VISUALIZATION OF INTERLOCKS WITH EPICS DATABASE AND EDM EMBEDDED WINDOWS

Evgeniy Tikhomolov, TRIUMF, Vancouver, Canada

Abstract

The control system for TRIUMF's upgraded secondary beam line M20 was implemented by using a PLC and one of many EPICS IOCs running on a multi-core Dell server. Running the IOC on a powerful machine rather than on a small dedicated computer has a number of advantages such as fast code execution and the availability of a large amount of memory. A large EPICS database can be loaded into the IOC and used for visualization of the interlocks implemented in the PLC. The information about interlock status registers, text messages, and the names of control and interlock panels are entered into a relational database by using a web browser. Top-level EPICS schematics are generated from the relational database. For visualization the embedded windows available in the Extensible Display Manager (EDM) are the EPICS clients, which retrieve interlock status information from the EPICS database. A set of interlock panels is the library, which can be used to show any chains of interlocks. If necessary, a new interlock panel can be created by using the visualization tools provided with EDM. This solution, in use for more than 3 years, has proven to be reliable and very flexible.

MOTIVATIONS

The new control systems at TRIUMF are implemented by using the Experimental Physics and Industrial Control System toolkit [1]. To control the devices in the upgraded M20 secondary beam line, Programmable Logic Controllers (PLCs) are used [2]. One of the tasks of the control system is to visualize the state of interlocks for various vacuum devices and power supplies. The present scheme for visualization of interlocks, which is used at TRIUMF was designed more than 15 years ago [3], when the old Display Manager with limited capabilities was used, and EPICS support was located in VME Input-Output Controllers (IOCs), which had only 16 Mb of memory, and 100 Megabit network connections.

At present EPICS support can be realized as an application running on a Linux server (we call it PLCIOC). This application communicates with the PLC by using the EPICS driver on one hand and with the EPICS clients on the other. Thus, PLCIOC plays the role of a transfer agent between EPICS clients and PLC: EPICS clients communicate via the network with this server, send/receive commands and set/get values from devices via PLCIOC (Fig. 1).

At present there is effectively no limitation for memory usage by PLCIOC. EPICS database can be of very large size and keep lots of information. At present we use a multi-core Dell PowerEdge 600 with 8 Gb of memory. Memory usage by M20 PLCIOC serving ~100 devices is only 1.5%. Processing information is very fast on such multi-core computers. CPU usage for M20 PLCIOC is ~12% for one core (out of 8). Transfer of data between EPICS clients and PLCIOC is also very fast for a Gigabit network connection.

Another favourable reason for new development was the addition of a new feature to the Extensible Display Manager (EDM) [4] in 2008, namely, possibility of assignment of macro value by a script or executable. Thus, it became possible to implement control device panels, in which interlocks can be represented by an embedded interlock panel, and all necessary information is retrieved from EPICS database (Fig. 2).

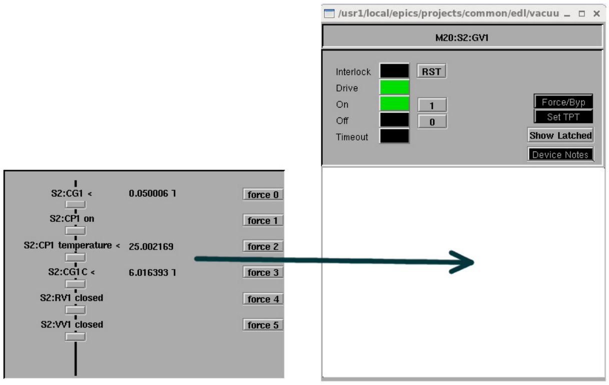


Figure 2: Interlock panel (on the left) is embedded in lower part of control device panel (on the right).

