channels
- Shine light on the detector be taking data!
- Take data with bias set to 0V for ~5 minutes
- Shine light on the detector again keep detector working well
- Take data with bias set to 1V



Use mhttpd to set ODB keys



MIDAS History to monitor detectors => adjust if needed

Sequencer

T1_V0.mid.gz

ODB snapshot
random pulses
triggered pulses
ODB snapshot

Start/Stop transitions to collect random pulses

ODB snapshot
triggered pulses
ODB snapshot

T3_V0.mid.gz

ODB snapshot
triggered pulses
random pulses
ODB snapshot

Frontends to control ODB

- **Precise timing**
- **locking**

T1_1V.mid.gz

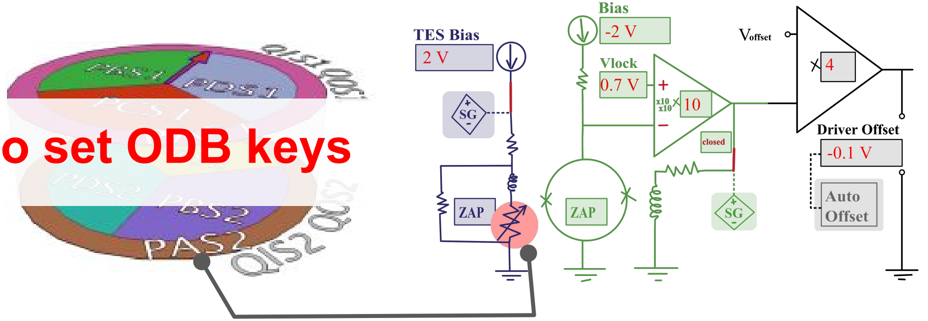
T2_1V.mid.gz

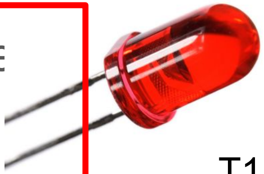
T3_1V.mid.gz

Take data with SuperCDMS

- User tunes 4+ detector channels
- Shine light on the detector be taking data!
- Take data with bias set to 0V for ~5 minutes
- Shine light on the detector again keep detector working well
- Take data with bias set to 1V

Use mhttpd to set ODB keys





MIDAS History to monitor detectors => adjust if needed

Start/Stop transitions to collect random pulses

T1_V0.mid.gz

ODB snapshot
random pulses
triggered pulses
ODB snapshot

T2_V0.mid.gz

ODB snapshot
triggered pulses
ODB snapshot

T3_V0.mid.gz

ODB snapshot
triggered pulses
random pulses
ODB snapshot

Frontends to control ODB

- **Precise timing**
- **locking**

Sequencer

T1_1V.mid.gz

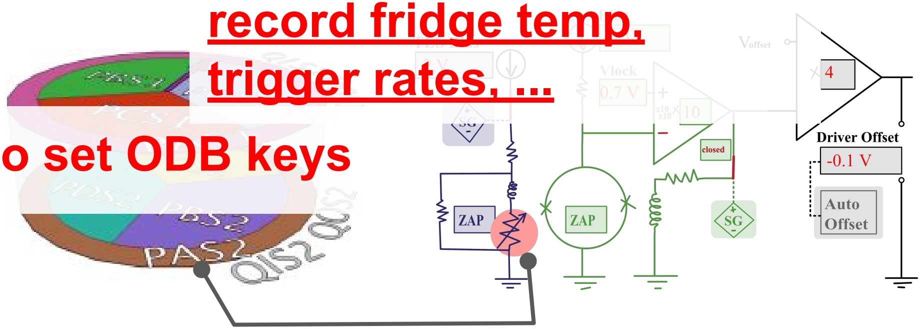
T2_1V.mid.gz

T3_1V.mid.gz

Lazylogger renames files

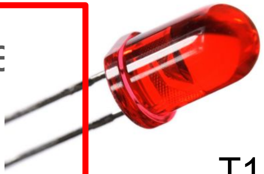
Take data with SuperCDMS

Use MIDAS History to record fridge temp, trigger rates, ...



