PROTOTYPE CONTROL SYSTEM FOR THE LOW ENERGY BRANCH ION BEAMLINE

M. Skobe*[1,2], Ž. Brenčič[1,3], M. Kelemen[1], K. Bučar[1,2,3], P. Pelicon[1], J. Simčič[1], A. Trost[4], A. Biasizzo[1,2]
[1]Jožef Stefan Institute, Ljubljana, Slovenia
[2]Jozef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia
[3]Faculty of Mathematics and Physics, University of Ljubljana, Ljubljana, Slovenia
[4]Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia

Abstract

At the tandem ion accelerator laboratory of the Jožef Stefan Institute (JSI) in Ljubljana, Slovenia we are developing a control system for the Low Energy Branch (LEB) ion beamline. This activity is ongoing simultaneously with the hardware construction of the ion beamline dedicated to the research with low-energy ion beams with energies up to 30 keV. In this work, we present the architecture of a prototype control system, focusing on test ion beamline, selecting the development and runtime environment, selecting hardware units, and integrating common devices using the Experimental Physics and Industrial Control System (EPICS) environment.

INTRODUCTION

The LEB ion beamline is brand-new and the first of its kind at our tandem ion accelerator laboratory, specialized for low-energy research. The core concept behind the LEB project is to expand our research scope, unite various research groups for joint experiments, and encourage contributions to the development of new analytical methods and instrumentation for low-energy ion beam research.

As these experiments are extensive and require a lot of operational and precise scientific work, we have decided to develop a control system that will help us control and monitor devices, and experiment with machine learning algorithms for automatic ion beam control [1, 2] in the future.

The LEB instrumentation is generally categorized into: a) Ion sources, b) Ion beam transport optics, and c) Control system, specialized detector systems, and hardware vacuum support. Key functionalities of the control system include the controlling of vacuum pumps, power supplies and other devices, data acquisition for readout of sensors and detector systems, analog and digital control, testing of custom-developed ion optics components and detector systems, and ensuring reliable operation for high-precision physics experiments [3].

We have chosen to develop a control system using EPICS [4] to meet our long-term control system requirements and provide the open-source tools necessary for developing a comprehensive and scalable control system [5]. Additionally, the selected contemporary development environment, hardware units, and prepared documentation will serve as a base point for future upgrades and development.

PROTOTYPE CONTROL SYSTEM ARCHITECTURE

As the duration of the LEB project spans over several years, a step-by-step development of the hardware components, the ion beamline, and a prototype control system is required.

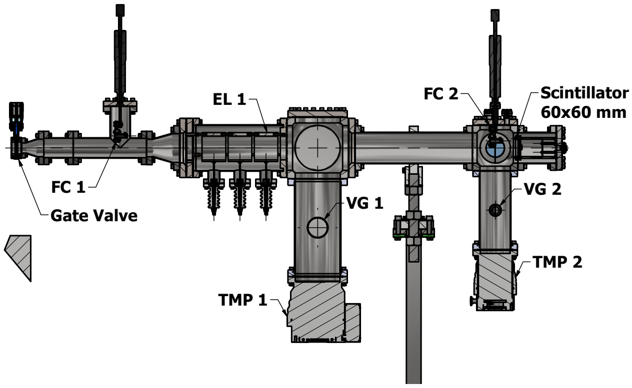

Figure 1: Test LEB ion beamline.

The current design of the LEB ion beamline, including all essential components is presented in Figure 1. The primary goal is to prepare and sustain optimal ultra-high vacuum conditions, along with testing the custom-developed Einzel lens, presented in Figure 2. Testing involves applying a wide range of high DC voltages to each of the three galvanically insulated Einzel lens electrodes. Simultaneously, changes in the ion beam's shape are monitored using a custom-developed scintillator system equipped with four silicon photomultipliers positioned at its edges. These changes are observed with the CCD camera installed at the vacuum chamber's side port. Additionally, periodic measurements of the ion current are taken using the custom-developed Faraday cup (FC). Those activities represent the key functionalities of the prototype control system:

- Control of the high-voltage (HV) power supply unit (PSU),
- Pressure readout from vacuum gauges,
- Control of the turbomolecular vacuum pump (TMP),

---

* matevz.skobe@ijs.si

ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-THPG17

- Ion current readout from FCs,
- Multi-client access,
- Simple graphical user interface (GUI).

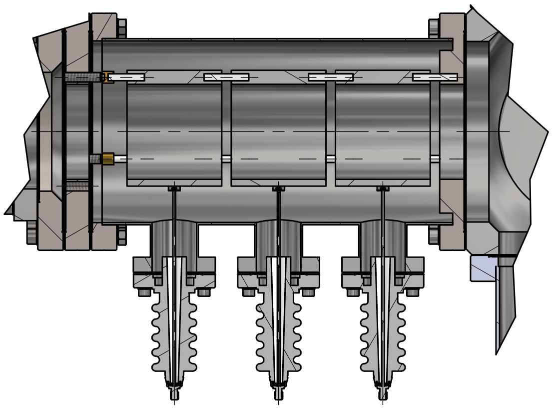

Figure 2: Cross-sectional view of the Einzel lens.

