ALBA Controls GUIs

Taurus Performance Improvements

Oriol Vallcorba, Miquel Navarro, José A. Ramos, Emilio Morales, Sergi Rubio & Zbigniew Reszela

on behalf of

ALBA Controls Section

Controls and Acquisition GUI Strategies @ NOBUGS 2024
23-27 September, ESRF and ILL, Grenoble

Outline

- ALBA Controls GUIs: Current technologies
- GUI strategy and next steps
  - Taurus
    - Performance Optimization
    - Upgrade to Qt6
    - Trainings - documentation
  - Other technologies (web, cameras/detectors)
- Questions to the community

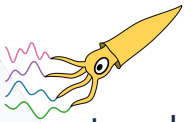
Outline

- **ALBA Controls GUIs: Current technologies**
- GUI strategy and next steps
  - Taurus
    - Performance Optimization
    - Upgrade to Qt6
    - Trainings - documentation
  - Other technologies (web, cameras/detectors)
- Questions to the community

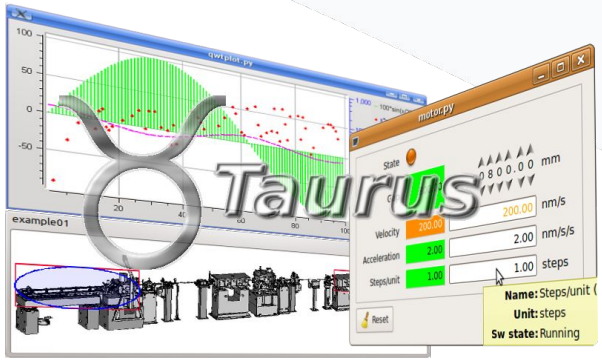
GUIs at ALBA

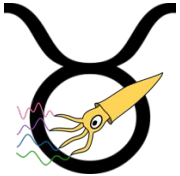
Main technology behind GUIs today at ALBA is

Qt/PyQt5



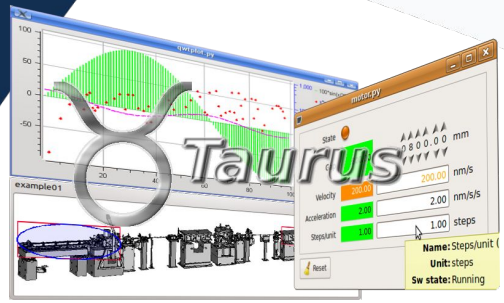
pyqtgraph





*"**Taurus** is a **python** framework for control and data acquisition **CLIs** and **GUIs** in scientific/industrial environments. It supports multiple control systems or data sources: Tango, EPICS, … New control system libraries can be integrated through plugins."*

