



CEMES

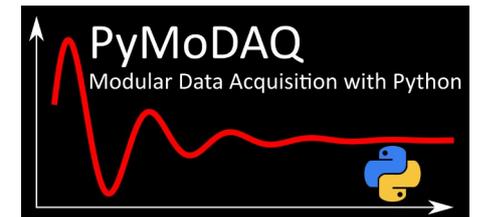
©
Patri

PyMoDAQ

Modular Data Acquisition with Python

Overview

Sébastien Weber



Contexte et vocabulaire

Je veux mesurer des données en fonction d'un ou plusieurs paramètres variables !

Decteurs

- Caméra
- Signaux analogiques
- Spectres
- ...

Actuateurs

- Platine de déplacement
- Rotation
- Contrôleur de température
- Courant
- ...

Camera

XY stage

Laser wavelength

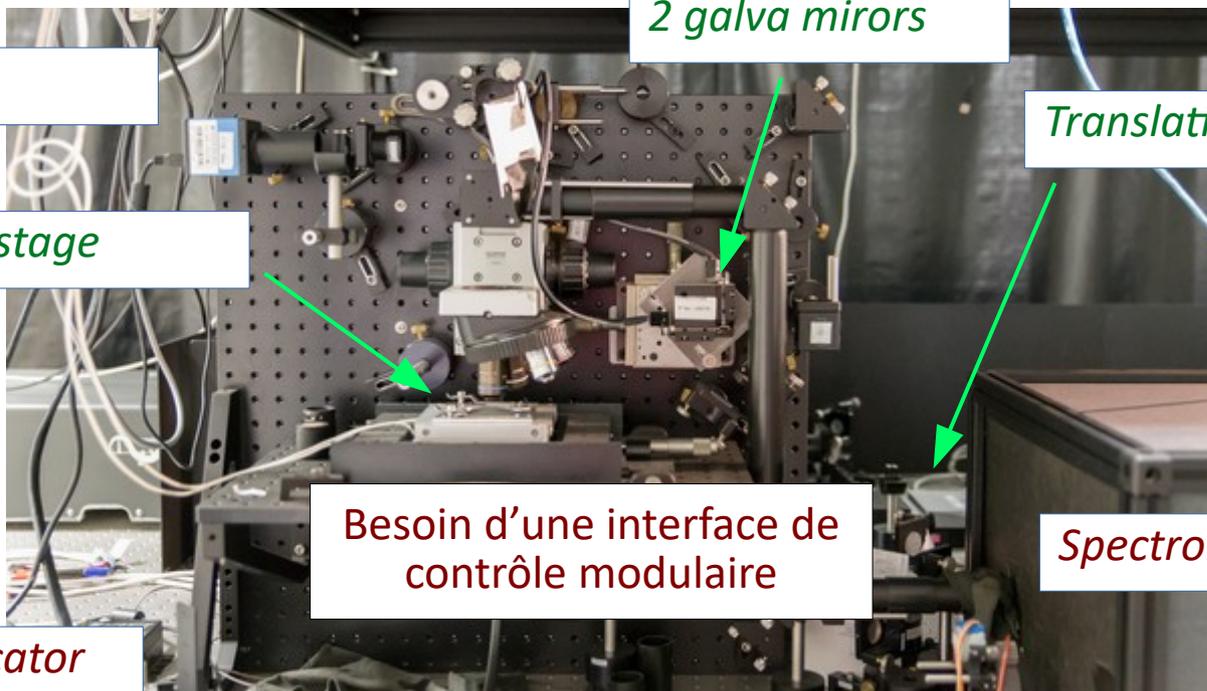
Photomultiplier

2 galva mirrors

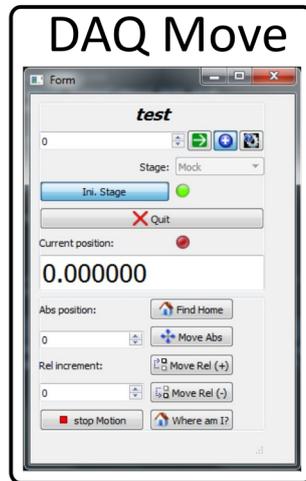
Translation stage

Besoin d'une interface de contrôle modulaire

