

# TANGO BASED SOFTWARE OF CONTROL SYSTEM OF LIA-20

A. Senchenko<sup>†1</sup>, G. Fatkin<sup>1</sup>, S. Serednyakov, P. Selivanov  
 Budker Institute of Nuclear Physics, Novosibirsk, Russia  
<sup>1</sup>also at Novosibirsk State University, Novosibirsk, Russia

## Abstract

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.

## LIA-20 PROJECT

Linear Inductor Accelerator LIA-20 is designed to produce three electron bunches with energy up to 20 MeV, current up to 2 kA and lateral size after focusing on the target less than 1 mm. It is planned to provide three consecutive bunches, with one of them divided into 9 angles. The accelerator will be used for the flash X-Ray radiography.

LIA-20 consists of the injector, 30 “short” accelerating modules (SAM) and 12 “long” ones (LAM). Injector generates beam with the energy up to 2 MeV. SAM increases the energy by 0.33 MeV and LAM increases the energy by 0.66 MeV. The total length is about 75 meters. Control units are placed along the installation. All units are based on uniform VME crate and connected via Ethernet. Structure of the control system is described in detail in [1].

## DATA RATES

All channels could be divided into following groups:

**Fast.** All measurements faster than 10 us: voltage on inductor, currents on lenses and beam position monitor.

**Slow.** This group includes measurements with duration up to several milliseconds (charging device, degaussing current).

**Timing.** These channels provide all devices with proper start pulse.

**Interlock.** These channels belong to subsystem that prohibit experiment in case of component malfunction or failure [2].

**Technological controls.** This group incorporates vacuum controls, optical system alignment, control of power supplies.

First four groups are bound to machine cycles, while the last one is continuous. Tables 1 present the summary of channels and data rates. Estimation provided for one-bunch cycle.

## SOFTWARE OVERVIEW

Experience of LIA-2 [3] shows that use of widely-used control system software could reduce costs. Taking into account that LIA-20 is more sophisticated than LIA-2 it could be crucial.

After some studies it was decided to use TANGO controls. A lot of tools (like rapid UI prototyping, Archive service, macros) are available out-of-the-box or in the form external library.

## User Software

User applications are created using Python language and PyQt/PyTango/Taurus libraries. There are two types of user application: “engineer” and “operator’s”. The first one provides access to “raw” tango device. It is designed for developing and testing purposes. The second one is high-level application that interacts with multiple tango devices and hides implementation details. Examples of operator’s applications is shown in Figure 1.

**Mimic Diagram** prototype visualizes a summary of all subsystem’s states. It is based on PyQt/QWebKit and SVG. The use of SVG allows to utilize ordinary vector editors like Inkscape to create and modify diagram.

**Time Editor** is a prototype of editing tool for timing diagrams [4]. It provides operator the ability to prepare, verify and apply timing diagrams. Current version directly applies values to the tango devices. Drawback of such approach is that it is quite difficult to track and revert changes. Management of changes is a common task and should be implemented as standalone service.

Table 1: Data Rates Estimation

Channel type	Number of channels		Data rate	
	whole system	per VME crate	whole system	per VME crate
Fast	594	22	5.7 MB/cycle	214 KB/cycle
Slow	1485	55	13.5 MB/cycle	0.5 MB/cycle
Timing	1485	55	13.5 KB/cycle	0.5 KB/cycle
Interlock	1485	55	13.5 KB/cycle	0.5 KB/cycle
Technological control	1000	~40	513 KB/min	19 KB/min
	6000	~280	19.3 MB/cycle +	3.5 MB/cycle +
			540 KB/min	19.5 KB/min