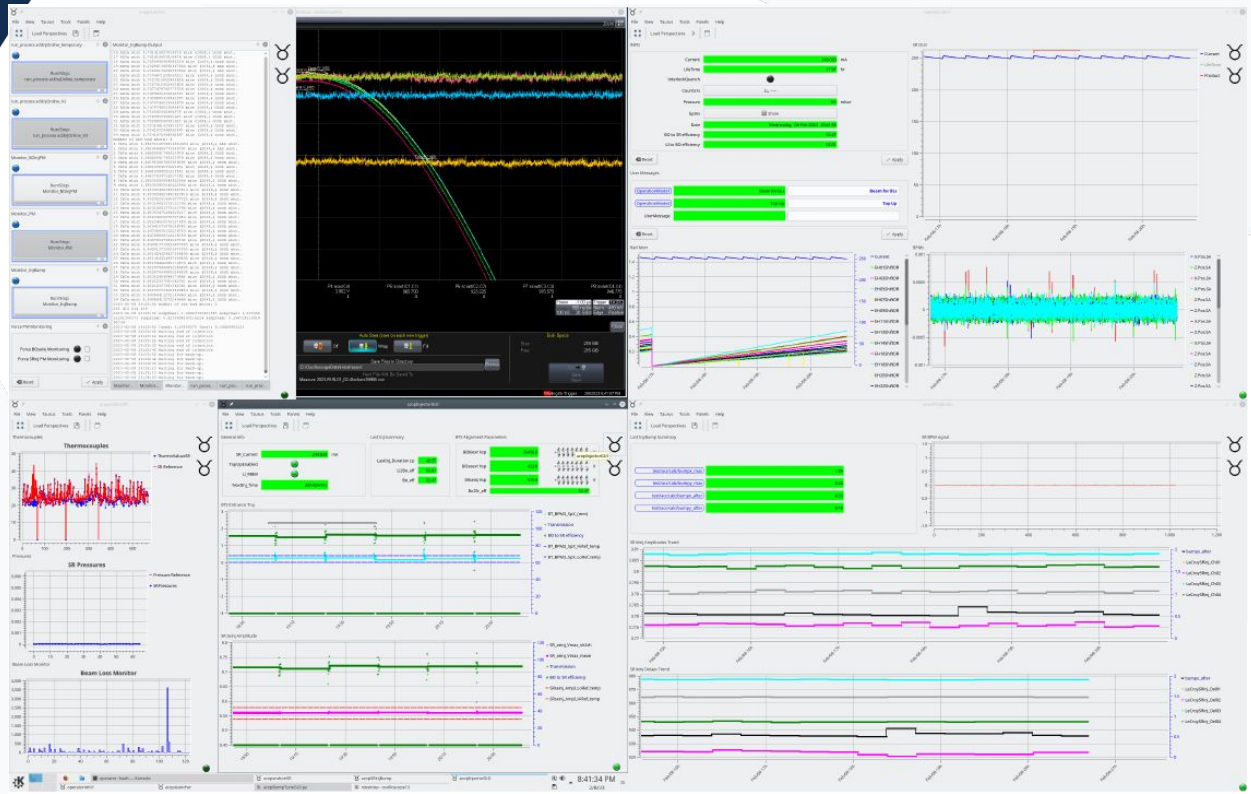
GUIs at ALBA



- Community-driven, free/Open Source and actively developed.
- @ALBA, more than 10 years of operation with Taurus. Extensively used.
- Modular and extensible with plugins and widgets (e.g. archiving using pyhdbpp library).
- Zero-code solutions (forms, trends/plots, GUIs). Taurus designer.
- Programmatic development (PyQt).
- Draggable attributes between applications.
- Synoptics (JDraw panel, svgsynoptics library).