Use mhttpd to set ODB keys

- User tunes 4+ detector channels
- Shine light on the detector be taking data!
- Take data with bias set to 0V for ~5 minutes
- Shine light on the detector again keep detector working well
- Take data with bias set to 1V



Use MIDAS History to monitor detectors => adjust if needed

Sequencer

T1_V0.mid.gz
ODB snapshot
random pulses
triggered pulses
ODB snapshot

T2_V0.mid.gz
ODB snapshot
triggered pulses
ODB snapshot

T3_V0.mid.gz
ODB snapshot
triggered pulses
random pulses
ODB snapshot

Start/Stop transitions to collect random pulses

Frontends to control ODB

- **Precise timing**
- **locking**

T1_1V.mid.gz

T2_1V.mid.gz

T3_1V.mid.gz

Lazylogger renames files

Open Issues

MIDAS History

Information that should be recorded together is not

Sequencer

Prone to crashing and will be too slow when we run with full set of detectors

Public API

Runs and sequencer are controllable through HTTP commands but these can change

mhttpd

Near-in-time requests over a typical public network often slow mhttpd response time to several minutes

History Bug Type 1: Bank-Writing Out-of-Syncness

1a) ‘SQL version’

Out-of-syncness between first and subsequent listed struct members when writing a struct to a history bank.

Present in:

- SQLITE history
- ODBC history
- MYSQL history (?)

1b) ‘MIDAS version’

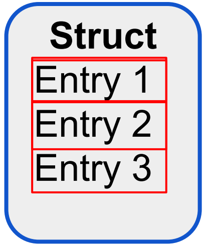
Out-of-syncness between first and subsequent banks when writing to multiple history banks at once.

Present in:

- MIDAS history
- FILE history

More on Bug Type 1a - ‘SQL Version’

Struct defined with multiple entries:



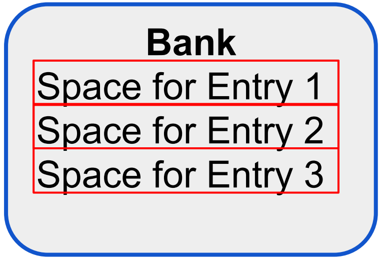
Struct

Entry 1

Entry 2

Entry 3

Bank defined to contain the struct:

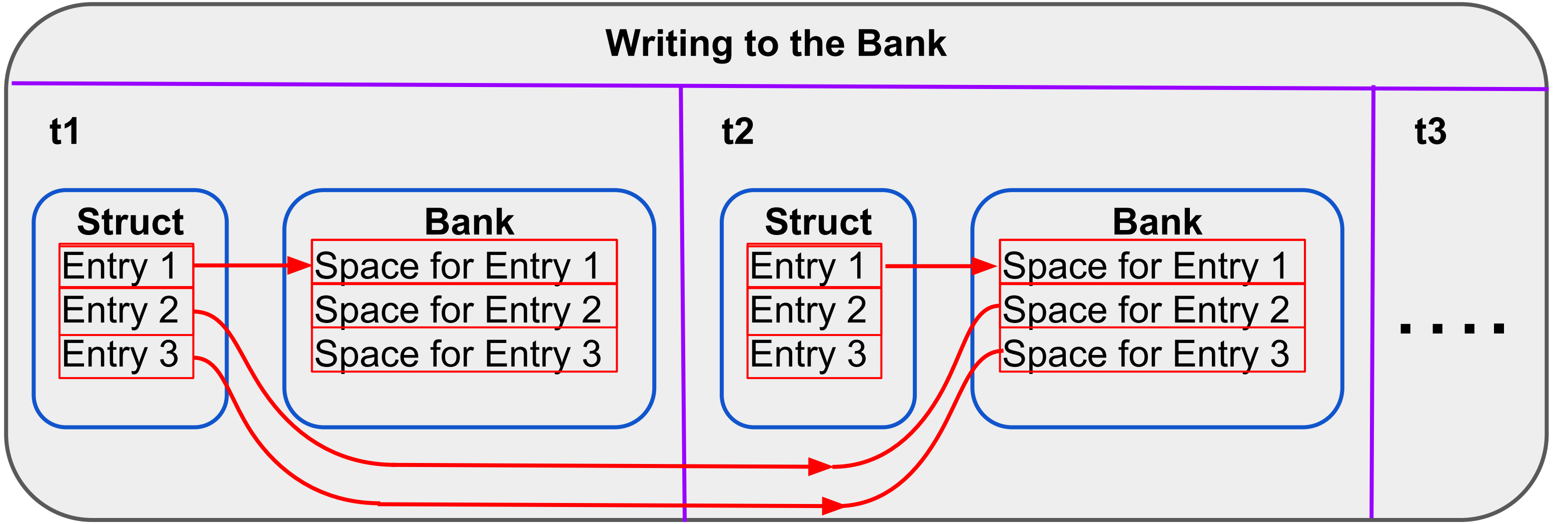


Bank

Space for Entry 1

Space for Entry 2

Space for Entry 3



Writing to the Bank

t1

Struct

Entry 1

Entry 2

Entry 3

Bank

Space for Entry 1

Space for Entry 2

Space for Entry 3

t2

Struct

Entry 1

Entry 2

Entry 3

Bank

Space for Entry 1

Space for Entry 2

Space for Entry 3

t3

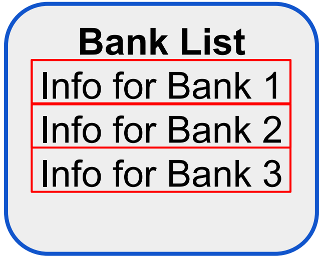
. . . .

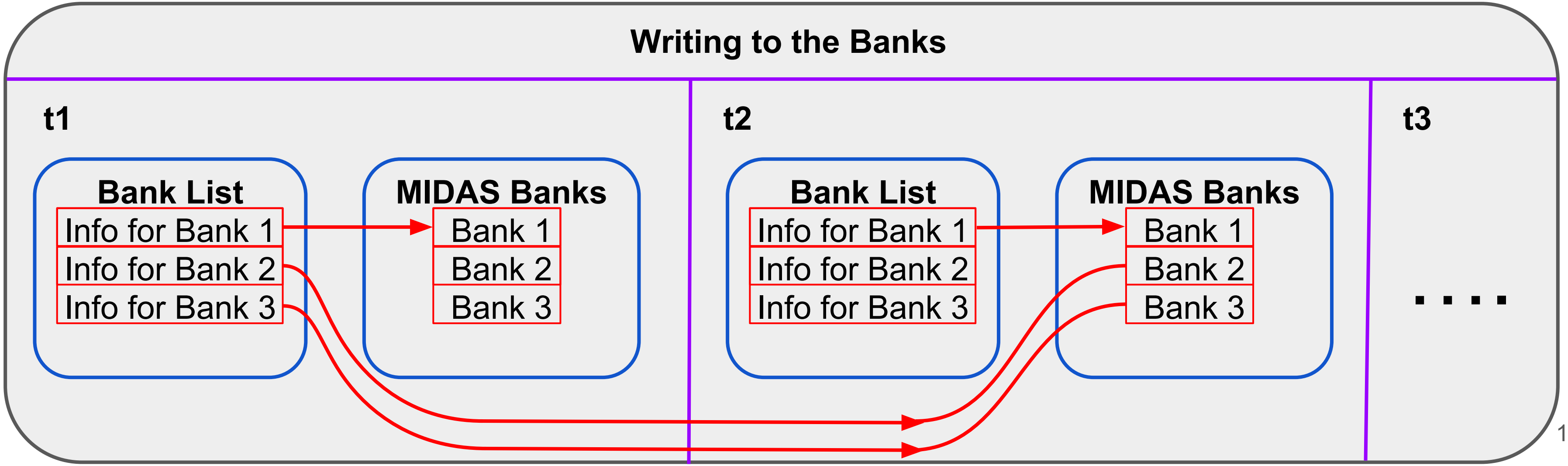
Point of Interest

This bug only appears when connecting to MIDAS through an mserver.

More on Bug Type 1b - ‘MIDAS Version’

Bank list defined to contain bank info. The banks may contain any variable types (int, struct, array, etc.)





History Bug Type 2: Slow Bank-Writing with SQLITE

When writing to the banks using SQLITE history:

- It takes ~6-12 seconds for bank entries to appear in and become readable from the bank.
- This is an issue for programs that need to read and write new bank data at $\mathcal{O}$(1s) rates.