The architecture of the prototype control system is presented in Figure 3 where the primary goal was to establish the development and runtime environment, select essential components for operation, and integrate them using the EPICS environment.

Development & Runtime Environment

The primary computer used for the control system development and operation is the industrial personal computer (IPC) Beckhoff C5240 with Debian 12 installed. Additionally, we installed the Docker visualization and allocated one container for the prototype control system application, featuring EPICS Base 7.0.7 and additional modules such as StreamDevice [6], and its requirements for interfacing with hardware. For analog and digital control and interlocks, we chose to utilize a programmable logic controller (PLC) Beckhoff CX5620 for its effectiveness and real-time operation. It operates autonomously from the IPC and EPICS application and shares data using the Automation Device Specification (ADS) device support [7]. For PLC programming, we opted for TwinCAT 3 software.

Hardware Units

The 10 kV PSU is used to apply voltages to the electrodes of the custom-developed Einzel lens. The PSU is controlled via analog voltage signals ranging from 0 to 10 V, enabling voltage control as well as voltage and electric current readout. For these functionalities, we employed digital-to-analog (DAC) and analog-to-digital (ADC) I/O modules with 16-bit signal resolution. As they communicate through EtherCAT communication, they require a master computer, in our case, the PLC Beckhoff CX5620. Due to the operation under HV conditions, I/O modules are connected to the PLC through plastic optic fiber and couplers to prevent costly hardware failures and protect end-users in the case of voltage breakdowns.

For pressure measurements, we utilize two cold cathode vacuum gauges IKR 251, connected to a 6-channel control unit MaxiGauge TPG 366 from the same supplier as the vacuum gauges. Since this control unit offers various connectivity options, we opted for the Ethernet communication interface.

TMP HiPace 300 is operated through an electronic drive unit TC 110 mounted on the pump's body. The electronic drive unit offers RS-485 communication. For connection and communication purposes we use an industrial module Moxa NPort 5650-8 with 8 input serial ports. Additionally, the module's serial ports support RS-232, RS-422, and RS-485 communication protocols, which could be beneficial for future device needs. The module is directly connected to the Ethernet switch.

For ion current measurements, we utilized two FCs on the test ion beamline. To connect them within the control system, we selected a commercial 4-input channels picoammeter Libera Current Meter, capable of low current measurements from a few nA to 2mA. The instrument itself runs a Linux operating system with EPICS Base 3.14 installed, making the data directly accessible through the standard EPICS network protocols.

In addition to the key hardware units depicted in Figure 3, another component is an Ethernet switch connecting networked components to the main IPC. It manages the assignment of IP addresses for each device and provides network address translation and firewall for network security. We selected the Mikrotik Ethernet switch CRS326-24G-2S+RM, which features 24 standard gigabit RJ45 ports and 2 Small Form-factor Pluggable ports, allowing for both copper and fiber optic cable connections and provides high-level of configurability.

EPICS IOC

The EPICS input-output controller (IOC) is the most important component of the EPICS application, acting as a server running *iocCore* software on the IPC. The EPICS Base software tool provides options for creating EPICS IOCs, and additional drivers and modules may be required for more complex functionalities. The LEB IOC comprises:

- **EPICS database**, a live database loaded and located within IOC's dynamic storage and available as long as the IOC is running. Inside the EPICS database process variables (PVs) and their metadata were defined in the form of records [8], where each device type has its own database file.
- **Device support**, for interfacing with the hardware. It consists of the protocol files for transferring data between the records and control units, and hardware startup scripts for configuring the hardware.
- **PV access server** for accessing PVs from various EPICS clients using the client-server model.

The values of PVs are exchanged through the control system network, and since EPICS does not prevent having

