Pymodaq for scanning NV center microscopy

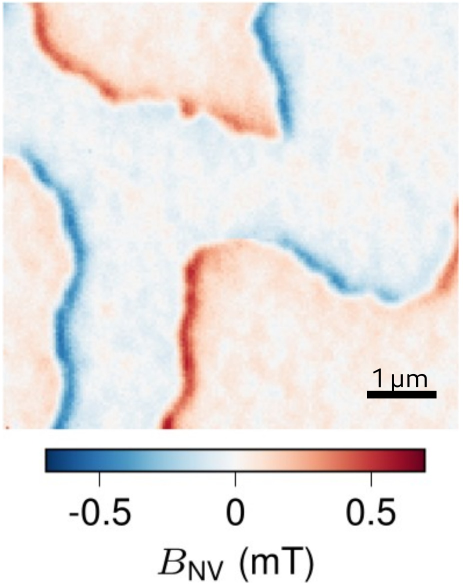
S2QT team, Laboratoire Charles Coulomb, Montpellier

Elias Sfeir, Roméo Beignon, Elijah Wane, Aurore Finco

Pymodays, October 22nd 2024, Lyon

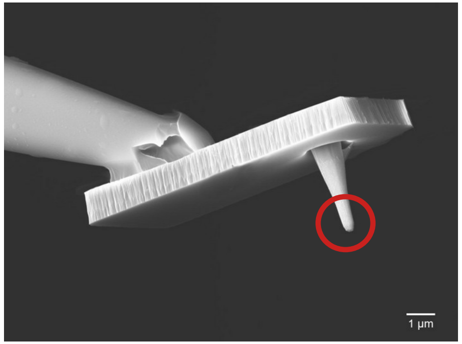
Scanning NV center microscopy

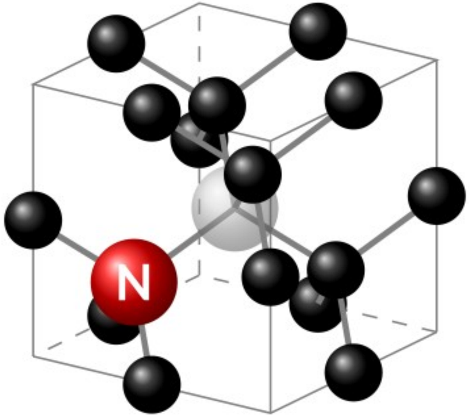
Scanning probe magnetometry technique = we measure **maps of the magnetic stray field** produced by a sample.



