



Progress in Porting CERN UNICOS to EPICS



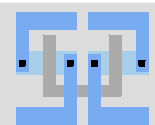
Michael Ritzert

michael.ritzert@ziti.uni-heidelberg.de

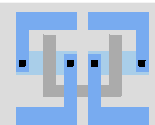
10.04.2014

Introduction

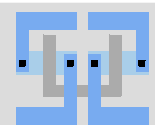
- The actual control of the cooling plant(s) is done in the PLC.
 - We are not involved in the development of this.
- EPICS/CSS will provide the supervision layer for the PLC, this is handled by PVSS/WinCC OA at CERN.
- CERN uses their UNICOS framework for the SC development.
 - Relevant functionality: List all hardware connected to the PLC in an Excel table, and generate both PLC code templates and PVSS databases from that.
 - Includes templates for control GUIs and animated widgets for schematic views.



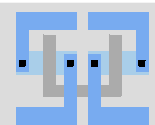
- Meeting at CERN with UNICOS developers and IBBelle PLC developers.
 - Bogdan Lobodzinski and Michael Ritzert present for VXD.
- Discussion on UNICOS SCADA requirements.
- Hands-on tests with the actual Schneider PLC hardware.
 - First connection between IBBelle-like hardware and EPICS.



- Combination of plain modbus, and TSPP (time-stamp push protocol, CERN-specific) on top of modbus.
- TSPP is not required to operate the PLC!
Advantages when it is used:
 - heartbeat signal
 - detection of short pulses on status bits

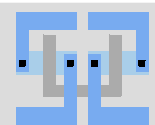


- At CERN, UNICOS outputs a database for PVSS.
- This is a text file with one line per device, containing all information.
- For EPICS, one line corresponds to a number of EPICS DB records.
- The text file is created by a number Python scripts that can easily be modified.
- It should be not problem at all to adapt the output format for EPICS.



PLC Connection Test

- A hand-written EPICS database has been used in this first test.
- No actual hardware connected to the PLC.
- Just a single device (a valve) configured in EPICS.
 - 10 PVs to represent the various registers in the PLC.
- OPI resembling the UNICOS „faceplate“ (dialog to control the device) and OPI widget following the UNICOS widget specifications.
- No problems observed with the modbus connection.
- Sending a command to the PLC worked in the first try.
- **Very successful proof of principle for the planned implementation.**
- The PLC remains at CERN to allow us to run PVSS in parallel to compare the behavior with our EPICS implementation.



„Faceplate“

- What is visible is working!
- Still some minor issues with disabling buttons when they are ineffective. To be implemented in the CSS OPI Editor.

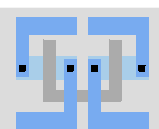
The screenshot displays the CV142 control panel interface. At the top, there are tabs for 'Status', 'Trends', and 'Debug'. The main area is divided into several sections:

- Status:** Includes input fields for 'Position' (18,0 %), 'Limit On' (90,0 %), and 'Limit Off' (10,0 %). A 'Failsave On / Opened' checkbox is checked.
- Operation Modes:** A table with columns for mode and status indicators.

Mode	Indicator 1	Indicator 2
Auto	<input type="checkbox"/>	<input type="checkbox"/>
Manual	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Forced	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Local	<input type="checkbox"/>	<input type="checkbox"/>
Hardware Local	<input type="checkbox"/>	<input type="checkbox"/>
- Alarms:** A list of alarm indicators, all with green status boxes: Full Stop Interlock, Temporary Stop Interlock, Start Interlock, Alarm, and Alarm Not Ack.
- Warnings:** A list of warning indicators, all with green status boxes: Event Masked, Allow restart needed, Alarm blocked, I/O Error, I/O Simulated, Manual <> Auto, and Position Warning.
- Ranges:** Input fields for 'Range Max.' (100,0 %) and 'Range Min.' (0,0 %).
- Requests:** Input fields for 'Auto' (18,1 %), 'Manual' (0,0 %), and 'Active' (18,4 %).

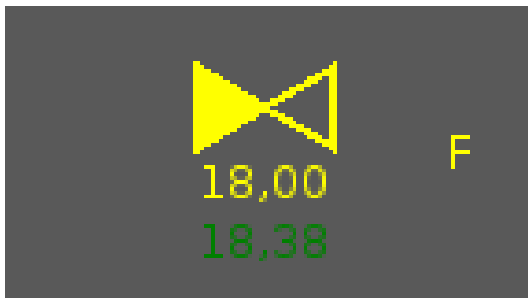
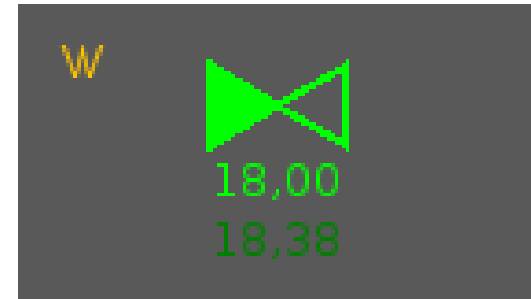
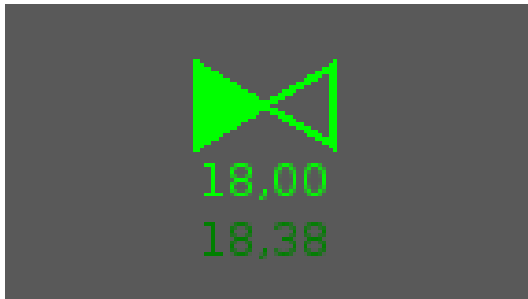
At the bottom, there are two rows of control buttons:

- Row 1: 'On', 'Off', 'Set Value...', 'Inc.', 'Dec.', 'Allow Restart'.
- Row 2: 'Auto Mode', 'Manual Mode', 'Forced Mode', 'Ack. Alarm'.



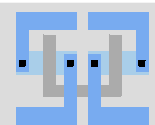
OPI Widget

- Implemented in Java as an Eclipse plugin to CSS.



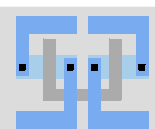
A lot of TODOs

- Adapt the UNICOS script to output in EPICS format.
 - Fairly easy
- Design more faceplates.
 - Some trivial, some requiring still-to-be-developed widgets.
- Extend the script to also output the archiver configuration.
 - Also quite easy.
- Integrate the PLC alarm handling with the CSS alarm handling.
 - No good idea, yet, especially regarding acknowledging alarms.
- Implement TSPP at least far enough to handle the heartbeat.
 - Still lacks documentation, but probably < 1 week of work.
 - Probably means extending the EPICS modbus driver.
- Understand the concept of recipes for the PLC and maybe implement the interface.



Conclusions

- The first test with IBBelle-like hardware went extremely well.
- The concept to integrate the PLC with the rest of our SC system is sound.
- The biggest open point is the alarm system integration.
- The rest is „just work“.
But far less than the few man years first estimate.
In principle, we could run within a few weeks with only few shortcomings.



Thank you!