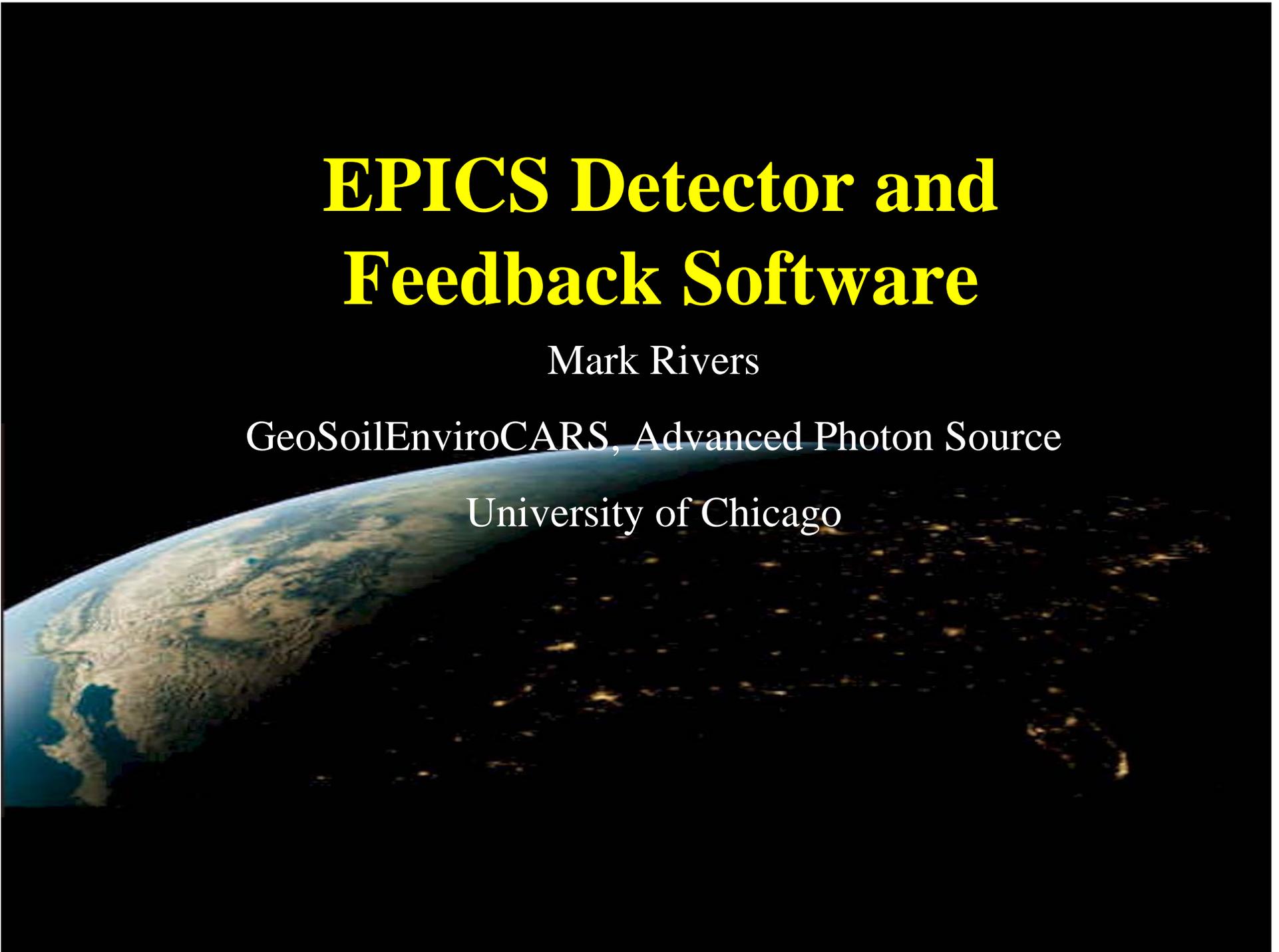


# **EPICS Detector and Feedback Software**

Mark Rivers

GeoSoilEnviroCARS, Advanced Photon Source

University of Chicago

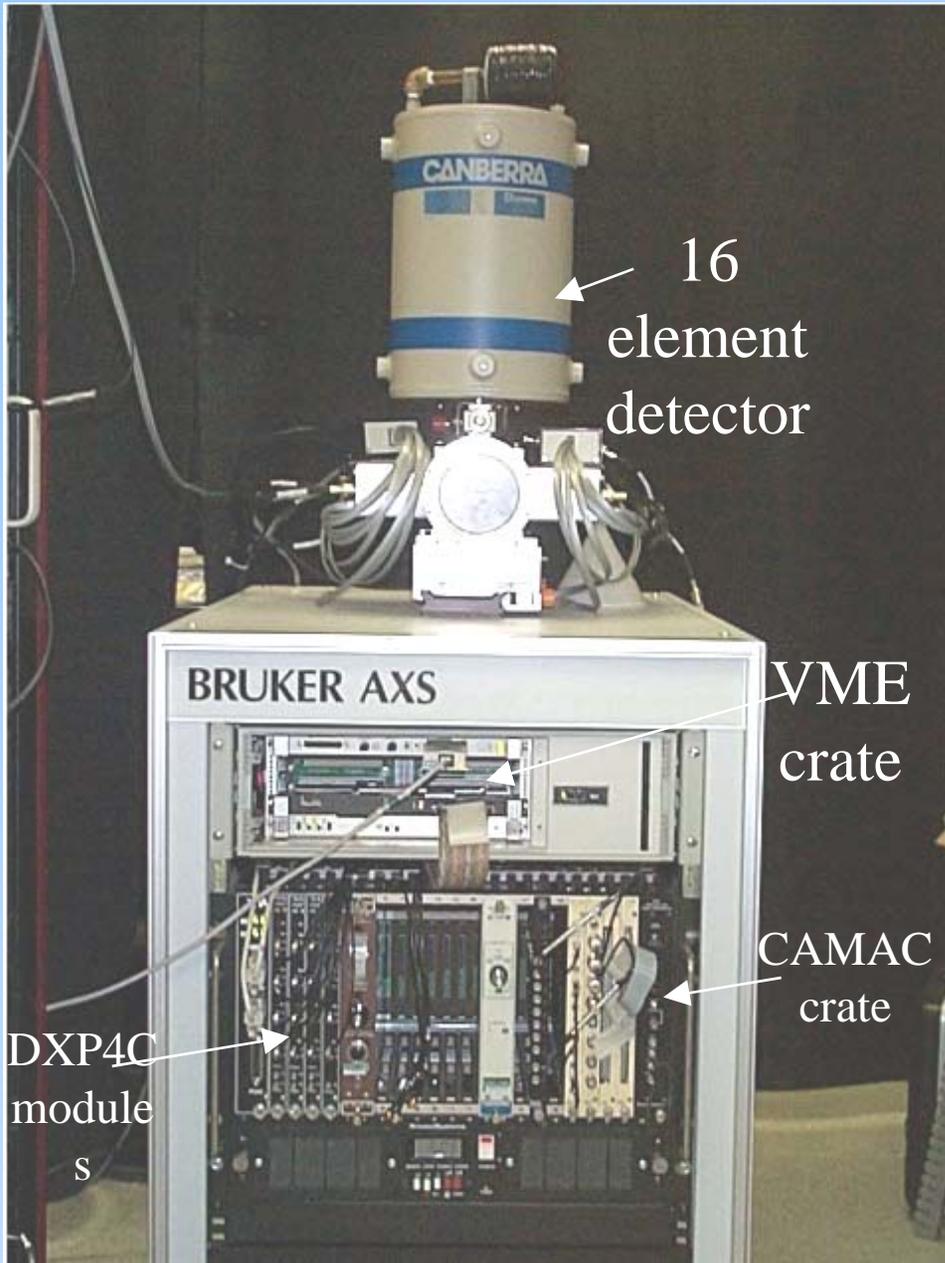


# Outline

- EPICS Interface to Canberra Electronics
- EPICS Interface to XIA DXP Electronics for Energy Dispersive Detectors
- ccdApp: Generic EPICS Interface to Area Detectors
- smartControl: Using Bruker SMART to Control EPICS Experiments
- Generic Feedback under EPICS
- APS Beam Position Monitor and EPICS Software

# Fast DSP Electronics for EDS Detectors

- DXP4C2X are digital signal processing based x-ray spectrometers from X-ray Instrumentation Associates (XIA). These CAMAC modules provide a very fast and cost-effective solution for data acquisition with multi-element energy-dispersive x-ray detectors.
- I have developed software to control these modules under EPICS. This software consists of device support for the standard **EPICS MCA record** and a new **DXP record**.
- MCA record
  - Start and stop data acquisition
  - Readout the spectra
  - Control and read the data acquisition time
  - Define up to 32 Regions of Interest (ROIs) for computing the net or total counts in each fluorescence peak.
- DXP record
  - Provides complete control over the internal operation of the DXP
  - More than 50 adjustable parameters.
- The EPICS software can in turn be controlled by Channel Access clients. We use IDL and medm to control, display and store data from our detectors.
- Next generation will be PXI based, rather than CAMAC.



16  
element  
detector

VME  
crate

CAMAC  
crate

DXP4C  
module  
s

# Software Control – using EPICS

16element\_dxp.adl

### 16 Element Detector Control

Start Stop Acquire  
Erase/Start Erase

Done Status  
14.67 Elapsed real time  
0.00 Preset real time  
.5 second Status rate  
5 second MCA Read rate  
5 second DXP Read rate  
Disable Wait for client  
Done Client Wait

DXPs  
 ROIs 0-4  
 ROIs 5-9  
 Calibration  
 Acquisition time  
 Combined Plots  
 Plots 1-8  
 Plots 9-16

16element\_ROI.adl

### 16 Element Detector - ROI 0

Det.	Label	Low	High	nAvg	Sum	Net
1	Mn Ka	1051	1177	0	0.000	0.000
2	Mn Ka	940	1053	0	0.000	0.000
3	Mn Ka	1027	1150	0	0.000	0.000
4	Mn Ka	1174	1315	0	0.000	0.000
5	Mn Ka	1098	1230	0	0.000	0.000
6	Mn Ka	1033	1157	0	0.000	0.000
7	Mn Ka	1080	1210	0	0.000	0.000
8	Mn Ka	865	969	0	0.000	0.000
9	Mn Ka	953	1068	0	0.000	0.000
10	Mn Ka	1275	1428	0	0.000	0.000
11	Mn Ka	1084	1214	0	0.000	0.000
12	Mn Ka	1062	1190	0	0.000	0.000
13	Mn Ka	1032	1156	0	0.000	0.000
14	Mn Ka	1058	1184	0	0.000	0.000
15	Mn Ka	1035	1159	0	0.000	0.000
16	Mn Ka	1042	1167	0	0.000	0.000



13GE2:med:dxp1

### Analog Signal Conditioner

COARSEGAIN

FINEGAIN

VRYFINGAIN

OFFDACVAL

SDACREF

SDACNOM

SLOPEVAL

TRACKLST

TRACKRST

### Readout Parameters

CODEREV

RUNIDENT

RUNERROR

ERRINFO

DECIMATION

BASEVAL

BASESLOPE

DACPERADC

DADCDT

Unused

Unused

Unused

Unused

Unused

Unused

### Run Statistics

EVTSINRUNO

UNDRFLOWSO

OVERFLOWSO

FASTPEAKSO

BASEEVTSO

BASEMEANO

LIVETIMEO

NUMASCINTO

NUMRETSO

NUMDRDOSO

NUMDRUPS0

NUMUPSETS0

NUMZIGZAGO

Unused

### Acquire

### FIPPI Parameters

POLARITY

FASTGAP

FASTLEN

SLOWGAP

SLOWLEN

MAXWIDTH

MINWIDTH

PEAKINT

PEAKSAMP

THRESHOLD

### DSP Parameters

BINPERADC

MCALOWBIN

RESETINT

LOOPCOUNT

WHICHTEST

RUNTASKS

Unused

Unused

Unused

Unused

### Run Tasks

Initially measure slope

Update slope

Initially meas. baseline

Update baseline

Collect MCA data

Correct for baseline

Sub. slope from baseline

Collect baseline history

Special task/calibration

Histogram delta baseline

Unused

### Record Processing

Scan Rate

### Calibration

Task

Status

### Baseline Parameters

BASEBINNING

Unused

Unused

Unused

Unused

Unused

Unused

### Peaking Time

# ccdApp: EPICS Interface to Area Detectors

- Goal: Uniform interface for controlling area detectors (CCD, online image plates) from EPICS
  - Any EPICS client (e.g. spec, IDL, scan record) can control (at a bare minimum).
    - Exposure time
    - File name
    - Start collection, wait for completion
  - Much more control for most detectors
- Current status:
  - MAR 165 CCD (complete, in use on Sectors 1, 8, 13, 18, others)
  - Roper CCD detectors (complete, in use on Sectors 13, 15, 20)
  - Bruker CCD detectors
    - Winview interface complete. In use on Sectors 13, 15 next run.
    - SMART Service interface in planning stage. To be done?
  - MAR 345 online image plate (soon)
    - Will be based on Keith Brister's work

# Implementation

- Use manufacturer's software for primary user interface.
  - Minimizes amount of new code
  - Uses existing file formats, unwarping algorithms, etc.
  - These programs include:
    - marccd for MAR165
    - Winview/Winspec for Roper cameras
    - WinView for Bruker cameras.
      - Requires PCI card to replace ISA card that Bruker supplies. Also new SCSI-type cable.
    - SMART for Bruker cameras
    - scan345 for MAR 345 image plate

# Implementation

## Control these programs from EPICS

Each of these programs has a “remote control” interface, typically using TCP/IP sockets

Using EPICS means each client (e.g. spec) does not have to know how to talk to each type of detector. Only has to know how to talk to EPICS.