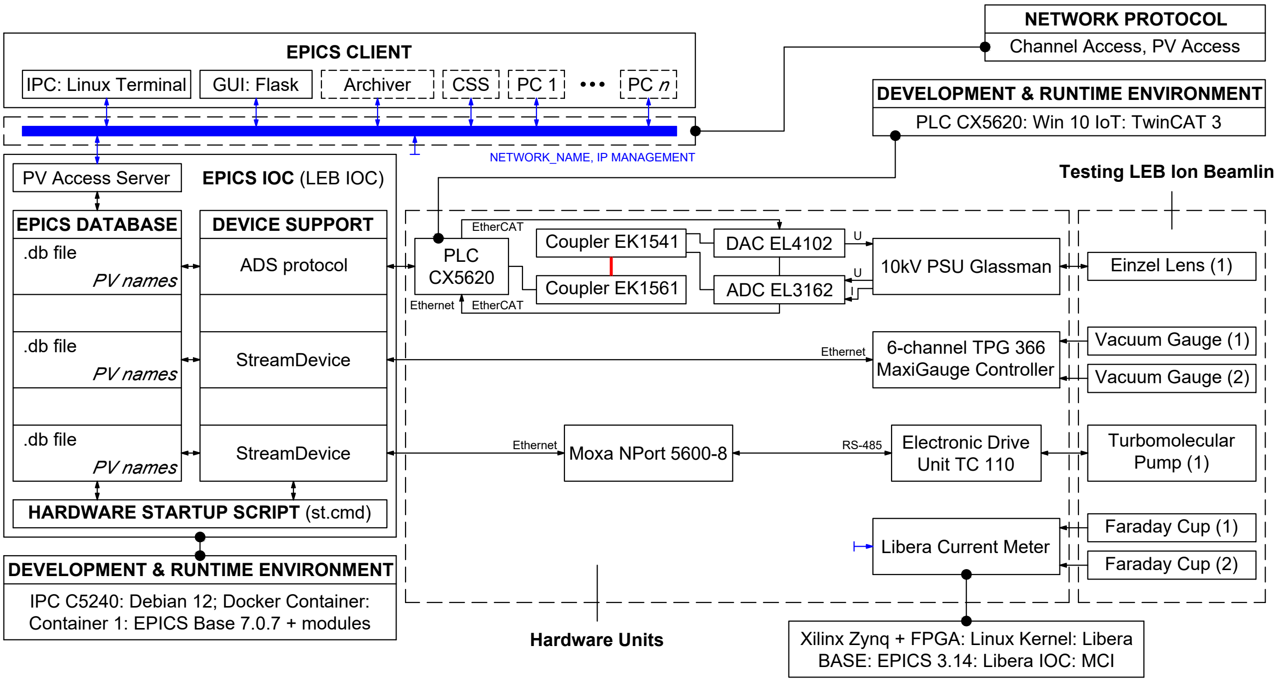

Figure 3: Block diagram of the prototype control system architecture.

different PVs with the same name, PV names were chosen wisely through a prepared rule called naming convention to prevent undesired control system behavior.

EPICS Clients

EPICS supports two network protocols, Channel Access (CA) and PV Access (PVA), to transfer data over the network. CA is an original EPICS protocol for communication, fully supported for both EPICS 3.x and EPICS 7.x versions. It facilitates communication between EPICS clients and servers. The EPICS IOC hosts the CA Server which holds, updates, and provides clients access to specific PVs stored on servers through the CA Client software. For our control system, we mainly use the PVA protocol while CA protocol is only used for the Libera Current Meter.

Our prototype control system application runs on the IPC. The initial EPICS clients to access the control system data include the IPC's Linux terminal, and a simple GUI based on the Flask framework. This GUI interface allows users to control and monitor live data from the test ion beamline, stored in EPICS PVs, and provides current time-dependency graphs of pressure, and ion current. Moreover, the connection to the control system's network from various PCs is both available and required for remote operation.

Later, we will also employ the Archiver for data storage, Control System Studio (CSS) for a more professional control system GUI, and the Alarm Handler for managing alarm conditions.

CONCLUSION

Once the control system is fully developed in combination with components following industrial standards, it will serve as a robust tool for testing the in-house developed instrumentation, training the students and an excellent showcase for controlling other physics experiments running in our laboratory.

ACKNOWLEDGEMENTS

Work at Jožef Stefan Institute was supported by the Slovenian Research Agency (ARIS) grants P1-0102, P1-0112, I0-0005, as well as by the Slovenian company Cosylab, which provided consulting services for the development of the prototype control system architecture.

REFERENCES

[1] P. J. Antsaklis et al, ”An introduction to autonomous control systems”, IEEE Control Systems Magazine, vol. 11, no. 4, pp. 5-13 (1991)

[2] V. Kain et al, ”Test of Machine Learning at the CERN LINAC4”, 61st ICFA ABDW (2021)

[3] Z. Brencic *et al.*, “Development of Low Energy Branch at Micro Analytical Centre, Ljubljana, Slovenia”, in *Proc. IPAC'23*, Venice, Italy, May 2023, pp. 5131–5133. `doi:10.18429/JACoW-IPAC2023-THPM113`

[4] S. Li, Y. Wang, Y. Chen, and F. Wang, “Development of the EPICS-based monitoring and control system for EAST fast control power system,”in 2016 IEEE-NPSS Real Time Conference (RT), Jun. 2016. `doi:10.1109/rtc.2016.7543180`.

[5] K. Ogata, “Modern Control Engineering”, 5th Edition, Prentice Hall, New Jersey (2002)

[6] SLAC National Accelerator Laboratory. StreamDevice. Retrieved from `https://www.slac.stanford.edu/grp/cd/soft/epics/site/streamdevice/`, Accessed on 2024-05-10

[7] Cosylab. EPICS Documentation: ADS Driver. Retrieved from `https://epics.cosylab.com/documentation/adsDriver/`, Accessed on 2024-05-10

[8] P. Stanley, “EPICS: Record Reference Manual”, EPICS Release 3.13, The University of California, The University of Chicago (1995)