Spectrometer

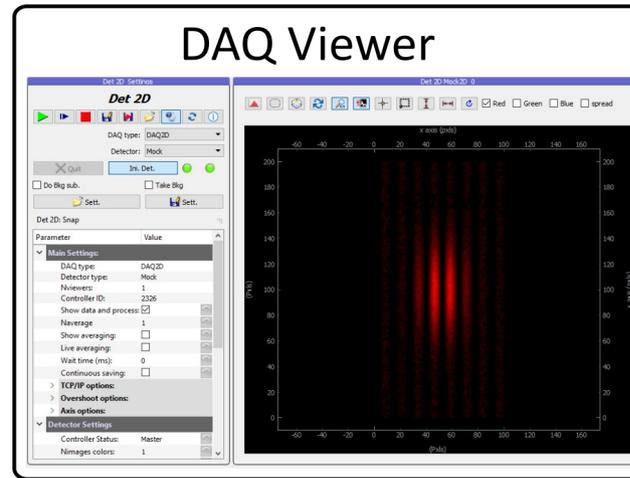


Control Modules



M
●●●

Actuator

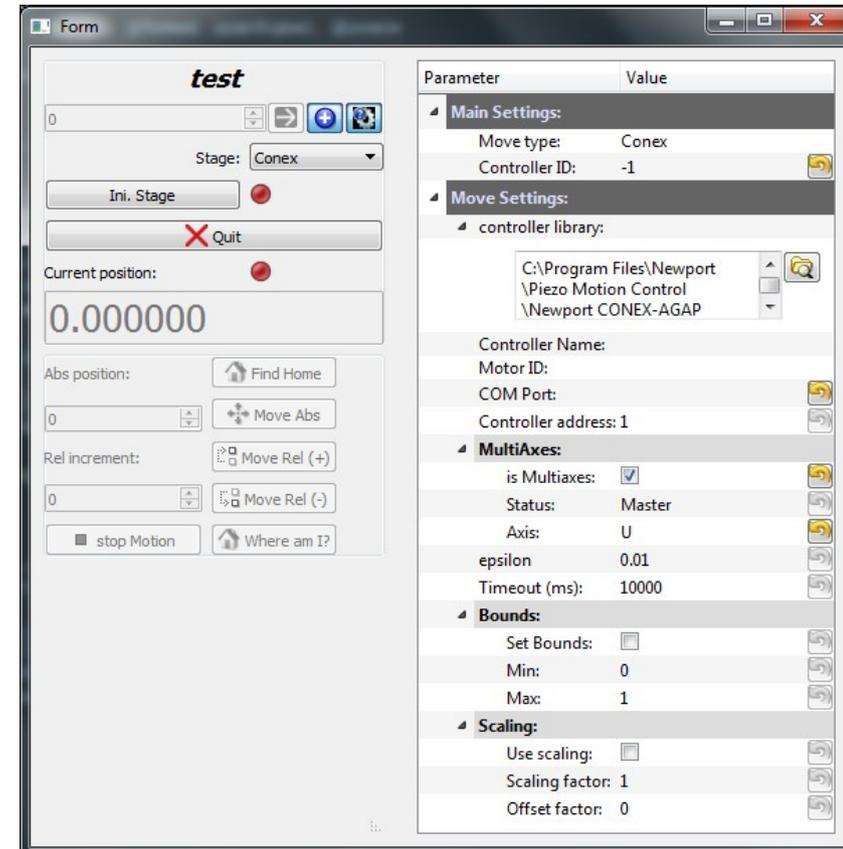
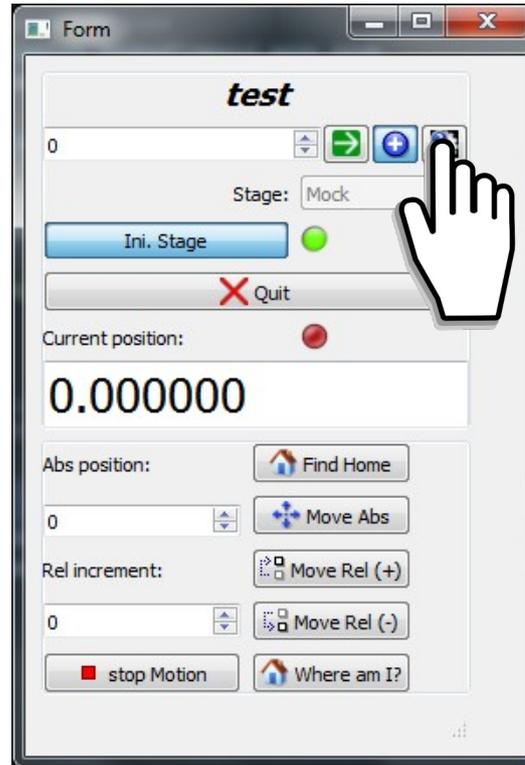
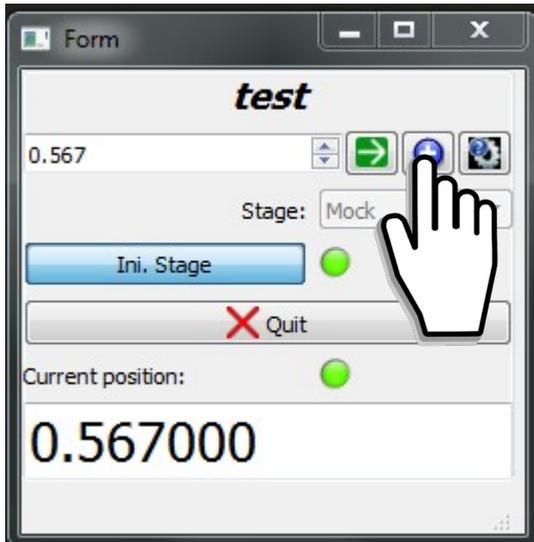


N
●●●

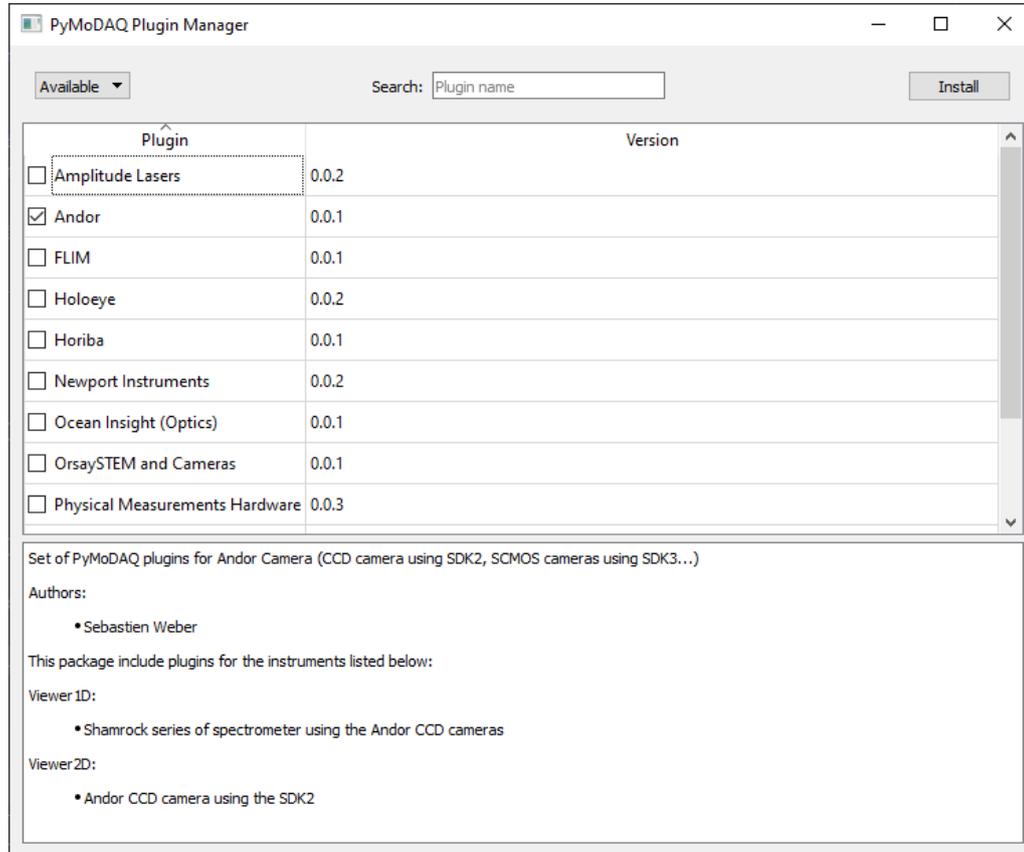
Detector

Dashboard

DAQ Move : « Déplacement » des actionneurs



1) Utiliser le Plugin Manager

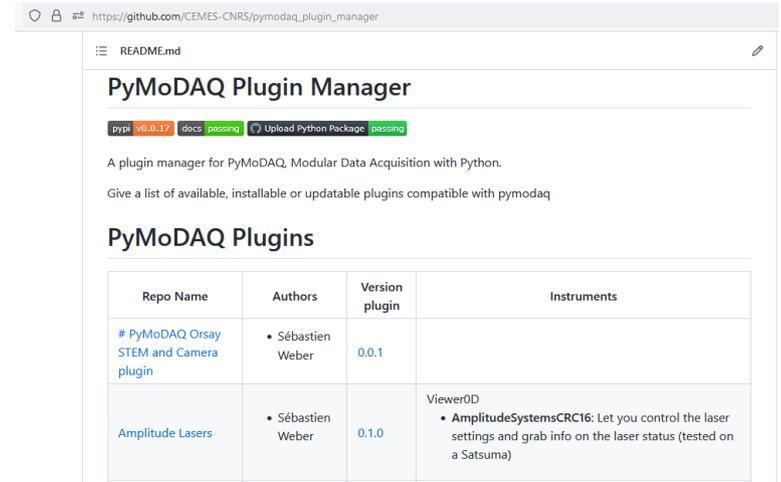


The screenshot shows the PyMoDAQ Plugin Manager window. It features a search bar with the text "Plugin name" and an "Install" button. Below is a table of available plugins:

