Software Architecture of a RadFET Dosimetry System

Garth Brown: 20 May 2015

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

SLAC National Accelerator Laboratory Overview

- New system offered many design decisions
- Collaboration with radiation physicists, electrical engineers, mechanical engineers, software engineers, to design at a system level
- Process of this system design, decisions needed to meet requirements
- Matching functionality to the part of the system best suited
- Chance to implement both sides of networked device control, custom serial protocol between instrument and IOC

Background

The Project:

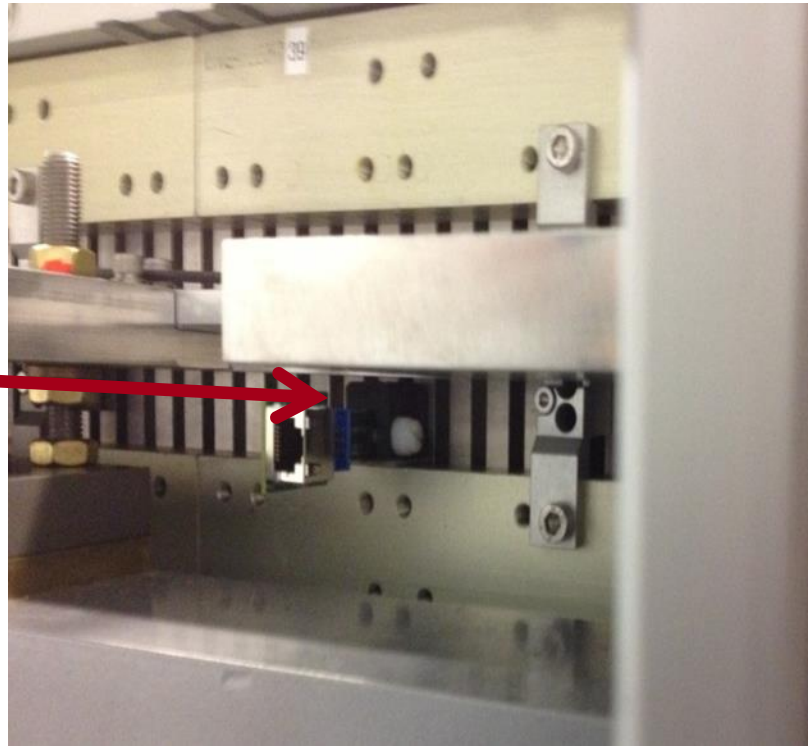
- Monitor radiation at many locations throughout the LCLS undulator
- Determine radiation damaging to undulator magnets, its sources, causes, and locations
- Extrapolate to LCLS II, and potential for radiation damage rate proportional to beam rate
- Track radiation accumulated dose over minutes to months

SLAC National Accelerator Laboratory
Background

The Sensors:

- RadFETs: When the metal oxide layer of MOSFETs are damaged, it changes the electrical characteristics in precisely measurable ways that persist for years. Thumbnail size. Well established technology

3D Printed Sensor Mount, Non-magnetic Materials



RadFET operation:

