SoLID Slow Controls

Brad Sawatzky

What are Slow Controls

- "Infrastructure support" systems and logging
  - Status monitoring of power, vacuum, temperatures, etc
    - Includes logging and alarms/notification services
    - Safety interlocks between systems
      - fast valve closure on vacuum problems
      - disable power on temperature/cooling failure, etc.
  - Remote control of motors, pumps, actuators, stepper motors, etc...
  - Typical meas./response time scale on the order of 1 Hz
- Examples include
  - High voltage / Low voltage power controls (R/W)
  - 'Read-only' logging of temperatures, pressures, B-field, flow rates, ...
  - Magnet/Target control systems
    - Complex control process loops: vacuum, temperature, power
  - Gas systems
    - simple "set and forget" open loop STP systems w/o recapture
    - complicated control systems running a distillation/purification system
  - Etc...

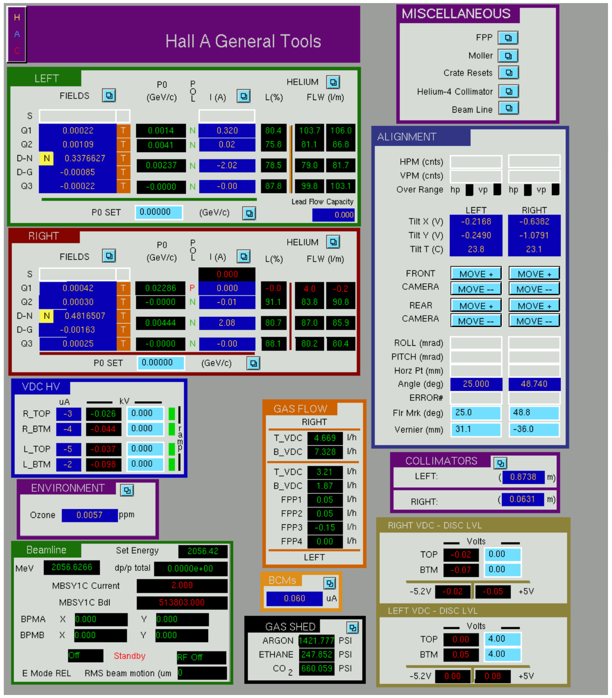
PLCs

- Programmable Logic Controllers
  - integrated hardware + firmware solutions
    - integrated systems often trade higher performance for the flexibility of 'hand-rolled' IOCs
  - modular, off-the-shelf components suitable for many common processes
  - Care needs to be taken to ensure good systems integration with the rest of the world
    - ie. built in EPICS interface very strongly encouraged
  - Allen-Bradley/Rockwell (ControlLogix, CompactLogix) PLCs in common use at JLab
    - will interface well with EPICS
  - Ideally, let's standardize on one vendor/system
- Magnet controls are good use case for PLCs