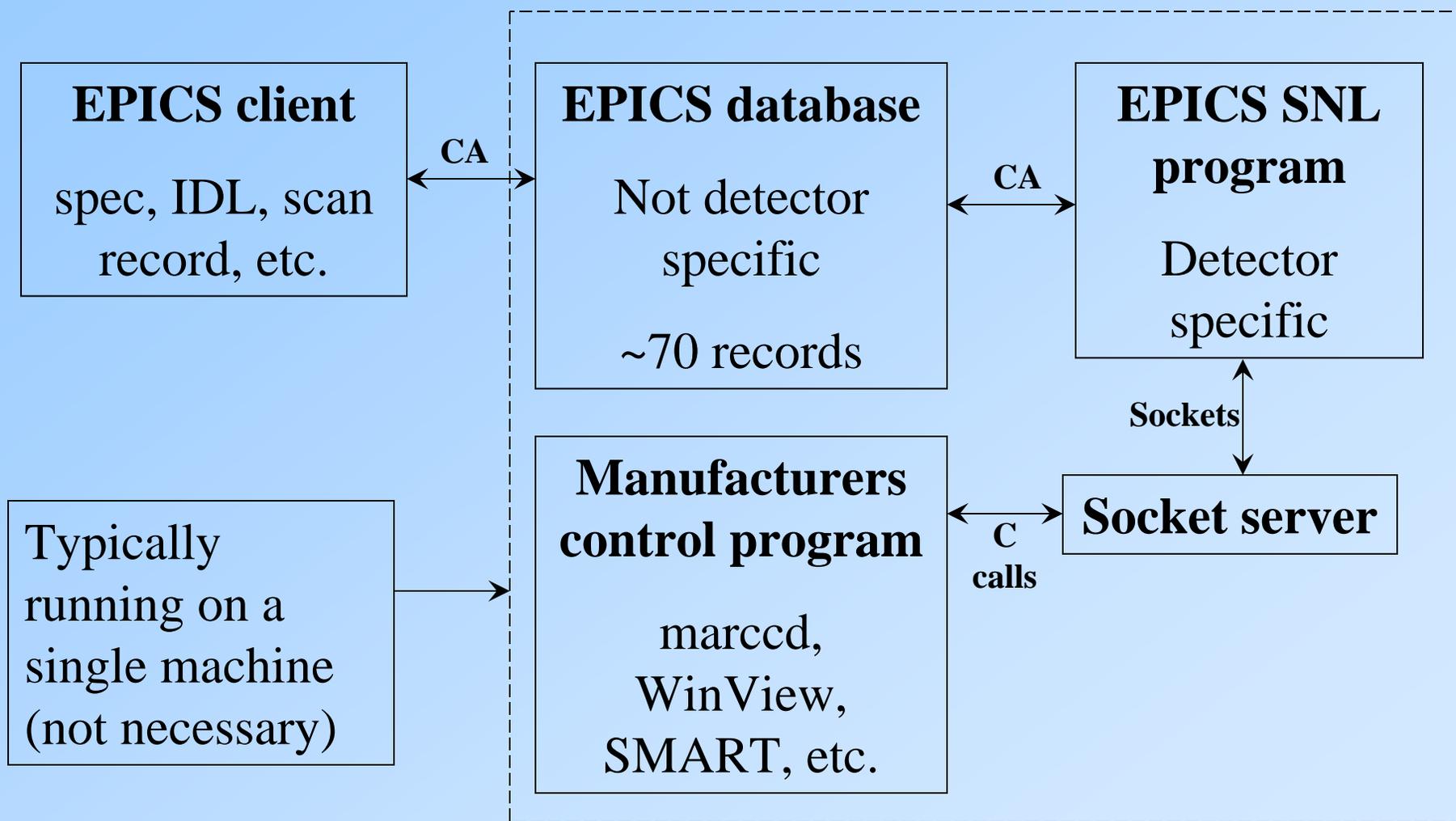
## EPICS software consists of

Database of records (PVs), identical for all detectors

State-notation-language (SNL) programs, unique for each detector.  
Reads/writes PVs and communicates with remote control interface over sockets.

The database and SNL programs are typically run on the same machine that the user interface software runs on (e.g. Linux box for MAR detectors, Windows for Roper and Bruker). **No VME crate required.**

# Schematic Architecture



# Relationship with Brian Tieman's Package

- Complementary to, and compatible with, Brian's package.
  - My software uses the same PV names as Brian's wherever possible, so EPICS clients that work with his software should work with mine
- Brian's uses the EPICS portable channel access server, rather than running a real EPICS database and SNL program on the server machine
  - Only option when Brian wrote his code, before EPICS 3.14.
- Brian's program calls library (DLL) layer directly, bypasses manufacturers user interface program
  - More flexible
  - More code
  - File format is HDF, not manufacturers format that many data processing applications expect
  - Unwarping is not implemented
- Brian's program is a Windows application, and so only works on Windows detectors (Roper, several others). Not MAR or other Unix detectors.

## “Expert” medm screen

The screenshot shows a software interface titled "Area Detector Control" with the following sections:

- Setup:** EPICS name: roperCCD:det1; Manufacturer: Roper; Model: ST138; SNL Program: Running; Server name: gselab1; Server port: 5001; Connect: Connected; Temperature: -20.00, 1.00; Print debugging: No; User input: ; To detector: save, c:\temp\test021; From detector: OK.
- Shutter:** Shutter Control: Camera output; Shutter PV: roperCCD:det1:Shutter; Shutter status: Closed; Open command: Open; Close command: Closed; Open/Close: Open, Close; Open delay: 1.000; Close delay: 0.050.
- Collect:** Exposure time: 0.200, 0.300; # frames: 1, 1; Frame type: Normal; Acquire: Start, Done; Detector state: Idle; Time remaining: 0.0; Polling rate: Passive; Abort: Abort.
- File:** File path: c:\temp\; Base filename: test; Next file #: 22; Filename format: %s%3d.SPE; Comment 1: test comment 1; Comment 2: test comment 2; Comment 3: another comment; Comment 4: ; Comment 5: ; Correct bkgnd: No; Correct flat: No; Correct spatial: No; Auto save: Yes; Save file: Save; Last filename: test021.SPE.
- Readout:** Binning (X): 2, (Y): 2; Top/bottom (T): 1, (B): 1152; Left/Right (L): 1, (R): 1242; Total counts: 2.63258e+008; Net counts: 1.29189e+006; Compute counts: Yes.

Many fields do not apply to all detectors. Simpler screens (e.g. for MAR 165 only) can easily be made.

## Process variables (PVs)

\$(P)\$(C)Abort	\$(P)\$(C)AcquireCLBK	\$(P)\$(C)AcquirePOLL	\$(P)\$(C)ActualBinX
\$(P)\$(C)ActualBinY	\$(P)\$(C)ActualNumFrames	\$(P)\$(C)ActualROIBottom	\$(P)\$(C)ActualROILeft
\$(P)\$(C)ActualROIRight	\$(P)\$(C)ActualROITop	\$(P)\$(C)ActualSeconds	\$(P)\$(C)ADC
\$(P)\$(C)AutoSave	\$(P)\$(C)BinX	\$(P)\$(C)BinY	\$(P)\$(C)BitDepth
\$(P)\$(C)CCDManufacturer	\$(P)\$(C)CCDModel	\$(P)\$(C)CloseShutter	\$(P)\$(C)CloseShutterDly
\$(P)\$(C)CloseShutterStr	\$(P)\$(C)Comment1	\$(P)\$(C)Comment2	\$(P)\$(C)Comment3
\$(P)\$(C)Comment4	\$(P)\$(C)Comment5	\$(P)\$(C)Compression	\$(P)\$(C)ComputeROICts
\$(P)\$(C)ConnectState	\$(P)\$(C)CorrectBackground	\$(P)\$(C)CorrectFlatfield	\$(P)\$(C)CorrectSpatial
\$(P)\$(C)DebugFlag	\$(P)\$(C)DetectorState	\$(P)\$(C)DetInStr	\$(P)\$(C)DetOutStr
\$(P)\$(C)FilenameFormat	\$(P)\$(C)FilePath	\$(P)\$(C)FileTemplate	\$(P)\$(C)FrameType
\$(P)\$(C)FullFilename	\$(P)\$(C)HDFTemplate	\$(P)\$(C)Hours	\$(P)\$(C)Initialize
\$(P)\$(C)MeasuredTemp	\$(P)\$(C)Milliseconds	\$(P)\$(C)Minutes	\$(P)\$(C)NumExposures
\$(P)\$(C)NumFrames	\$(P)\$(C)OpenShutter	\$(P)\$(C)OpenShutterDly	\$(P)\$(C)OpenShutterStr
\$(P)\$(C)PollDetState	\$(P)\$(C)ROIBottom	\$(P)\$(C)ROILeft	\$(P)\$(C)ROINet
\$(P)\$(C)ROIRight	\$(P)\$(C)ROITop	\$(P)\$(C)ROITotal	\$(P)\$(C)SaveFile
\$(P)\$(C)Seconds	\$(P)\$(C)SeqNumber	\$(P)\$(C)ServerName	\$(P)\$(C)ServerPort
\$(P)\$(C)SetTemp	\$(P)\$(C)Shutter	\$(P)\$(C)ShutterMode	\$(P)\$(C)ShutterStatus
\$(P)\$(C)SNLWatchdog	\$(P)\$(C)TimeRemaining	\$(P)\$(C)UserInStr	

# marccd remote control

The screenshot displays the marccd remote control software interface. The main window shows a large circular field of view with a central spot. A dialog box titled "Acquire/Remote Control dialog" is overlaid on the main window, containing the following fields and buttons:

- Device Database Host: localhost
- Device Database Server: /home/marccd/source/servers/marccd\_serve (with a Browse button)
- Device Personal Name: 3000
- Command Log: (with a Browse button)
- Buttons: Start, Stop, Save, Cancel, Dismiss, Help