The goals of new implementation for interlock visualization are:

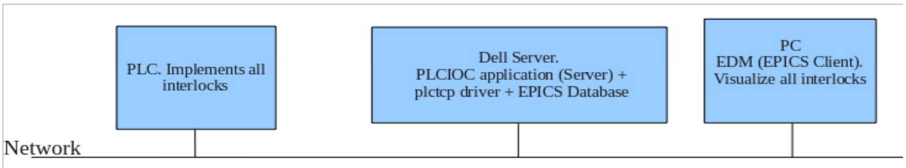

PLC. Implements all interlocks
Dell Server.
PLCIOC application (Server) +
plctcp driver + EPICS Database
PC
EDM (EPICS Client).
Visualize all interlocks
Network

Figure1: Communication between PLC, PLCIOC, and EPICS clients.

ISBN 978-3-95450-148-9

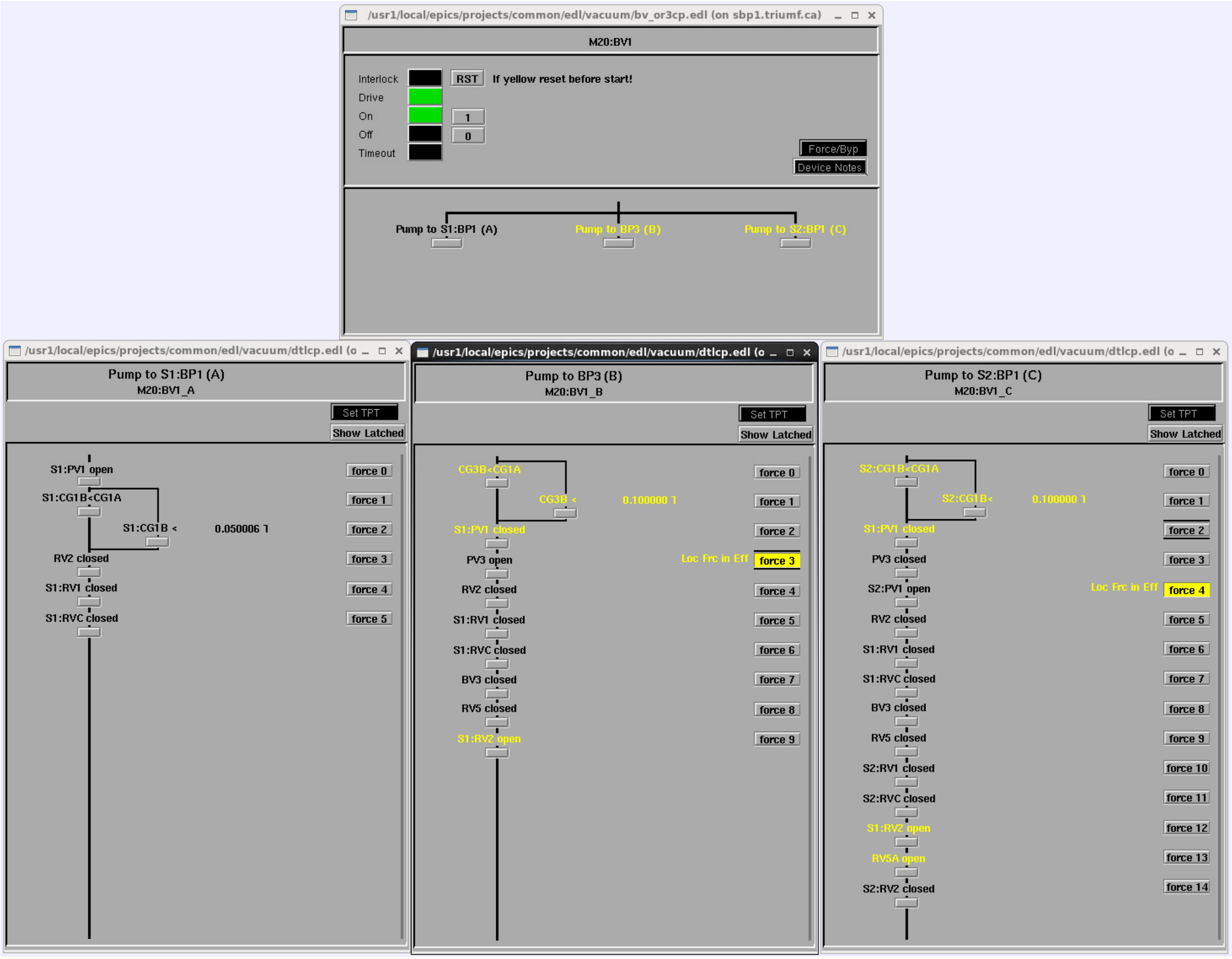


Figure 3: Example of hierarchical interlock chains implemented for M20 Secondary Beam Line.

- Less development time for new Control Systems
- More flexible scheme for adding new types of devices and types of interlocks
- Simplification of maintenance (fixing errors in PLC code and relational database)
- Simple handling of very complicated interlocks
- Simple adding of new features (like variable trip points, local force/defeat of the interlocks, applying different colour rules, etc.)

