PyMoDAQ
Modular Data Acquisition with Python

Sébastien Weber

Ingénieur de Recherche CNRS
Centre d'Elaboration de Matériaux et d'Etudes Structurales

CNRS — *dépasser les frontières*

CEMES
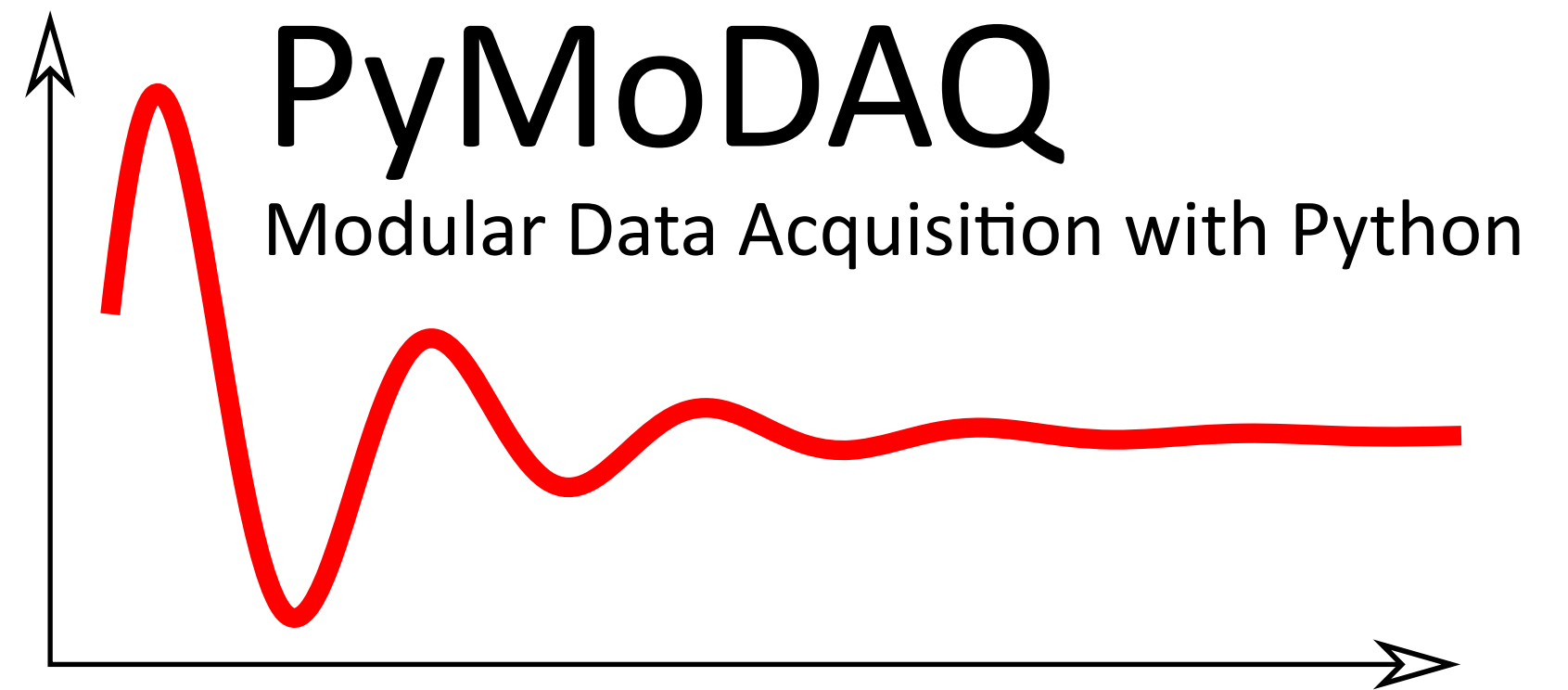
PyMoDAQ features

Base language: Python
- *large community*
- *extensive libraries*

Support via Github
- *continuity of the product*
- *project*
- *documentation*

Double fonction:
- *manual*
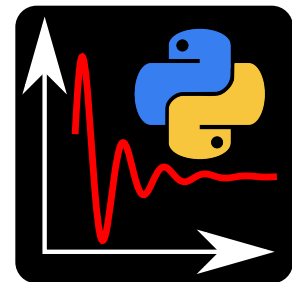- *automated*