Our B field sensor

The nitrogen vacancy (NV) defect in diamond

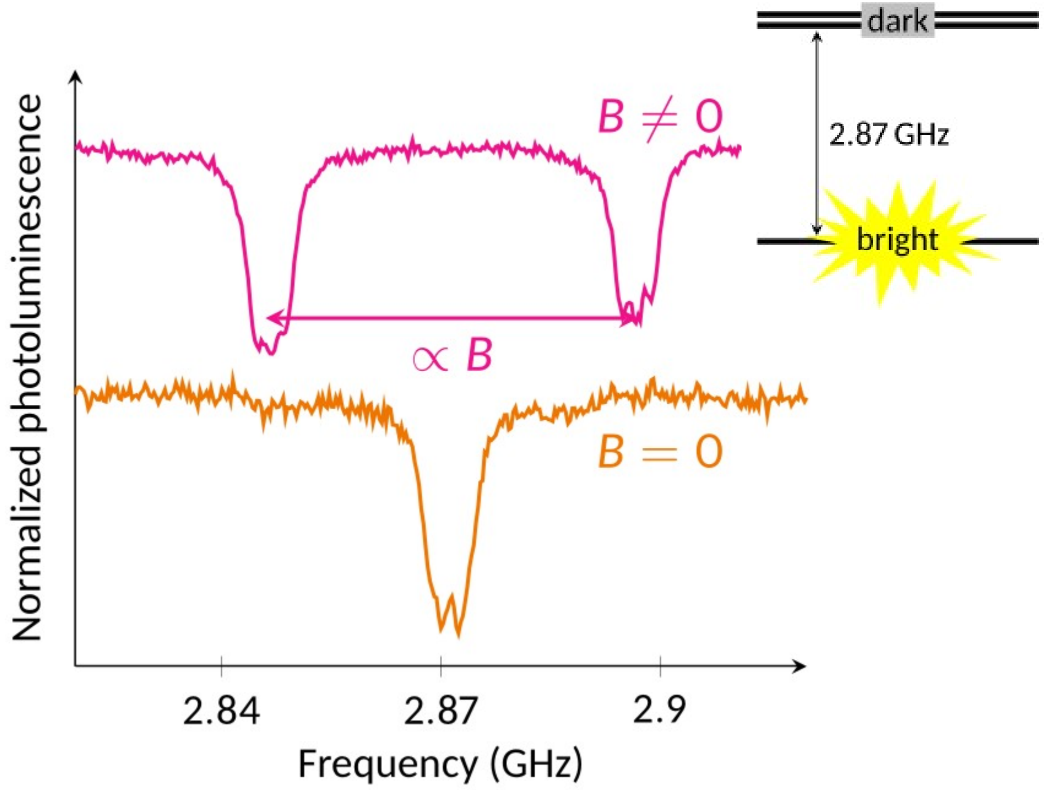




We have tips made of diamond with a single defect at the end

Basic principle of the measurement

Optically detected magnetic resonance



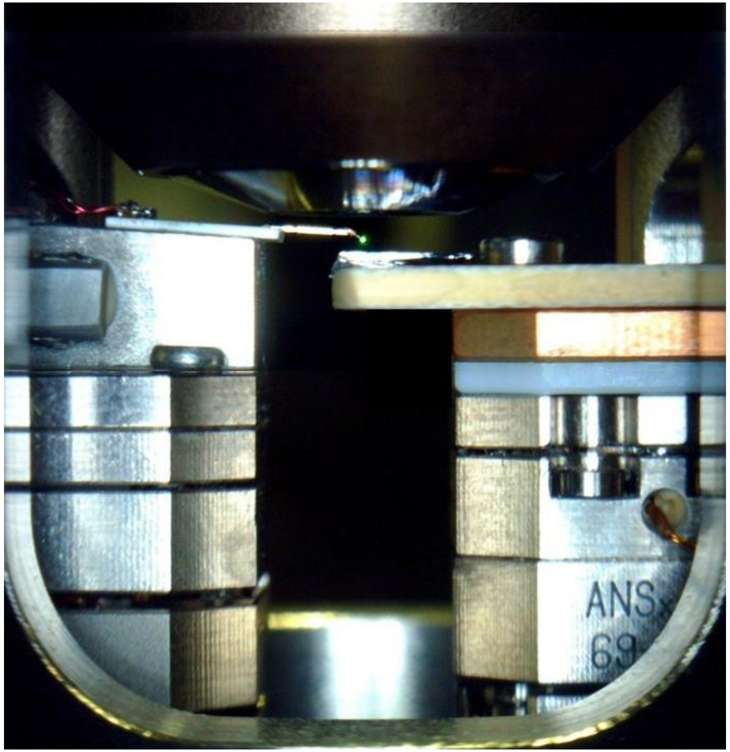
At each pixel of the map :

i. Sweep the MW excitation frequency (actuator)
ii. Record the photoluminescence at the same time (detector)
iii. Extract the position of the resonance
iv. Obtain the value of B
v. Move to the next pixel (actuator)