Plugin	Version
<input type="checkbox"/> Amplitude Lasers	0.0.2
<input checked="" type="checkbox"/> Andor	0.0.1
<input type="checkbox"/> FLIM	0.0.1
<input type="checkbox"/> Holoeye	0.0.2
<input type="checkbox"/> Horiba	0.0.1
<input type="checkbox"/> Newport Instruments	0.0.2
<input type="checkbox"/> Ocean Insight (Optics)	0.0.1
<input type="checkbox"/> OrsaySTEM and Cameras	0.0.1
<input type="checkbox"/> Physical Measurements Hardware	0.0.3

Below the table, there is a description: "Set of PyMoDAQ plugins for Andor Camera (CCD camera using SDK2, SCMOS cameras using SDK3...)" and a list of authors: "Sebastien Weber". It also lists viewer IDs: "Shamrock series of spectrometer using the Andor CCD cameras" and "Andor CCD camera using the SDK2".

2) Aller sur Github

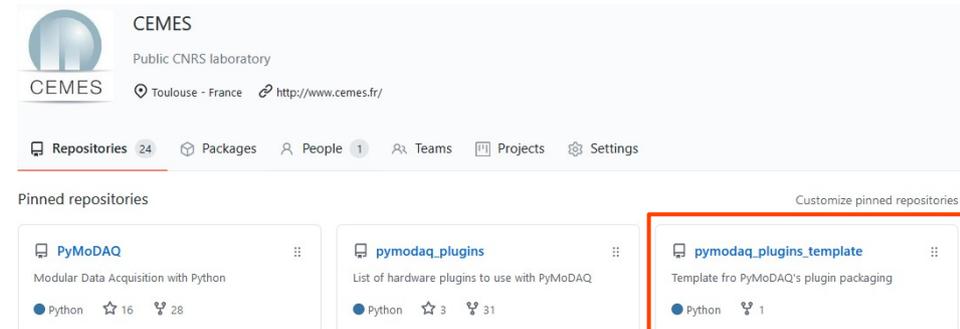


The screenshot shows the GitHub repository page for "PyMoDAQ Plugin Manager". The repository is under the user "CEMES" and is a public repository. It has a README.md file and a "pypi v0.0.17" badge. The repository description is: "A plugin manager for PyMoDAQ, Modular Data Acquisition with Python. Give a list of available, installable or updatable plugins compatible with pymodaq".

Below the description is a table of PyMoDAQ Plugins:

Repo Name	Authors	Version plugin	Instruments
# PyMoDAQ Orsay STEM and Camera plugin	• Sébastien Weber	0.0.1	
Amplitude Lasers	• Sébastien Weber	0.1.0	Viewer0D • AmplitudeSystemsCRC16 : Let you control the laser settings and grab info on the laser status (tested on a Satsuma)

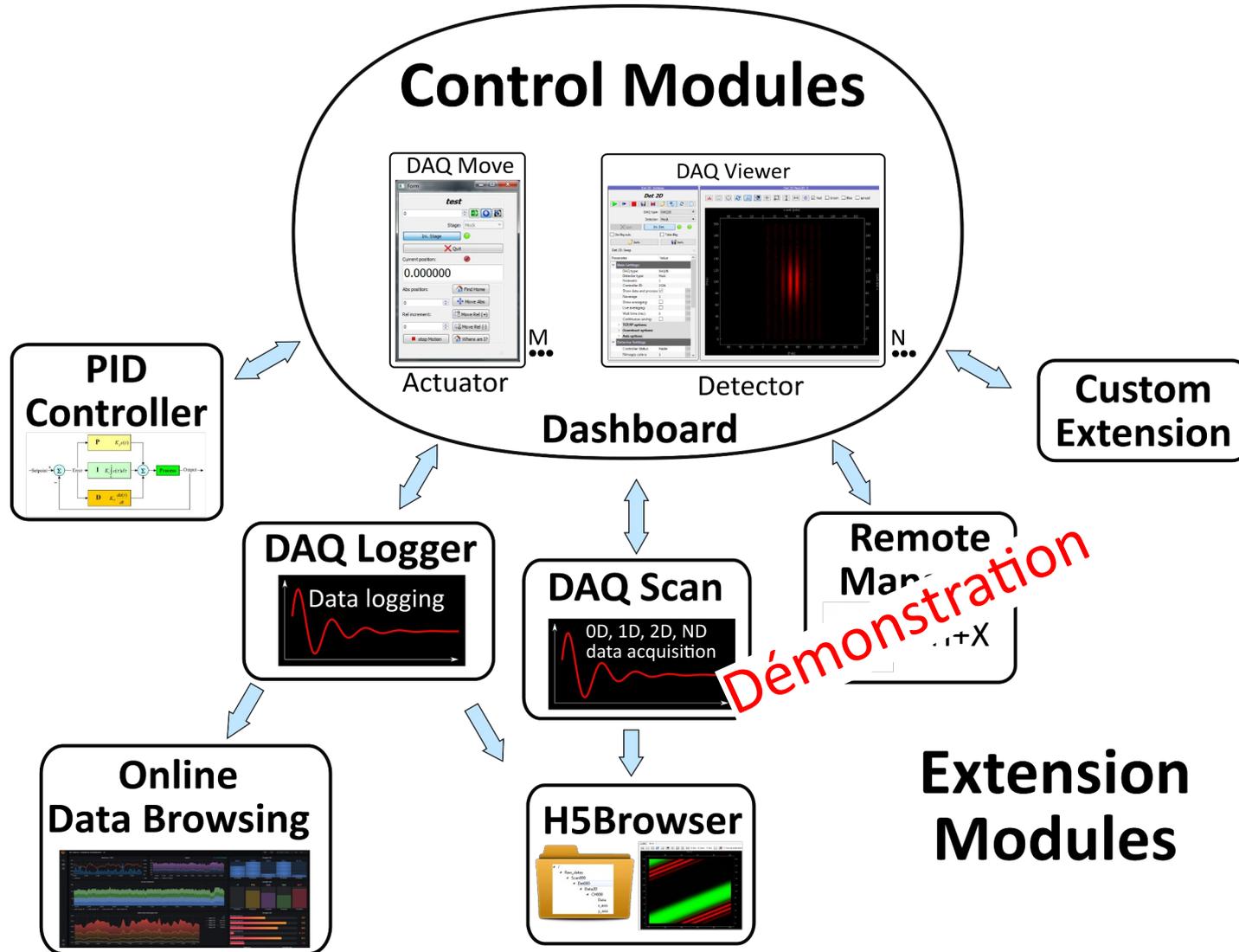
3) Le construire à partir d'un template



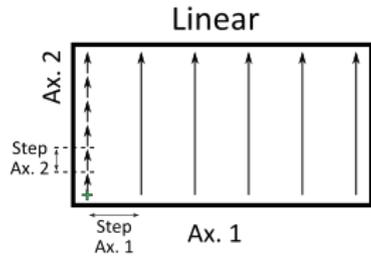
The screenshot shows the GitHub profile page for "CEMES", a public CNRS laboratory based in Toulouse, France. The profile has 24 repositories, 1 package, 1 person, 1 team, 1 project, and 1 setting.

