

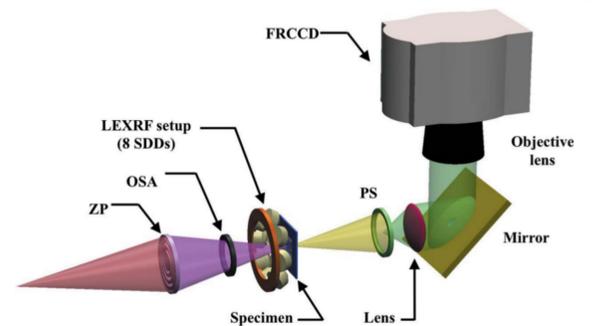
# Evolving a Labview end-station software to a Tango-based solution at the Twinmic Elettra beamline

R. Borghes, G. Kourousias, V. Chenda, A. Gianoncelli

Elettra Sincrotrone Trieste S.C.p.A., Basovizza, ITALY

The **TwinMic Elettra Beamline**, operating since 2007, hosts the European twin X-ray microscopy station. The typical end-station configurations are the transmission X-ray microscopy (TXM), the scanning transmission X-ray microscopy (STXM) and some coherent diffractive imaging modalities like ptychography. Its experimental setup covers a wide range of applications in diverse research fields such as biology, biochemistry, medicine, pharmacology, environment, geochemistry, food, agriculture and materials science.

Control system instrumentation: Andor Ixon and Princeton EEV cameras, XGLab MCAs, Zaber and NI stepper motors, NI I/O cards, PI piezo stage controller, 4 industrial PCs.



## Reasons for an upgrade

- Windows 2000/XP unsupported
- Monolithic LabView application
- Lack of documentation
- New instruments to integrate
- New custom experiments

Why?

1

## Technical choices

- TANGO distributed control system
- Python and PyQt
- DonkiOrchestra Experiment control
- HDF5 scientific data format

4

How?

virtualization  
**DonkiOrchestra**  
 reliability efficiency  
**Tango**  
**Python**  
 upgrade  
**Linux**  
 distributed flexible  
 customizable  
 ZMQ

What?

2

## Beamline requirements

- Everything we do now plus...
- Reuse current instrumentation
- Software reliable and flexible
- Maximize parallel operations
- Standard scientific data format

3

When?

## Time and resource constrains

- Avoid any beamline downtime
- Developments during shutdown periods
- Dual boot Windows/Linux CentOS6
- Long periods between test sessions

The **DonkiOrchestra** framework is a scalable system for data collection and experiment management based on ZeroMQ distributed messaging. Any experiment is organized as a sequence of independent phases started by a synchronization software trigger. As an *orchestra*, the DonkiDirector organizes the execution of multiple tasks performed independently by a DonkiPlayer. Each DonkiPlayer belongs to a priority group and has a specific task to execute. A data acquisition Player, after the execution of the readout task, tags the datum with the actual trigger number and sends it back to the Director that stores the incoming data in suitably structured HDF5 archives.