Now a commercial technique

Qnami ProteusQ

Qzabre QSM



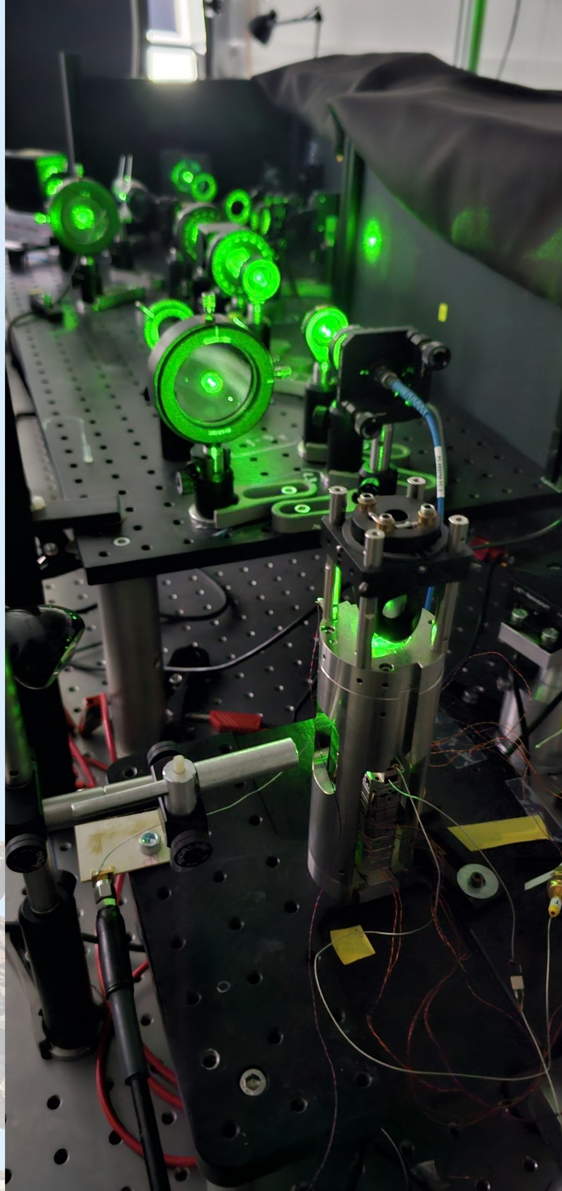
However, we also have two custom-built systems based on Attocube AFMs, including a cryogenic setup

→They need a control software!

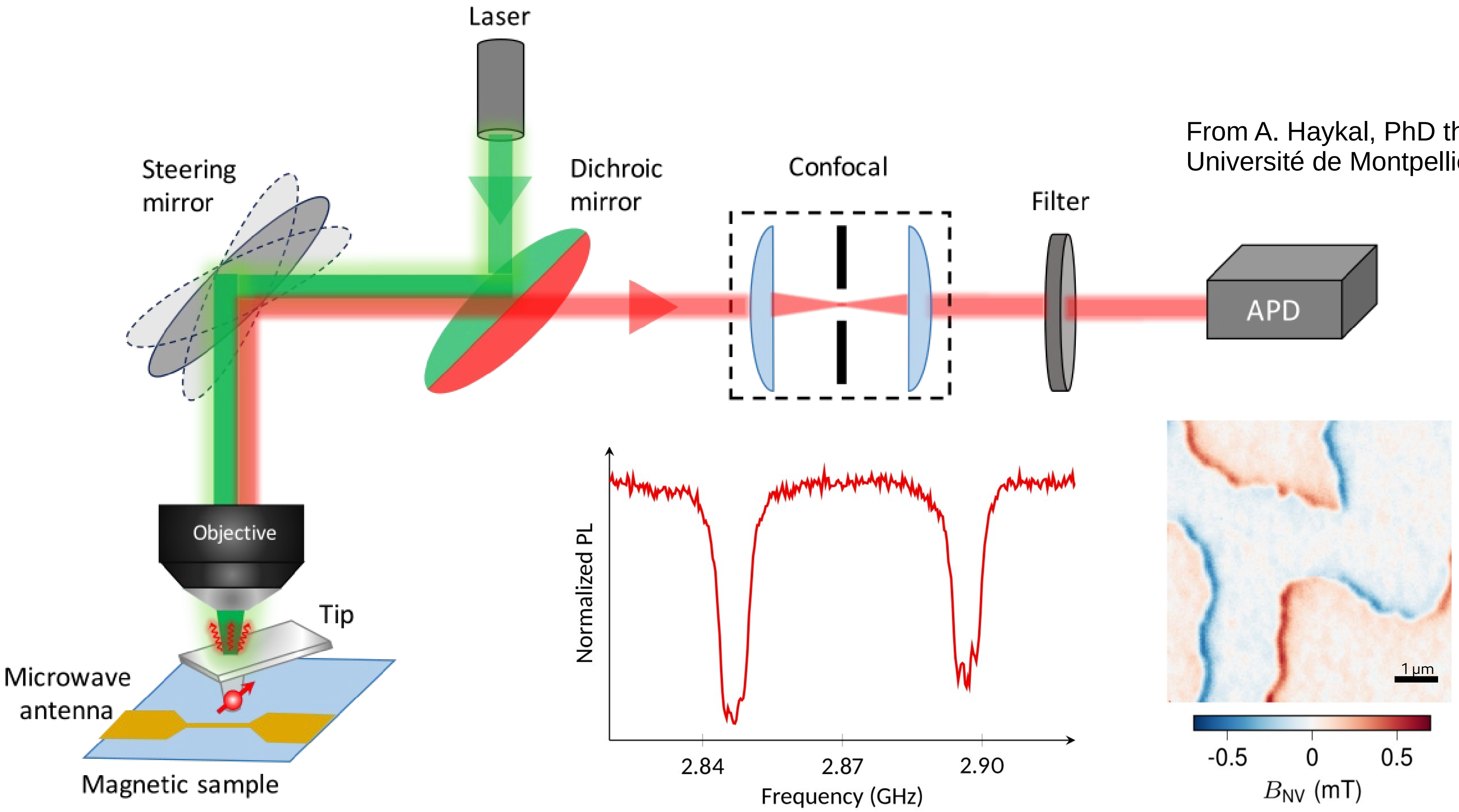
PyMoDAQ
Modular Data Acquisition with Python