New hardware?
Small python plugin using base resources and template

New experiment?
Easy preset modes building using available plugins

Hierarchical binary data saving
- *hdf5 file format*
- *metadata*

DAQ Scan: a unique module to rule them ALL!
- Interface initialising actuators and detectors as needed for a particular experiment
- On line creation of preset modes for specific experiments
- Automatic scan involving 1 or 2 actuators (1D or 2D scans) and multiple detectors
- Various mode of scans possible
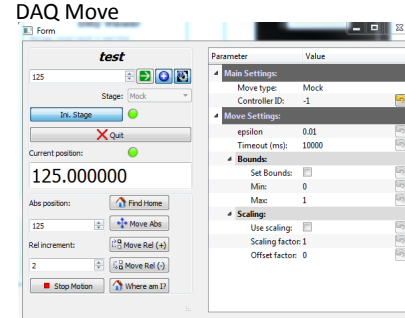- Save data as hierarchical binary HDF5 files containing

1..n — 1..m

DAQ Move
- Manage actuators: relative or absolute positioning
- One single interface for each actuator
- Specific parameters loaded on the fly
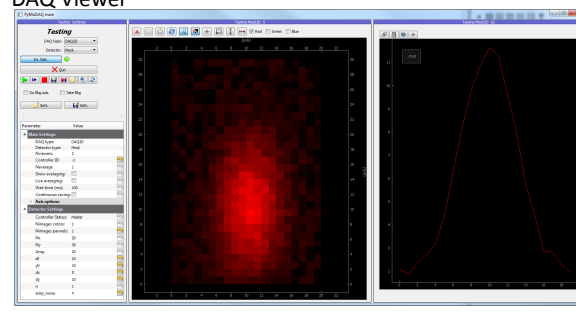- Options: limited range, axis scaling...

DAQ Viewer
- Manage detectors: live visual acquisition
- One single interface for each detector
- Saving and quick analysis features
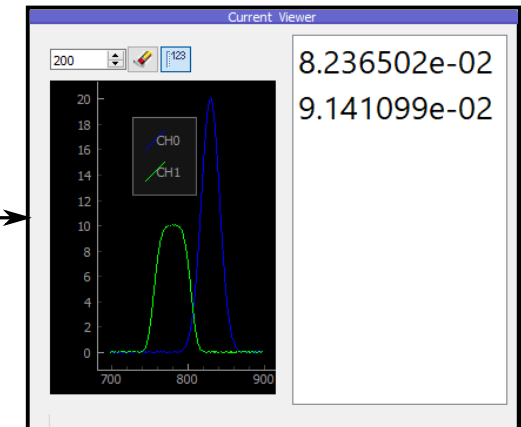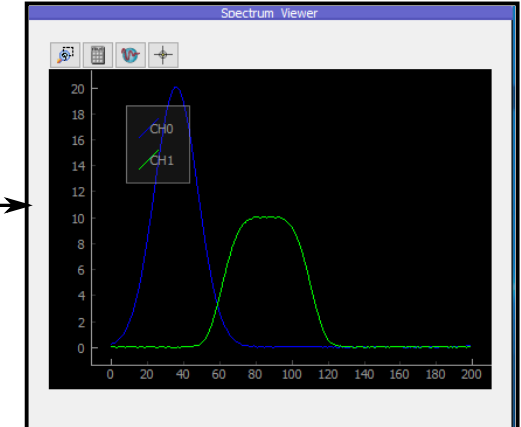- Specific parameters loaded on the fly

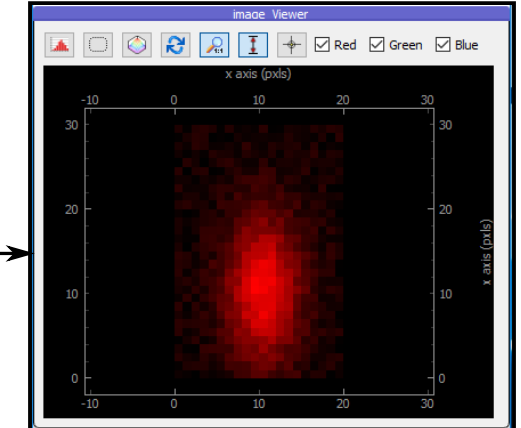**Hardware**
- One simple script to write following a template

