Argonne
NATIONAL LABORATORY

Summary of EPICS Beamline Controls SIG Workshop

Mark Rivers

University of Chicago

June 14, 2006

THE UNIVERSITY OF CHICAGO

Office of Science
U.S. DEPARTMENT OF ENERGY

A U.S. Department of Energy laboratory managed by The University of Chicago

Organization of Workshop

- Organized and chaired by Bob Dalesio
- Presentations on the synApps EPICS beamline controls package
  - Tim Mooney
    - *Overview of what synApps is*
    - *Presentation on the autosave, calc, ip, sscan modules*
  - Mark Rivers
    - *Presentation on the asyn, ip330, dac128V, ipUnidig, mca, dxp, ccd, epid modules*
  - Ron Sluiter
    - *Presentation on the motor module*
  - David Kline
    - *Presentation on the love, vme, ebrick modules*
- Presentations on experiences and plans elsewhere with synApps
  - Wayne Lewis, Australian Synchrotron
  - Peter Dennison, Diamond
- Discussion of problems, future plans, etc.
  - Use of Source Forge as repository? Nick Rees to pursue and report.

What is synApps?

- A collection of EPICS **App**lication**s** for use at **syn**chrotron beamlines
- Commissioning-level software for ~80% of a new beamline
  - Basic support for common beamline devices
- Support for higher level, beamline-specific applications
  - Code libraries, implementation tools, examples
- Run-time tools to help users meet unanticipated needs
  - Can build software machines, feedback loops, serial support
- Support for collaborative development of beamline software
  - A vehicle by which developers at one beamline can easily contribute to software running on other beamlines.
- On the web:
  - www.aps.anl.gov/aod/bcda/synApps

Scope

- Mostly infrastructure and generic capabilities, as opposed to experiment/technique-specific programs
  - E.g., things like motors, scalers, and scans, as opposed to things like EXAFS, small-angle scattering, and protein crystallography
  - Why?
    - ***Economics*** *– Generic has the wider audience*
    - ***Information & expertise*** *– beamline developers have it; we don't.*
    - ***Control*** *– Generic software allows us to contribute to user software without taking control.*
- Mostly IOC-resident code, as opposed to client-side programs
  - E.g., scan software could run on workstations, but instead it runs on IOC's
  - Why?
    - *Access/coordination – IOC-resident code can be driven by anyone; clients generally cannot be driven by other clients.*
    - *Distribution/deployment – EPICS handles this for ioc-resident code.*

...Scope

- Some statistics:
  - 19 EPICS modules
    - *autosave, calc, camac, ccd, dac128V, dxp, ebrick, ip, ip330, ipUnidig, love, mca, motor, optics, quadem, sscan, std, vme, xxx*
  - 16 record types
    - *aCalcout, sCalcout, swait, transform, camac, dxp, motor, mca, table, busy, sscan, scanParm, epid, scaler, sseq, vme*
  - ~156 device types (~118 hard, ~38 soft)
  - ~197 EPICS databases
  - ~458 MEDM display files
  - ~526 c, c++, SNL source files (~281,000 lines of code)
  - ~77 documentation files (~38,000 lines of documentation)

synApps modules

autosave	parameter save/restore; maintains continuity through reboot
calc	run-time expression evaluation
camac	CAMAC support
ccd	CCD support
dxp	XIA DSP-based x-ray detector pulse analyzer
ebrick	EPICS-brick application
love	Love controllers (digital and analog I/O via serial)
Ip*, dac*	Serial, analog, and digital I/O
mca	multichannel analyzers and multichannel scalers
motor	stepper/servo motor
optics	monochromators, optical table, etc.
quadEM	four-channel fast analog input
sscan	scans, data storage
std	scalers, feedback, misc.
vme	VME hardware
xxx	runnable example of how everything in synApps is configured & used; typically controls a single experiment station

Who uses synApps?

- Synchrotron beamlines
  - ~20 sectors of APS
  - Swiss Light Source
  - NSLS (several beamlines)
  - Diamond
  - Australian Synchrotron
  - CLS, LCLS ?