Under "Pinned repositories", three repositories are listed:

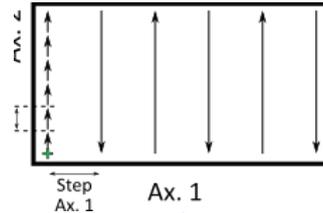
- PyMoDAQ**: Modular Data Acquisition with Python. 16 stars, 28 forks.
- pymodaq_plugins**: List of hardware plugins to use with PyMoDAQ. 3 stars, 31 forks.
- pymodaq_plugins_template**: Template for PyMoDAQ's plugin packaging. 1 fork. This repository is highlighted with a red box.



Scan Linéaires : 1, 2,..., N
actuateurs



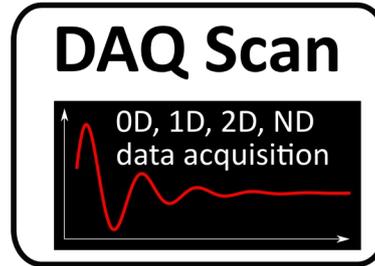
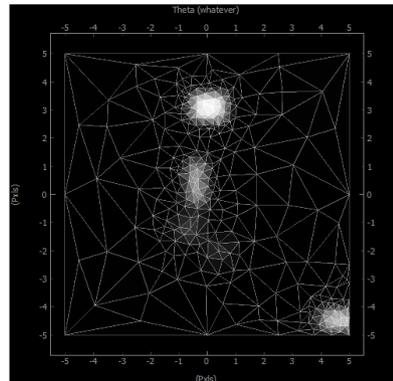
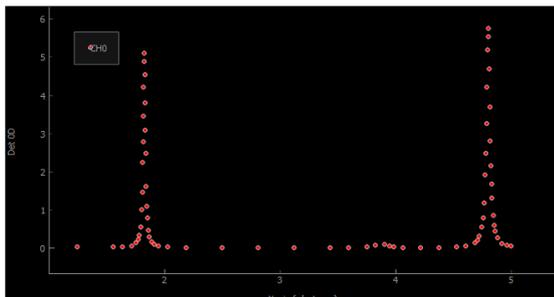
Scan Linéaire Back and
Forth : 2 actuateurs



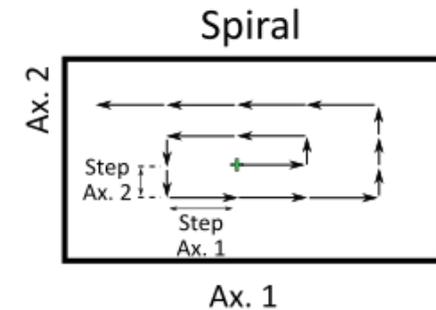
Scan Random :
1, 2,..., N
actuateurs

Scan Tabulaire : 1,
2, ..., N actuateurs

Scan Adaptatif : 1
ou 2 actuateurs



Scan avec référence :
1 ou 2 actuateurs



Scan Spiral : 2 actuateurs

python Scan

File Settings

Parameter	Value
Actuators/Detectors Selection	
▼ detectors	
Det 1D	
Det 0D	
Det 2D	
▼ Actuators	
Theta Axis	
Yaxis	
Xaxis	
Moves done?	<input checked="" type="checkbox"/>
Detections done?	<input checked="" type="checkbox"/>
Data dimensions	
Actuators positions	

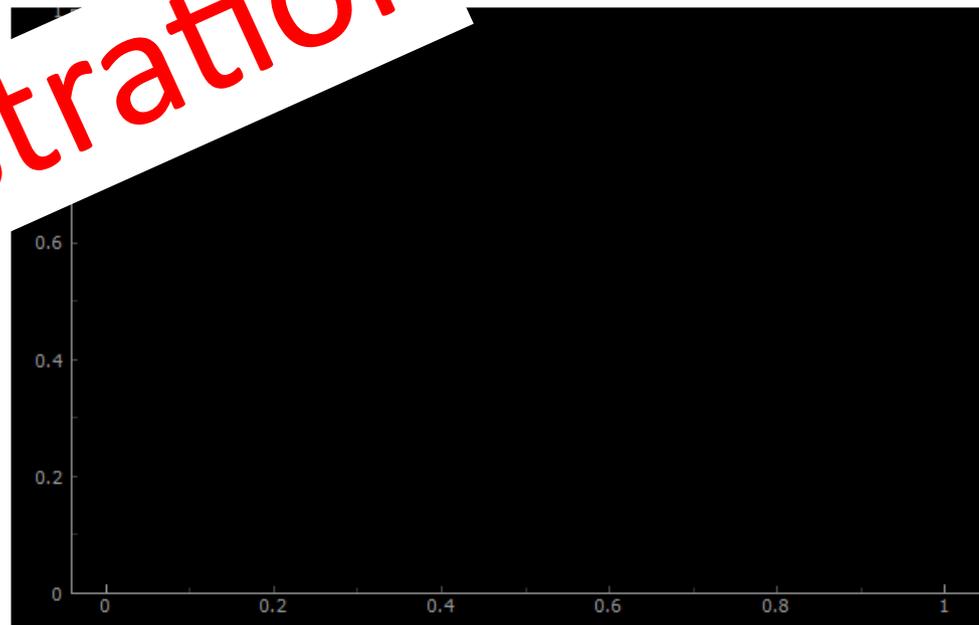
General Settings

Save Settings

Scanner Settings

Parameter	Value
▼ Scanner settings	
calculat	
	-2
stop:	3
Step:	0.5
<input type="button" value="load_xml"/>	
<input type="button" value="save_xml"/>	