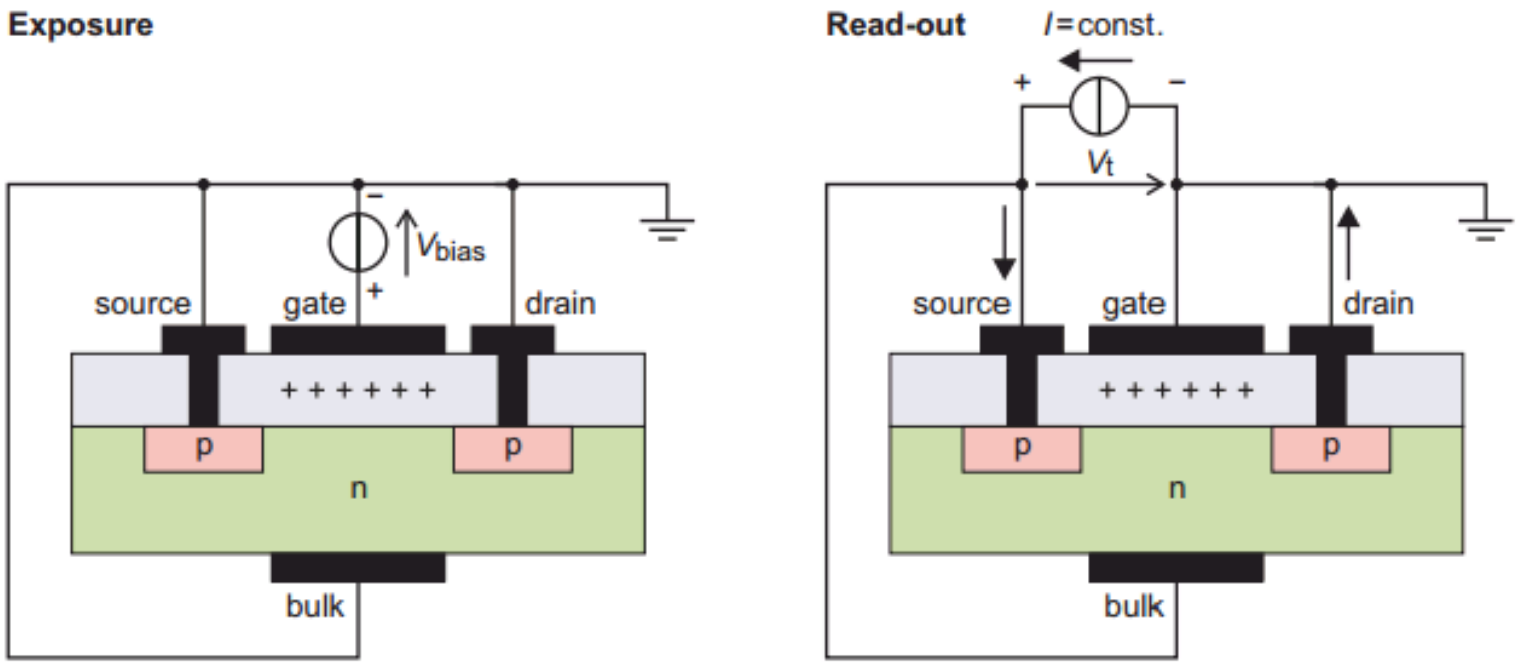


Figure 1: Schematic representation of a RADFET in exposure and readout configuration.

Source: http://accelconf.web.cern.ch/AccelConf/ibic2013/papers/tupc45.pdf

The Sensor System:

- Keithley 2600B: supports TestScript Lua, voltage and current sources, voltage and current measurements
- Keithley 3700S: provides an array of relays, controllable by TestScript
- The Keithley 2600B/3700S system: Source meter orchestrates sampling sequence, sample one RadFET connected to current source while biasing the rest with voltage source. **Up to 168 RadFETs**.
- Readout cycle relay switching sequence designed to read voltage across one RadFET at a time