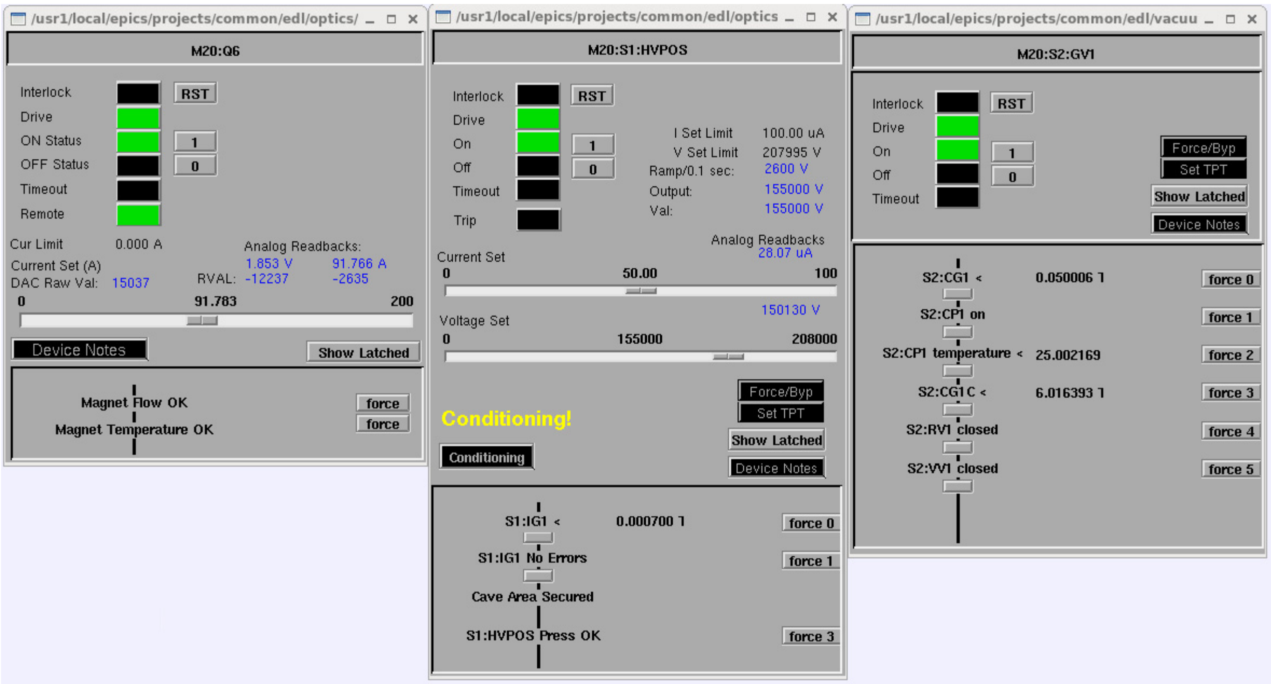
BACKGROUND AND SOME DEFINITIONS

All interlocks are implemented in the PLC. The EPICS screens (device control panels + interlock panels) are used only for visualization of the interlocks. The strict correspondence between implementations of interlocks in the PLC and visualization in EPICS must be maintained during the development. Software support for M20 control system was implemented mostly by 2 people. One person developed PLC code and another EPICS support. The approved Interlock Specification (like shown in Table 1) was used by both developers to work on their tasks in parallel. At the commissioning stage the full implementation of control system (PLC+EPICS) was tested automatically by using developed Perl scripts.