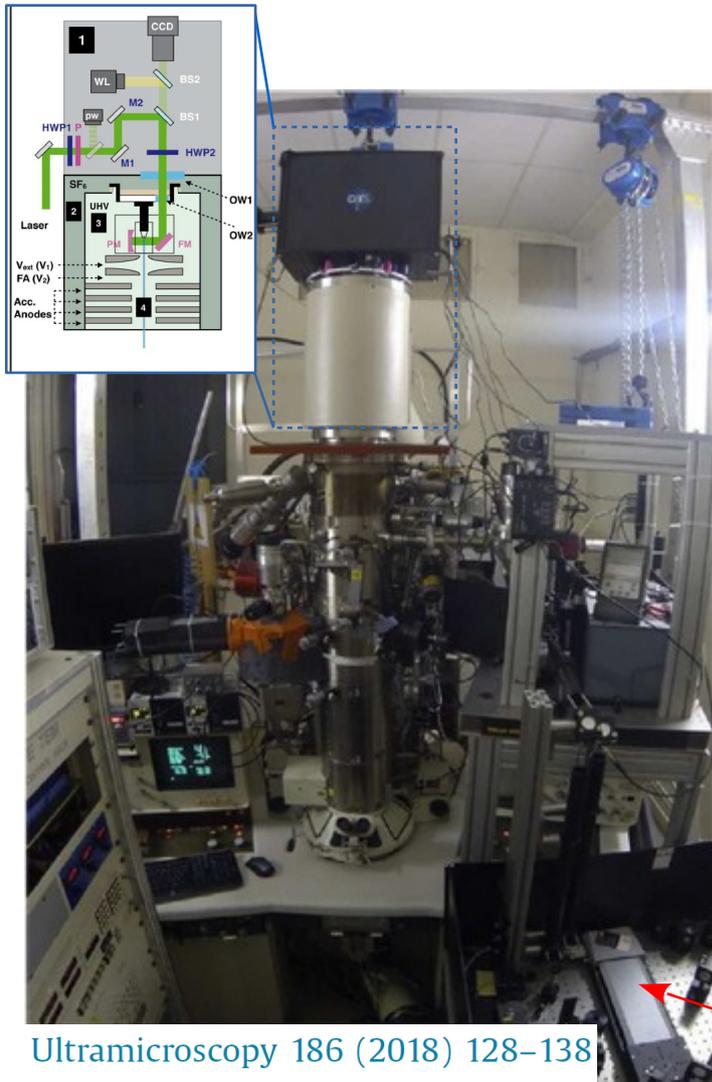
1D plot 2D plot



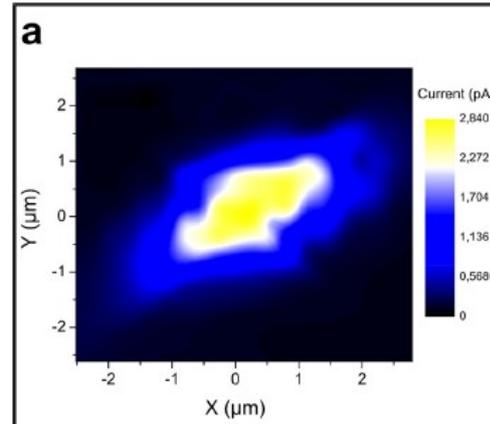
Initializing

Démonstration

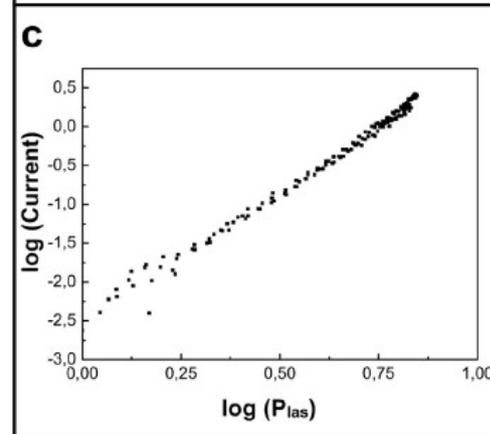
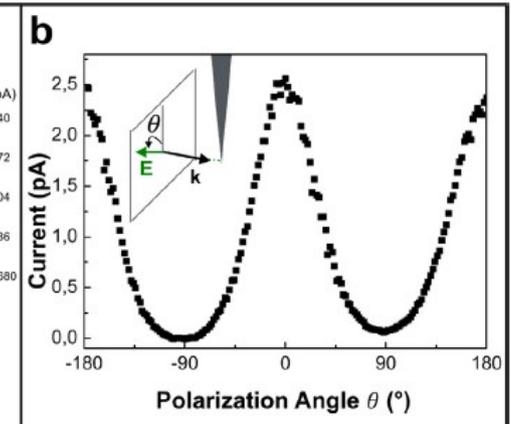
Exemples de scans sur le microscope ultra-rapide FemtoTEM



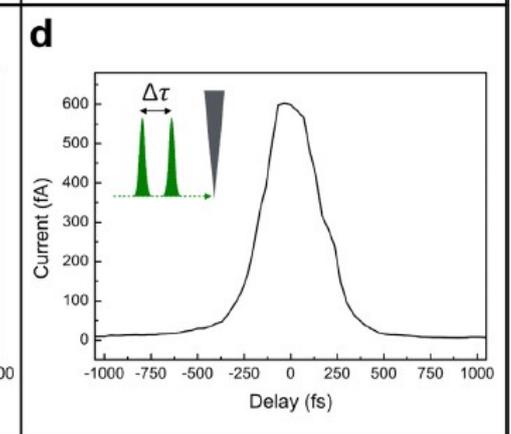
Probe current (Pico-amperemeter) as a function of laser pulse displacement, axes XY mirror M2



Probe current (Pico-amperemeter) as a function of laser polarisation (HWP2)

