Evolution of Control System and PLC Integration at the European XFEL

Ayaz Samadli, Tobias Freyermuth, Patrick Gessler, Gabriele Giovanetti, Steffen Hauf, David Hickin, Navid Mashayekh, Alessandro Silenzi
European XFEL GmbH, Schenefeld, Germany

Introduction

As one of the world's leading light sources, The European X-ray Free Electron Laser facility is opening up completely new research opportunities for scientists and industrial users by generating **ultrashort** X-ray flashes - **27 000** times per second and with a peak brilliance of 5×10^{33} photons / s / mm2 / mrad2 / 0.1% bandwidth. Unique characteristics of the facility enable the researchers to study tiny structures, ultrafast processes, extreme states, and small objects.
Operating such a complex facility without a robust control system is impossible. At the European XFEL, Karabo is used: a pluggable, distributed control system that offers rapid control feedback to meet the complex requirements of the facility [1]. The main interface with the existing hardware infrastructure are Programmable Logic Controllers (PLC) which use Beckhoff technology. The primary objective of this project is to evolve the communication between Karabo and the Beckhoff PLCs.

Current communication between Karabo and PLC

- The communication between Karabo and PLC currently uses an in-house developed protocol over **TCP/IP**.
- **The same port** for operational-related communications and self-description (the description of all available devices sent by PLC) is used.
- Both the self-description and operational-related data (data updates from hardware) are in **binary format**.

In the current production environment, there are approximately::
- ~ 15,000 Beckhoff devices
- ~ 1,000,000 Beckhoff properties
- ~ 39 different Beckhoff device types