Table 1: Interlock Specification

Device name	**Device Type**	**OK to open/set on**
M20:RV2	Roughing valve	A and B A: M20:CG1A< M20:CG2 or M20:CG1A < 50 mTorr B: M20:RV1 closed M20:RVC closed

The concept of 8-bit and 16-bit Status Registers was used both in PLC code and EPICS schematics. The



/usr1/local/epics/projects/common/edl/optics/

M20:Q6

Cur Limit 0.000 A

Current Set (A)

DAC Raw Val: 15037

Analog Readbacks:

1.853 V 91.766 A

RVAL: -12237 -2635

0 91.783 200

Magnet Flow OK

Magnet Temperature OK

/usr1/local/epics/projects/common/edl/optics

M20:S1:HVPOS

I Set Limit 100.00 uA

V Set Limit 207995 V

Ramp/0.1 sec: 2600 V

Output: 155000 V

Val: 155000 V

Analog Readbacks 28.07 uA

Current Set

0 50.00 100

150130 V

Voltage Set

0 155000 208000

Conditioning!

S1:IG1 < 0.000700 1

S1:IG1 No Errors

Cave Area Secured

S1:HVPOS Press OK

/usr1/local/epics/projects/common/edl/vacuu

M20:S2:GV1

S2:CG1 < 0.050006 1

S2:CP1 on

S2:CP1 temperature < 25.002169

S2:CG1C < 6.016393 1

S2:RV1 closed

S2:VV1 closed

Figure 4: Visualization of interlocks for M20 optical and vacuum devices.

Status Registers contain bits which reflect status of interlock signals. Usually, “1” means interlock condition (i.e. NOT OK) and “0” means no interlock condition (i.e. OK). Some devices can have several status registers to incorporate more than 16 interlocks. Each signal used in registers is latched after getting NOT OK status by PLC code. Reset signal is sent from EPICS device Control Panel to PLC and clears latched interlocks. A single Web Form for entering all kinds of information into relational database was used for any interlock chains.

IMPLEMENTATION

Interlock Visualization in EPICS Control Panels

Device control panels were designed manually by using EDM. Each Control Panel has a container. The reason for using a container is a limitation of EDM: it doesn't allow a script in the File field for an Embedded Window. Otherwise, only a Control Panel with an embedded window for interlocks could be used.

All interlock panels used for visualization of interlocks were also created manually by using EDM. A library of interlock panels can be used for development of any control system. All necessary information for visualization is taken from EPICS Process Variables (PVs), which reside in PLCIOC by using small scripts called by the container. The size of the scripts is very small, ~20 lines.

Production versions of device panels with embedded interlock panels for rather complicated hierarchical vacuum interlocks are presented in Fig. 3. The visualization scheme is universal: for M20 it is used both for optical and vacuum devices in similar manner (see Fig. 4).

Implementation of Interlock Visualization in EPICS Database

A relational database was used for all necessary interlock information and to generate an EPICS database [5]. Special EPICS symbols with the names like plcdevint.sym were created by using TDCT [6]. Related schematic contains all interlock information. All device interlock symbols are aliased in relational database to the names like DEVINT, which are associated with symbol files (CGINT, CPINT, FGINT, etc.)

Using Multiple Status Registers

For each STATUS register the corresponding PV name is constructed as shown here:

<Device Name>INT<Device ID>_S<NSTATUS>,

where NSTATUS is the unique instance of the Status Register. For example, M20:RVINT2_S2.

