ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THMPL06

CAMERAS IN ELI BEAMLINES: A STANDARDIZED APPROACH

B. Plötzeneder†, O. Janda, V. Gaman, P. Pivonka, P. Bastl
ELI Beamlines, Prague, Czech Republic

Abstract

The ELI Beamlines facility is a Petawatt laser facility in the final construction and commissioning phase in Prague, Czech Republic. The central control system connects and controls more than 40 complex subsystems (lasers, beam transport, beamlines, experiments, facility systems, safety systems) with hundreds of cameras.

This paper describes the approaches and solutions used in ELI Beamlines to manage the selection, integration and maintenance of cameras by providing a comprehensive set of standards. Hardware interface standards guarantee ad-hoc software integration (using vendor-independent drivers), for commonly used models, auxiliary hardware (triggering: optical/TTL, power supplies) is available. Information on key parameters (vacuum compatibility, noise levels) is collected. By using a strict model-based approach and a component-based design, all cameras and 2D-detectors can be controlled with the same C++-API.

INTRODUCTION

ELI Beamlines [1] is an emerging high-energy, high-repetition rate laser facility located in Prague, Czech Republic. Four laser beamlines (ranging from the in-house developed L1 with <20fs pulses exceeding 100mJ at 1kHz based on DPSS technology to the 10PW-L4, developed by National Energetics) will supply six experimental halls which provide various secondary sources to users. Facility commissioning, and installation work of lasers and experiments is progressing, and first user experiments are expected in 2018.

The central control system group connects, supervises and controls all technical installations used for the operation of this facility, which are more than 40 complex subsystems with hundreds of cameras.

In 2017, we worked with ca 40 stakeholders on the topic of cameras: This means supporting selection and procurement, helping with operation, software / hardware development and integration into the central control system. While 90% of the requests can be fulfilled using standard off-the-shelf machine vision cameras, we also work with highly specific detectors (for example: photon counters, x-ray detectors, wavefront sensors,..) and have even developed custom cameras.

This paper describes the approaches we chose for supporting our users, our process of camera selection and interface standards, and the soft- and hardware we developed to support them.

† birgit.ploetzeneder@eli-beams.eu

SUPPORT APPROACH

Integrating a single camera into a control system is easy – vendor SDKs are freely available and can be wrapped into any middleware without problem. Almost all industrial cameras are supported by Matlab and LabVIEW, allowing easy access for users.

The key challenge for a control system team is scaling up and supporting potentially hundreds of different cameras. (in 2014, we counted 117 different models in ELI, a number that is probably much higher now). Individual solutions are simply not feasible any more.

In such situations, industrial SCADA teams often enforce **standard component catalogues** and restrict device choice to a few well-known models. Any deviation needs a good justification, explicit permission and a maintenance plan (spare parts, expertise).

We would have preferred this approach, because it reduces complexity, maintenance cost and unit-price (due to high-volume procurement); however, we did not succeed with introducing it in ELI: First, there was low acceptance from users, who are not used to such restrictions; and second, component prices in machine vision are currently falling rapidly, making yesterdays "top-notch" model obsolete and overpriced.

Therefore we settled for different, three-tier approach:

1. **Strictly enforced interface standards**
2. **Support with camera selection**, with guidance towards common models that can be borrowed for testing, prototyping and bridging delivery times.
3. **Complete and standardized vendor-independent software and hardware solutions**

CAMERA SELECTION

In ELI, any camera that fulfils the interface standard described below is called a **standard camera**. These cameras are known to work within our control system environment, and users can purchase them without previous consultation.

Any other camera is called a **special camera**, typically supplied by commercial-research companies or university spinoffs, performing niche measurement applications. These companies either use outdated/cheap cameras models, or developed their own interface for more complex detectors. Often there are no alternative suppliers, and we deal with these systems on a case-by-case basis, and try to wrap their acquisition solutions into our standard APIs.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THMPL06

Interface Standard

ELIs interface standard can be summarized in one sentence: **We allow only USB3, GigE, 10GigE and CameraLink cameras.**

These interfaces cover the full spectrum of requirements in terms of acquisition speed, real-time capability and usability – and we as control systems team can cover them with a single driver solution that requires no further adaption for new models (*Pleoras eBUS SDK*[*] *[2]*). USB3 and GigE do not require any special hardware; for 10GigE we use the *Myricom 10G-PCIE-8B-S* network card and are currently testing the *Komodo 10GigE Frame Grabber;* CameraLink is converted to 10GigE using Pleoras *iPORT CL-Ten Full* external framegrabber.