EPICS

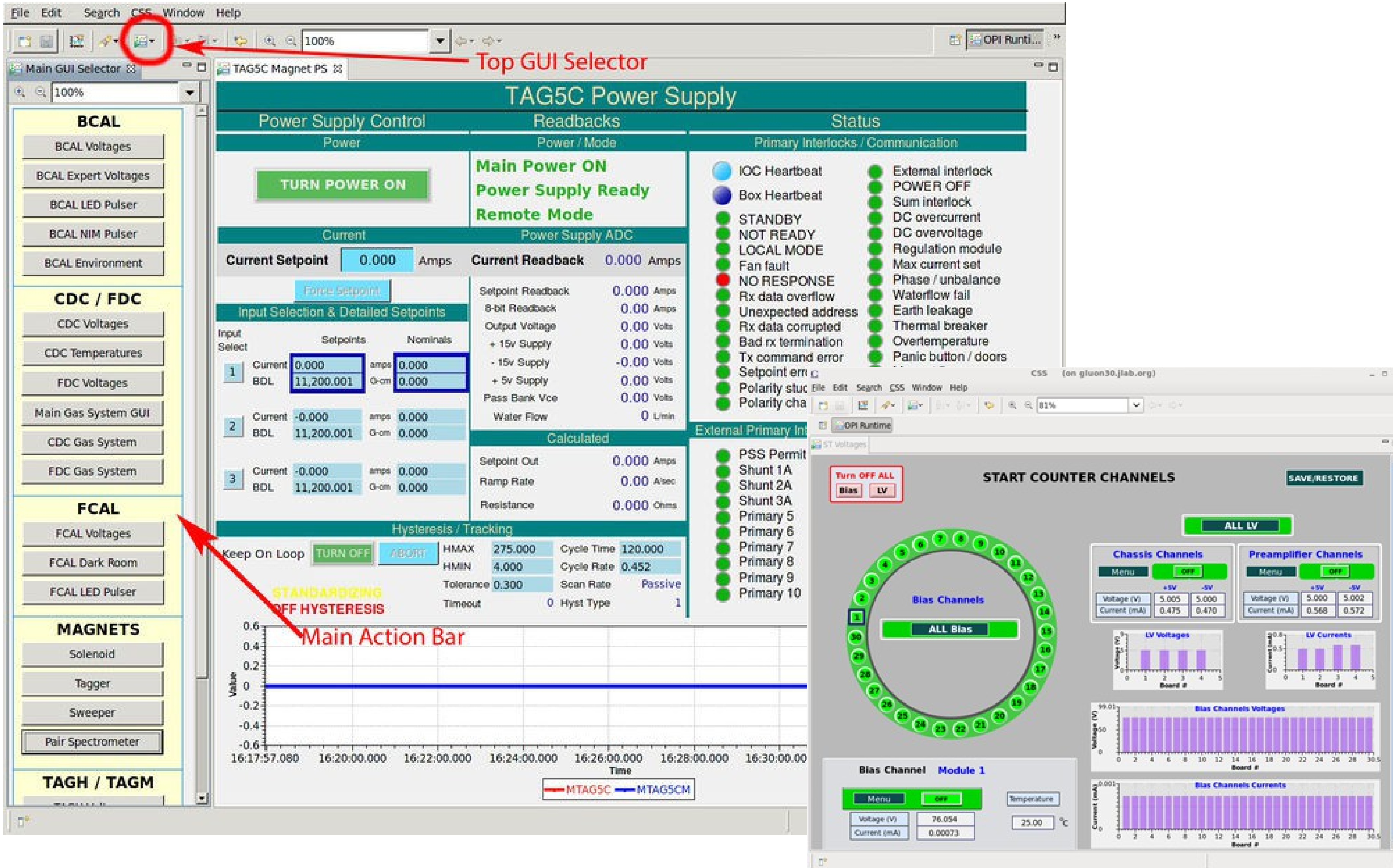
- Experimental Physics and Industrial Control System
  - http://www.aps.anl.gov/epics/
    - Open source, actively developed, lots of users
    - Based on C; APIs available for Java, Python, LabView, etc...
  - Covers both input/output controllers (IOCs) that do the real work
    - ie. poll for and respond to data in real time
    - publish data for other systems to consume
    - IOCs can be single board computers running vxWorks, embedded devices that supprt the EPICS protocols, or 'softIOCs' which are applications that can run under conventional OSes (linux, etc)
- Main slow controls 'backend' used at JLab
  - A lot of expertise in Accel Div. that we can leverage
    - However, we need to schedule (and budget for) the developer time well in advance!
  - Archiving of slow controls data can be integrated with existing (Accel) MYA Archiver

Frontend GUIs



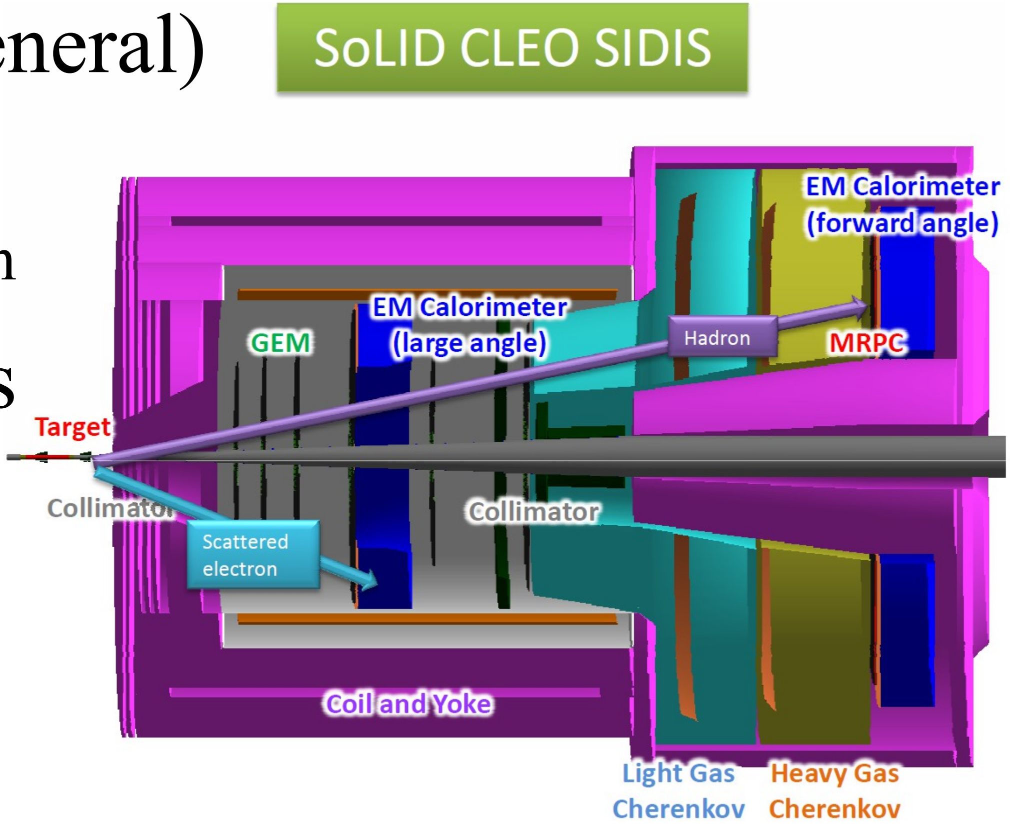
- EDM (MEDM) / JTABS
  - Forward-port of JLab's 6 GeV EPICS screens
  - Still developed, but dated
- Control Systems Studio
  - http://controlsystemstudio.org/
  - Eclipse-based toolkit designed for systems like this
    - SNS, BNL, FRIB, DESY using this system
    - JLab: Hall D (in use), Hall B (evaluating)
- Let's settle on some standards
  - Avoid LabView
  - Avoid custom/proprietary code as much as possible
    - if not possible, provide EPICS interface for integration