Outline

1. Our experimental setup
2. Welcome to our dashboard
3. Review of our PyMoDAQ modules
   - Newport Picomotor 8742
   - Rohde Schwarz SMB100A
   - NI DAQmx controlled scanners
   - NI DAQmx based single photon counter
   - Our custom ODMR detector
4. Demo



Our experimental setup



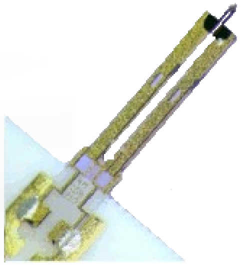
From A. Haykal, PhD thesis,
Université de Montpellier (2020)

Our experimental setup

Actuators
Detectors
Can do both

Atomic force microscope

Attocube scanners and steppers



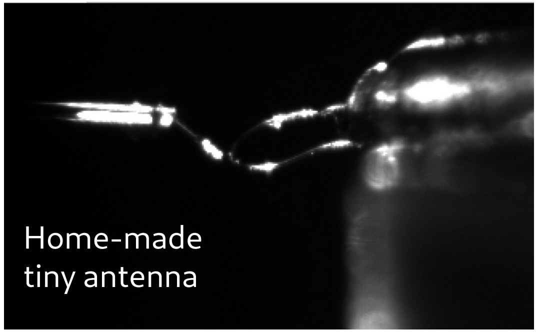
Akiyama probe

Optical confocal microscope

Single photon counter (APD)