Hardware
- One simple script to write following a template

Type of viewers

Viewer 0D

Viewer 1D

Viewer 2D

PyMoDAQ usage

Start a new experiment:

1) Define a preset mode
- *all needed plugins to load*
- *configure their settings*
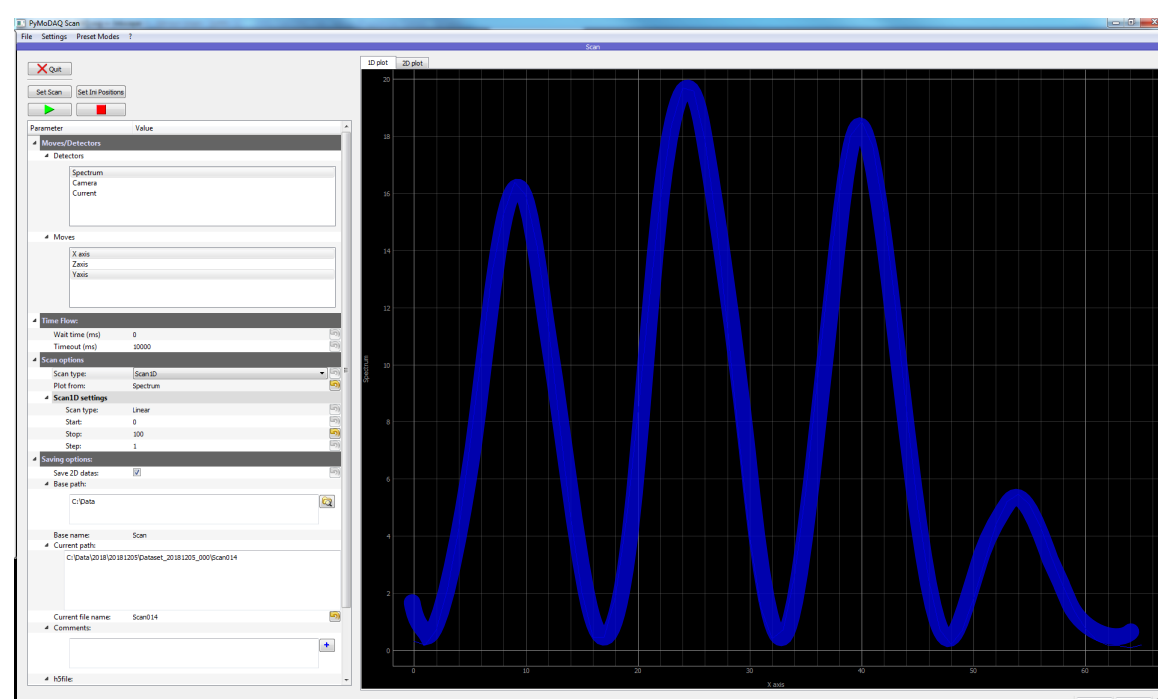- *all set in one click for later use*