Hall D CSS example



SoLID Subsystems

- Magnet
- DAQ / Detectors (general)
  - Power (HV, LV)
  - Crate / Chassis selection
- Detector Subsystems
  - Ecal
  - LA/FASPD
  - Cherenkov
  - GEMs
  - MRPC



Magnet

- Complicated, lots of fast interlocks, high-risk, needs to be expert driven
  - Expert will pick what works best for them, hard to impose outside constraints…
  - One request:
    - Please allow for EPICS interface for easier integration into logging and DAQ systems

Detectors / Crates

- We want remote access to:
  - crate status: temperatures, fans, remote resets
- Standardize on a crate model:
  - all crates should have (at minimum) an ethernet interface on their controller
    - typically have SNMP support, etc, for monitoring/controls
  - select common (high-power spec'd) power supply module
- Wiener 60xx series in common use at JLab (VME/VXS standard)

Detectors / High Voltage

- High Voltage hardware should be standardized
  - CAEN SYx527 system (~$350/channel, incl crate)
    - Hall B / Hall D / Hall C
    - Built-in EPICS support, supplied controls GUI (java), other GUIs available on-site (Hall C)
  - Wiener MPOD system (Option 'B')
    - Hall D, Hall B SVT HV/LV
      - Hall B had some difficulty getting dedicated CAEN boards to work well with SVT (cooling, power, vacuum interlock related challenges)
    - SNMP-based EPICS interface exists
    - Cost ??
- Low Voltage
  - ??

Detectors – EC (Calorimeter)

- HV / LV power (previous slide)
- Fast interlocks / shutdowns?
- LED / Gain monitoring?
- Temperatures?
  - SiPM (if feasible) would likely require cooling

Detectors – LA/FASPD

- HV / LV power (previous slide)
- Fast interlocks / shutdowns?
- LED / Gain monitoring?
- Temperatures?

Detectors – Heavy Gas Cherenkov

- HV / LV power (previous slide)
- LED/Gain monitoring
- Gas flow/purity monitoring?
- Gas Temperature/Pressure regulation?
- Gas purification/recirculation systems
  - gets complicated/expensive quickly
  - pressure systems / code requirements typically mean proffessional engineering/designer support is mandatory
    - custom PLC/IOC design needed
  - C_4F_8O scarce, (C_4F_{10} even worse), long term options unclear...
    - All of the above gases expensive enough to need purification/distillation system on the scale needed for SoLID

Detectors – Light Gas Cherenkov

- HV / LV power (previous slide)
- LED/Gain monitoring
- Gas flow/purity monitoring?
- Gas Temperature/Pressure regulation?
- Gas purification/recirculation system?
  - CO_2 (SIDIS) can just flow (cheap, easy)
  - $CO_2 + C_4F_8O$ (PVDIS) mixing + purification system
    - mixing is easy, purification/reuse is complicated...

Detectors – MRPC

- HV / LV power (previous slide)
- Fast interlocks / shutdowns?
- Gain monitoring
- Gas flow, purity

Detectors – GEM

- HV / LV power (previous slide)
  - “Wiener-Iseg” system used at UVa
  - Can a 'standard' system be used for production, or is this the best technical choice?
- Fast interlocks / shutdowns?
- Gas system?
  - Ar/CO2 gas mix that just flows through is easy

Summary

- Think about and document slow control needs
  - Feed what you need/want to <brads@jlab.org>
  - I'm happy to do research and answer questions!
- Standardize, standardize, standardize
  - Avoid investing time in 'quick' solutions for local implementation. Stick with the standards – steeper learning curve, but it'll save time in the long run (build trained people as well as software).
  - Hacks and workarounds tend to become 'permanent' and unintended dependencies get baked in – good to avoid these.
- EPICS should be our common API/Protocol
- Frontend GUIs/software take time to develop
  - Can be good student projects, but needs sufficient lead time.