<sup>†</sup> a.i.senchenko@inp.nsk.su

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

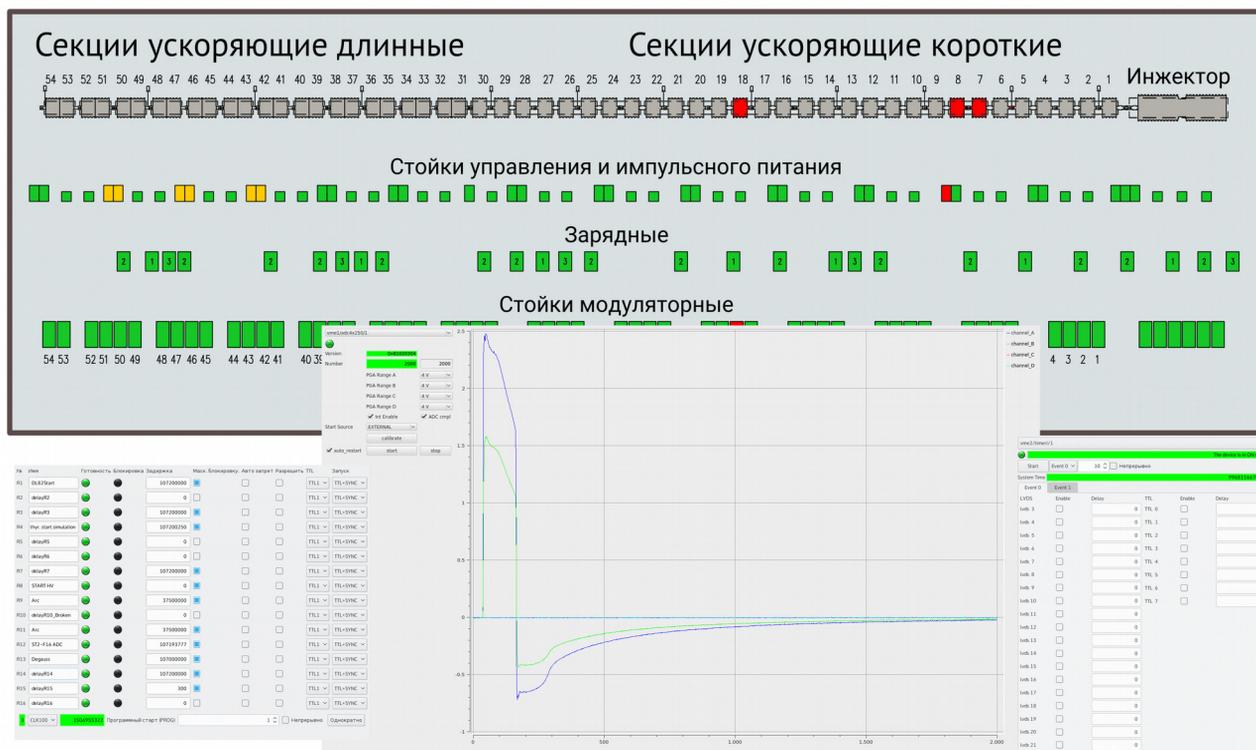


Figure 1: Examples of operator's applications.

### Hardware Access

Software of the VME crates is a media that connects hardware and software of the control system. To simplify software maintenance and reuse, it separated into three layers (see Fig.2).

**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 utilities is also included. For example: CANBus abstraction library is supplied with asynchronous (publisher-subscriber) and synchronous (request-reply) modules.

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.

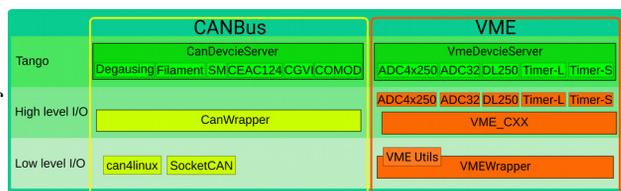


Figure 2: Software of VME crate.

To date, tango-devices (with underlying drivers) and are provided almost for all VME modules [5]:

ADC4x250 [6], ADC32VME, DL-250VME, Timer-LVME, and for majority of CANBus devices.

### High-level Control

Developed tango devices and user applications is intended to interact directly with hardware. This approach suits well for acceptance tests but significantly increase development costs with facility assembling. To solve this problem, a concept of the high-level control are introduced. The high-level control should comprises several subsystem:

**Timing** subsystem is responsible for synchronization. It includes Time Editor.

**Facility State Machine** manages states and transition sequences for entire facility and subsystems.

**Facility State and Regime Management** is intended to solve above-mentioned problem of tracking, reverting and managing changes.

**Modeling** is an essential component of the control system. It allows operator to fine-tune facility.

These subsystems in the design phase and could be implemented as a single tango device or a set of hierarchically organized tango devices.

## ARCHIVE SYSTEM

Tango controls provides an archival system called HDB++. It supports Cassandra and MySQL backends. Minimal recommended installation of Cassandra consists of 3 independent nodes. Lack of experience of Cassandra utilization and a higher hardware requirements lead to choosing of MySQL backend.

LIA-20 is a pulsed facility and significant amount of data is oscillograms. Oscillogram could be treated as an array. But this type isn't supported by MySQL out-of-the-box. To overcome this restriction, backend stores array as multiple rows. Such approach result in excess resource consumption in long term.

Another DBMS used in BINP is Postgresql. It proved to be a reliable solution and supports array type. It therefore seems reasonable to develop Postgresql backend for HDB++. Preliminary estimation gives reduction by a factor of 10.

## CONCLUSION

Almost all hardware of the control system was provided with tango devices and "engineer" user applications. To facilitate configuration of the timing system, prototype of Time Editor was developed. HDB++ was put into operation. All these components forms minimum system necessary for initial LIA-5 testing.

Further development will be directed to the implementation of high-level control and creation of Postgresql backend for HDB++.

## REFERENCES

- [1] G. Fatkin et al., "LIA-20 Control System project" TUPHA052, this conference.
- [2] A. Panov, G. Fatkin "LIA-20 Experiment protection System", TUPHA103, this conference.
- [3] Starostenko D. A. et al., "Results of operating LIA-2 in radiograph mode", Phys. of Particles and Nuclei Letters, 2014, Vol.11, No. 5, pp. 660-664
- [4] G.Fatkin, A. Senchenko "The Timing Diagram Editing and Verification Method," TUPHA087, this conference.
- [5] G. Fatkin et al., "New VME-Based Hardware for Automation in BINP", THMPL10, this conference.
- [6] E. Kotov et al., "VME Based Digitizers for Waveform Monitoring System of Linear Induction Accelerator (LIA-20)", THMPL09, this conference.