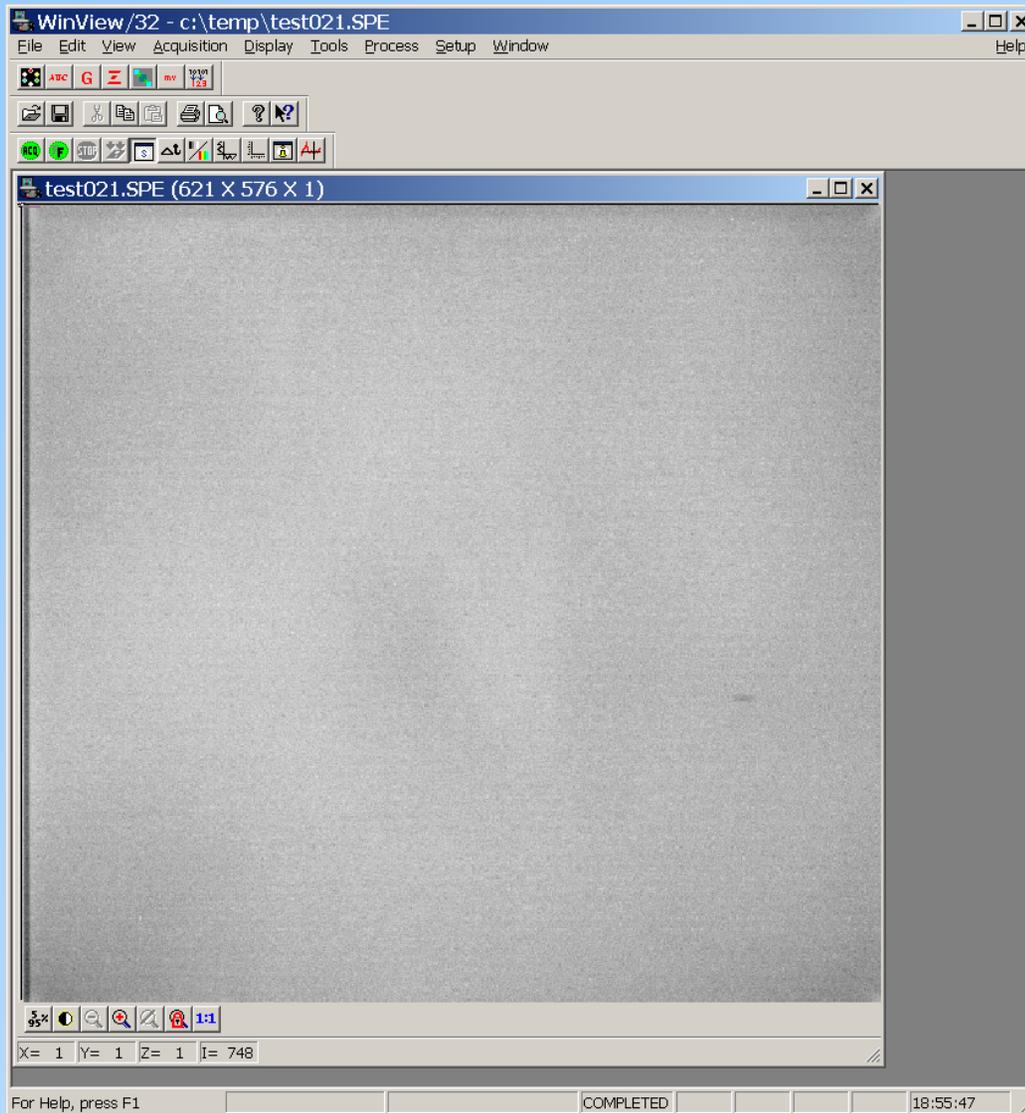
The main window also displays various status and control elements:

- Protocol: None (with a Stop button)
- Detector: Clearing
- Intensity: (with a Shutter: Closed button)
- Temperature: -79.8 C
- Pressure: 0.02 Torr
- Status: Cooler ON
- File: <untitled>
- Title: Spatially Corrected <untitled>
- Max: 0 Mean: 9.1
- Min: 1 N Sat.: 0
- Detector: (dropdown menu)
- Resolution: 5.21 Angs.
- Resolution: 11.38 degrees
- Resolution: 256 pixels
- Resolution: 20 pixels
- Resolution: 741 pixels
- Spacing: 11.32 Angs.
- Misset: -524 -524 pixels
- Misset: -41.21 -41.21 mm
- Misset: -22.40 -22.40 degrees
- Zone: 512 512 pixels
- Zone: 40.26 40.26 mm
- Beam: 1036 1036 pixels
- Beam: 81.47 81.47 mm
- Pointer: 63.86 148.47 1.73
- Pointer: 812 1888
- DENZO X Y
- Resolution: 11

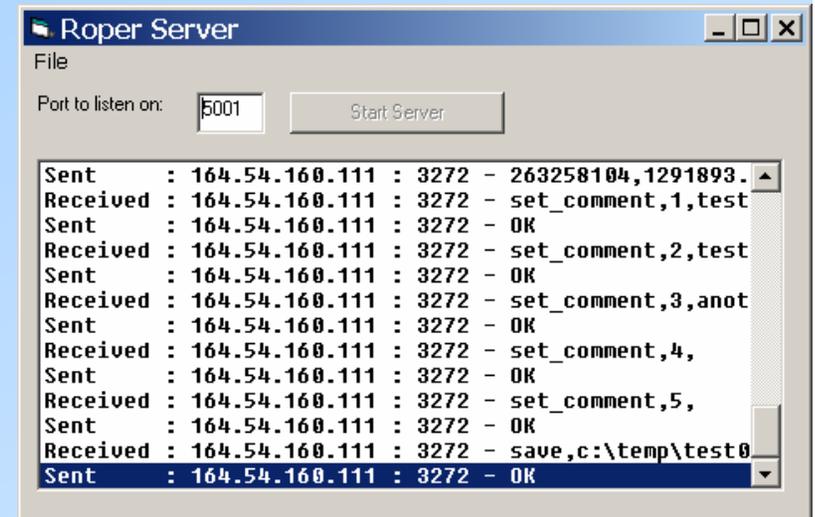
The interface also includes a terminal window at the bottom right showing the following output:

```
File Edit View Terminal Go Help
gunzip: stdout: Broken pipe
gunzip: stdout: Broken pipe
```

# Roper Interface



← WinView (from Roper)



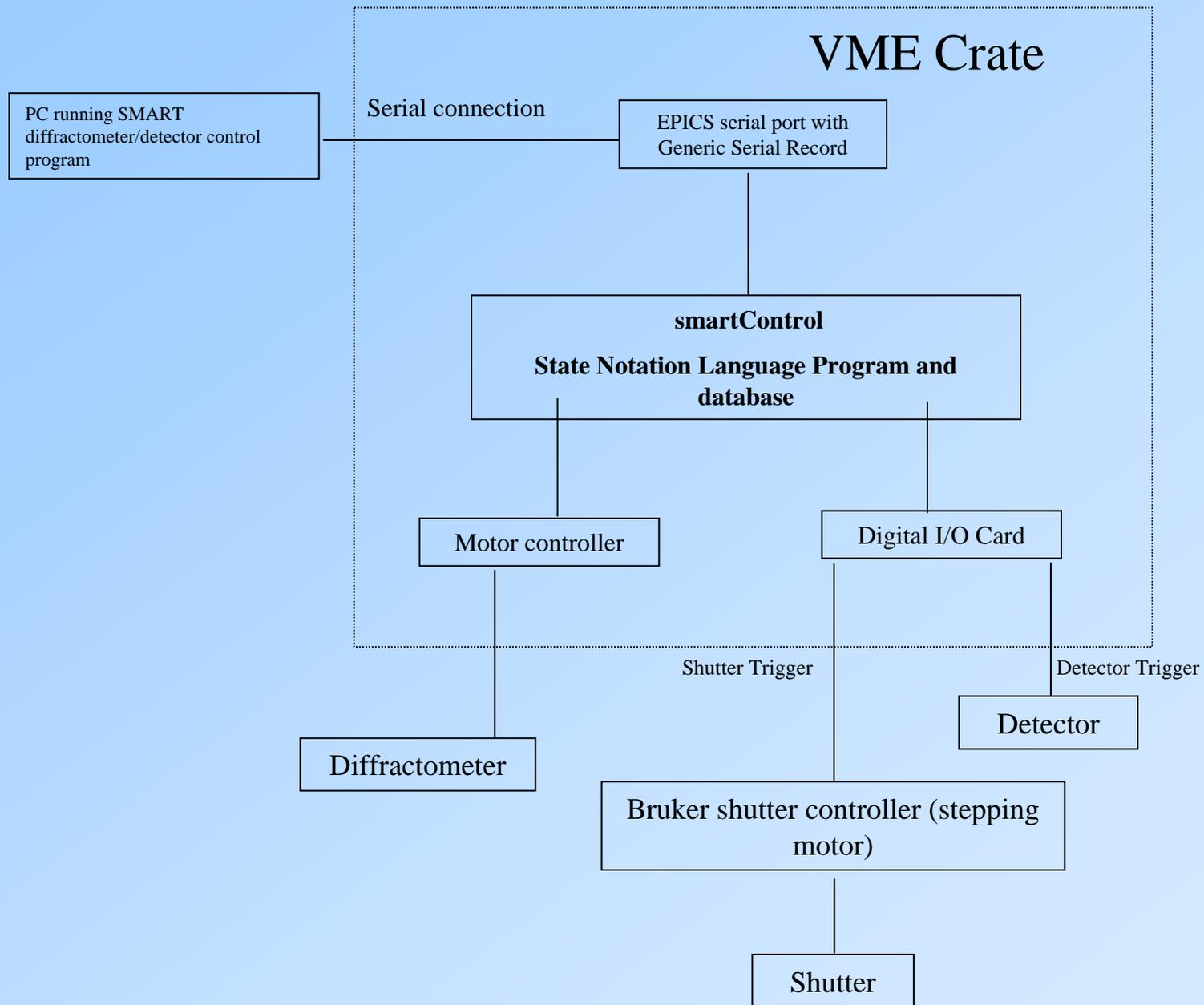
↑  
Socket server written in Visual Basic. Simple ASCII commands. Calls COM interface to automate WinView & WinSpec). Not EPICS specific, other applications can talk to it.

# **smartControl: Interfacing the Bruker SMART Software with EPICS**

# Introduction

- Bruker makes x-ray detector systems for single-crystal and powder diffraction, and small-angle scattering applications.
- Widely deployed in crystallography laboratories world-wide
- Large user community who are familiar with the Bruker “SMART” control and analysis software.
- SMART software is only capable of controlling goniometers through the Bruker General Goniometer Control System (GGCS), which is a specific hardware controller manufactured by Bruker.
- GGCS is not generally used to control goniometers at synchrotron facilities.
- smartControl permits the standard Bruker SMART software to control any goniometer, with any number of axes, through EPICS. Also provides
  - Shutter control
  - Normalization information, such as from an ion chamber, to the SMART software where it is stored in the frame headers.

# smartControl Architecture



# smartControl Implementation

Communicates with GGCS goniometer via RS-232

Runs in an EPICS IOC and *emulates* the GGCS.

- Listens to commands from SMART on the RS-232 port
- Translates those commands into EPICS channel access calls to:
  - Move goniometer motors
  - Open and close shutters
  - Trigger the CCD detector
  - Start and read scalers and timers for beam intensity normalization
- Sends responses back to the SMART system in the format that SMART expects from the GGCS.

Thus, SMART *thinks* it is talking to a GGCS, while it is *actually* talking to an EPICS IOC.