We explicitly forbid FireWire and USB2.0 for standard use-cases. Some users request exceptions, mainly because of price, availability or previous positive experiences with the model. So far we have not encountered a technical reason why those interfaces would be necessary, so we help with selection of a replacement model.

The lack of support in the *eBUS SDK* is a driving factor for this decision, but there other reasons that speak against those interfaces:

- FireWire is outdated and requires additional interface cards without providing an advantage
- USB2.0 is not standardized "on the wire", which means we would need to provide in the best case DirectShow/Video4Linux plugins and in the worst, drivers for every single vendor.

[*] Commercial run-time licensed SDK

Table 1: Users Choose Sensor Parameters According to their Image / Stream Quality Requirements – This Affects the Amount of Produced Data (and Thereby Sometimes Interfaces Choices)

Sensor parameter	User Concerns	Support Concerns
Spatial resolution	Image size	Amount of data
Framerate	Acquisition speed	Amount of data, Highspeed-Interface/ Framegrabber needed?
Parameters related to input spectrum	Application-specific: Mono/Color, without glass cover, NIR,..	-
Image depth	Sensitivity / Noise	Amount of data (12/14bit often transmitted as 16bit)
Sensor (Pixel) size	Diffraction limits Sensitivity (low-light applications)	-
Area / Line	Application-specific	-
CCD vs CMOS	Linearity / Sensitivity vs Cost / Speed	Users often have outdated information: CMOS recently improved a lot

Table 2: Users Just Want to Operate Cameras in a Chosen Environment without much Effort – We Choose Interfaces and Power-Supplies in Accordance to those Needs

Environment	User Concerns	Support Concerns
Use in laboratories	Plug-and-play possible No extra power-supply	USB3: Easier for users, directly powered, incompatible with older laptops GigE: Network configuration difficult for users; PoE reduces cabling
Integration in central control system	-	USB3: Limited cable length, expensive GigE + switches with SFP+/PoE: Ideal
Use in vacuum	Overheating Application-specific: Outgassing	Temperature: Use Peltier elements and USB3 (lower voltage than GigE/PoE) Outgassing: build extra chambers Feedthrough: buy (signal quality/effort)
Multi-camera environments	Ease-of-use	USB3: Still driver / firmware problems GigE: Network configuration difficult for users, network load management

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THMPL06

Selection of Standard Cameras

When helping users to select cameras, we typically discuss image/stream parameters and environment (Table 1 and 2) with them, and choose sensor, interface and powering options according to their requirements.

Often, multiple vendors can provide "the same" cameras. The subtle differences lie in cost, noise level, temperature behaviour, compactness and – which can be decisive –software implementation: A Basler camera may be more expensive than a Smartek camera, but it offers more advanced Ethernet parameters (useful in multi-camera environments) or allow you to set a picture-by-picture exposure time using the trigger length.

Multi-camera environments are best considered already during procurement. In our experience, the USB3 interface is not yet suitable for these applications: Drivers and firmware implementations are immature (especially for Linux) and we still see problems with identification and power management.

We typically use GigE cameras with local switches (PoE: less cabling), with SFP+ / optical fibre transmission (signal integrity) to our data acquisition systems. We use dedicated Intel PRO/1000 network cards (availability of "high-performance" filter drivers to decrease CPU load). Limiting factors are copper bandwidth (realistic in non-EMP environments: 700MB/s), CPU loads (rule of thumb: no more than 6 cameras per system) and temporary bandwidth spikes especially in triggered systems.

Network load has to be proactively managed using Ethernet parameters (availability depending on vendor: interpacket delays, desired peak bandwidth,..); otherwise cascades of resent packets lead to data loss.