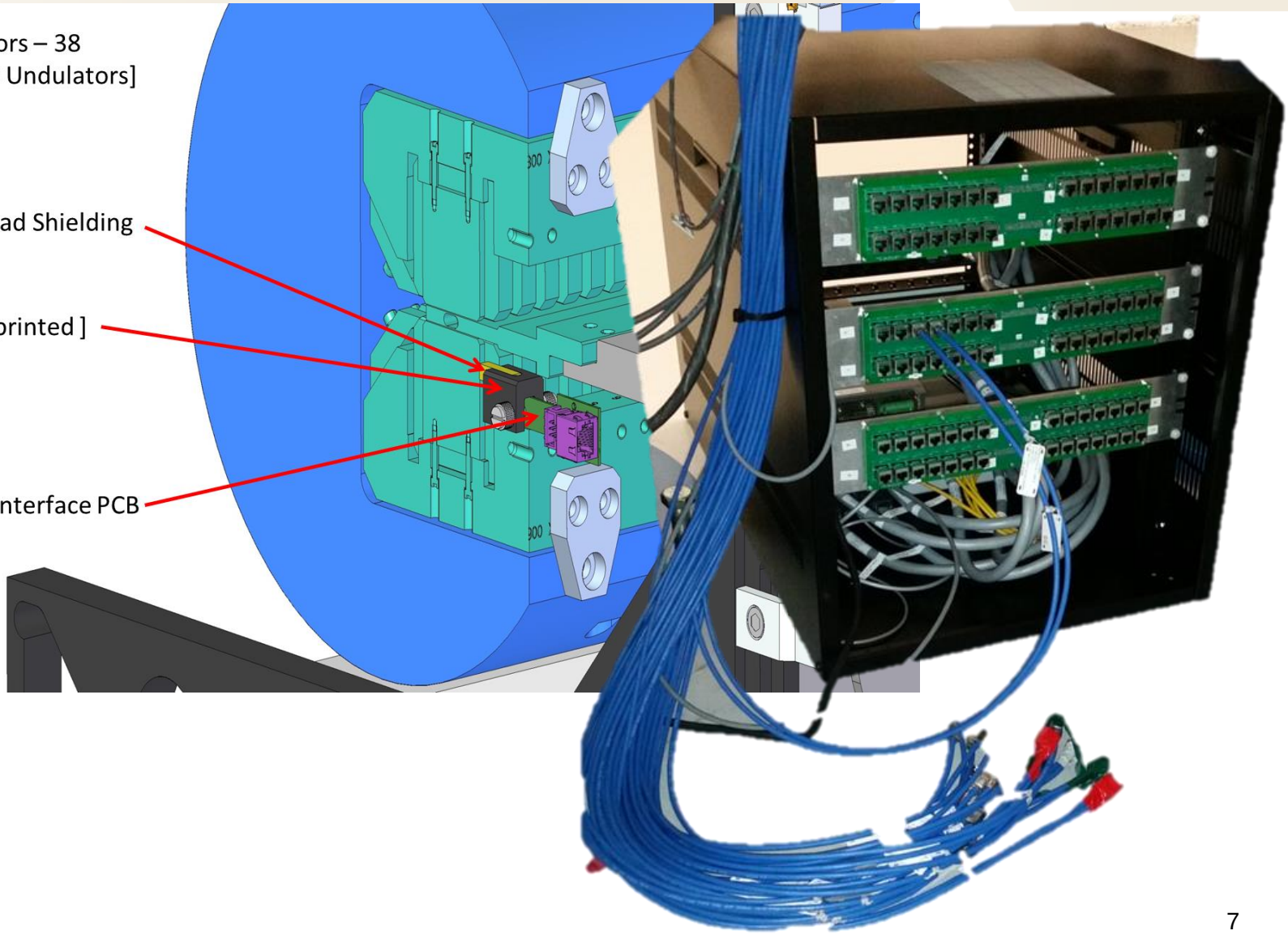
SLAC National Accelerator Laboratory

of Sensors – 38
[2 Per 19 Undulators]

Lead Shielding

Sensor Mount [3-D printed]

RADFET Interface PCB



RadFET readout

- Readout sequence for 1 RadFET
  - Open switch for bias voltage
  - Close switch for current source
  - Pause for precisely “Delay” seconds
  - Read voltage at drain
  - Open switch for current source
  - Close switch for bias voltage
  - Repeat for each RadFET at regular intervals (minutes)

Design considerations

- Use Keithley's Test Script Builder (TSB) Lua extension to implement protocol on the instrument side, EPICS streamdevice on the soft IOC side
- Voltage read is sensitive to the time between applying constant current source and taking voltage reading
- Dose isn't accumulated during reading cycle. It should be a small and consistent fraction of total time.
- Physicist want flexibility in converting raw reading to dose, so make the equation easily configurable.
- Keep serial interface between streamdevice and Keithly system clean and simple

Design considerations

- Initial plan: IOC requests reading, instrument takes reading and sends it back
- Conflicting requirement: EEs want all-channel read cycle, make-before-break switching from one RadFET to the next less stress and noise from constant current source
- Next idea: instrument returns array of all readings
- Software complication: With expected 1 second per RadFET read cycle, up to 168 RadFETs, that’s almost 3 minutes from request to reply. Long timeout! Data timestamped late.
- Better approach: IOC initiates read cycle. Instrument asynchronously sends readings. No timestamp problem.

SLAC National Accelerator Laboratory
Design Considerations: In the TestScript

Relay switching control and time

- Voltage across RadFET is not constant when connected to constant current course. Time from applying current source to reading voltage must be consistent across readings for consistent data.
- In all-sensor read cycle, keep time to switch from one sensor to the next to a minimum, and consistent.
- Design choice: switching/reading cycle implemented completely in TestScript
  - No network delays to worry about
  - No EPICS processing time variations to worry about
  - Keeps interface between IOC and instrument simpler and cleaner
  - Required learning more TSB/Lua programming