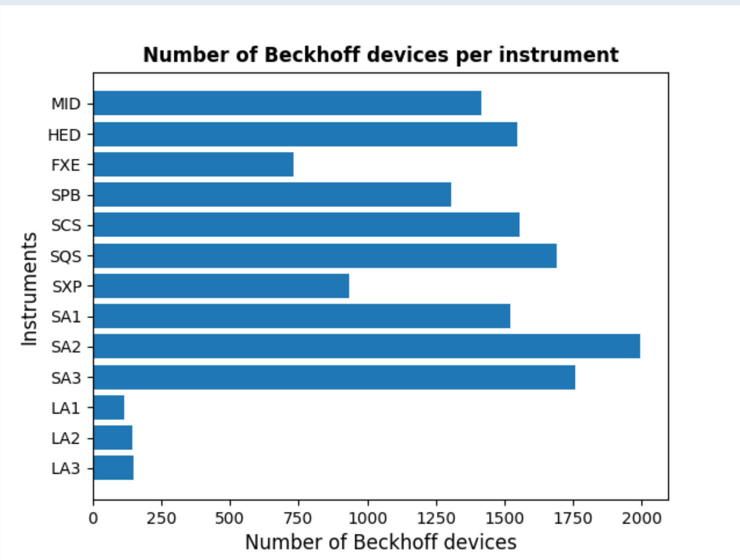

Figure 1. Number of Beckhoff devices per instrument in the infrastructure

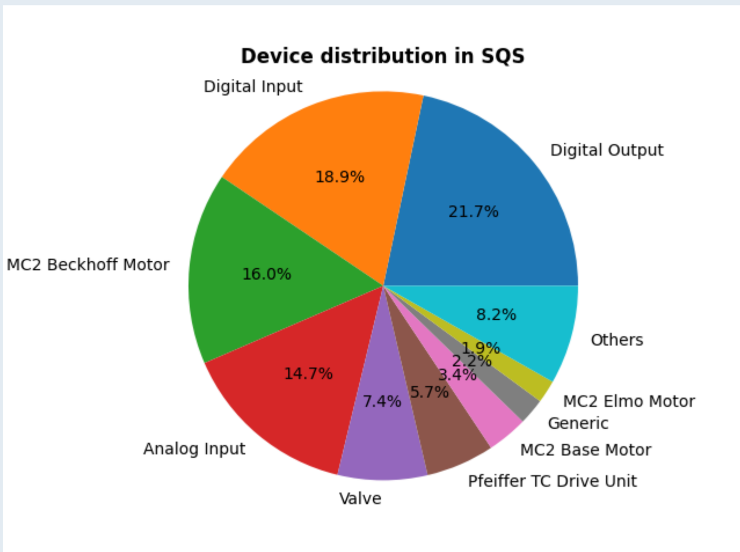

Figure 2. Distribution of the Beckhoff devices in SQS instrument

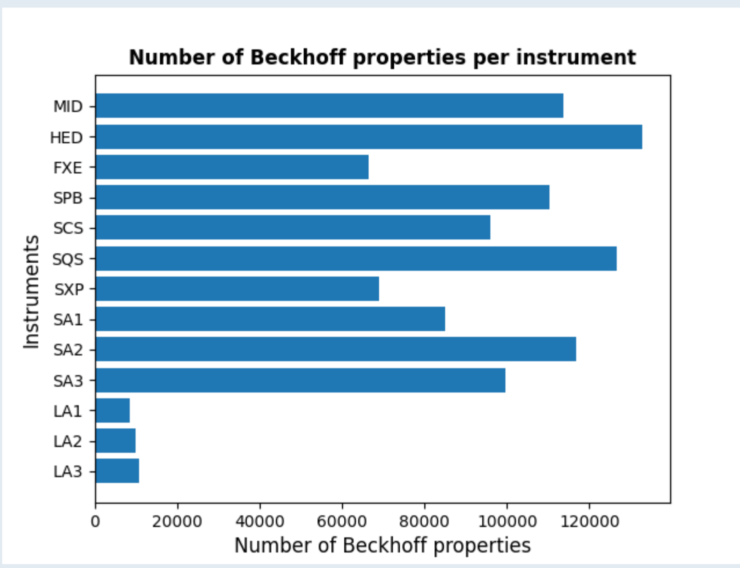

Figure 3. Total number of Beckhoff properties per instrument in the infrastructure

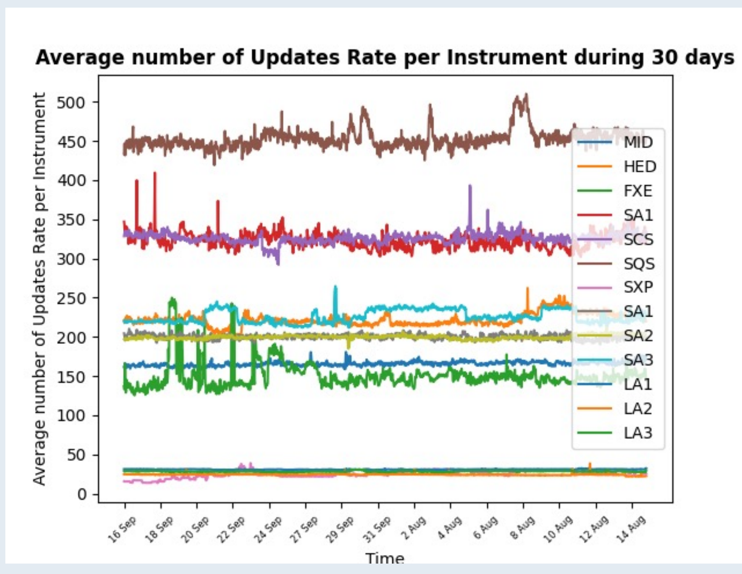

Figure 4. Average number of Beckhoff update rates per instrument during 30 days

Issues in the current implementation

- **Limitation in maintainability:**
  - **Python's Increasing Popularity:** **A Python implementation** of the PLC integration in Karabo can leverage the growing Python usage throughout the group and facility.
  - **Challenges in Maintaining Legacy Code:** The existing C++ code is dense and has **multilevel model-view-presenter (MVP)** design, which makes maintenance and addition of new features tedious if they weren't foreseen in the original design.
- **Cumbersome C++ implementation:**
  - **The presence of Verbose Code:** The MVP design results in a necessity of modifying many classes, even for simple additions. Compared to a more stream-lined design, this could be considered **boilerplate**.
  - **Challenge in Bug Identification:** Debugging and bug detection are made more challenging due to the **non-negligible time** required for **compilation**, potentially leading to inefficiencies in the development process.
  - **Less straightforward asynchronous callbacks:** The **boost::asio** callbacks that are used in C++ can be **less straightforward** and intuitive than **Python's asyncio**.
  Incomplete Self-Description: The self-description lacked crucial components as:
  - **Error Codes:** Absence of error codes along with their explanatory details.
  - **State Machine:** The PLC state machine and the Software state machine are not coupled, Current protocol does not include any information about state machine.
  - **Value Ranges and Options:** Critical information such as minimum and maximum value ranges and available configuration options were missing.
- **Mixing of static and dynamic data:** Combining self-description (static) and operational communication data (dynamic) in a **single protocol** and **data stream** make the protocol complex. Sending the self-description data upon connection may cause operational data exchange delays due to **increased traffic**.

Python Asyncio based event-driven proposal

- **Comprehensive Self-Description:** The new implementation aims to enhance user interaction by providing an extensive self-description delivered through a **web service** in **JSON** format [2], effectively separating static data from dynamic condition updates.
- **Python Asyncio Approach:** The implementation will adopt a Python Asyncio-based approach for the Karabo device, characterized by **event-driven** behavior.
- **Responsibilities:** The new implementation will be responsible for **message encoding/decoding**, facilitating **bidirectional communication** with the PLC, and establishing connections with pertinent software devices within the Karabo environment.
- **Incorporating Lessons Learned:** It will incorporate insights from **past design decisions** as mentioned above, with a focus on supporting future updates and bolstering **developer productivity**.
- **Testability:** The implementation is designed with testability in mind, using **PyTest** for both unit and integration tests. The current test coverage is **92 percent**.