## Implementation Details

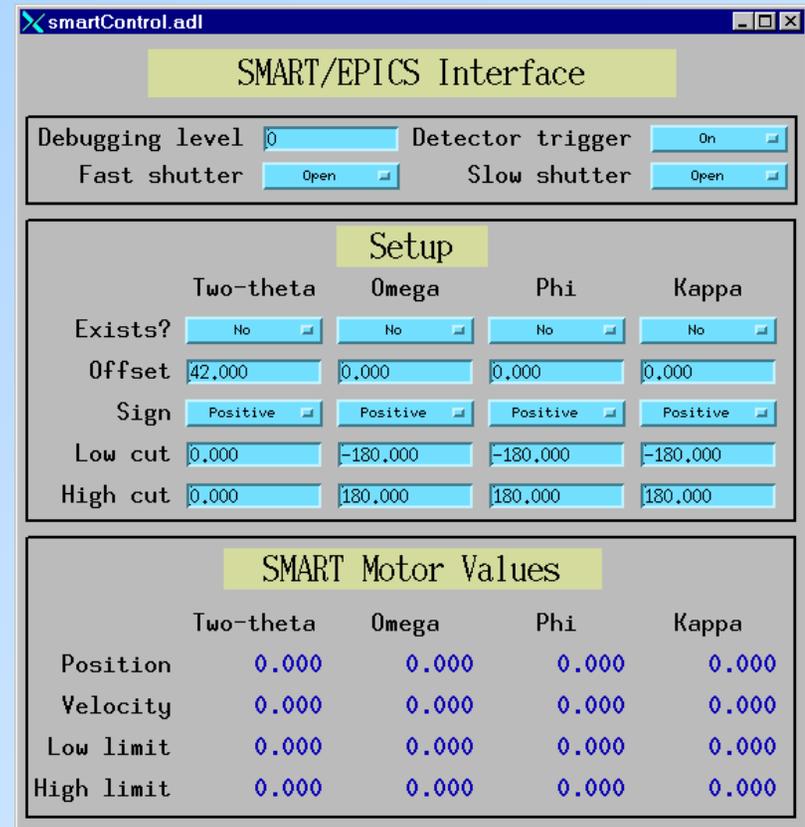
- smartControl consists of:
  - An EPICS database, smartControl.db
  - A State Notation Language program, smartControl.st
- Motor control is done via the EPICS “motor” record.
- RS-232 communication is done via the EPICS “asyn” module
- Scaler/timer control is done via the EPICS “scaler” record.
- Shutter control is done via the EPICS “binary output” record.

## Advantages

- Any motor with EPICS device support can be used for the goniometer.
- Any type of beam monitor (ion chamber, photodiode, etc.) can be used.
- Any shutter can be used by simply constructing a simple database with a binary output record controlling it.

# smartControl User Interface

- Display and manual control of the state of the fast shutter, slow shutter (if implemented), and detector trigger
- Setup of the goniometer motors
  - Each of the (up to) 4 axes that SMART controls is assigned to an EPICS motor.
    - Exists\ (Yes/No)
  - Offset and sign difference between the EPICS motor coordinate system and the SMART coordinate system.
    - Useful when the same goniometer is used with SPEC and with SMART, since these use different sign conventions.
  - High and low “cut” that control where the limits of motion of the actual EPICS motor are.
    - A command from SMART to move omega to +140 degrees can be translated into a move to –220 degrees.
- Display of the motor parameters for the goniometer axes from the SMART system’s perspective.



# Systems Controlled

- Single axis rotation stage for diffraction with SMART 1500 at NSLS X-26A.

- Newport 6-axis General Purpose Goniometer at APS Sector 13 with the SMART 1500 and SMART 2K systems. Applications include single-crystal micro-diffraction and single-crystal diffraction in the diamond-anvil cell.

- Huber 4-circle goniometer at APS Sector 15 with the SMART 6000 system. Applications include microcrystal and time-resolved diffraction.

- Single-axis goniometer (Newport UR-100 stage) at APS Sector 13 with the SMART 2K detector. Applications include single-crystal micro-diffraction and high-temperature, high-pressure powder diffraction in an externally heater diamond-anvil cell.



# Closing the Loop: Using Feedback in EPICS

Mark Rivers, Center for Advanced Radiation Sources

- Many applications for feedback on APS beamlines
- Dedicated feedback controllers are expensive and relatively inflexible
- A new EPICS record for performing feedback
  - Enhanced Proportional Integral Derivative (EPID)
  - Flexible and fast feedback under EPICS

# EPID record: Enhancements over the standard EPICS PID record

- Separation of device support from the record.
- Soft Record device support which uses EPICS database links
  - Very similar to the PID record
- EPID record can also be used with other device support
  - Communicate with faster feedback software
  - Hardware controllers.
  - Device support is provided in the Message Passing Facility for fast feedback (> 1 kHz) using an Acromag IP330 ADC and a Systran DAC128V DAC.
- Addition of many fields (OUTL, DRVH, DRVL) to simplify construction of databases

- The PID expression is computed as an absolute number, rather than a differential number to be added to the present output value.
  - Simplifies database construction, and also permits the record itself to perform limit checking on the output.
- Limits are placed on the magnitude of the integral term (I) which are lacking in the PID record.
- Monitors are posted for the CVAL field
  - Simplifies construction of user-interface tools, such as plotting.
- The CVL field has been renamed INP
  - This field can now be modified (a feature of EPICS R3.12 and higher)
  - A single EPID record can be used to control different processes at different times.
- Changed the time units of the KI and KD terms from minutes to seconds

# PID Equation

The discrete form of the PID expression is as follows:

$$M(n) = P + I + D$$

$$P = K_P * E(n)$$

$$I = K_P * K_I * \text{SUM}_i ( E(i) * dT(n) )$$

$$D = K_P * K_D * ( ( E(n) - ( E(n-1) ) ) / dT(n) )$$

Where

$M(n)$  = value of manipulated variable at  $n$ th instant.

$P$  = Proportional term

$I$  = Integral term

$D$  = Derivative term

$K_P$  = Proportional gain

$K_I$  = Integral gain

$K_D$  = Derivative gain

$E(n)$  = Error at  $n$ th sampling instant

$\text{SUM}_i$  = Sum from  $i=0$  to  $i=n$

$dT(n)$  = Time difference between  $n-1$  instance and  $n$ th instance

# Sanity checks on Integral Term

- Integral term is a sum from time=0 to the present time
- Can grow extremely large if not subject to some "sanity checks"
- The EPICS PID record does not perform any such checks, which is a serious limitation.

- The EPID record device support performs the following checks to prevent the integral term from growing too large:
  - I is not allowed to increase if the computed output,  $M(n)$ , is at the high limit, DRVH.
  - I is not allowed to decrease if the computed output,  $M(n)$ , is at the low limit, DRVL.
  - I is not allowed to be less than DRVL or greater than DRVH.
  - I can be modified from database access or channel access.
    - Allows the user to set I to a specific value to improve response time, rather than waiting for the normal time constant associated with this term.
  - If KI is 0. then set I to DRVL to 0
  - If feedback is off don't change integral term.
  - When feedback is changed from OFF to ON set I to the current value of the output PV.

# “Slow” Feedback

- The EPID record has two kinds of device support.
- “Soft” device support allows the readback input and control output to be any EPICS process variables.
  - Very flexible
  - Any type of device can be used for input (analog to digital converter, RS-232, GPIB, scaler, etc.)
  - Any type of device can be used for output (digital to analog converter, RS-232, GPIB, etc.)
  - Can be reconfigured on the fly, changing the input and output process variables, feedback coefficients, etc.
  - Limited to standard EPICS scan rates, typically 10 Hz maximum
  - Sufficient for many applications

# Slow feedback - D/A connected to A/D

**pid\_control.adl**

## PID feedback control

Readback PV   
Control PV

Setpoint	Readback
<input type="text" value="4.000"/>	<input type="text" value="3.996"/>
Feedback	Update rate
<input type="text" value="On"/>	<input type="text" value=".1 second"/>

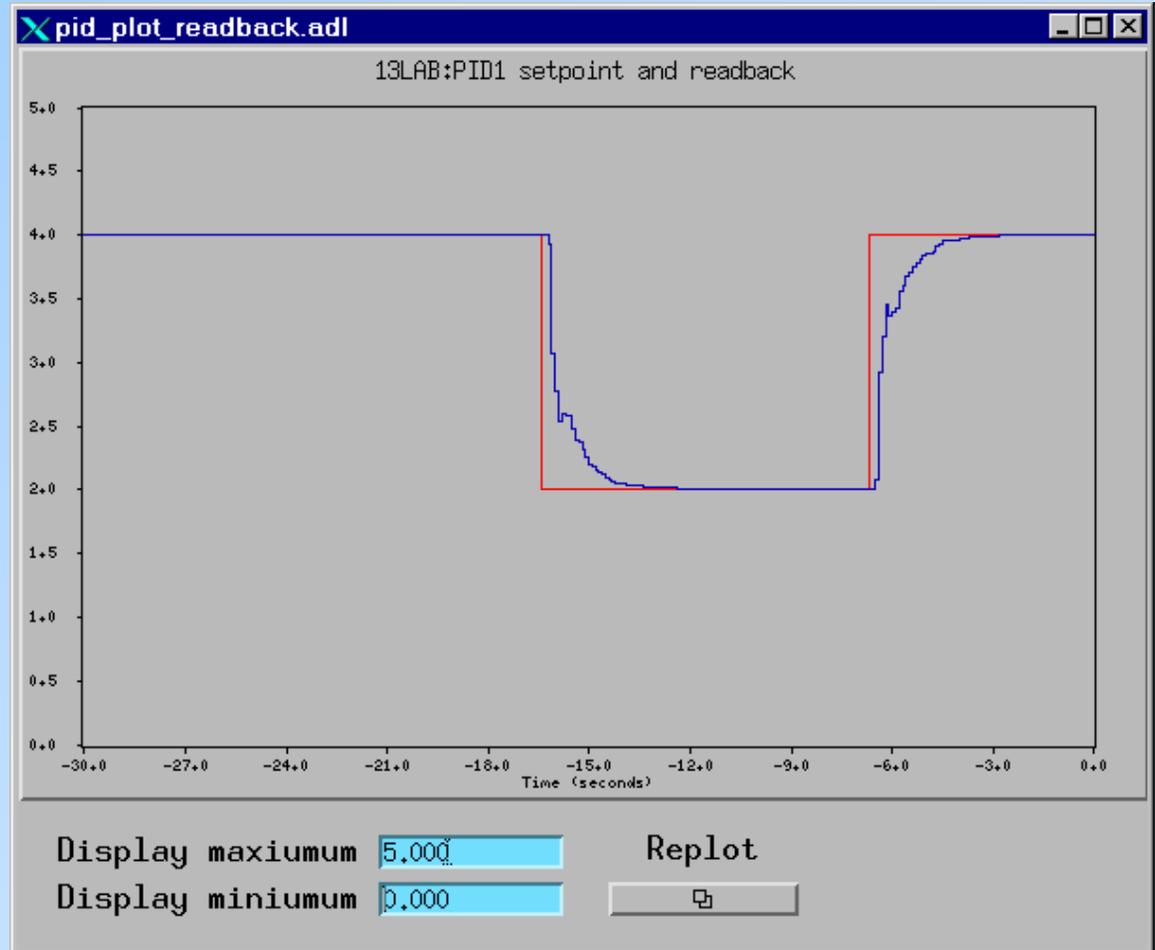
More

**pid\_parameters.adl**

## PID feedback parameters

KP	<input type="text" value="0.300"/>	P	<input type="text" value="0.001"/>
KI	<input type="text" value="5.000"/>	I	<input type="text" value="4.002"/>
KD	<input type="text" value="0.000"/>	D	<input type="text" value="0.000"/>

Delta time   
Error   
Output   
Low limit   
High limit



# “Fast” Feedback

- “Hardware” device support is presently limited to using the Acromag IP-330 A/D converter for the input and the Systran DAC-128V digital to analog converter for the output
- Inexpensive Industry Pack modules
  - ADC is about \$1000 for 16 channels
  - DAC is about \$300 for 8 channels
- Very fast
  - Up to 10 kHz feedback rate
- Feedback coefficients and feedback rate be reconfigured on the fly
- Uses the Message Passing Facility to communicate with EPICS
  - The feedback loop can be running on a dedicated inexpensive CPU, such as a MVME162 with only 1 MB of memory

# Fast feedback - D/A connected to A/D

**pid\_control.adl**

## Fast\_Feedback

Readback PV

Control PV

Setpoint	Readback
<input type="text" value="45000.000"/>	<input type="text" value="44995.000"/>

Feedback	Update rate
<input type="text" value="On"/>	<input type="text" value=".1 second"/>

More

**pid\_parameters.adl**

## PID feedback parameters

KP	<input type="text" value="0.020"/>	P	<input type="text"/>
KI	<input type="text" value="300.000"/>	I	<input type="text" value="2430.653"/>
KD	<input type="text" value="0.000"/>	D	<input type="text" value="0.000"/>