Design Considerations: In the IOC

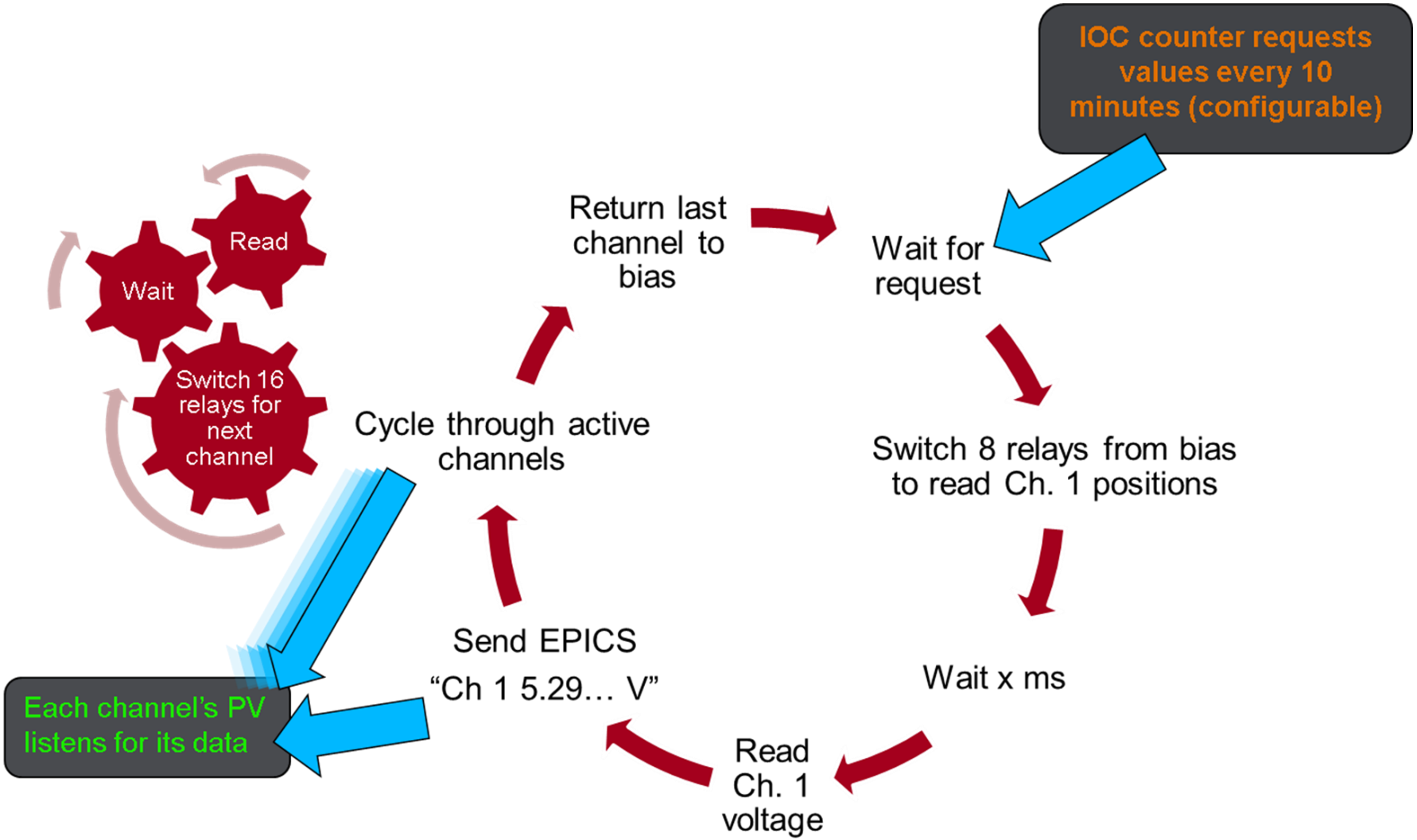
Read cycle interval

- Doesn't need precision below 1s
- We have already implemented "process this PV every x seconds", code reuse
- Design decision: implement this in the IOC
  - If the IOC stops, instrument doesn't know or care

Raw reading to dose calculation

- Calc record is flexible and has screens that let physicists try whatever they want
- One per RadFET, for flexibility and per-RadFET calibration

SLAC National Accelerator Laboratory
Keithley TestScript & IOC Architecture



RadFET System Protocol

- Simple readout protocol:
  - Init – IOC request to create a list of attached sensors by checking each channel for an open circuit
  - ReadAll – IOC request to cycle through all active channels. Asynchronously send each reading to IOC.
  - ReadCh – Receive asynchronous data from ReadAll, keyed on channel #.
    - Protocol file looks like: **ReadCh { in "Sensor \$1 %f";}**
    - PVs corresponding to disconnected channels exist, but never process