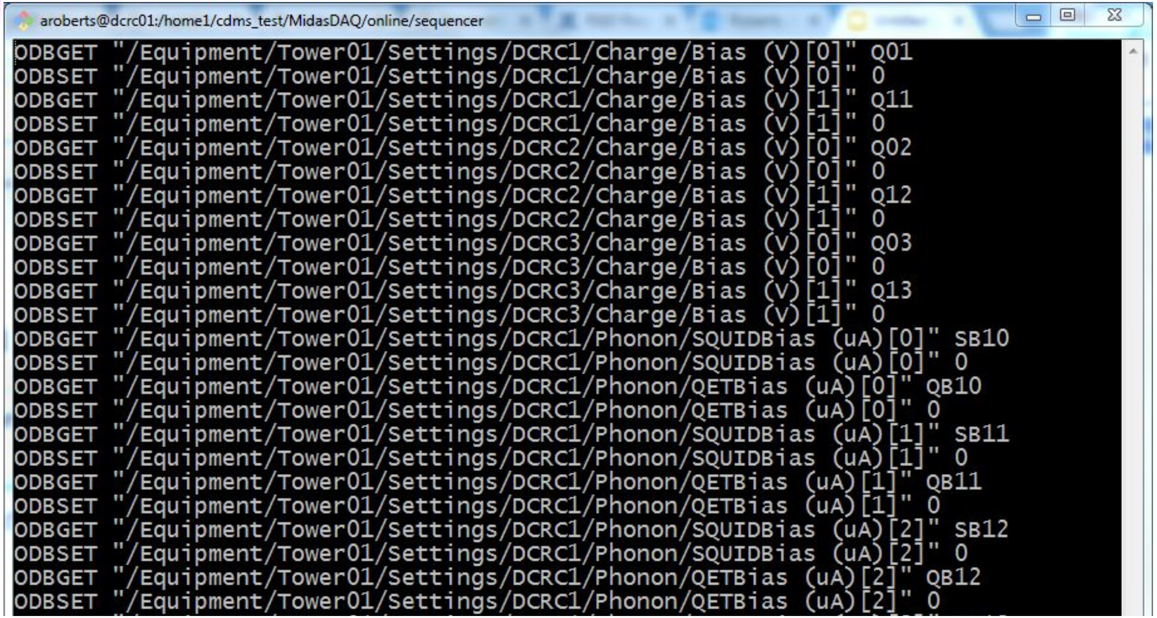
When writing to the banks with other history systems, it takes less than 1 second for bank entries to appear and become readable after they are written to the bank.

Sequencer: fantastic for detector development

Parses commands

Controllable via HTTP requests

Provides status info



```
ODBGET "/Equipment/Tower01/Settings/DCRC1/Charge/Bias (V)[0]" Q01
ODBSET "/Equipment/Tower01/Settings/DCRC1/Charge/Bias (V)[0]" 0
ODBGET "/Equipment/Tower01/Settings/DCRC1/Charge/Bias (V)[1]" Q11
ODBSET "/Equipment/Tower01/Settings/DCRC1/Charge/Bias (V)[1]" 0
ODBGET "/Equipment/Tower01/Settings/DCRC2/Charge/Bias (V)[0]" Q02
ODBSET "/Equipment/Tower01/Settings/DCRC2/Charge/Bias (V)[0]" 0
ODBGET "/Equipment/Tower01/Settings/DCRC2/Charge/Bias (V)[1]" Q12
ODBSET "/Equipment/Tower01/Settings/DCRC2/Charge/Bias (V)[1]" 0
ODBGET "/Equipment/Tower01/Settings/DCRC3/Charge/Bias (V)[0]" Q03
ODBSET "/Equipment/Tower01/Settings/DCRC3/Charge/Bias (V)[0]" 0
ODBGET "/Equipment/Tower01/Settings/DCRC3/Charge/Bias (V)[1]" Q13
ODBSET "/Equipment/Tower01/Settings/DCRC3/Charge/Bias (V)[1]" 0
ODBGET "/Equipment/Tower01/Settings/DCRC1/Phonon/SQUIDBias (uA)[0]" SB10
ODBSET "/Equipment/Tower01/Settings/DCRC1/Phonon/SQUIDBias (uA)[0]" 0
ODBGET "/Equipment/Tower01/Settings/DCRC1/Phonon/QETBias (uA)[0]" QB10
ODBSET "/Equipment/Tower01/Settings/DCRC1/Phonon/QETBias (uA)[0]" 0
ODBGET "/Equipment/Tower01/Settings/DCRC1/Phonon/SQUIDBias (uA)[1]" SB11
ODBSET "/Equipment/Tower01/Settings/DCRC1/Phonon/SQUIDBias (uA)[1]" 0
ODBGET "/Equipment/Tower01/Settings/DCRC1/Phonon/QETBias (uA)[1]" QB11
ODBSET "/Equipment/Tower01/Settings/DCRC1/Phonon/QETBias (uA)[1]" 0
ODBGET "/Equipment/Tower01/Settings/DCRC1/Phonon/SQUIDBias (uA)[2]" SB12
ODBSET "/Equipment/Tower01/Settings/DCRC1/Phonon/SQUIDBias (uA)[2]" 0
ODBGET "/Equipment/Tower01/Settings/DCRC1/Phonon/QETBias (uA)[2]" QB12
ODBSET "/Equipment/Tower01/Settings/DCRC1/Phonon/QETBias (uA)[2]" 0
```

Simple commands in a text file, => scientists testing detectors can write useful DAQ sequences.