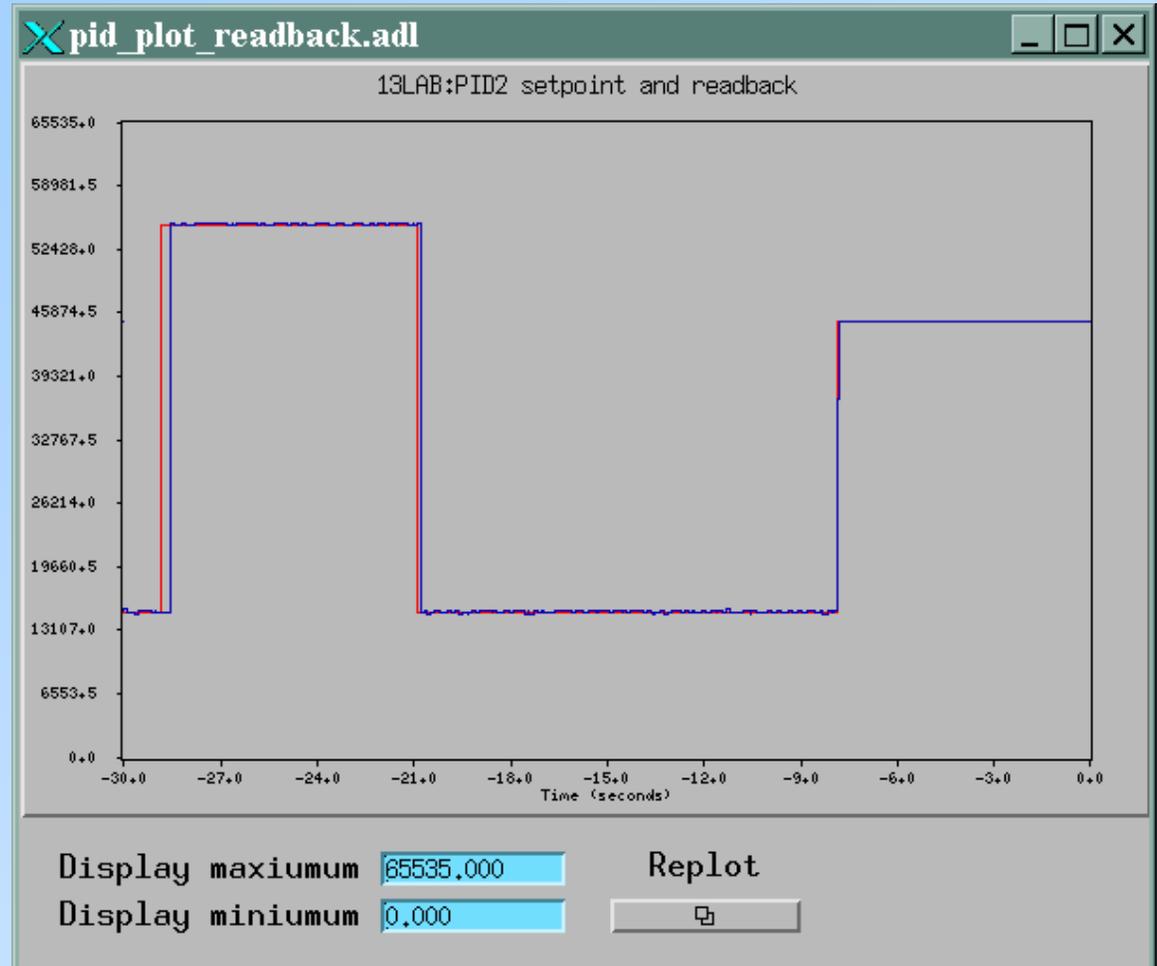
Delta time

Error

Output

Low limit

High limit



# GSECARS Applications of EPID record

- Monochromator second crystal feedback:
  - Feedback on beam position on 13-ID, using photo-diodes in-vacuum slits, measuring scattered radiation from in-vacuum slits
  - Feedback on beam intensity on 13-BM, using table-top ion chamber.
  - Recovers gracefully from beam dumps. PV available to indicate “feedback locked”, which data acquisition programs can wait for.
- Position feedback on large Kirkpatrick-Baez mirrors with piezo actuators. Stabilizes beam position at sample.
- Furnace temperature control in the large-volume press in 13-BM-D and 13-ID-D. Safety checks to limit voltage, current, and power.
- Pressure control in the large-volume press, via hydraulic pump, in 13-BM-D. Can ramp pressure up and down using scan record to control setpoint
- Temperature stabilization via laser power control in the laser-heated diamond-anvil cell in 13-ID-D.

# Feedback Tuning: Quick guide to selecting the optimal values for KP, KI and KD.

- Turn off feedback (FBON=0).
- Set KI and KD to zero initially in order to first determine the optimum value for KP.
- If possible calculate theoretically or empirically the "correct" value of KP, e.g. the required change in OVAL to produce a unit change in CVAL.
  - For example if controlling a heater power supply with a DAC which has 10 Watt/volt response, and the heater response is 10 degrees/Watt, then  $KP=.01$  volts/degree.

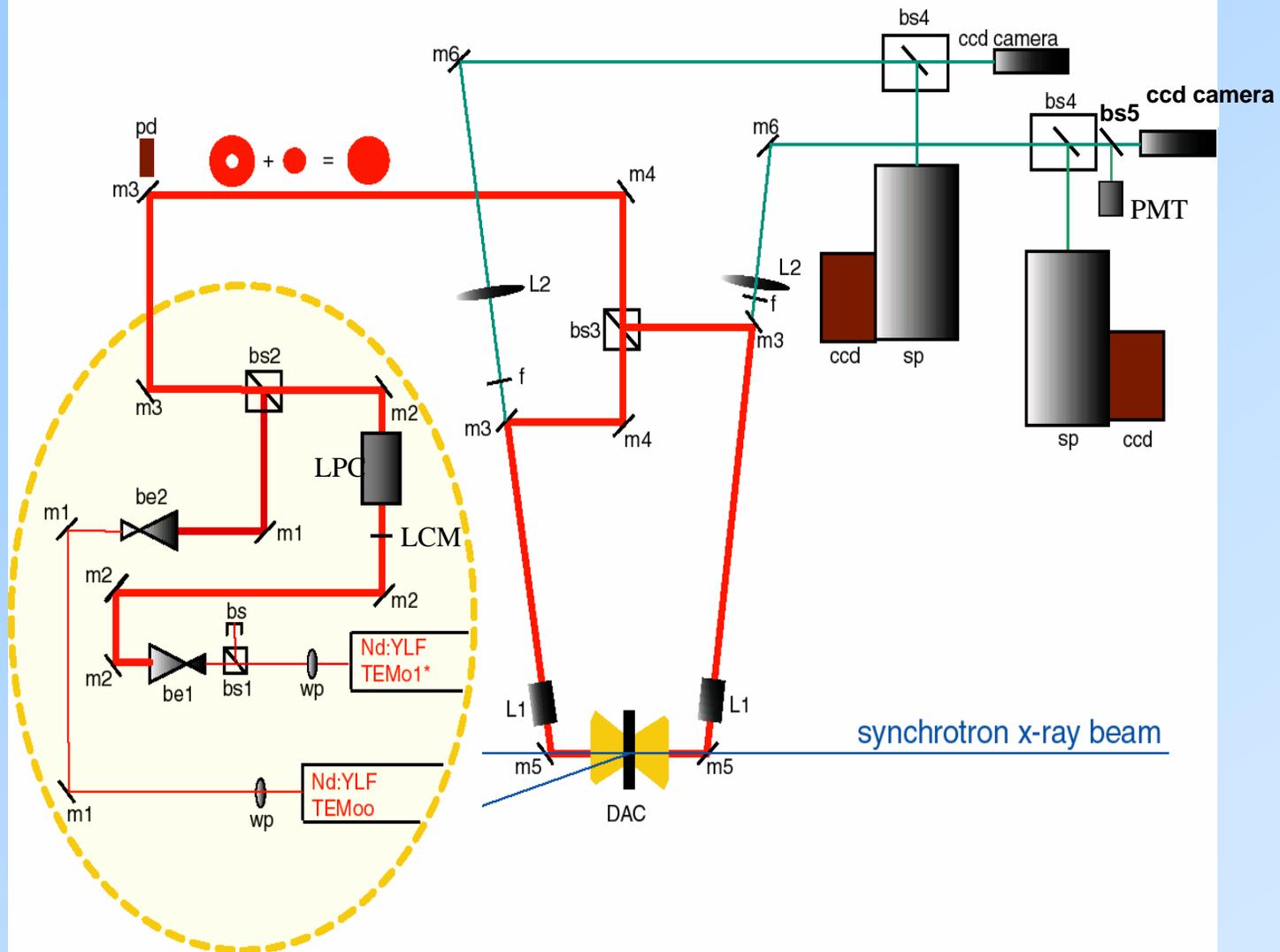
- Set KP to about 10%-25% of the "correct" value computed above.
- Turn on feedback (FBON=1) and make changes in the setpoint (VAL) and observe the system response.
  - Don't worry about system droop (CVAL  $\neq$  VAL), since this is unavoidable when KI=0.
  - Rather look for oscillations and instability.
  - Gradually increase KP while making changes in the setpoint.
  - When KP is too large the system will begin to oscillate.
  - Decrease KP until the oscillations just disappear.

- Increase KI (units of Hz) to eliminate the system droop.
  - The optimum value of KI depends upon the time constant of the system and the update rate of the feedback loop.
  - Increase KI until the system responds as rapidly as possible to changes in the setpoint without overshoot or oscillation.
- For systems with no significant inertia KD should be left at zero.
  - For systems with large inertia increase KD (units of seconds) to minimize overshoot.

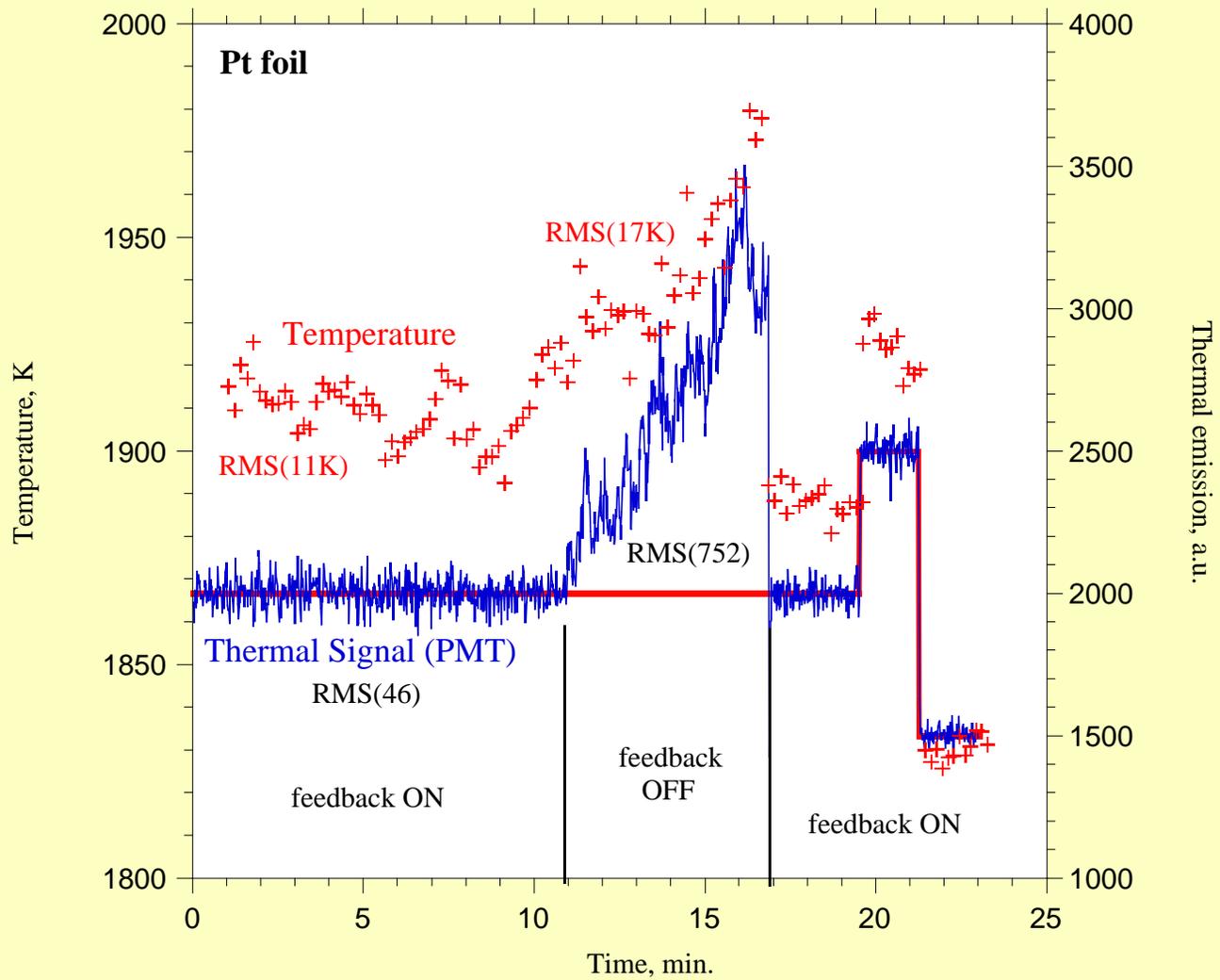
## Example Application: Laser power stabilization

- Laser-heated diamond anvil cell apparatus
- YLF lasers are used to heat the sample in the diamond anvil cell
- Temperature is measured from the blackbody radiation with an optical spectrometer
- Important to maintain the sample at a constant temperature during the diffraction measurement.
- Read thermal radiation with a photomultiplier
- Control polarization (transmission) of an optical element
- Do not feed back directly on temperature, because this takes several seconds to measure
- Temperature is stable if the thermal radiation is stable.
- Currently using the Acromag A/D converter to directly measure the photomultiplier output, with “fast” feedback.