- Other EPICS sites
  - Many non-synchrotron sites use a few synApps modules, notably
    - *motor*
    - *autosave*
    - *calc*
    - *mca, dxp, ccd*

synApps depends on software written by others

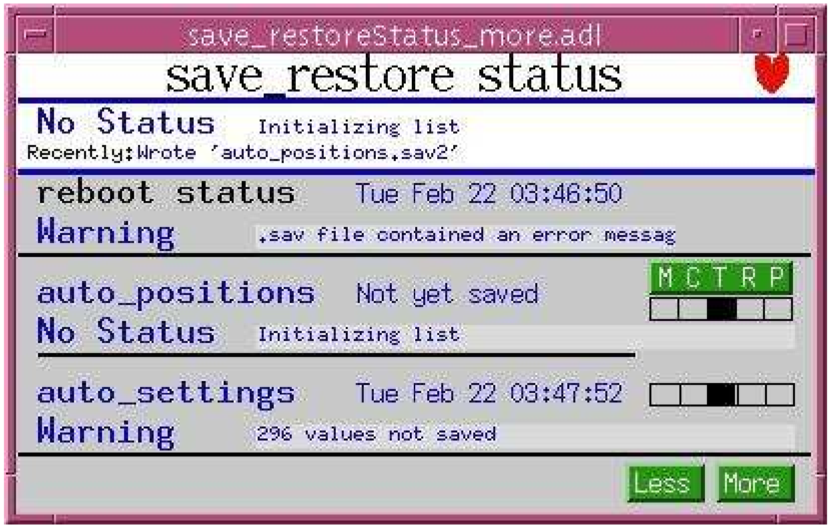
Module	Developer Org.	Purpose
asyn	APS Controls & CARS-CAT	support for asynchronous message-based control
ipac	APS Controls	IndustryPack carrier support
seq	SLAC	State-Notation-Language compiler
genSub	Observatory Sciences	General-purpose subroutine record
vxStats	SNS/ORNL	vxWorks status/statistics
allenBradley	APS Controls	support for communicating with Allen-Bradley PLC's

- synApps also uses (“*contains”*? “*second sources”*?) some software written by others that is not in the form of a module
- The idea is to avoid requiring synApps users to find, configure and test software that someone else has already found, configured, and tested.

autosave module

- Records values of selected EPICS Process Variables -- periodically, or in response to user-specified trigger
- Restores saved values when the computer restarts
- Can save/restore any scalar or array-valued PV
- Developer chooses default PV's to be saved; user can override
- Defends saved values from incompetent file server, crash, etc.

- Recent work:
  - v4.1.3 (synApps 5.2 candidate)
  - reduce sensitivity to errors
  - status-PV name length fixed
- Plans:
  - no immediate plans



```
save_restoreStatus_more.adl
save_restore status
No Status      Initializing list
Recently:Wrote 'auto_positions.sav2'
reboot status        Tue Feb 22 03:46:50
Warning        .sav file contained an error messag
auto_positions     Not yet saved
No Status      Initializing list
auto_settings        Tue Feb 22 03:47:52
Warning        296 values not saved
```

calc module

- Evaluate expressions entered at run time
- Records
  - **sCalcout** – like calcout, but also supports string expressions; user can specify wait-for-completion.
  - **aCalcout** – like sCalcout, but for arrays instead of strings;
  - **swait** – like calcout, but uses recDynLink (no “PP MS” link attributes)
  - **transform** – like 16 calcout records that share a PV data pool
- Other code
  - string/array-calc engines
  - sCalcout soft device support (with wait-for-completion option)
  - interpolation (lookup table), based on the genSub record

ip module

- Originally, all IndustryPack modules
- Now, only support for message-based devices
- device support, SNL code, databases, and MEDM displays for message-based devices
  - digital multimeters, current preamplifiers, temperature controllers, etc.

- deviceCmdReply
  - Used to write support at run time for one command/reply message
  - sCalcout to format output string
  - asyn record to write/read device
  - sCalcout record to parse reply

- devXxStrParm device support
  - to be replaced by streamDevice/asyn

optics module

- Slits and mirrors
- Monochromators
  - Nondispersive double-crystal
  - Dispersive double crystal
  - Spherical Grating
- Optical table
- Orientation matrix
  - (H, K, L) ←→ ($2\theta,\theta,\phi,\chi$) + constraint
  - User/client can write to underlying motors
- Automated alignment for zone-plate microscope

sscan module

- Support for user-programmable data-acquisition
  - **sscan** and **busy** records
  - saveData
  - recDynLink
- Recent work:
  - number of data points limited only by IOC memory
  - pipelined data storage
  - can mix scalar and array detectors
  - fixed some link-management bugs
  - added Python code to read, write, and operate on scan-data files
- Plans:
  - Include Dohn Arms' (APS/XOR-7) C-code utilities for scan-data files
  - Support 2D-array detectors