Rules for Construction of Interlock Panel Names

The Interlock Specification provides logical equations for interlocks. Enumeration of bits starts with zero. For example:

(Bit_0 and Bit_1) and (Bit_2 or Bit_3 or Bit_4) and Bit_5 and Bit_6 and ... (some more bits)

Such equations correspond to the ladder logic of the PLC code. The name of interlock panel can be constructed by using logical equations and applying the following simple rules:

- dash sign represents "and" for groups before and after the sign (corresponds to vertical lines on interlock panels)
- if several bits are used in "and' statements they are combined into the range of bits by using colon sign, for example, 0:1
- the word "or" is used every time to avoid confusion

Thus, the name of the interlock panel, which represents the above mentioned logical equation, looks like:

0:1-2or3or4-5:15.edl (see Fig. 5)

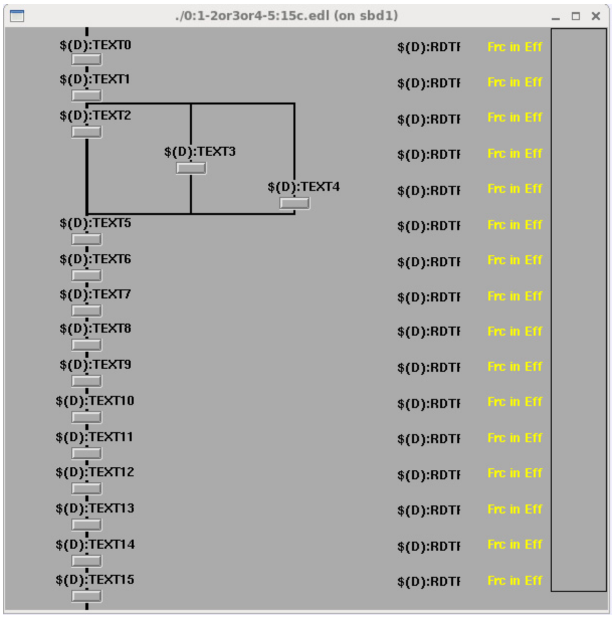


Figure 5: Construction of name for interlock panel from logical equation.

CONCLUSION

A complete scheme for interlock visualization was created. It was used for 3 years on the TRIUMF M20 Secondary Beam Line and proved to be easily maintainable and extensible. It can be recommended for development of any new control system, which uses the EPICS toolkit. The major steps for developing a new system are presented in the Appendix.

ACKNOWLEDGEMENT

The author is thankful to TRIUMF Controls groups and personally to Rolf Keitel and Mike Mouat for providing a very good creative environment and basis for development of the project presented.

APPENDIX

Major Steps to Implement Interlock Visualization with Embedded Windows

- Define PV name for each device. It should have a structure:

<Device Name><Device ID>,

where <Device Name> includes names for Subsystem, sub-subsystem, and device type, for example: M20:S1:RV2

- For each device construct a corresponding interlock device name:

<Device Name>INT<Device ID>, for example: M20:S1:RVINT2.

This name is used for creation of a device using the standard initial web form for relational database

- Enter all required information into relational database via web page form
- build EPICS database and load into PLCIOC
- if necessary, create new device control panel and interlock panel by using EDM

REFERENCES

[1] http://www.aps.anl.gov/epics/. Experimental Physics and Industrial Control System.

[2] http://www.schneider-electric.com/en/product-range/1468-modicon-m340/

[3] R. Keitel, R. Nussbaumer, "Automated Checking and Visualization of Interlocks in the ISAC Control System", Proceedings of ICALEPCS 2001, Nov 27-30, 2001, http://arxiv.org/abs/physics/0111094

[4] http://ics-web.sns.ornl.gov/edm/

[5] Keitel, R., "Generating EPICS IOC databases from a relational database: A Different approach", Proceedings of ICALEPCS 2001, Nov 27-30, 2001, http://arxiv.org/abs/physics/0111048

[6] Keitel, R., "TDCT – A Configuration Tool for EPICS Runtime Databases," Proceedings of ICALEPCS 2009, Oct 12-16, 2009