Even without expert DAQ knowledge!

This has never happened before and significantly shortens the time to do on-the-fly tests.

Sequencer: issues

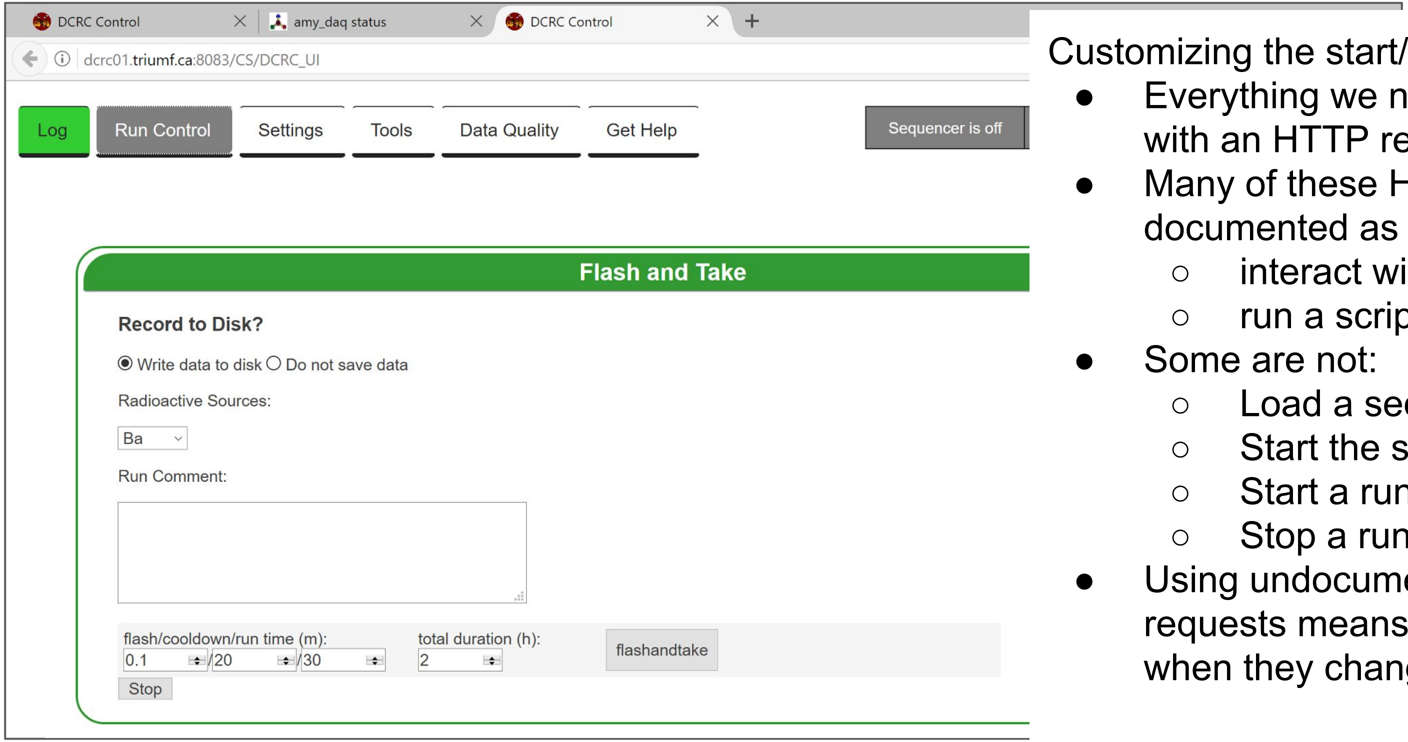
- **If mhttpd crashes, the sequencer crashes.**
    - Separate sequencer from mhttpd?
- **Fixed delay (1 second) between line execution.**
    - Setting up detectors could dominate deadtime!
- **Flow control for the production DAQ will be complex**
    - Example: the fridge system wants to take a noisy measurement.