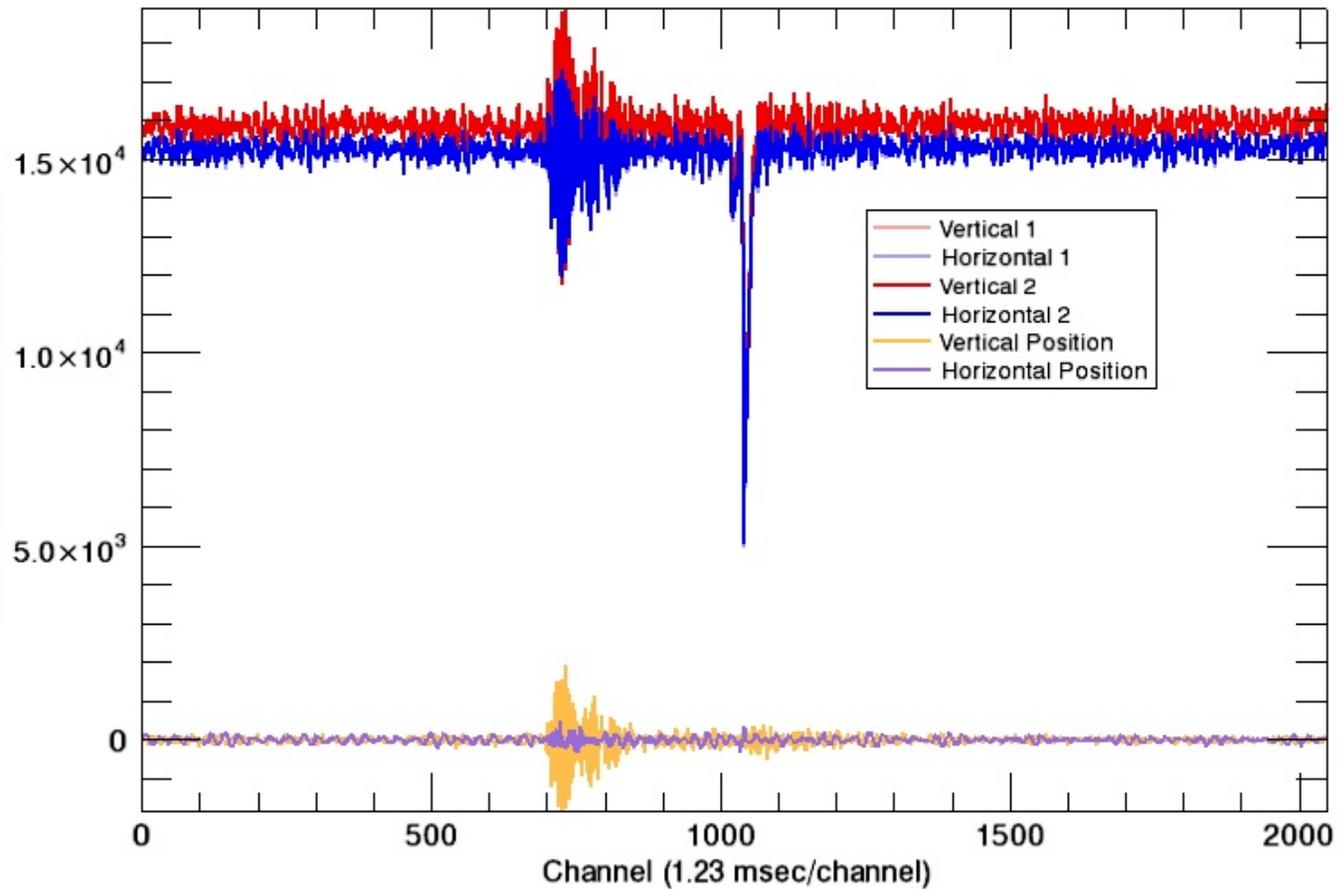
# Double Sided Laser Heating System at the GSECARS (Sector 13, APS)



# laser\_logger2



## Example Application: Monochromator Second Crystal Stabilization

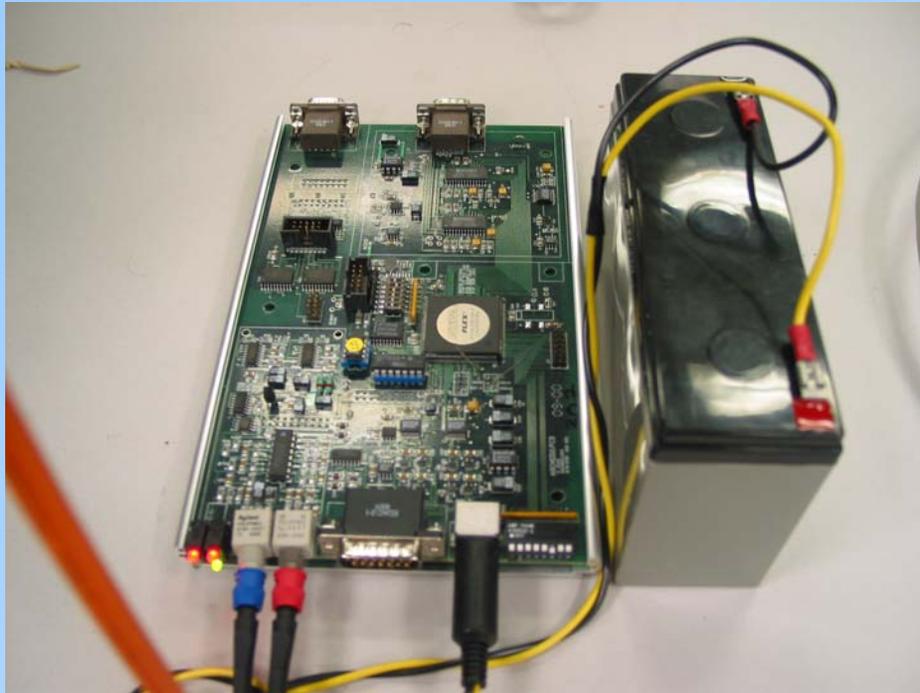


# Hardware and EPICS Software for the APS Quad Electrometer for X-ray Beam Position Monitors

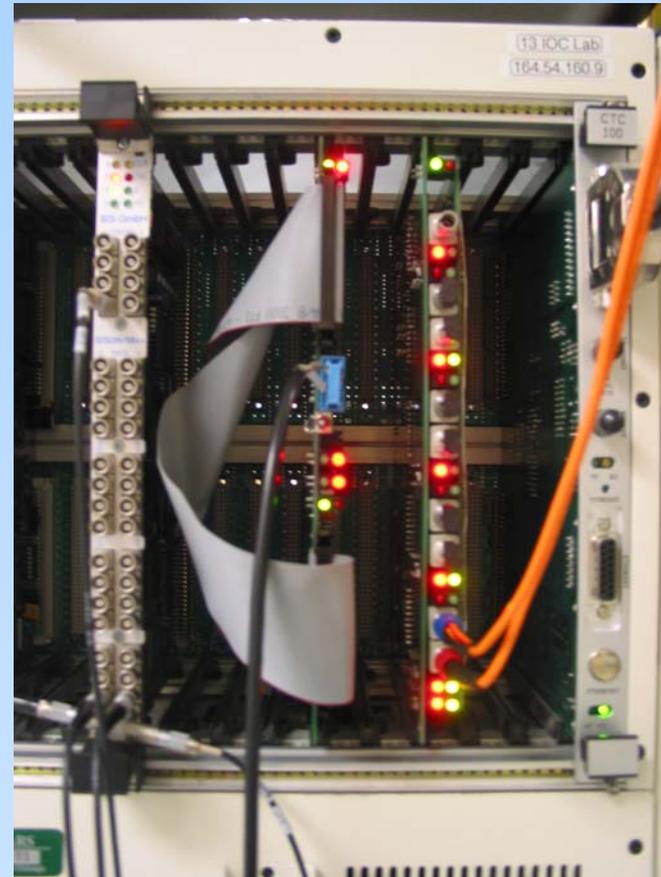
Mark Rivers (CARS) and Steve Ross(APS)

- Steve has designed a 4-channel electrometer for measuring currents in the nA to uA range.
- Intended primarily for reading x-ray beam positions using 4 photodiodes or split ion chambers.
- Compact and inexpensive, and can be placed close to the position monitor hardware to keep signal leads short.
- Outputs digital data at up to 815Hz over a fiber-optic cable
- Read by a pair of VME boards.
- Fiber allows reliable data transmission over long distances, for example from an experiment station to a VME crate in the FOE, where feedback to a monochromator crystal can be implemented.

