Tango Based Software of Control System of LIA-20

A. Senchenko1,2, G. Fatkin1,2, S. Serednyakov1,3, P. Selivanov1
BIP SB RAS, NSU
e-mail: a.isenchenko@inp.msk.ru

Introduction

The linear induction accelerator LIA-20 for radiography is a pulsed machine designed to provide three consecutive electron bunches. Since every pulse is a distinctive experiment, it is of high importance to provide coherence of the facility state and the experimental data. This paper presents overall software architecture. Challenges and particular approaches to designing of a pulsed machine control system using Tango are discussed.

Control System Project

For more details visit T1PHA052

The software of the control system is based on Tango 8. At the current moment almost all VME modules are provided with tango devices and GUI clients. Low-level tango devices are implemented in C++, Client applications are created using Python language and PyQt/PyTango/Taurus. HDB++ with Mysql backend was chosen as an archival system.

Further development will be directed to the creation of high-level tango devices and possible introduction of Sardana. Another important field of research is the Facility State And Regime Management System.

Application Software

User software comprises Mimic Diagram, Time Editor, GUI for tango devices and common tango utilities. Mimic Diagram visualizes a summary of all subsystem's states. It is based on PyQI/WebKit and SVG. Time Editor is a editing tool for timing diagrams. It provides operator the ability to prepare, verify and apply timing diagrams.

For more details visit TUPHA 087

Dedicated virtual servers

Virtualization

KVM

TFTP Server

Tango DB
TFTP, NFS

Archive Server

HDB++ MySQL / Postgresql

High-Level Control Server

FacilityStateMachine
Interlock, Timing, etc.

Data Rates

<table>
<thead>
<tr>
<th>Channel type</th>
<th>Number of channels</th>
<th>Data rate (l-bunch)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>whole system</td>
<td>per VME crate</td>
</tr>
<tr>
<td>Fast (&lt;10us) oscillograms</td>
<td>594</td>
<td>22</td>
</tr>
<tr>
<td>Slow oscillograms</td>
<td>1485</td>
<td>55</td>
</tr>
<tr>
<td>Timing system</td>
<td>1485</td>
<td>55</td>
</tr>
<tr>
<td>Interlocks</td>
<td>1485</td>
<td>55</td>
</tr>
<tr>
<td>Technological controls</td>
<td>1000</td>
<td>~40</td>
</tr>
<tr>
<td></td>
<td>6000</td>
<td>~280</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Low level I/O layer consists of CANBus driver (can4linux or SocketCAN) and VME wrapper.

High level I/O layer provides an abstraction libraries for access to CANBus and VME and hides implementation details. A set of Device Drivers is implemented on top of VME wrapper.

Tango is the top layer. It contains tango interfaces to the underlying Device Drivers. To reduce VME controller’s resource consumption and simplify hardware access appropriate tango devices are aggregated in two device servers: CANBus and VME.

ICALEPCS 2017 - TUPHA169