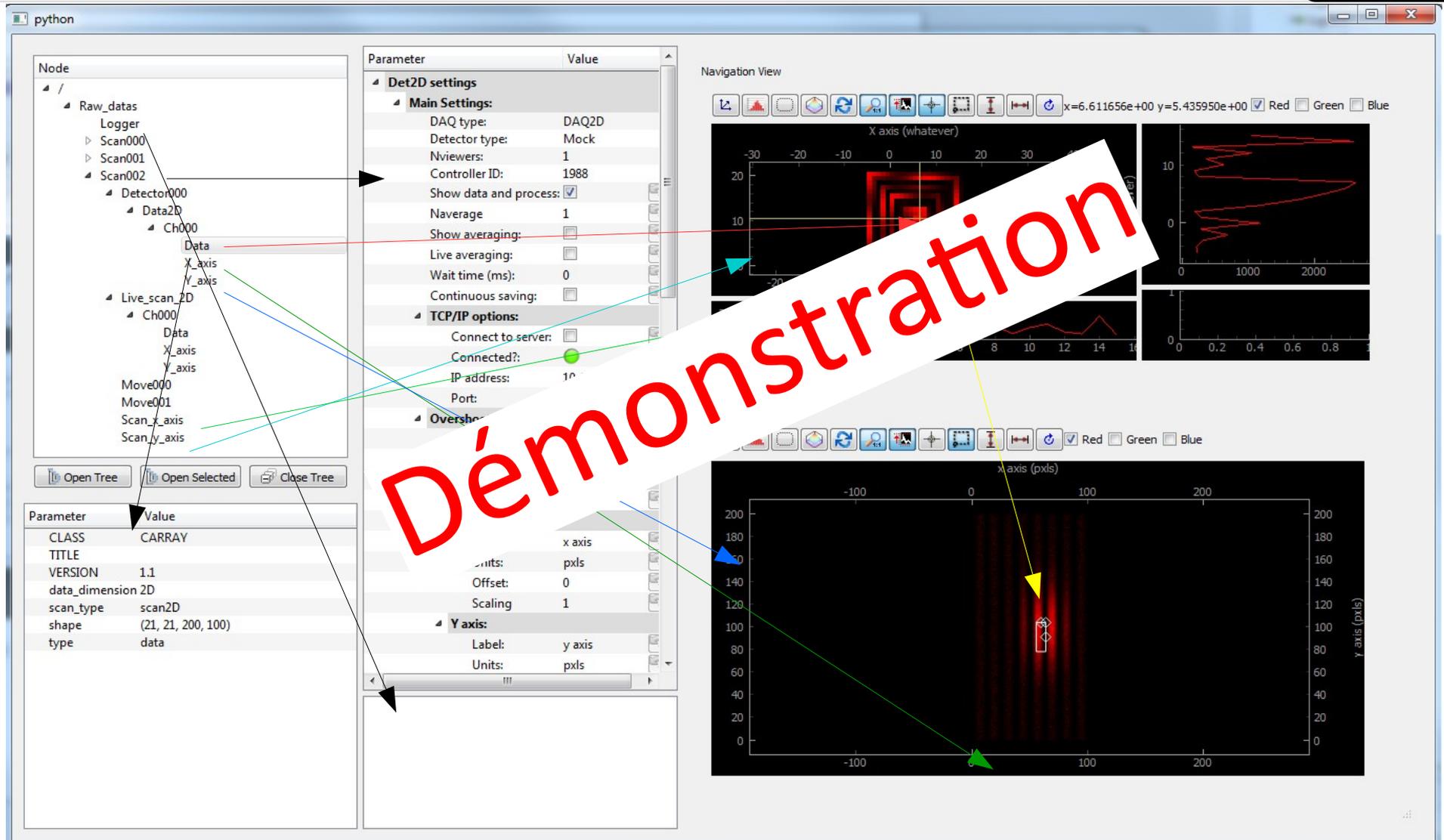
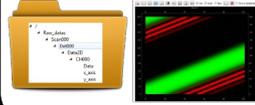


Probe current (Pico-amperemeter) as a function of laser intensity (HWP1)



Probe current (Pico-amperemeter) as a function of pump probe delay femtosecond

Pump-probe delay



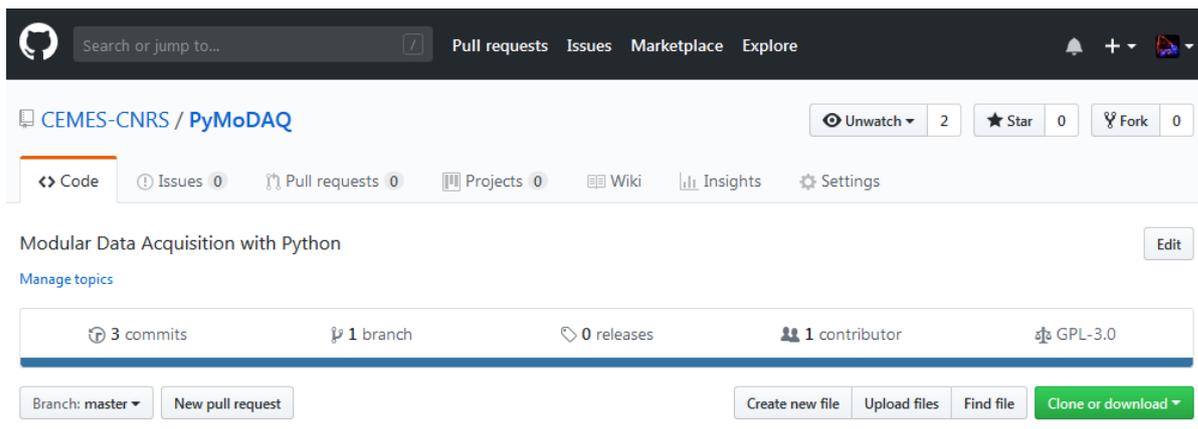
The screenshot displays the H5Browser interface within a Python environment. The interface is divided into several panels:

- Node Tree:** A hierarchical tree on the left showing the data structure. The selected node is `Scan002` / `Detector000` / `Data2D` / `Ch000` / `Data` / `X_axis`.
- Parameter List:** A table listing parameters and their values. The selected parameter is `X_axis`.
- Navigation View:** A panel on the right showing a 2D plot of the selected parameter. The plot shows a 2D histogram with a red overlay. The axes are labeled `X axis (whatever)` and `Y axis (pxls)`. The plot is zoomed in to show a specific region.
- Parameter Settings:** A panel in the center showing the configuration for the selected parameter. The settings include:
 - Main Settings:** DAQ type: DAQ2D, Detector type: Mock, Nviewers: 1, Controller ID: 1988, Show data and process: , Naverage: 1, Show averaging: , Live averaging: , Wait time (ms): 0, Continuous saving: .
 - TCP/IP options:** Connect to server: , Connected?: , IP address: 10..., Port: ...
 - Overshots:** X axis: Units: pxls, Offset: 0, Scaling: 1, Y axis: Label: y axis, Units: pxls.

A large red watermark reading "Démonstration" is overlaid diagonally across the center of the interface.

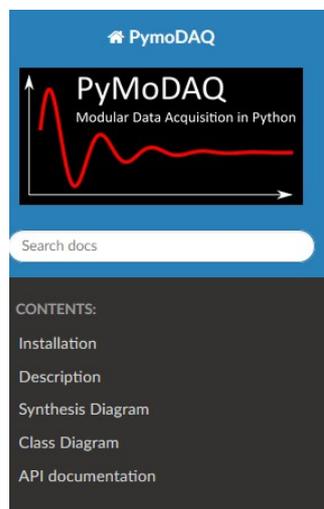
Documentation et contributions

<https://github.com/CEMES-CNRS/PyMoDAQ>



The screenshot shows the GitHub repository page for CEMES-CNRS/PyMoDAQ. At the top, there is a search bar and navigation links for Pull requests, Issues, Marketplace, and Explore. Below this, the repository name is displayed along with statistics: Unwatch (2), Star (0), and Fork (0). The main navigation bar includes Code, Issues (0), Pull requests (0), Projects (0), Wiki, Insights, and Settings. The repository description is "Modular Data Acquisition with Python" with an Edit button. Below the description, it shows 3 commits, 1 branch, 0 releases, 1 contributor, and the GPL-3.0 license. At the bottom, there are buttons for "Branch: master", "New pull request", "Create new file", "Upload files", "Find file", and "Clone or download".

<http://pymodaq.cnrs.fr/>



The screenshot shows the header of the PyMoDAQ documentation website. It features a blue header with the PyMoDAQ logo and the text "Modular Data Acquisition in Python". Below the header is a search bar labeled "Search docs". On the left side, there is a dark sidebar with a "CONTENTS:" section listing: Installation, Description, Synthesis Diagram, Class Diagram, and API documentation.