# Electrometer Hardware



Remote ADC unit and  
battery



VME boards

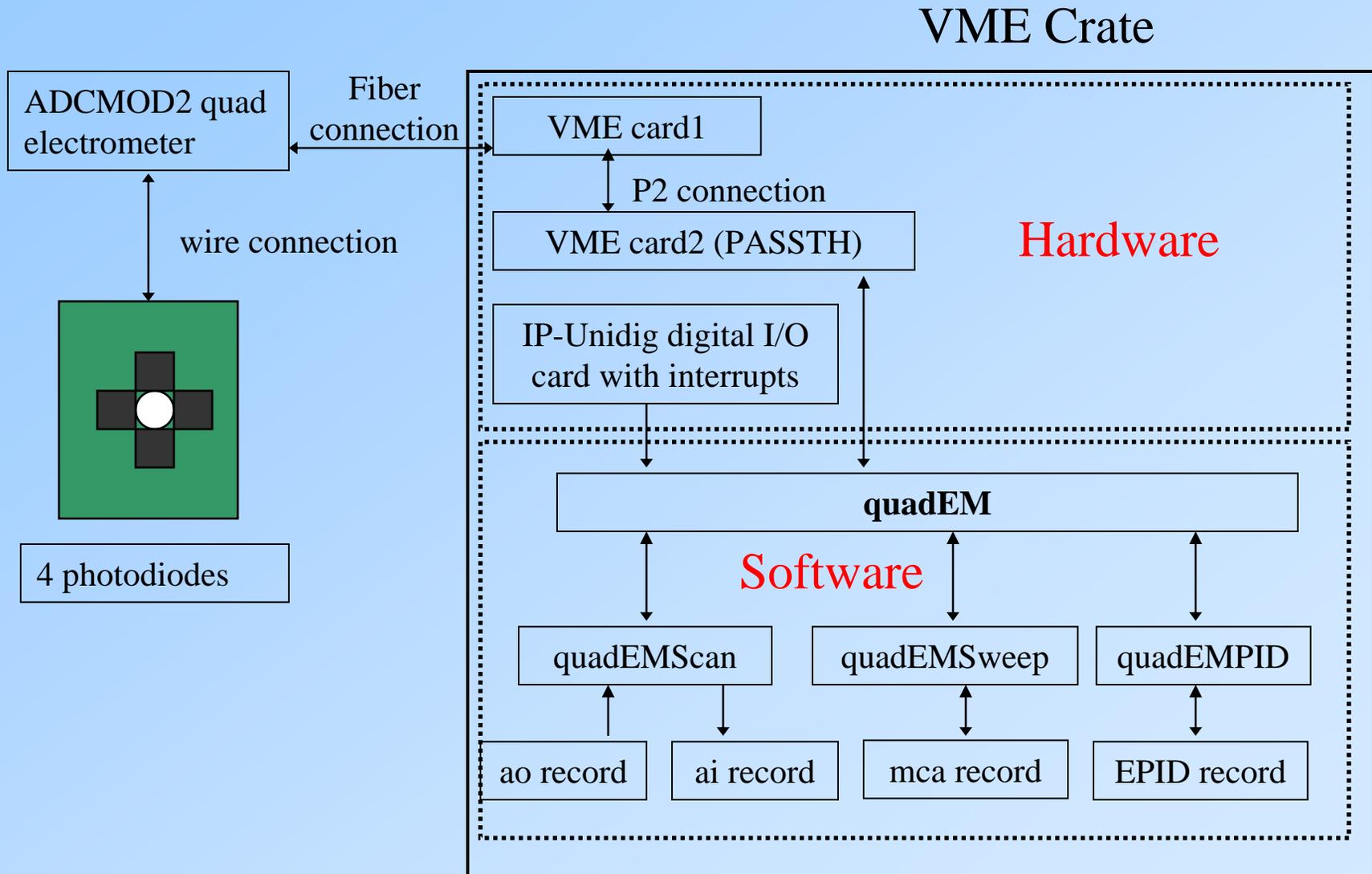
# Applications

- Feedback of the pitch and roll of monochromator crystals based on the beam position in the beamline or experimental station.
- Feedback on mirror pitch for stabilizing the position of the beam downstream of a focusing mirror.
- In-vacuum fluorescent foils allow I0, beam position, and energy calibration to always be available
- Replaces:
  - 4 SRS570 current amplifiers
  - 4 ADCs, or 4 V/F converters and 4 scaler channels

# EPICS Software

- I have developed a EPICS software (quadEM) to read the digital data from the electrometer.
- Interrupt driven, reads the digital data stream at 815Hz.
- Provides the current in each of the 4 photodiodes, as well as the sum, difference and position for opposite pairs of diodes.
- Device support is provided for 3 types of EPICS records:
  - analog input (ai) record at up to 10 Hz
  - multichannel analyzer (mca) record which functions as a “digital scope”, capturing the values at up to 815Hz
  - feedback (epid) record for fast feedback through an A/D converter at up to 815Hz.
- The mca and epid records can run slower than 815Hz as well, in which case they provide signal averaging.

# System Architecture



# Main medm screen for analog input

quadEM\_full.adl

Quad Electrometer (13LAB2:EM1)

Diode #	1	3	2	4
Current	7577	10721	18835	16668
Offset	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Sum	18298		35503	
Difference	3148		-2167	

---

Feedback

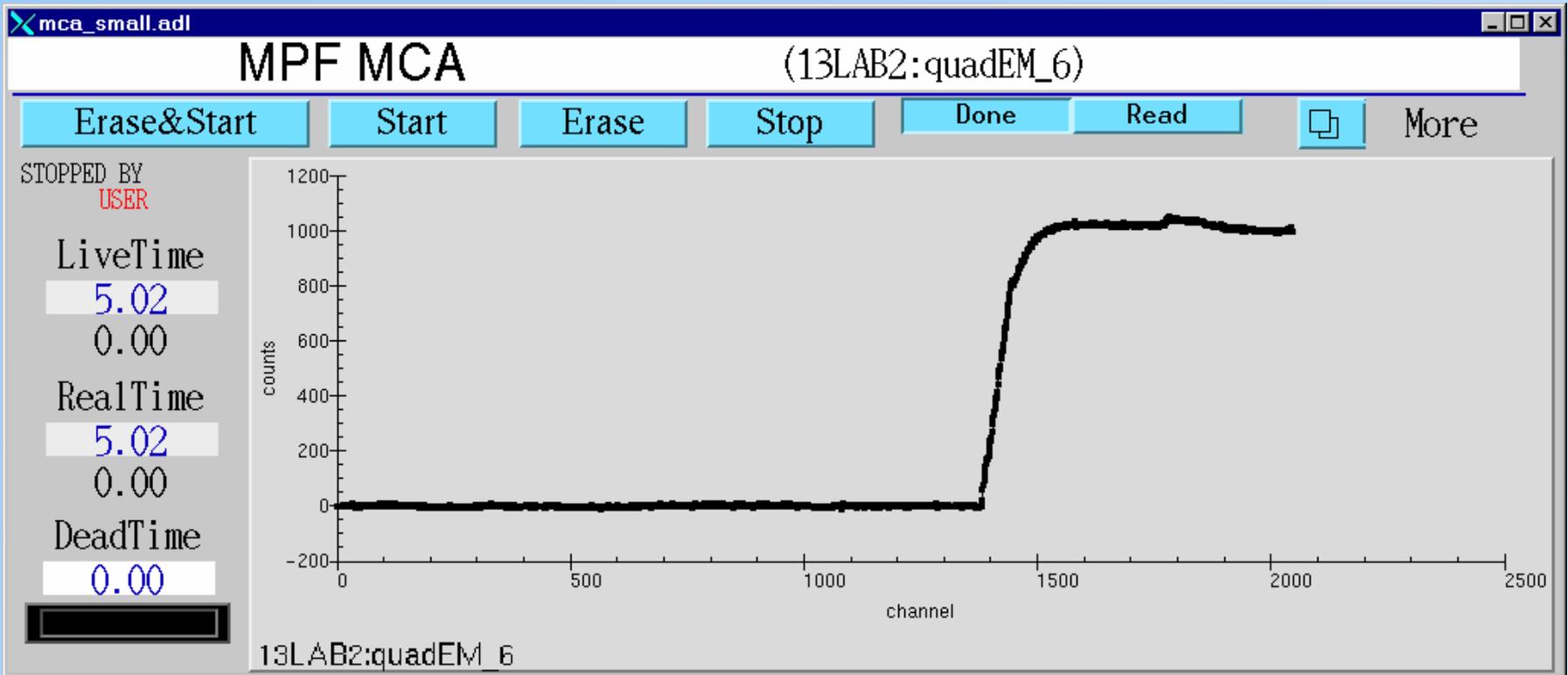
	Horizontal	Vertical
Position	5636	-1999
Setpoint	<input type="text" value="1500.000"/>	<input type="text" value="-2000.000"/>
Feedback	<input type="text" value="Off"/>	<input type="text" value="On"/>
Rate	<input type="text" value="1.226e-003"/>	<input type="text" value="1.226e-003"/>
Options	<input type="text" value=""/>	<input type="text" value=""/>
D/A	<input type="text" value=""/>	<input type="text" value=""/>

---

Conversion Time (usec)

Gain

Read rate

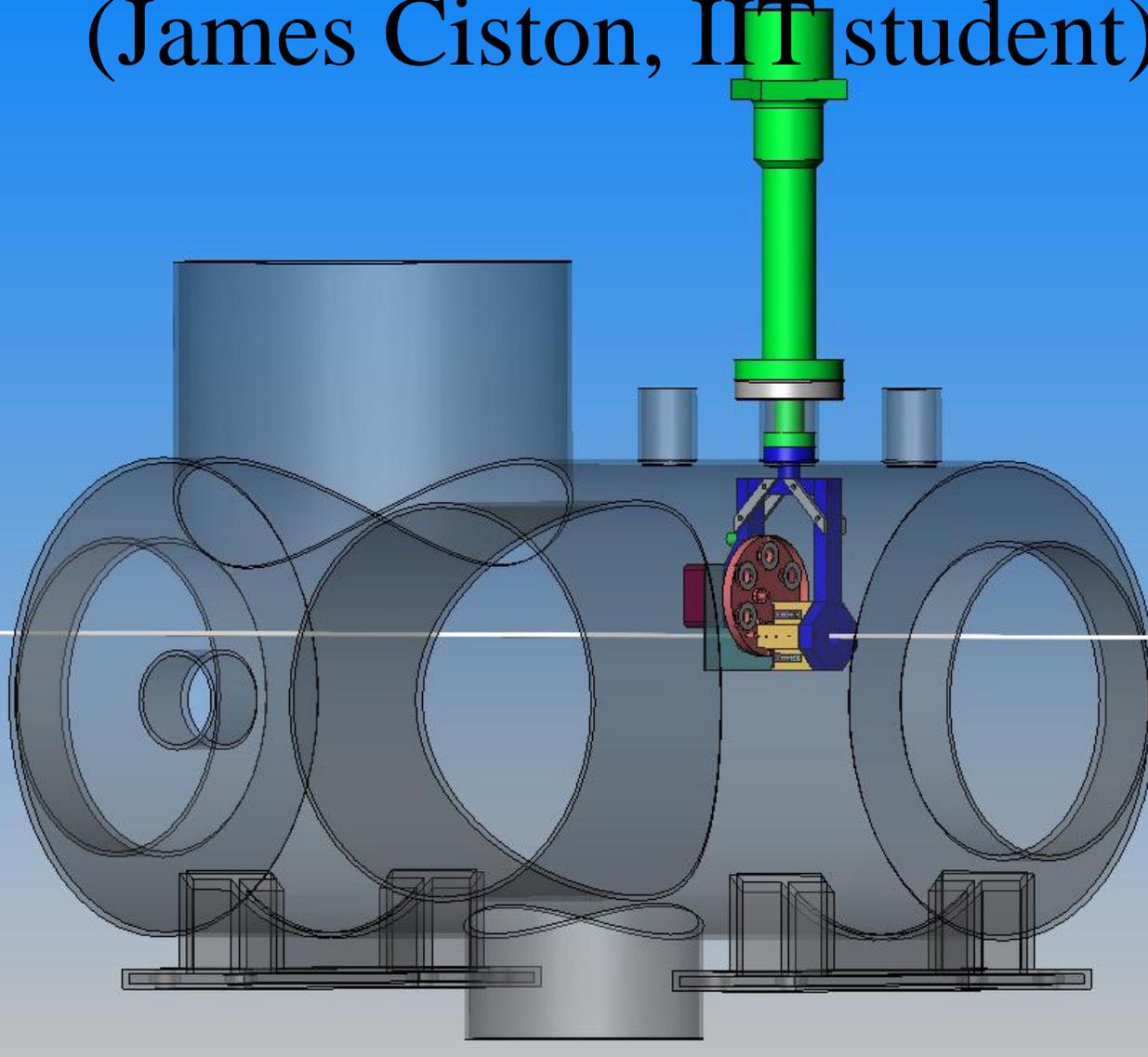


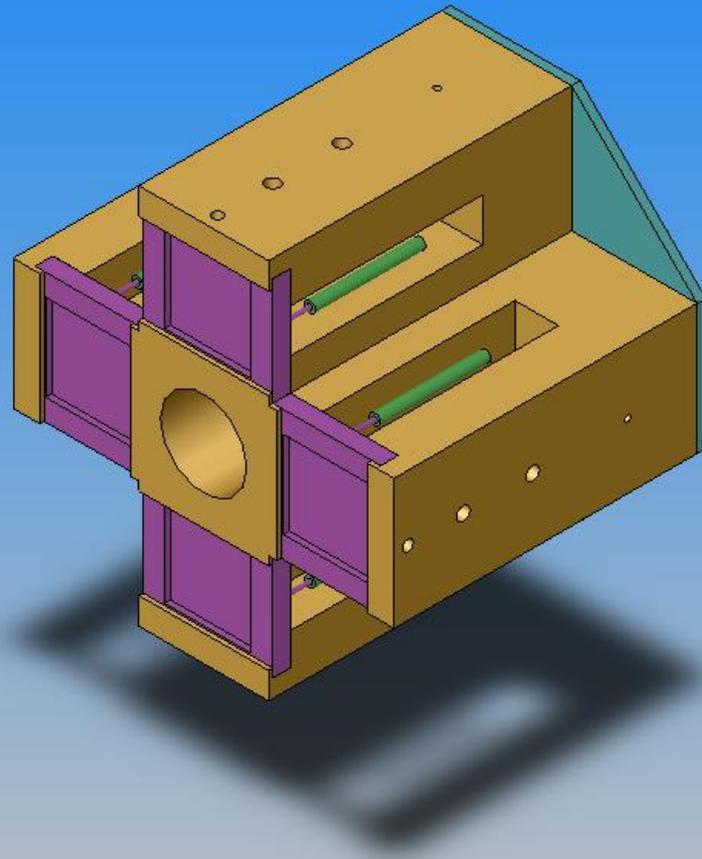
Plot of the laser beam position measured at 2.5 msec/point as the mirror is moved. This is the quadEMSweep server being read with the EPICS MCA record. Times as fast as 1.2 msec/point are available. These data can be read into an application like IDL to do spectral analysis of noise components, etc.