Tendering can be challenging, and it is best to involve distributors early on. We recommend specifying the above parameters, and requesting cables, power-supplies and potentially mounting plates directly within the tenders. This does not increase cost significantly, but allows users to work with their cameras immediately after receiving them.

SYSTEMS PROVIDED

Software

Most facilities use a generic software interface for cameras, such as LIMA [3] for TANGO, which implements an abstract layer and generic functionality on top of vendor SDKs (LIMA supports ca twenty different camera models developed and supported by multiple collaborating institutions). We believe this abstraction should happen one layer lower – on driver level. By enforcing interfaces that already standardize communication between camera and host "on the wire", we are able to use a generic and vendor-independent SDK (Pleoras eBUS – there are alternatives, but Pleora is the main supplier of IP cores for machine vision interfaces on the European market, which promises best compatibility). After going through some initial iterations as part of a careful software engineering process, we didn't need to touch the driver implementations since 2015.

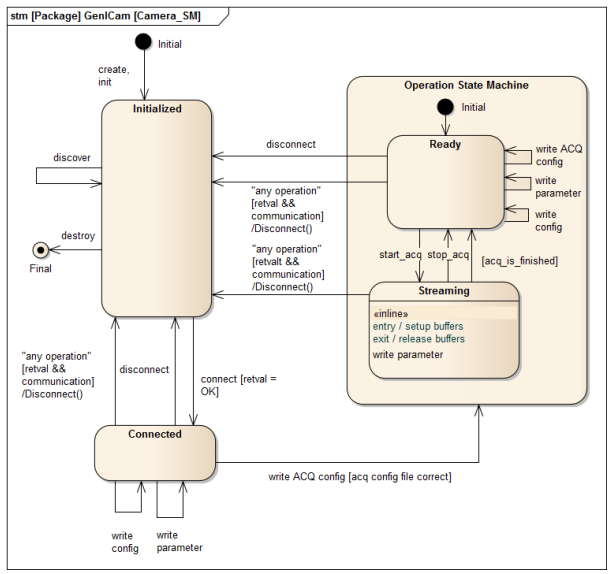

Figure 1: Camera state machine.

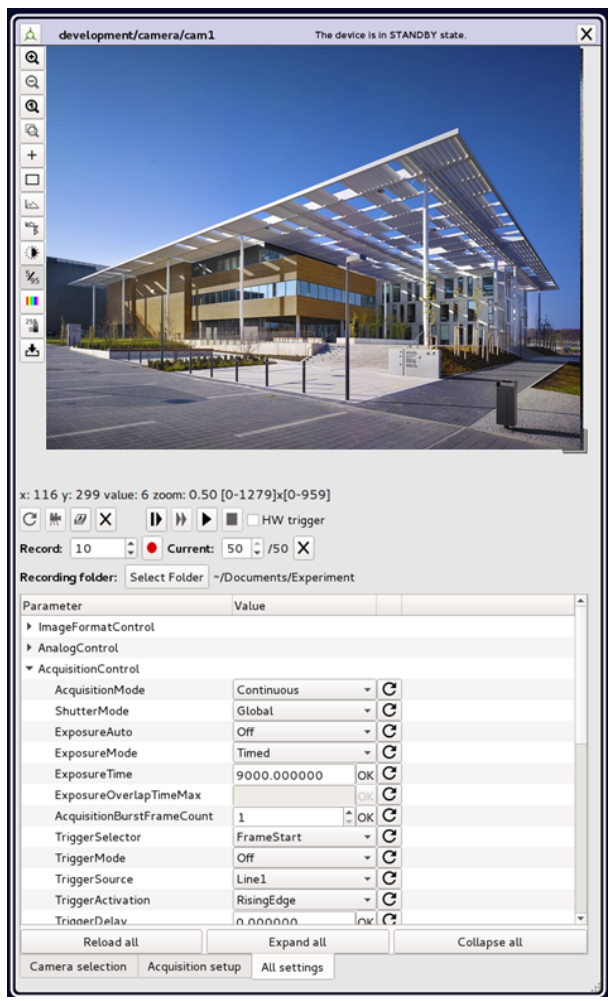

Figure 2: Camera GUI.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THMPL06

Figure 1 shows the state machine that is the core of our camera – APIs, which are implemented in C++ and exposed over many interfaces, including Matlab / Python / TANGO for which a specific GUI was developed (shown in Figure 2). As discussed in [4], we use a strict model-based software development process to produce high-quality, reusable components - figure 3 gives an overview over the camera class (and also shows our abstract device / component classes and related software).