        (1) wait until we’ve finished the current run

        (2) take the noisy measurement

        (3) start the next run

Public API: HTTP requests



Record to Disk?

● Write data to disk ○ Do not save data

Radioactive Sources:

Ba

Run Comment:

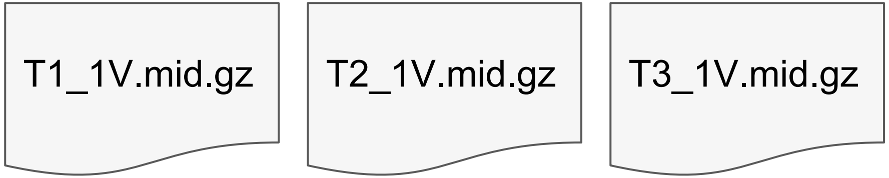
flash/cooldown/run time (m): 0.1 /20 /30

total duration (h): 2

Customizing the start/stop run interface:

- Everything we needed can be done with an HTTP request!
- Many of these HTTP requests are documented as a public API
  - interact with ODB
  - run a script
- Some are not:
  - Load a sequencer file
  - Start the sequencer
  - Start a run
  - Stop a run
- Using undocumented HTTP requests means our code breaks when they change

Public API: MIDAS C/C++ library



T1_1V.mid.gz

T2_1V.mid.gz

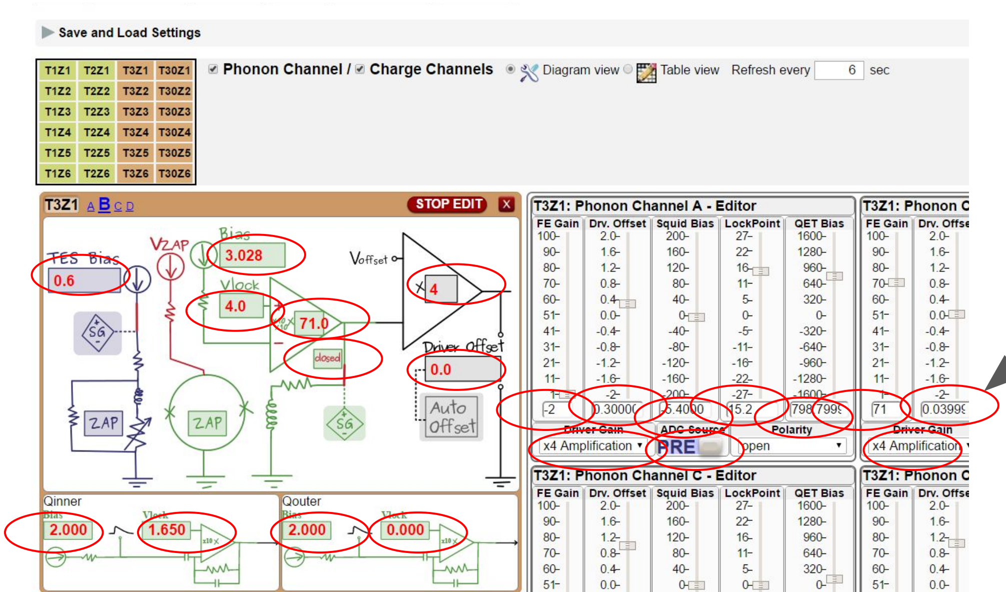
T3_1V.mid.gz

Detector Monte-Carlo programs also generate data files

- Ideally, format would be identical to the format of the DAQ data files!
- The MIDAS bank structure is simple and we could replicate it
- Or link to existing MIDAS bank-writer functions?

mhttpd: two users => slow response

- We send HTTP requests for ODB data about every second
- Two users working with the detector-tuning interface => mhttpd can take minutes to respond
- Slow-down may be triggered by close-in-time requests



▶ Save and Load Settings

T1Z1	T2Z1	T3Z1	T30Z1
T1Z2	T2Z2	T3Z2	T30Z2
T1Z3	T2Z3	T3Z3	T30Z3
T1Z4	T2Z4	T3Z4	T30Z4
T1Z5	T2Z5	T3Z5	T30Z5
T1Z6	T2Z6	T3Z6	T30Z6

☑ Phonon Channel / ☑ Charge Channels ● Diagram view ○ Table view Refresh every 6 sec

