LNLS decided to adopt MXCuBE [1] for its macromolecular crystallography beamline, named MX2, considering it is a Python based solution, which is being largely used in our laboratory, its basic support to EPICS, the control system adopted for the LNLS beamlines and because of its stability. Then, existing MXCuBE implementation has been customized to fit LNLS requirements, considering that previously it was mainly ready to control systems other than EPICS. MXCuBE has been used on MX2 beamline of LNLS since the end of 2016 with positive feedback from researchers.

Main customization effort

- There were two principal challenges after have been take decision to adopt MXCuBE in LNLS:
  1. Upgrade MXCuBE from Python 2.7, that was default version when we started working with it, to Python 3.4, which was the minimal version required by our Py4Syn [2] library to operate over EPICS control layer;
  2. Develop customized classes to work with EPICS to perform common procedures of macromolecular crystallography beamlines.
- Both of them have been successfully overcome, working together MXCuBE developers team.

Architecture Overview

![Architecture Overview Diagram]

- User Interface Layer:
  - MXCuBE (Qt4/PyQt4)
    - Brick: graphical control connected to a device;
    - HardwareObjects.xml: configuration of all Bricks instances on control interface;
- Logical (Abstraction) Layer:
  - EPICS (base 3.14.12.4)
    - CA: Channel Access
    - CAS: Channel Access Server
    - IOC: Input/Output Controller
  - Python (version 3.4)
    - Implemented classes by LNLS inside MXCuBE solution
      - Py4Syn
      - PyEPICS
- Electronic (Technical) Equipment

Some original LNLS Bricks

- Attending some requirements of MX2 staff, some original widgets were implemented, like these:
  - Beam intensity automatic optimization
    - Py4Syn and Matplotlib Python libraries;
  - Embedded CBF viewer
    - cbf Python library of PSI;
  - Amptek MCA control of dead-time
    - Py4Syn Python library;

Screenshots of current operational UIs

![Main LNLS MX2 beamline operation UI]

![Beam intensity optimization Brick]

![Embedded CBF viewer]

References