These models implement the behaviour of all standard cameras, special cameras only extend it.

Our scientists were closely involved in the development– as stakeholders in the early modelling phases and design reviews, and as testers / domain experts in later usability tests. For example, based on their feedback, we added a set of commonly requested tools to the GUI (histogram, zoom functions, cross-sections, centroid calculations, recording functions,..)

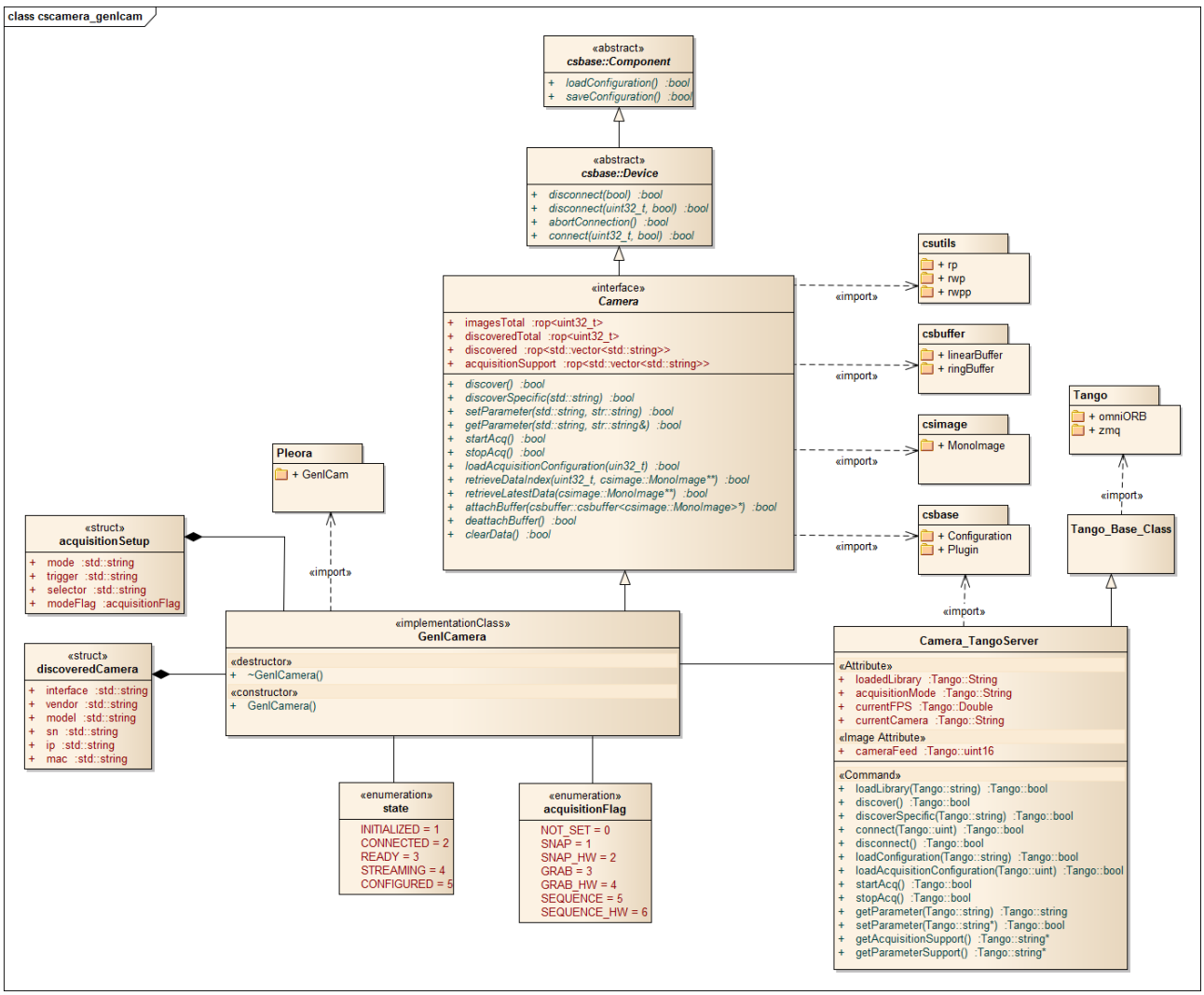

Figure 3: Class overview / camera component.