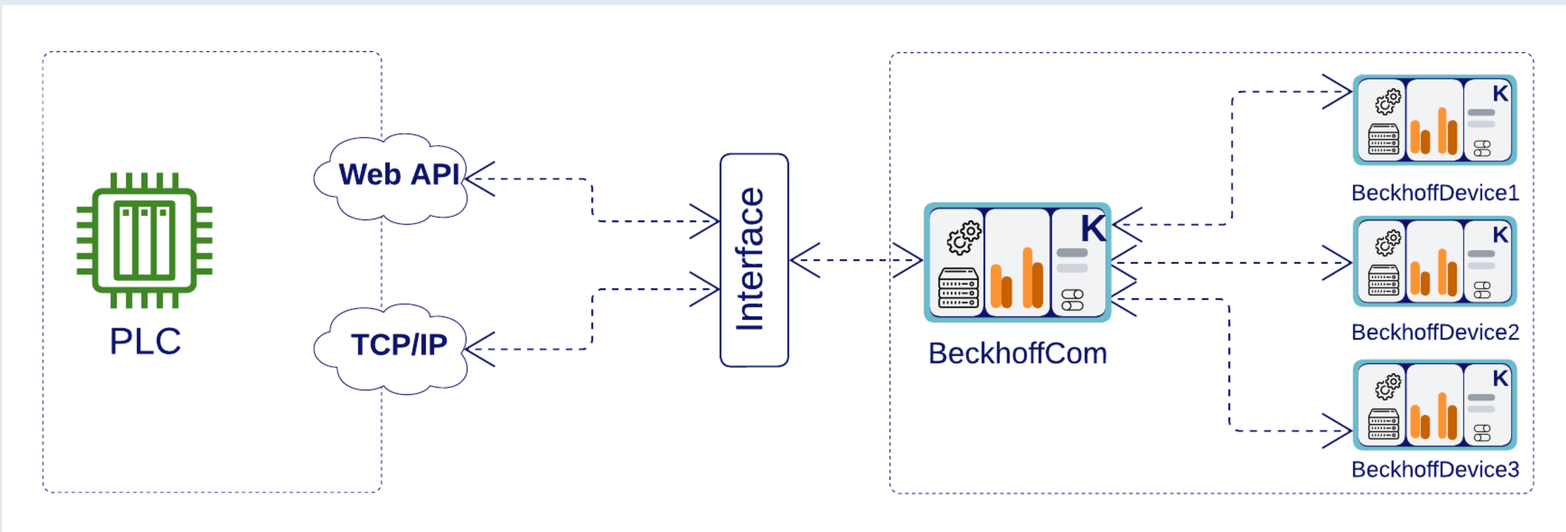

Figure 5. The Karabo-agnostic Interface connects asynchronously to PLC/Web service, enabling communication between BeckhoffCom (leading) and BeckhoffDevices (subordinate) via event-driven signals.

Table 1. The description of properties in JSON format

Name	Value	Description
key	triggerDuration	Keyname of property
alias	0x5011	32-bit hexadecimal PLC property code
displayedName	Trigger Duration	Property's displayed name
description	Trigger duration	Property's description
dataType	Int16	Property's data type
unitSymbol	SECOND	Property's unit
metricPrefixSymbol	MILLI	Property's metric prefix
allowedStates	[ON, PROCESSING]	Allowed property state changes
requiredAccessLevel	EXPERT	Property's access level for visibility
accessMode	RECONFIGURABLE	Modifiability of property value

Table 2. The description of commands in JSON format

Name	Value	Description
key	sendAll	Keyname of command
alias	0x80100000	32-bit hexadecimal PLC command code
displayedName	Send All	Command's displayed name
description	Send all available values	Command's description
allowedStates	[ON]	Allowed command execution states

Outlook

The forthcoming implementation is expected to be deployed in production in 2025 during the long shutdown of the infrastructure.

References

[1] Hauf, et al. The Karabo distributed control system J.Sync.Rad.26.5(2019): 1448ff
[2] T. Freyermuth et al. Progression Towards Adaptability in the PLC Library at the EuXFEL, PCaPAC'22, pp. 102-106.

European XFEL GmbH, Ayaz Samadli, Holzkoppel 4, 22869 Schenefeld, Germany, Phone +49 40 8998-6570, ayaz.samadli@xfel.eu
www.xfel.eu

ENLIGHTENING SCIENCE