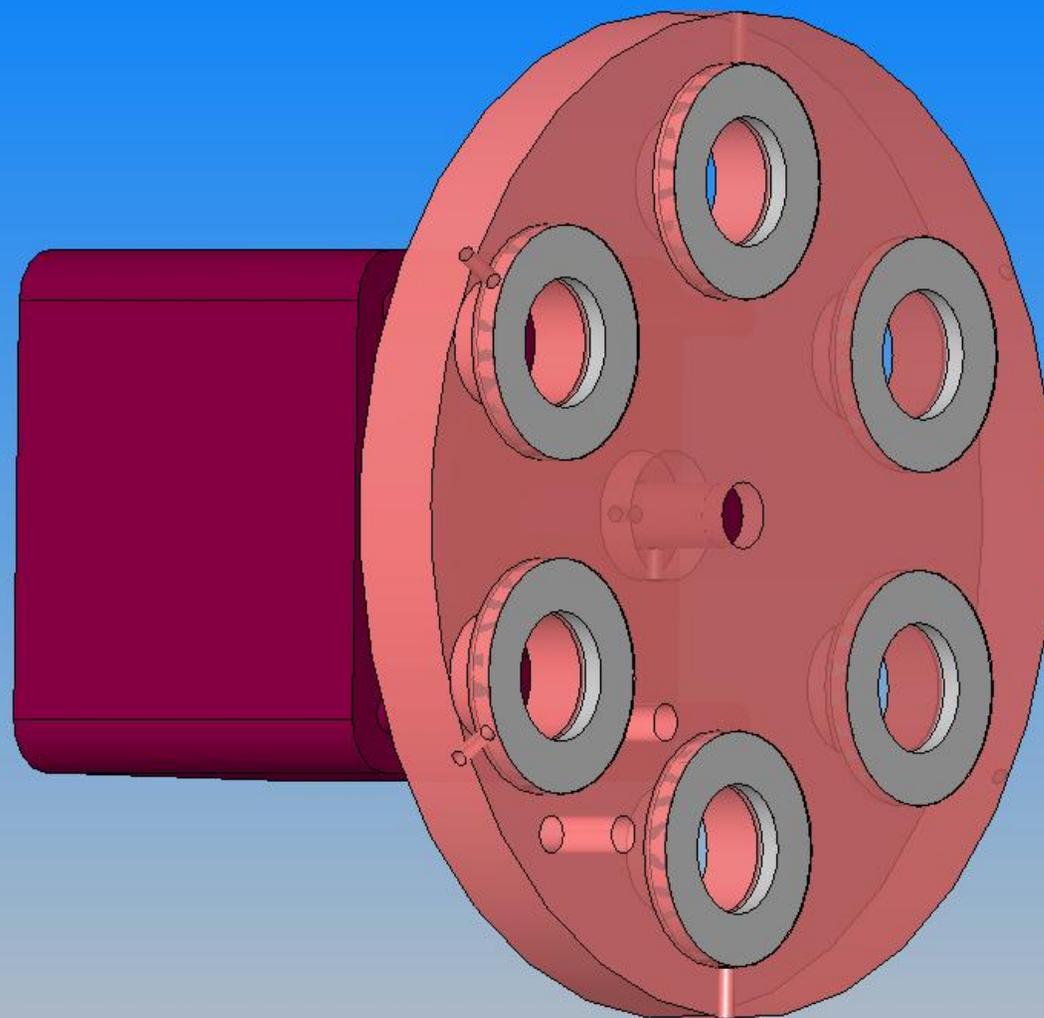
# Implementation Details

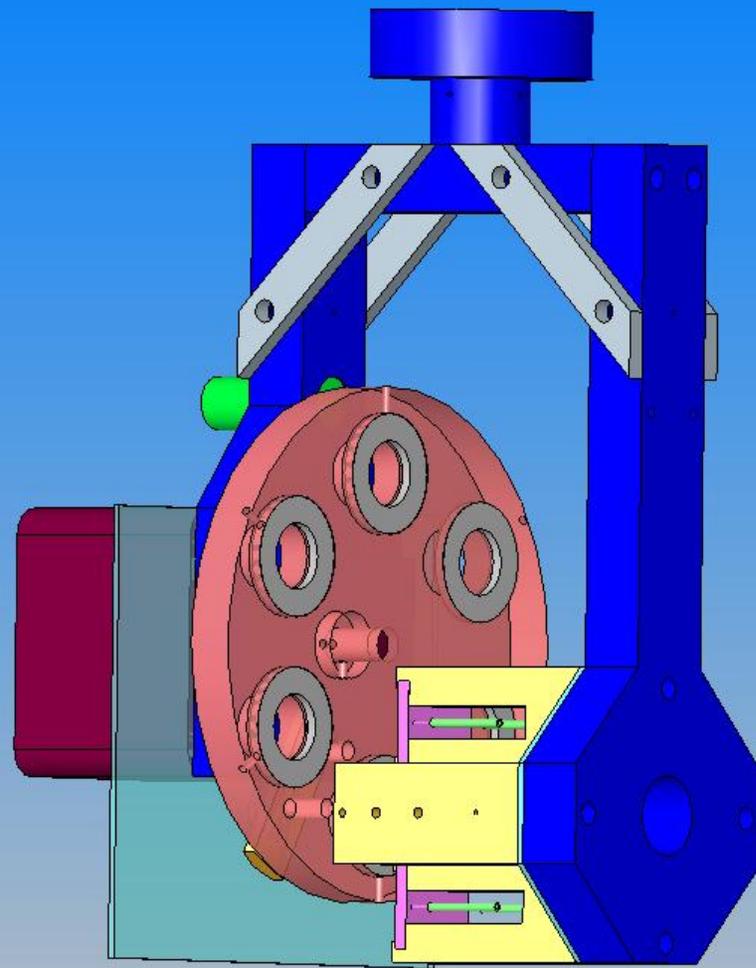
- Data comes from the electrometer into the VME system at up to 815Hz.
- Current VME boards do not support interrupts
- They do put out a TTL pulse when new data arrives, up to 815Hz.
- This pulse is input to an IP-Unidig Industry Pack I/O module, which does support interrupts.
- IP-Unidig interrupt routines calls the function to read the quad electrometer VME board.
- On each interrupt up to 3 quadEM functions are called
  - quadEMScan averages the current reading and returns averaged readings to EPICS “analog input” records
  - quadEMSweep puts the current reading into an array for an EPICS “mca” record. Performs averaging if the channel advance time is less than the electrometer clock rate.
  - quadEMPID uses the current reading to perform fast feedback via a Systran IP DAC. Performs averaging if the feedback rate is less than the electrometer clock rate.

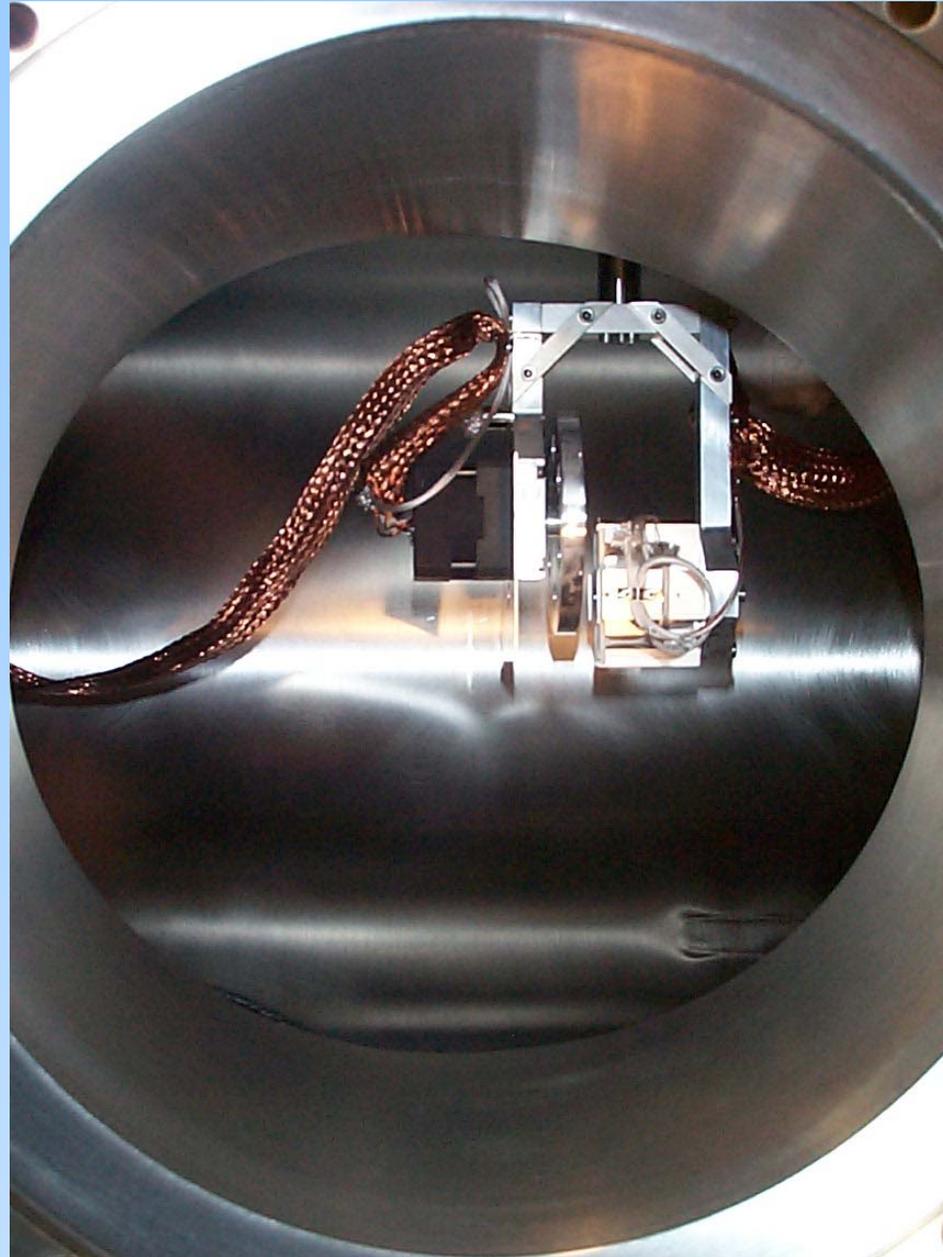
# GSECARS In-Vacuum BPM Design (James Ciston, IIT student)











# Current Status

- System “works” now but Steve Ross is working on the following problems/enhancements
  - Clean powerup/reset behavior
  - Interrupt support directly on VME board, no IP-Unidig needed
  - Convenient battery charge/switchover
  - Reduced gain by increasing capacitor. We saturated it with full undulator beam with 0.5 micron fluorescent foils
- EPICS software for 3.13 and 3.14 available now