[Docs](#) » Welcome to PyMoDAQ's documentation!

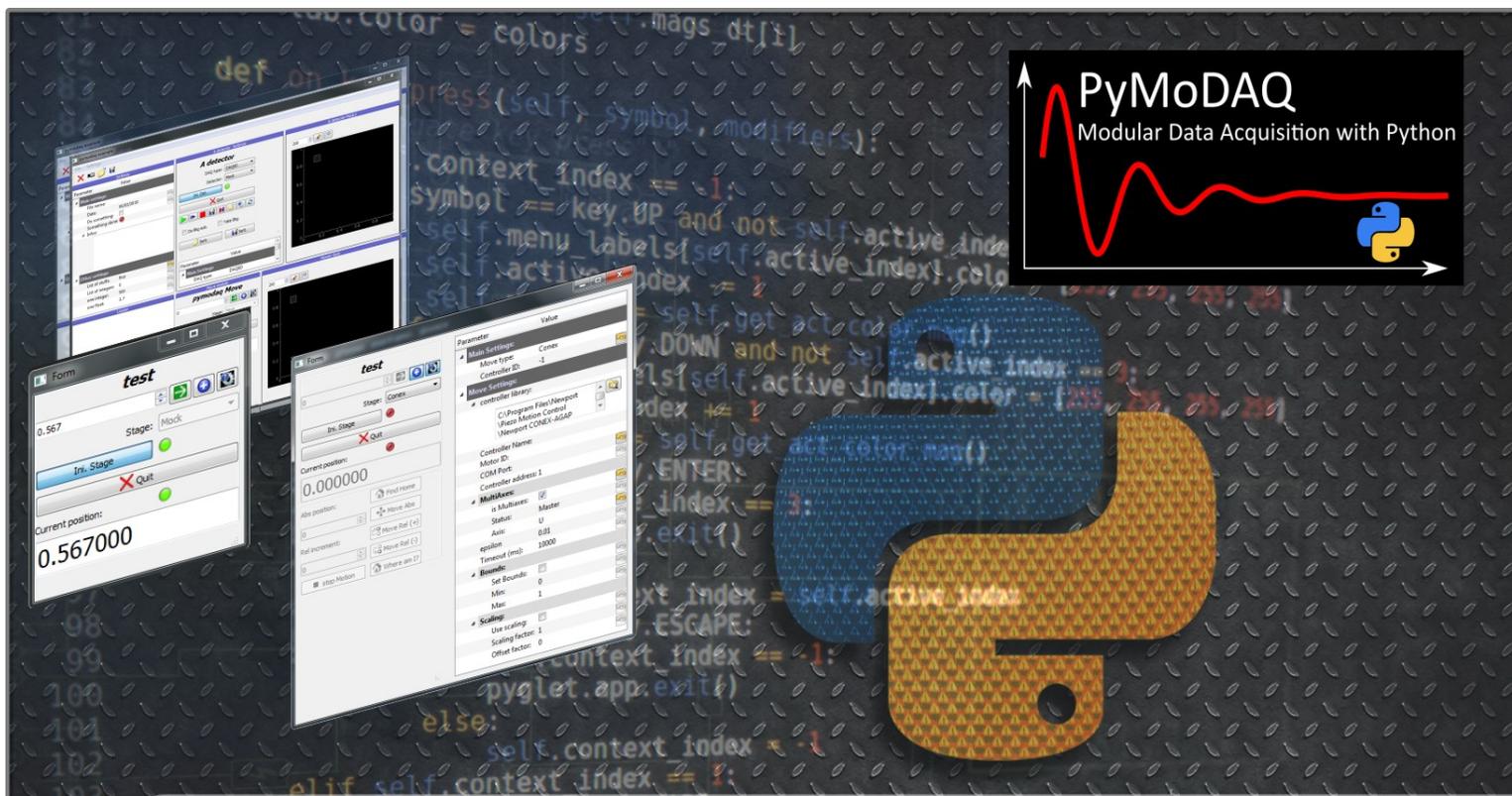
[View page source](#)

Welcome to PyMoDAQ's documentation!

Contents:

- Installation
 - Automatic setup
 - Manual setup
- Description
 - Main Modules
 - DAQ_Move
 - Introduction
 - A paragraph
 - Another paragraph
 - DAQ_Scan

Merci de votre attention



Pré-Formation
Outils de versionage:
Git, Github
Programmation objet
et interfaces graphiques

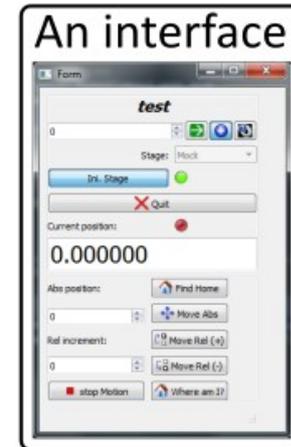
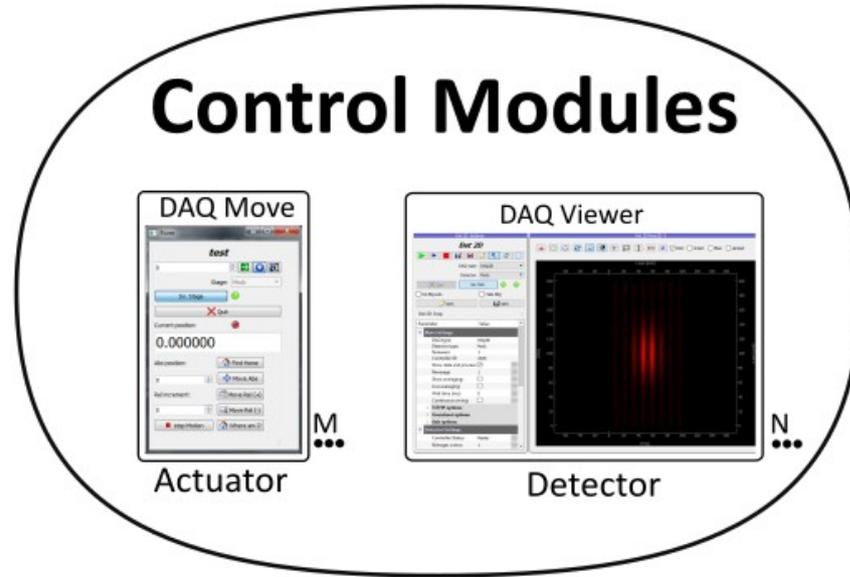
En visio en septembre 2022

Action Nationale de Formation
Maîtriser et développer
PyMoDAQ
Les 17/18/19 octobre 2022
au CEMES à Toulouse