T3Z1 A B C D STOP EDIT X

TES Bias 0.6

Bias 3.028

Vlock 4.0

71.0

x 4

Driver Offset 0.0

Qinner Bias 2.000 Vlock 1.650

Qouter Bias 2.000 Vlock 0.000

T3Z1: Phonon Channel A - Editor

FE Gain -2 | Drv. Offset 0.30000 | Squid Bias 5.4000 | LockPoint 15.2 | QET Bias 798.7999

Driver Gain x4 Amplification | ADC Source PRE | Polarity open

T3Z1: Phonon C…

FE Gain 71 | Drv. Offse… 0.03999

Driver Gain x4 Amplification

T3Z1: Phonon Channel C - Editor

T3Z1: Phonon C…

All this data is in the ODB and comes from an HTTP request to mhttpd.

The End!

1. MIDAS has allowed us to build a usable DAQ for detector testing

2. The HTTP API and Sequencer have been extremely useful

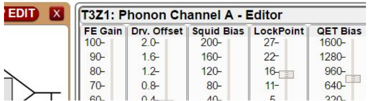
3. Our production DAQ has to address mhttpd slowness, sequencer slowness, and History bugs

4. Our production DAQ has to allow complex flow-control

The SNOLAB DAQ

Approximate Dependency

DCRC UI (dcrc_ui.git)



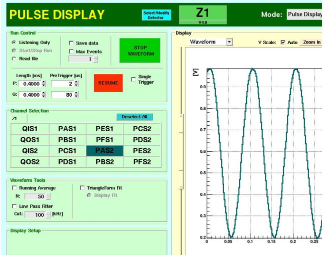
T3Z1: Phonon Channel A - Editor

FE Gain	Drv. Offset	Squid Bias	LockPoint	QET Bias
100	2.0	200	27	1600
90	1.6	160	22	1280
80	1.2	120	16	960
70	0.8	80	11	640
60	0.4	40	5	320

- Control detectors, data runs
- Amy, Xuji et al.
- Uses code in MidasDAQ to Flash, etc.

Midas Tools (MidasTools.git)

- Tune SQUIDs, measure Rp/Rn
- Bruno, Bill et al.
- Reads data provided by frontends

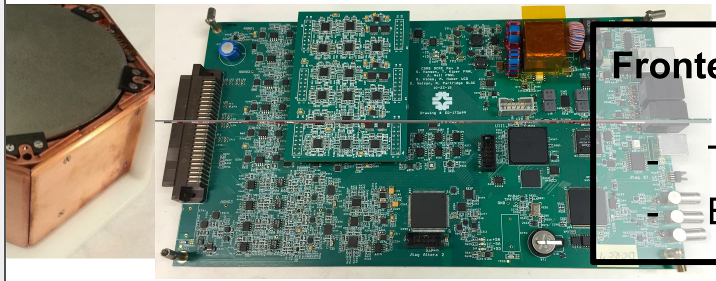


● Listening Only
○ Start/Stop Run
○ Read file
□ Save data
□ Max Events 1
Length [ms] P: 0.4000 Q: 0.4000
PreTrigger [us] P: 2 Q: 80
□ Single Trigger
☑ Auto

Z1

QIS1	PAS1	PES1	PCS2
QOS1	PBS1	PFS1	PDS2
QIS2	PCS1	PAS2	PES2
QOS2	PDS1	PBS2	PFS2

□ Running Average N: 50
□ Low Pass Filter Cut: 100 [kHz]
□ TriangleForm Fit
○ Display Fit



Frontends (MidasDAQ.git)

- Talk to our DCRC hardware
- Belina et al.

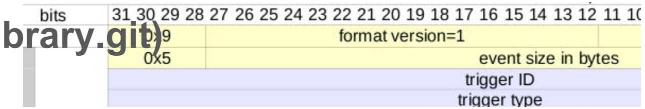
cdmsbats (cdmsbats.git)

- Read and process the data
- Bruno, Anthony et al.

MIDAS (midas_fork.git)

- General framework, not specific to CDMS
- Thomas Lindner et al.
- Using MIDAS Sequencer, History (Danika), etc.

Data I/O (IOLibrary.git)



bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10
	0x9				format version=1																	
	0x5				event size in bytes																	
	trigger ID																					
	trigger type																					

- Packs/Unpacks CDMS data
- Matt Wilson et al.
- First use in frontends