Software and Computational Infrastructure of LIA-20 Control System

A. Senchenko, G. Fatkin, S. Serednyakov (1,2), P. Selivanov (1) 1 - BINP SB RAS, 2 - NSU

Abstract

The linear induction accelerator LIA20 for radiography is currently under construction at Budker Institute of Nuclear Physics. This paper presents software architecture and computational infrastructure for the accelerator controls. System and application software are described. Linux operating system is used on PC and embedded controllers. Application software is based on TANGO. Overall data transfer rate estimations are provided.

Regimes

1. Cycle interval: 10.15 sec (4.6 cycles/min)
2. 1/2/3: bunches per cycle
3. Test regimes

Data Rates

<table>
<thead>
<tr>
<th>Channel type</th>
<th>Number of channels</th>
<th>Data rate (1 bunch)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Whole system</td>
<td>gigaMbyte/second</td>
</tr>
<tr>
<td>Fast (shorter than 10 us)</td>
<td>594</td>
<td>5.7 MByte/second</td>
</tr>
<tr>
<td>Slow, Sequencing, line PS</td>
<td>1465</td>
<td>135 MByte/second</td>
</tr>
<tr>
<td>Timing Delay Lines, Timers</td>
<td>1465</td>
<td>135 MByte/second</td>
</tr>
<tr>
<td>Interlocks</td>
<td>1000</td>
<td>55 MByte/second</td>
</tr>
<tr>
<td>Technological controls</td>
<td>6000</td>
<td>13.3 MByte/second</td>
</tr>
</tbody>
</table>

Computational infrastructure

Server:
- CPU 2.0 GHz, Cores 4
- Intel x86-64
- RAM 32GB
- Gigabit Ethernet
- 4TB SCSI
- RAID 5
- Virtualization: KVM

VME Crate Controller:
- PowerPC-based
- Diskless network boot

Operator’s PC:
- CPU 2.2 GHz, Cores 4
- Intel x86-64
- RAM 4GB
- Up to 4 monitors

OS:
- Server: Ubuntu LTS
- VM: Ubuntu LTS
- Operator’s PC: Ubuntu LTS
- Controller: Debian

Network:
- 3 * 24 port switches

System and Application Software

TANGO

The use of virtualization facilitates maintenance operations and provides dedicated OS for group of services. VM1 is used for network boot. Kernel image is served via TFTP. Root file system of controllers is exported via NFS. VM2 contains Tango specific services: TANGODB and Sardana.

Application software

ID abstraction layer.
- There are libraries that provide access to CANbus, VME and hide implementation details.

Device Driver layer.
- This layer is set of libraries (device-space drivers) that implement interaction with particular device and facilitate reuse.

Low Level TANGO Devices.
- This layer consists of TANGODevices that wrap Device Drivers and expose them to Tango.
- They are arranged in Tango servers by underlying bus type.

High level TANGO devices.
- The highest layer: Tango Devices and user applications that control subsystems rather than particular device. Timing server will provide proper delays and start times to underlying Timer-0 and Dr20.