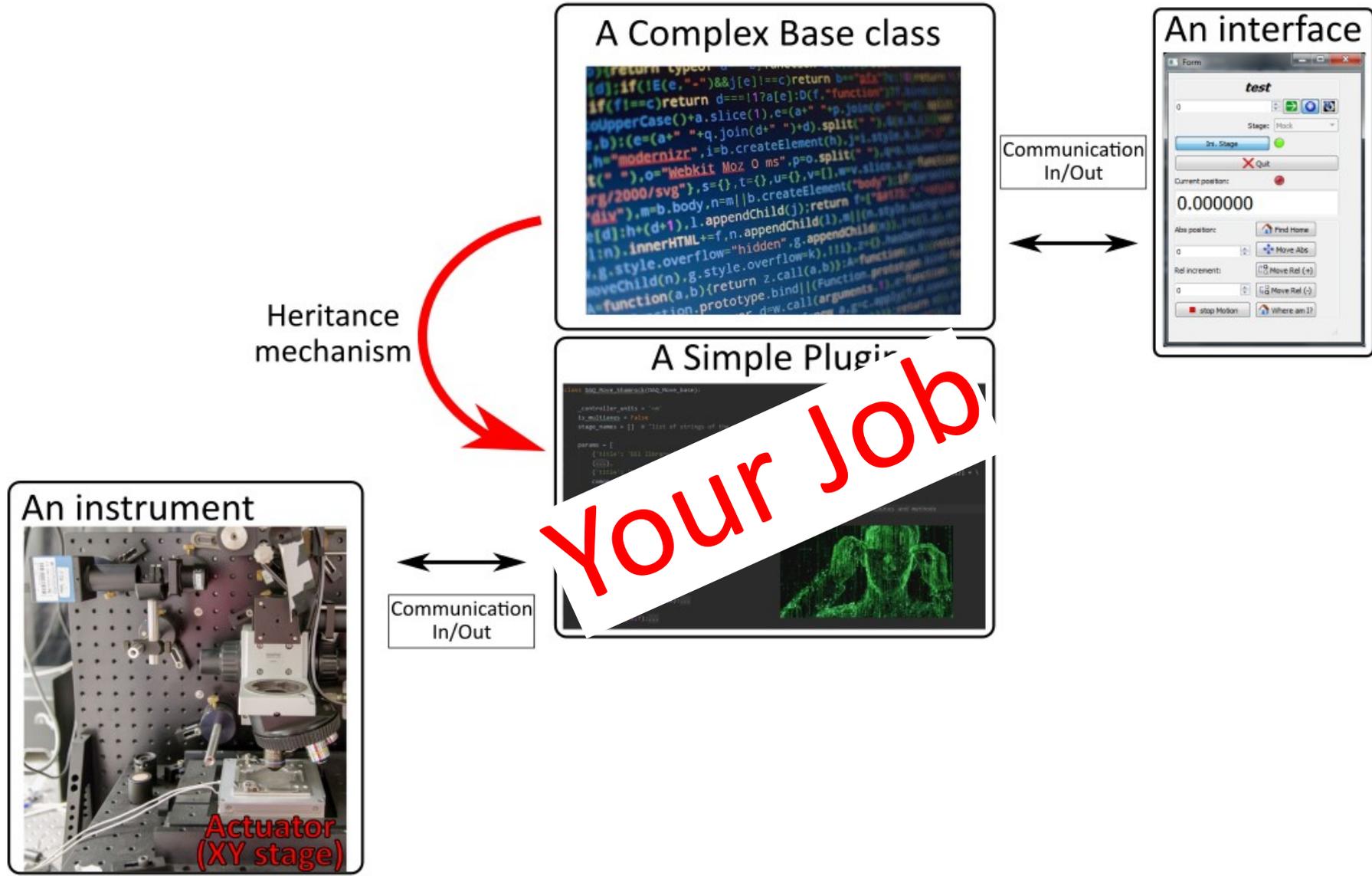
Préinscriptions avant le 10 septembre: sebastien.weber@cemes.fr

Journées PyMoDAQ
Réalizations, tables rondes et
évolutions du logiciel
Les 20 et 21 octobre 2022
au CEMES à Toulouse

What's behind a Control module

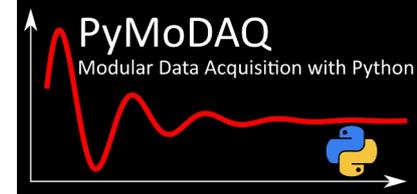


What's behind a Control module





Getting a template and understanding its structure



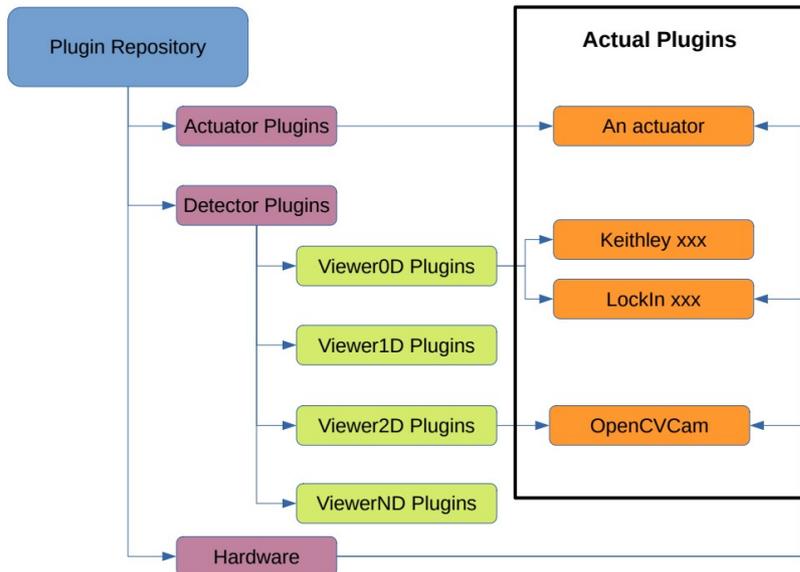
CEMES
Public CNRS laboratory
Toulouse - France <http://www.cemes.fr/>

Repositories 24 Packages People 1 Teams Projects Settings

Pinned repositories

- PyMoDAQ
Modular Data Acquisition with Python
Python 16 28
- pymodaq_plugins
List of hardware plugins to use with PyMoDAQ
Python 3 31
- pymodaq_plugins_template**
Template from PyMoDAQ's plugin packaging
Python 1

Customize pinned repositories



```
└─ pymodaq_plugins_physical_measurements
  └─ pymodaq_plugins_physical_measurements
    └─ daq_move_plugins
      └─ daq_viewer_plugins
        └─ plugins_0D
          ├── __init__.py
          ├── daq_0Dviewer_Keithley2110.py
          ├── daq_0Dviewer_Keithley_Pico.py
          ├── daq_0Dviewer_LockIn7270.py
          └─ daq_0Dviewer_LockInSR830.py
        └─ plugins_1D
          └─ plugins_2D
            ├── __init__.py
            ├── daq_2Dviewer_GenICam.py
            ├── daq_2Dviewer_OpenCVCam.py
            └─ daq_2Dviewer_TIS.py
          └─ plugins_ND
            ├── __init__.py
            └─ hardware
```