



# SC Progress Update



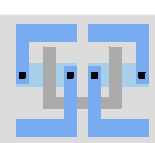
Michael Ritzert

[michael.ritzert@ziti.uni-heidelberg.de](mailto:michael.ritzert@ziti.uni-heidelberg.de)

16<sup>th</sup> International Workshop on  
DEPFET Detectors and Applications

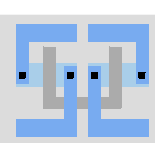
28.05.2014

- Hardware interfaces
  - DHH(C, +JTAG): see talk by Igor
  - ONSSEN
  - IBBelle
  - Environmental Sensors
- Interface to Belle II
  - New global RC/PSC scheme
- Alarms, Interlocks



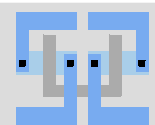
(Report from Bjoern Spruck)

- Reminder: The IOC for ONSSEN runs directly on the PowerPC in the FPGA.
- Cross compiling EPICS for the PowerPC was much harder than expected.
  - Now it is working reliably (Klemens Lautenbach).
  - Before we were using a version which was compiled on a NFS mounted development system, which tends to be time consuming, and had problem with the library compatibility to our Linux system.  
⇒ static linking, or additional copies of libs.
- Until now: Monitoring of few hardware registers (link up, IP/MAC address, free memory buffers...).
- Now we can start to move the complicated things to EPICS:
  - Initialization code/run start/stop
  - Access to monitor device (interrupts) etc.



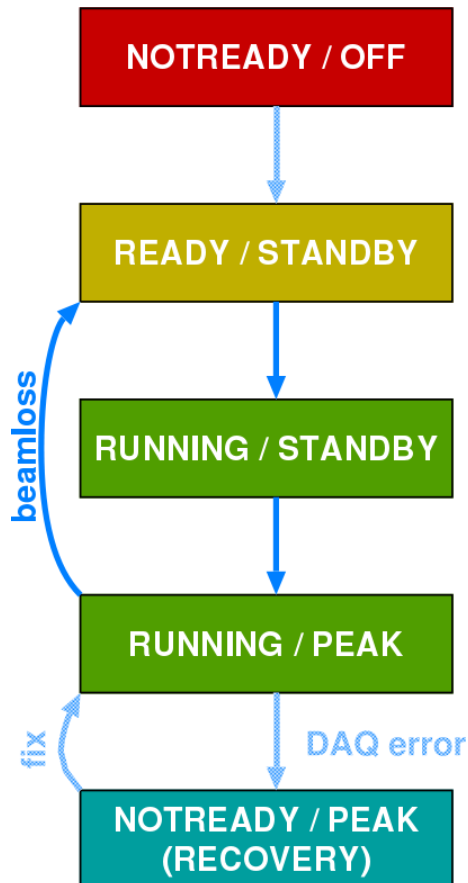
- Meeting at CERN to test EPICS with the actual PLC hardware. No problems on first try.  
⇒ Working proof-of-concept implementation available for a subset of the UNICOS objects<sup>1</sup>.
- For other objects: 50% copy-and-paste, 50% new work.
- Adding EPICS support to UNICOS is easy. We need to modify ~20 python scripts. 3 already done. 1 day of work left.
- On EPICS/CSS side:
  - Implement new CSS widgets following UNICOS conventions.
  - Implement OPI screens for device control.
- Work to be distributed between DESY, MPI, UH.
- First goal: All of IBBelle under EPICS control in time for commissioning of IBBelle.
- Second goal: Integration with the EPICS alarm system. (Development of suitable widgets by CSS core developers ongoing.)

<sup>1</sup> Object types: 4 Main categories, 20 types: Field (e.g. valve), Interface (e.g. config data), ....



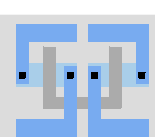
- Discussion with the Triest group about env. sensors started.
- Radiation: Diamond sensors with readout by AH501B (BEAST1) or successor (BEAST2).
  - Communication via TCP/IP.
  - Integration into EPICS for monitoring (should be) easy.
- Temperature (SVD):  
NTC thermistors read out by ELMB (from Atlas).
  - Communication via CAN bus.
  - Usual software Windows-only.
  - To be evaluated in detail (CANbus driver for EPICS is available).
- Temperature / Humidity: FOS.
  - As in the DESY testbeam.
  - But with an optical multiplexer to read more fibers.  
⇒ Some programming still to be done.

- New interface to Belle II: Power-supply Control.
- In the PXD case:
  - Controls all voltages that reach the DEPFET matrix.
  - Strong dependence on the RC.
- Definition not final, yet(?).



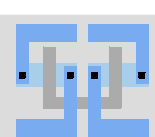
- Master state transitions shown on the left (from M. Nakao): RC-State / PSC-State
  - Mapping to PXD straightforward:
    - RC **READY**: All device SC connections established, no interlock triggered.
    - RC **RUNNING**: All systems configured, ready to take data. Implies LV fully on.
    - PSC **PEAK**: All voltages on.
- ⇒ same sequence as before, just new grouping into steps

- We need to collect information from our devices about
  - Abnormal conditions  $\Rightarrow$  alarms
  - Dangerous conditions  $\Rightarrow$  interlocks
- The required data is roughly the same:
  - Description of the condition
  - How it can be detected in the SC system
    - What data needs to be available in EPICS for monitoring?
    - How does it have to be processed?
  - What has to be done
    - Automatic shutdown (interlock)
    - Guidance for the operator to recover the situation (later automatic recovery)
- I'm looking for a good platform to manage the data.
  - Collect in the wiki?
  - Simple PHP interface to a database?



## Example Alarm Condition

- Description: ONSEN stops delivering data
- How detected:
  - throughput on network interface < threshold.  
PV Oxx:eth0:RATE < threshold
  - or reported data rate (hits per second) < threshold.  
PV Oxx:hitrate:RATE < threshold
- **Can be ignored:**
  - If not in physics run
- Type: major alarm (justification: data quality compromised)
- Recovery: send IPMI reset to crate on PV lxx:Oxx:reset:S
- Tree structure of errors:
  - Data flow broken
    - DHH stops delivering data
    - ONSEN stops delivering data
    - backpressure from B2 DAQ





Thank you!