Acousto-optic modulator

Microwave excitation



Attocube ASC500 controller

NIDAQmx card

Steering mirror

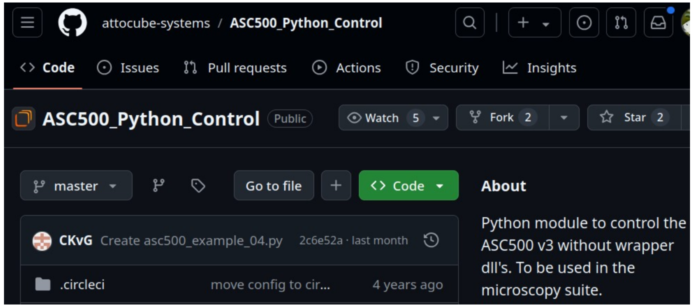
Rohde Schwarz SMB100A MW source

Picomotor motion controller

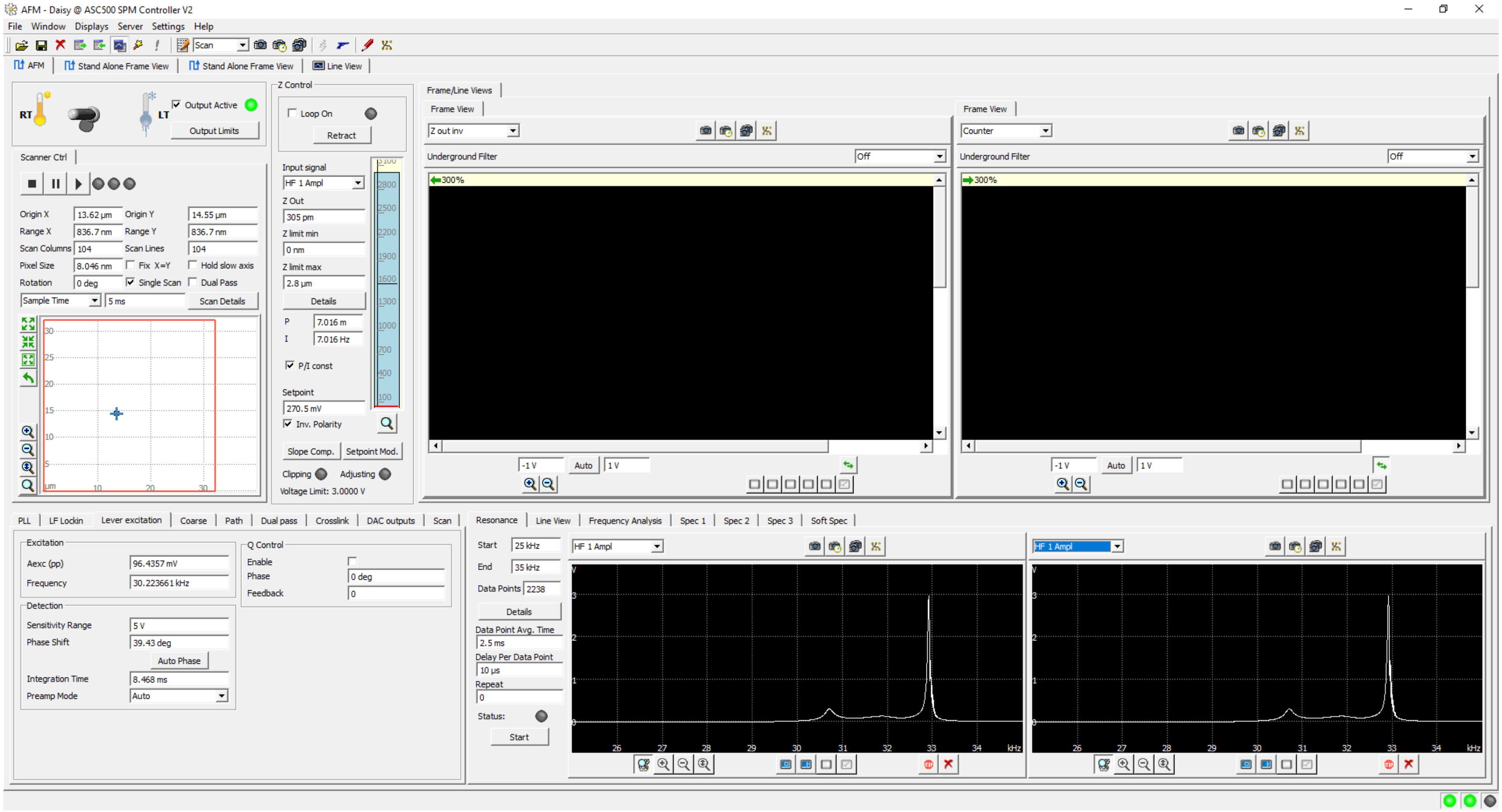
The atomic force microscope

- Steppers for coarse positioning: manual control
- Scanners :
  - X, Y scanners driven with the analog output of the NI card
  - Z scanner is driven by the Attocube AFM controller which takes care of the feedback loop
- Mostly actuators, but we also read the topography data from the AFM controller through the NI card