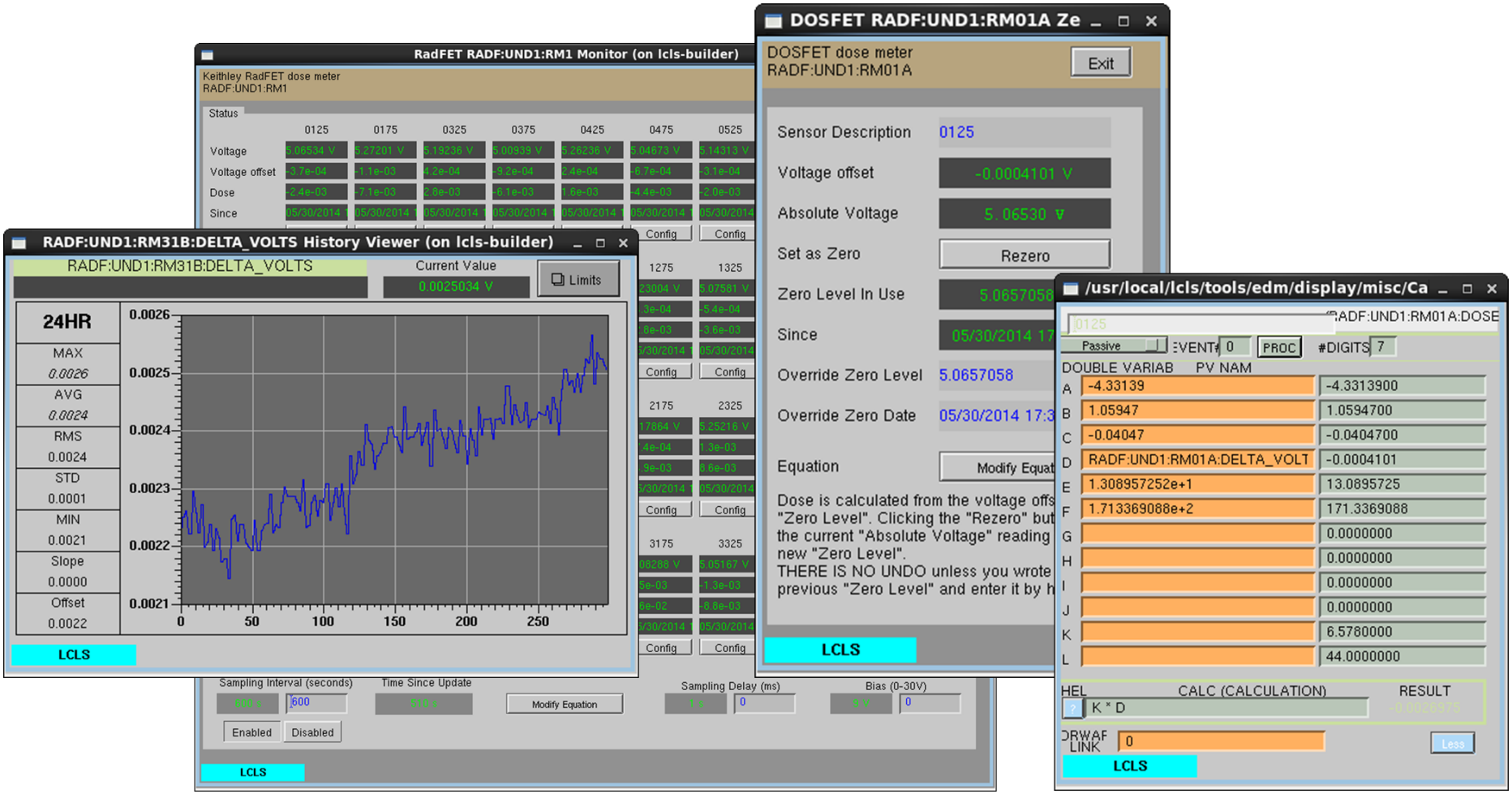
Other commands

- Get/SetDelay – time from current on to voltage measurement
- Get/SetBias – Voltage applied during “exposure”
- Get/SetSampleI – What current to apply during reading
- A few others added for testing, commissioning, trouble shooting

Other data in the IOC

- History of dose readings, over various time scales, using Bruce Hill’s history module
- Initial raw voltage reading – dose is calculated as a ΔV from initial reading
- Date & time of initial reading
- Channel description

EPICS EDM Screens: Configure and Readout



Conclusion

- Looking at a design from a system level solves and simplifies problems that look messy on a smaller scale
- Dive in and learn new tools and languages. It increases your options
- Keep it simple. This system has been running mostly trouble free since last August