Hardware

Finally, we provide the so-called „Breakout Box“ (Figure 4), which standardized power-supply and triggering. It can be connected to any 24 V source, and accepts both TTL and optical triggers, adapting them to different camera models (currently supporting 5 different). A version supporting our White Rabbit Timing system is in preparation.

Figure 4: Breakout box.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THMPL06

User Support

In addition to providing hardware, software and procurement support, we keep a rotating stock of ca 20 standard cameras (plus cables, switches, and PoE injectors) that users can borrow and we use as a maintenance reserve and resource for testing and development.

This concept has proven to be highly successful: Users can immediately verify that a camera delivers the desired image quality, physically assess thermal and mechanical properties (mounting, physical dimensions, and cabling), and bridge lead-times. Almost all users purchase exactly the same camera models they test, allowing us to subtly direct them towards commonly used standard models.

We collect data on requirements (most commonly seen: 10-20fps, mono, 12bit, 2MPixel) and camera behaviour (noise, vacuum, software irregularities), and have standard solutions for cooling and cabling.

Users are also encouraged to integrate their cameras using our optical laboratory (Figure 5). following the test-bed principle introduced in [5].

At the moment, we record ca 5 support incidents per week, most commonly regarding camera selection and configuration. We have produced educational material / troubleshooting guides for the most commonly seen problems(Figure 5) .

REFERENCES

[1] ELI Beamlines, https://www.eli-beams.eu

[2] Pleora, http://www.pleora.com/our-products/ebus-sdk

[3] Homs, A., *et al.,* "LIMA: A generic library for high throughput image acquisition.", in *Proc. ICALEPCS'11*, Grenoble, France, 2011.

[4] P. Bastl, O. Janda, A. Kruchenko, P. Pivonka, B. Plötzeneder, S. Saldulkar, J. Trdlicka, "Control System Software Environment in ELI Beamlines " in *Proc. ICALEPCS'2017*, Barcelona, Spain, paper THPHA171, this conference.

[5] P. Bastl, O. Janda, A. Kruchenko, P. Pivonka, B. Plötzeneder, S. Saldulkar, J. Trdlicka, "Control System Software Environment in ELI Beamlines " in *Proc. ICALEPCS'2017*, Barcelona, Spain, paper THPHA171, this conference.

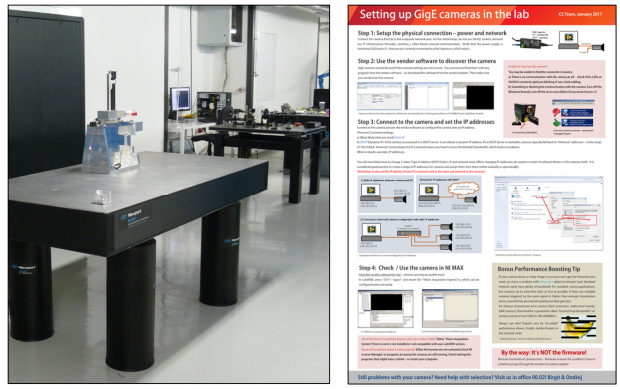

Figure 5 Left: Optical lab for integration tests, Right: educational material for camera support topics, available from birgit.ploetzeneder@eli-beams.eu.

OUTLOOK

The camera hardware interface standard and software implementation we currently provide serves our needs very well and no larger modifications are expected in the near future. However, we are planning to develop two additional tools: First, a generic IP configuration tool for GigE (this function is not provided by the eBUS SDK), second, a machine-learning based tool for automatic network load management by modifying Ethernet parameters in small-to-medium scale multicamera networks.