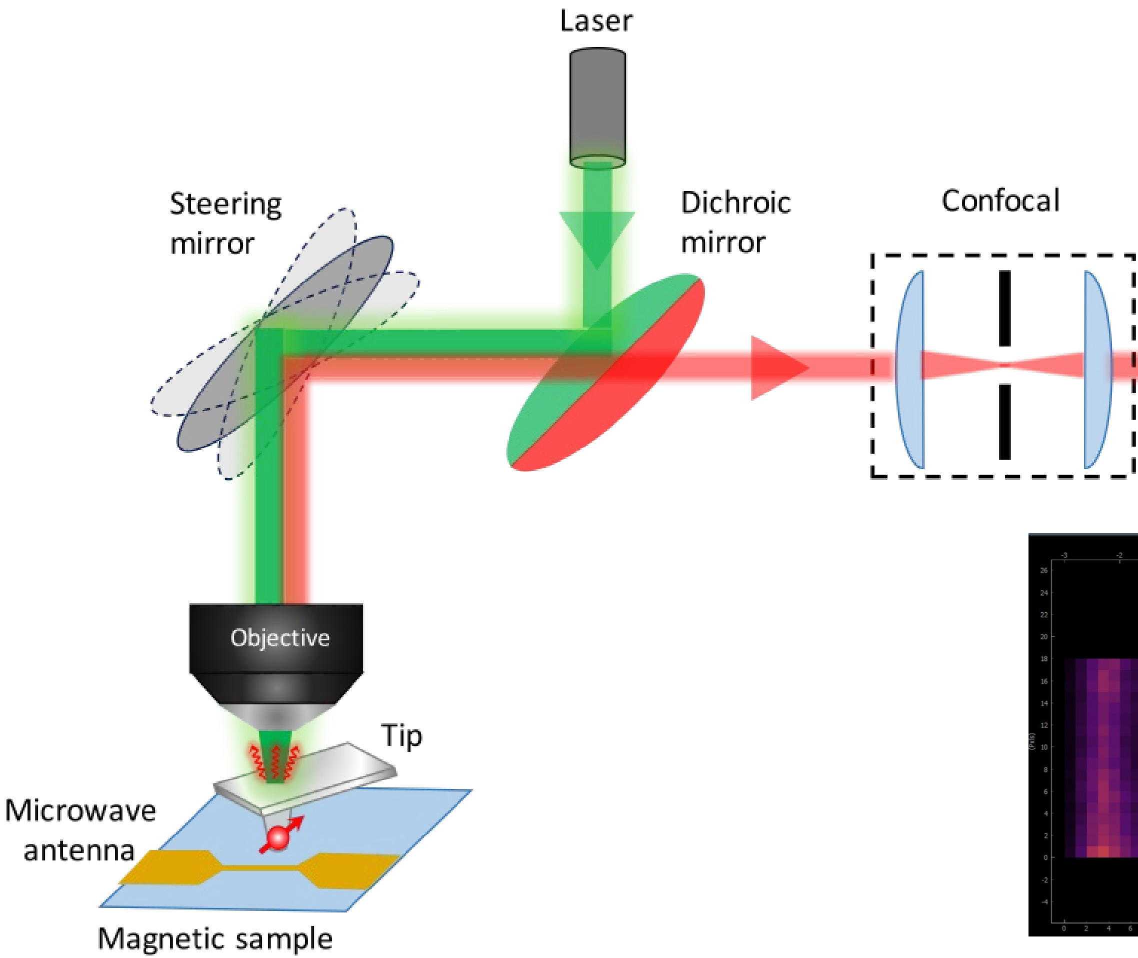
Plan for the future: control the sample XY scanners with the Attocube scan controller ASC500 and PyMoDAQ



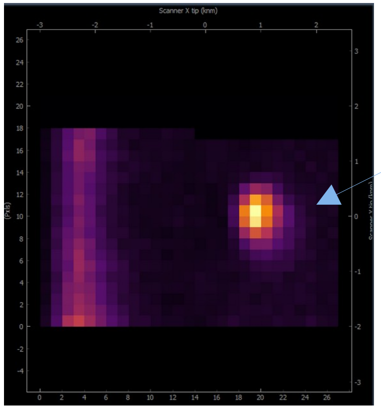
Daisy : Attocube control software for the AFM



The confocal microscope



From A. Haykal, PhD thesis,
Université de Montpellier (2020)



NV center

The confocal microscope control and detection

Scan the laser beam to adjust the position of the NV defect at the focal point

Generate laser pulses to manipulate the NV defect

Detect the emitted photons (usually about 100 kcts/s)

Count the voltage pulses from the APD and control the mirror's position

Steering mirror

Acousto-optic modulator

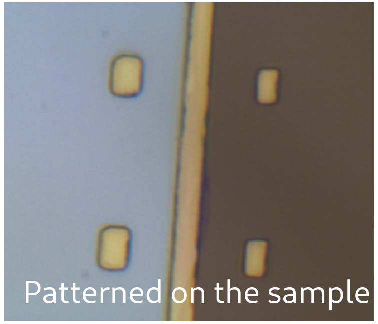
Single photon counter (APD)