2) Align experiment
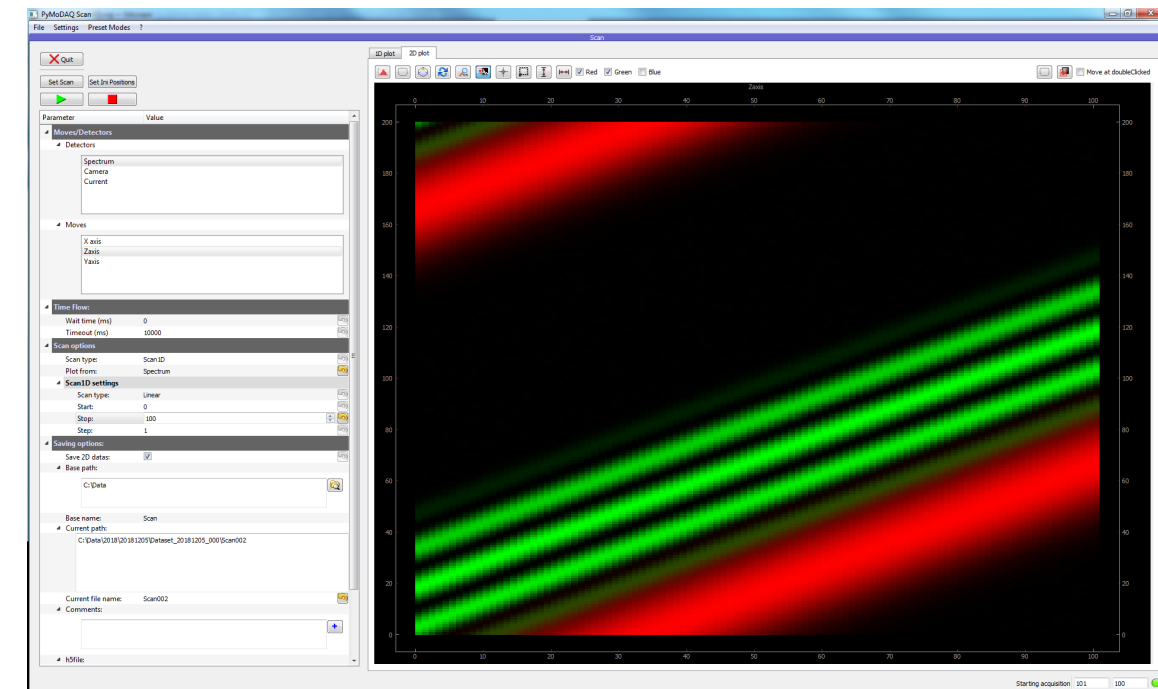*Manual actuators alignment while recording live data*

3) Perform scan
Chose modules and set Scan settings
Set live data plotting
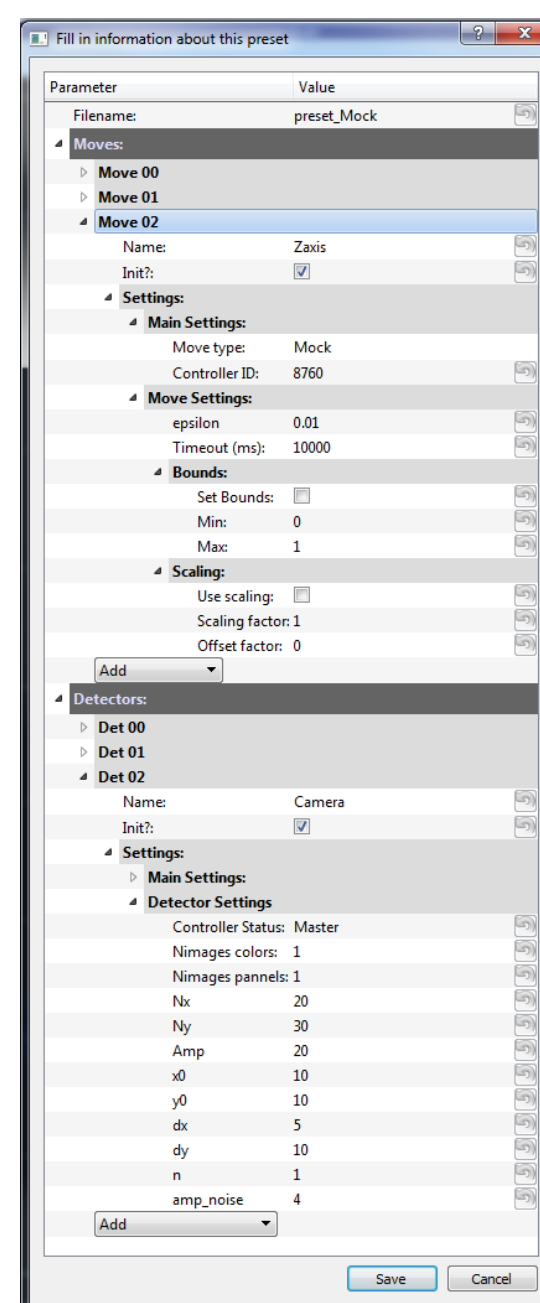*Record metadata about scan*
Run scan