mca module

- Support for multi-channel analyzers
- mcaRecord
  - Like waveform record with lots of additional fields
    - *Start/stop acquisition*
    - *Preset live/real time*
    - *Regions of interest – total and net counts, can be used as EPICS scan record detector like a scaler*
  - Device independent
  - Primary device support uses asyn – also device independent
    - *Drivers implement int32, float64, int32array asyn interfaces*
- Devices supported
  - Canberra Ethernet AIM MCAs and programmable instrumentation
  - SIS multichannel scaler
  - Rontec XFlash (serial) MCA
  - XIA Saturn, DXP-2X and xMAP (support in dxp module)

ccd module

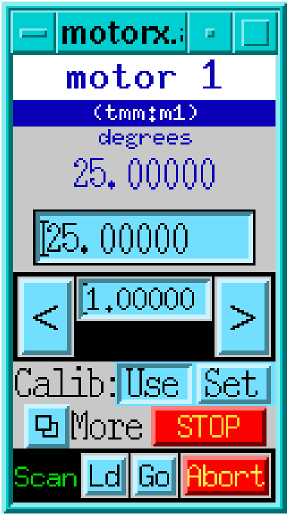
- Support for area detectors (CCD’s and image plates)
- Supported devices
  - MAR 165 CCD
  - MAR 345 image-plate reader (coming soon)
  - Roper (all WinView-supported CCD’s, including former Princeton and most former Photometrics devices)
  - Bruker SMART CCD
- Can control, at minimum
  - exposure time
  - file name
  - data-acquisition start
  - wait for acquisition to complete
  - much more for most devices

Modules for Industry Pack cards (dac128V, ip330, ipUnidig)

- Asyn drivers for A/D, D/A and digital I/O cards
- Use generic asyn device support for ai, ao, bi, bo, mbbi, mbbo records
- Support mca records (A/D as a 16-bit waveform digitizer)
- Support epid record for fast (up to 10 kHz) feedback
- Good models for how to write asyn drivers for other hardware

motor module - motor record

- Device independence – motor hardware is transparent to users.
  - Same medm screens and record level software, for all devices.



Available operations from this display;

1. Make absolute or incremental moves.
2. Define the current position.
3. Stop the current move.

without any controller specific information.

- Common device and driver level software for most devices.
- Unsupported motor record features are handled at the device level; typically, by ignoring the request.

Cont'd Feature list

- The scope of the motor record is limited to single axis, non-coordinated, point to point moves.
- Absolute, relative and incremental moves.
- Supports stepper, DC, piezo and Soft Channel motors. Very few fields are motor type specific;
  - PID parameters (PCOF/ICOF/DCOF) for DC motors.
  - Velocity base (VBAS) for stepper motors.
  - Done Moving Input Link (DINP) for Soft Channel.
- Three different position coordinate systems; user, dial and raw.
- Record level backlash correction.
- Homing.
- Device and driver support for 29 controller modules from 13 vendors

vme module

- Record support *vmeRecord*
  - Provides run-time access to VME bus (supported/unsupported)
  - Test and evaluate new hardware

- Device support
  - Acromag 9440 16-bit digital input/output
  - Acromag 9210 8 channel analog output (12-bit)
  - APS bunch-clock generator
  - Generic A32 VME interface
  - Heidenhain IK320 encoder/interpolator
  - HP 10895A Laser interferometer
  - Joerger scaler
  - APS machine-status link (MRD100)
  - Varoc SSI encoder
  - VMI4116 16-bit D/A

Australian Synchrotron

- **Benefits of synApps**
  - Reduced development time
    - *Common record types already developed*
    - *Existing drivers for wide range of hardware*
    - *Ability to put together a functioning experiment in a very short space of time*
    - *Interface screens already developed*
  - Consistency
    - *Maintain consistency with EPICS community*
    - *Access to support from developers*
    - *Retain ability to utilize future developments*
    - *Able to use existing developments that assume presence of synApps records*
    - *Interactions between records have been considered and allowed for*
    - *Creates the possibility of contributing back to the EPICS community*

Diamond

- Using some synApps modules
- Contributing enhancements to motor and mca modules already
- Doing some beamline things differently from synApps model

Discussion

- Problems:
  - Use of APS CVS repository is a bottleneck – impedes collaboration
  - Use of SourceForge being investigated
- RTEMS tested drivers needed. No RTEMS testbed at APS beamlines.