NIDAQmx card

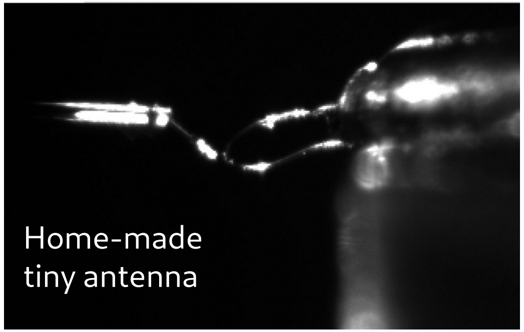
PyMoDAQ
Modular Data Acquisition with Python

The microwave excitation

Rohde Schwarz SMB100A
MW source



or



PyMoDAQ
Modular Data Acquisition with Python

Trigger to synchronize
with photon detection

NIDAQmx
card

Picomotor
motion
controller

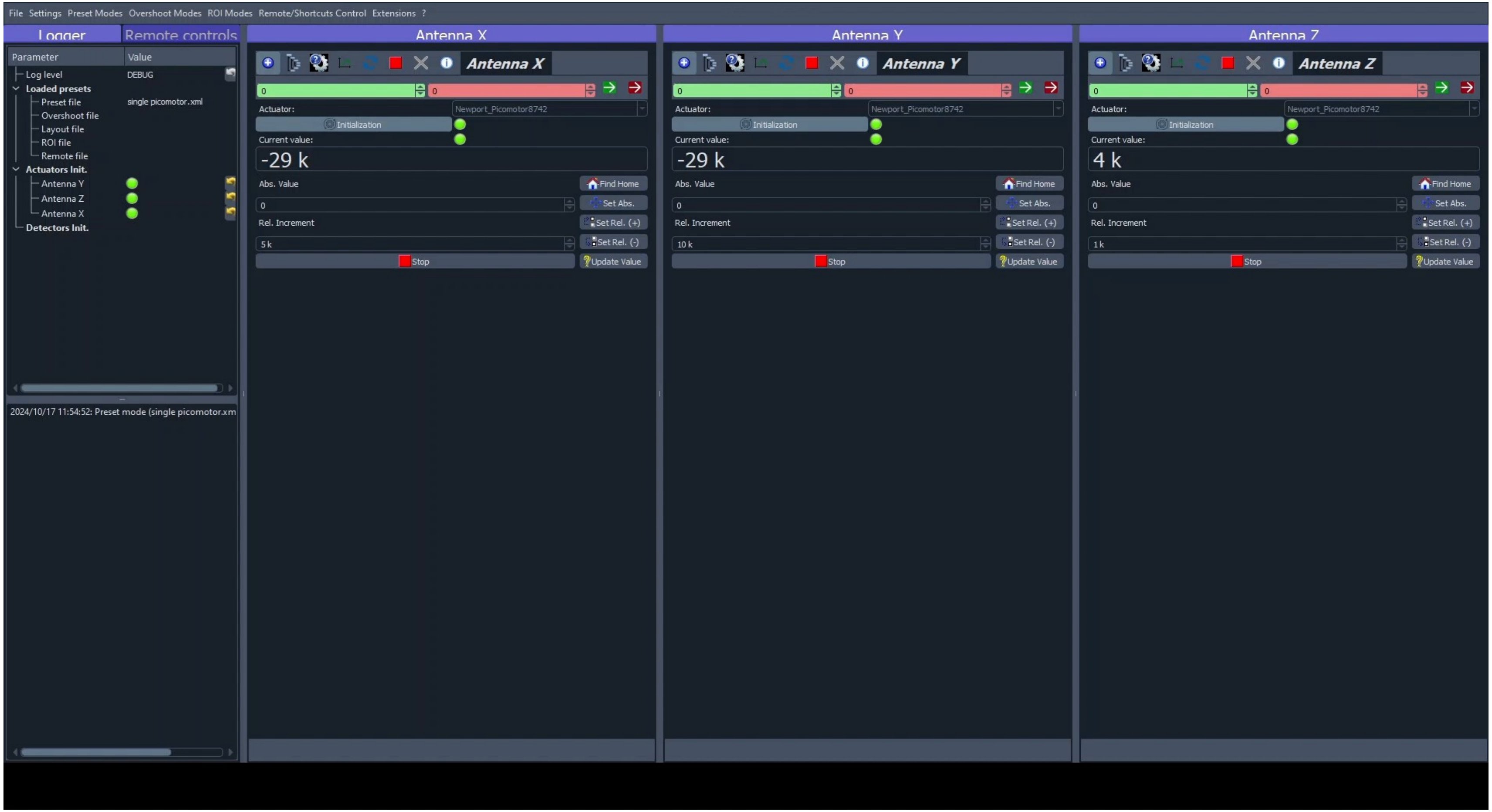
PyMoDAQ
Modular Data Acquisition with Python

Positioning with low risk
to break the precious tip

Welcome to our dashboard!