4) Review saved data

Preset manager

Modules control pannel

Scan 1D of 0D Data

Scan 1D of 1D Data
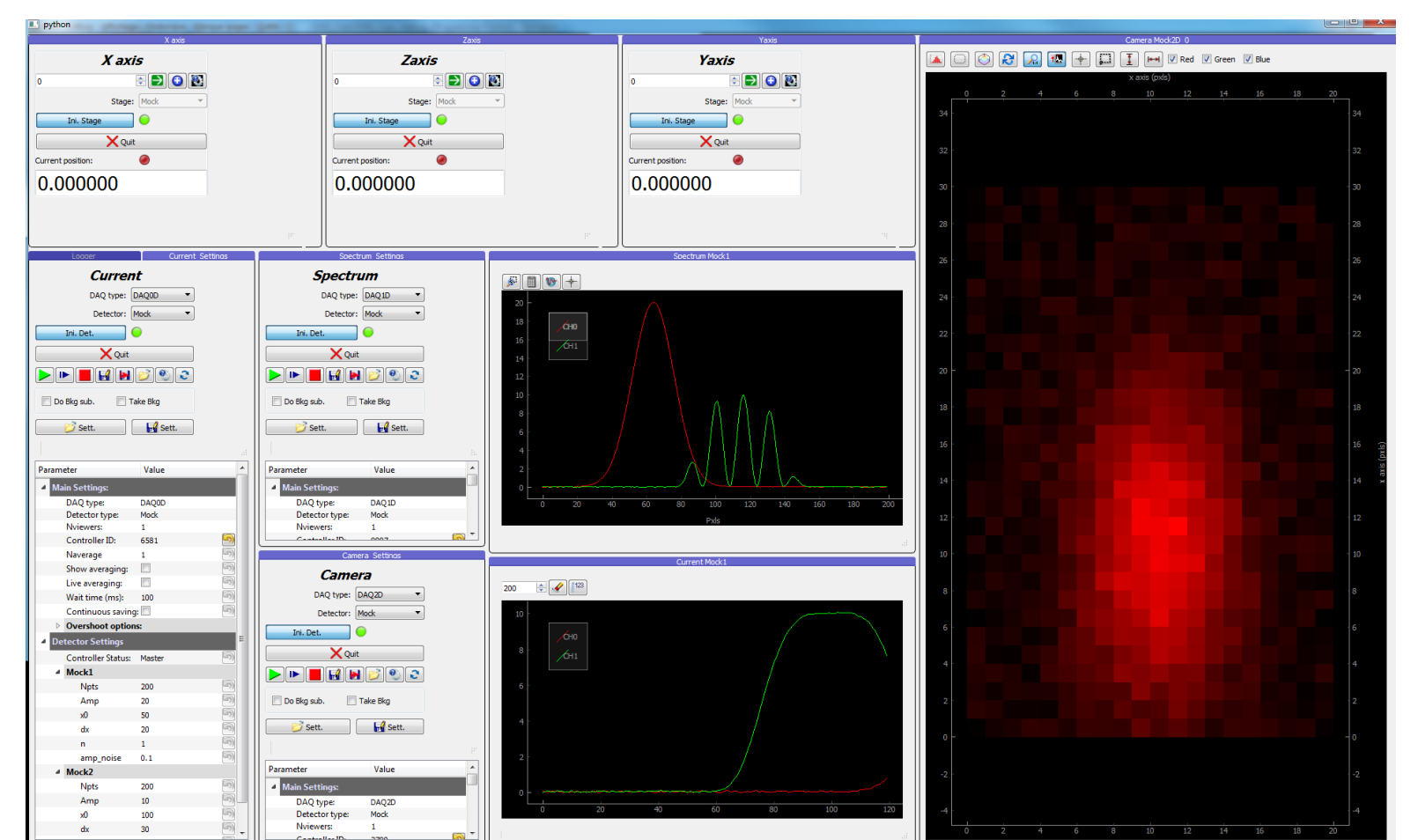
Scan 2D of 0D Data

h5 file browser - ND viewer