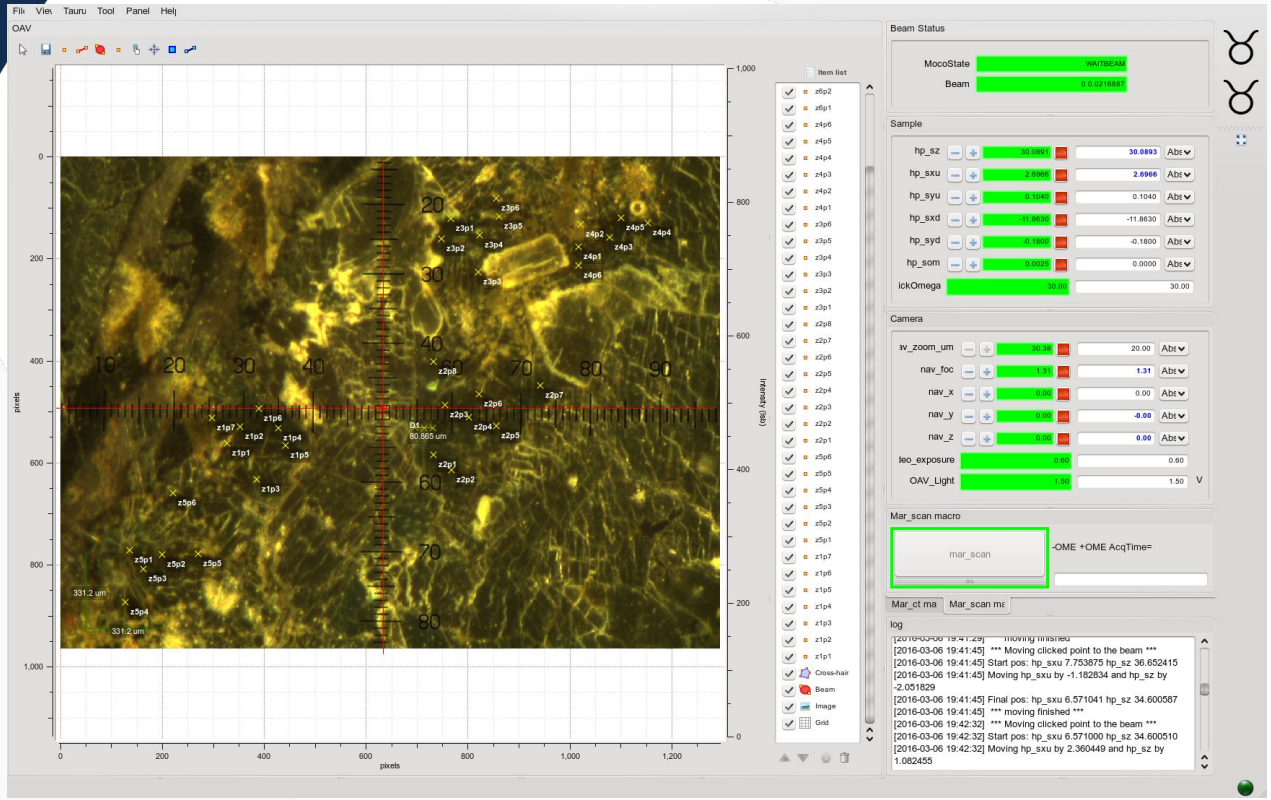
GUIs at ALBA

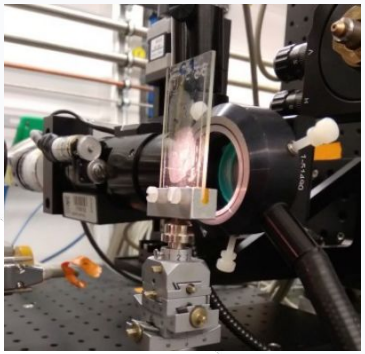
Control Room



- Device control
- Device status
- Device config
- Plotting area
- …

Experiment control



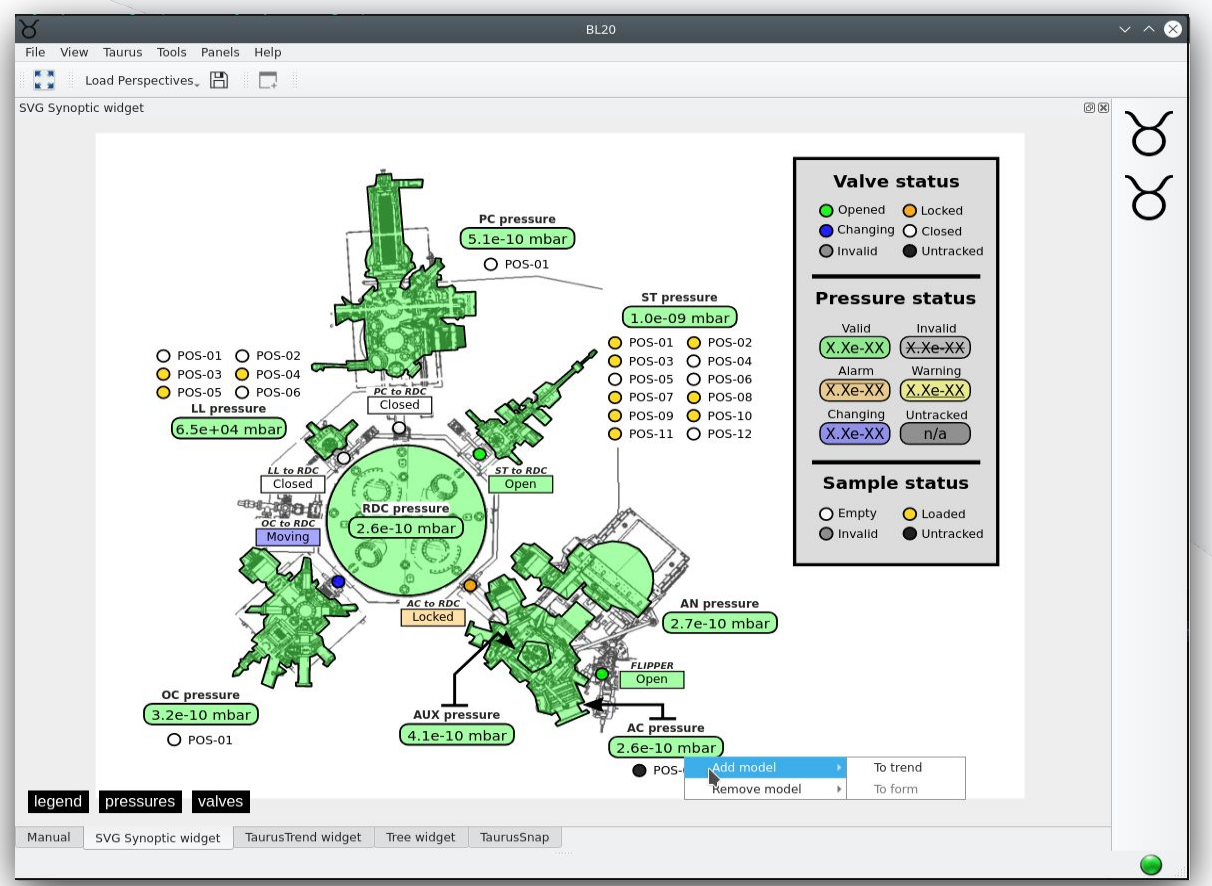


- LImA cameras
- Sardana widgets

GUIs at ALBA

svgsynoptic

MAX IV LABORATORY

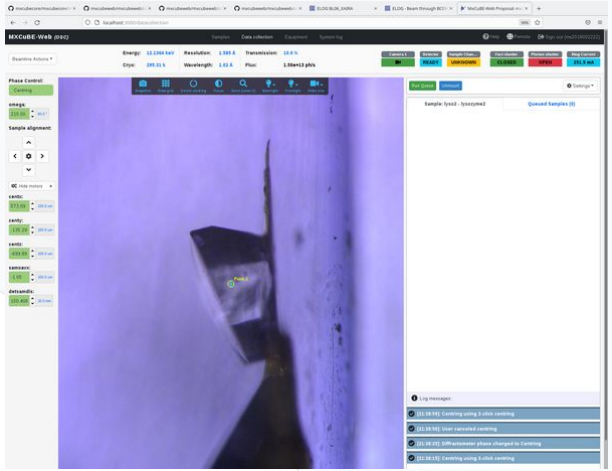


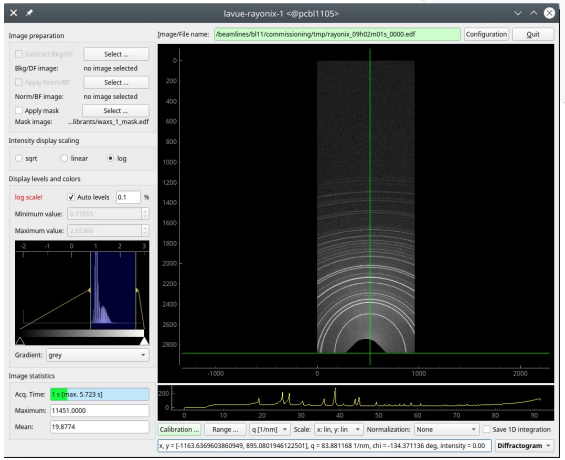
GUIs at ALBA

Relevant non-taurus GUIs used in production

- MxCuBE with Qt and web frontents
- Other *pure* PyQt apps
- Archiving plotting with E-Giga
- LaVue
- Proprietary endstation software:

  Xradia, Prodigy, UView...





Issues and limitations

Taurus

- Keep up-to-date with Qt versions. QtDesigner and custom widgets
- Plotting efficiency (e.g. marks in pyqtgraph)
- General performance

Other

- Standardization of *Ad hoc* solutions
- No-control over closed solutions

Outline

- ALBA Controls GUIs: Current technologies
- **GUI strategy and next steps**
  - Taurus
    - Performance Optimization
    - Qt6 compatibility
    - Trainings - documentation
  - Other technologies (web, cameras/detectors)
- Questions to the community

Taurus Performance Optimization (TPO)

TEP21 – Taurus startup and polling performance optimization (started on 2023)

- **Scalability issues**
  - Slow starts (core optimization, GUI starting issues)
  - Improve polling
  - Responsiveness
  - …
- **Profiling tools** (e.g. cProfile, snakeviz, tuna) helped in finding performance issues
- **Comprehensive benchmark tests:** Crucial in the performance optimization process & future maintenance of performance (to avoid regressions)
- Oct24 release **Taurus 5.2** with TPO additions up-to-date

Taurus Performance Optimization (TPO)

	Attr events 500 / Attr w/o events 0			Attr events 0 / Attr w/o events 500		
	pre-TPO	TPO		pre-TPO	TPO	
Tango Startup time [s]	2.3			1.4		
TaurusCore Startup Time [s]	5.2	3.5	*1.5x*	4.2	2.5	*1.7x*
TaurusLabel Startup Time [s]	5.3	4.1	*1.3x*	3.8	2.1	*1.8x*
TaurusForm Startup Time [s]	14.3	11.3	*1.3x*	11.7	6.2	*1.9x*
		~25% time reduction			*~43% time reduction*	

Taurus Performance Optimization (TPO)

	Attr events 496 (7) / Attr w/o events 81			Attr events 500 / Attr w/o events 0			Attr events 0 / Attr w/o events 500		
	pre-TPO	TPO		pre-TPO	TPO		pre-TPO	TPO	
Tango Startup time [s]	2,6			2.3			1.4		
TaurusCore Startup Time [s]	5.8	3.8	*1.5x*	5.2	3.5	*1.5x*	4.2	2.5	*1.7x*
TaurusLabel Startup Time [s]	5.9	4,3	*1.4x*	5.3	4.1	*1.3x*	3.8	2.1	*1.8x*
TaurusForm Startup Time [s]	16.2	12.0	*1.4x*	14.3	11.3	*1.3x*	11.7	6.2	*1.9x*
		~29% time reduction			*~25% time reduction*			*~43% time reduction*	

Taurus Performance Optimization (TPO)

	Attr events 496 (7) / Attr w/o events 81			Attr events 500 / Attr w/o events 0			Attr events 0 / Attr w/o events 500		
	pre-TPO	TPO		pre-TPO	TPO		pre-TPO	TPO	
Tango Startup time [s]	2,6			2.3			1.4		
TaurusCore Startup Time [s]	5.8	3.8	*1.5x*	5.2	3.5	*1.5x*	4.2	2.5	*1.7x*
TaurusLabel Startup Time [s]	5.9	4,3	*1.4x*	5.3	4.1	*1.3x*	3.8	2.1	*1.8x*
TaurusForm Startup Time [s]	16.2	12.0	*1.4x*	14.3	11.3	*1.3x*	11.7	6.2	*1.9x*
		~29% time reduction			***~25% time reduction***			***~43% time reduction***	

	Real ALBA RF GUI Attr events 496 (7) / Attr w/o events 81			*Real ALBA PCGrid SRMain GUI* Attr events 488 / Attr w/o events 2		
Tango Startup time [s]	5,7			5.1		
TaurusCore Startup Time [s]	14,3	8,9	*1.6x*	10.2	6.6	*1.5x*
TaurusLabel Startup Time [s]	16,4	8,9	*1.8x*	11.4	7.9	*1.4x*
TaurusFrom Startup Time [s]	28,7	17,6	*1.6x*	22.6	17.9	*1.3x*
		~40% time reduction			***~29% time reduction***	

More Taurus improvements

- Continue with TPO: Explore options of optimizing subscription to events in Tango
- Qt6 compatibility (PyQt6/PySide6)
  - MR ready to be reviewed. Targeted for 5.3 release (Q1-25)
- Taurus & pyqtgraph documentation/trainings

Evaluation of other technologies

- **Web**: Several PoC during last 2 years.
  - Taranta  MAX IV LABORATORY SKAO
  - Graphana  (data sources: archiving via pyHDBPP, tango attributes, …)
  - JupyTango + ipywidgets

TANGO

Evaluation of other technologies

- **Web**: Several PoC during last 2 years.
  - Taranta
  - Graphana (data sources: archiving via pyHDBPP, tango attributes, …)
  - JupyTango + ipywidgets
- **Detector/cameras** control integration & visualization:
  - Migration of current solutions
  - BPM GUI
  - DevVimba
  - LaVue

Summary

- ALBA Controls GUIs: Current technologies
- GUI strategy and next steps
  - Taurus
    - Performance Optimization
    - Upgrade to Qt6
    - Trainings - documentation
  - Other technologies (web, cameras/detectors)
- **Questions to the community**

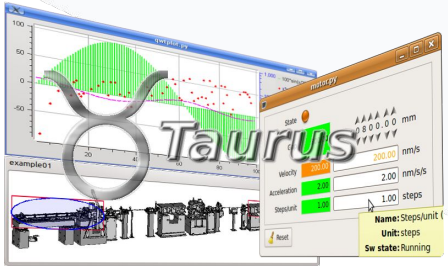
Questions to the community

- General testing strategies for GUIs and regarding the Qt layer (e.g. using pytest-qt and Qtbot).
- Experiences with automatic performance tests (to avoid regressions and performance drops). e.g. https://codspeed.io/
- Use of QML/QtQuick to design Qt GUIs in Control Systems. QtDesignStudio/QtCreator (qml) vs QtDesigner (ui)

Acknowledgements

Miquel Navarro
José A. Ramos
Emilio Morales
Sergi Rubio
Zbigniew Reszela

& all Controls section



Taurus Community