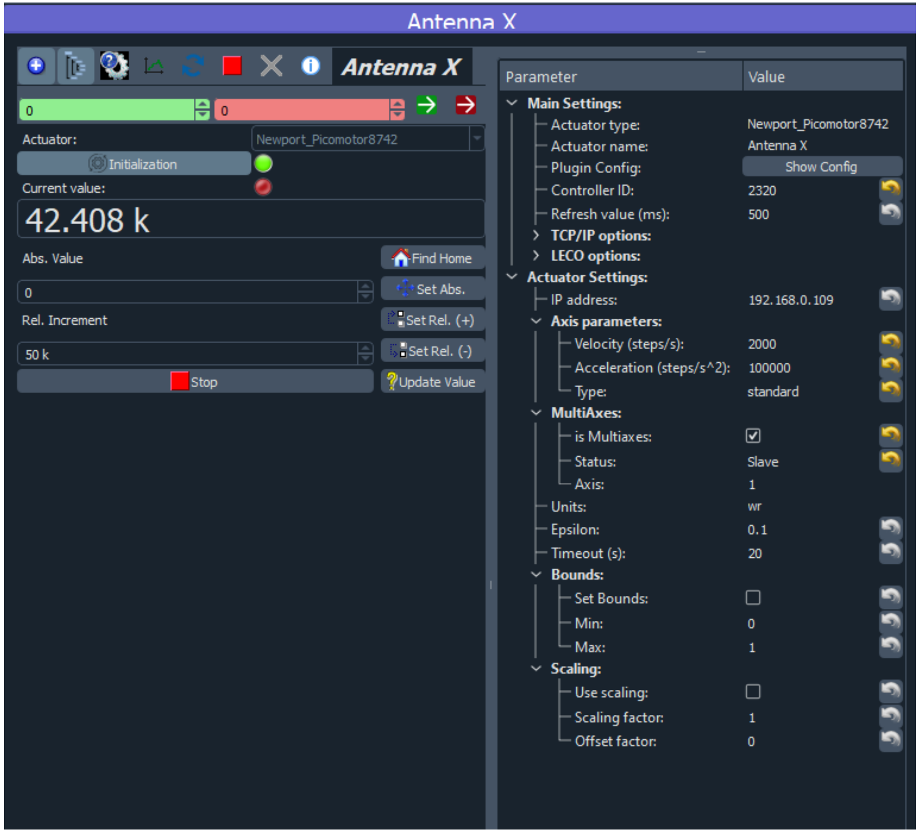

PyMoDAQ Dashboard: Balbuzard_octobre2024_4_4
File Settings Preset Modes Overshoot Modes ROI Modes Remote/Shortcuts Control Extensions ?
Scanner Y tip
Logger
Remote controls
MW source
Scanner X tip
ODMR settings
ODMR ODMR_fit
ODMR Topo
Scanner Y tip
Actuator:
DAQmx_MultipleScannerControl
Initialization
Current value:
0 m
MW source
RSMWsource
0
Scanner Y sample
Scanner Y sample
15 µm
Scanner X sample
Scanner X sample
15 µm
ODMR
DAQ1D : ODMR
DAQ type:
DAQ1D
Init. Detector
Detector:
ODMR
Do Bkg
Take Bkg
Parameter
Value
Main Settings:
Detector Settings
Controller Status:
Master
MW source settings
Address:
TCPIP0::
Power (dBm):
20
Counter settings:
Count time (ms):
10
Counting channel:
Dev1/ctr(
Source settings:
Enable?:
Photon source:
/Dev1/PF
Acquisition settings
Sweep mode?
Number of ranges
1
Range 1 parameters
Start (MHz):
2.82e+0:
Stop (MHz):
2.86e+0:
Step (MHz):
Dual iso-B mode?
Further NI card settings
Clock channel:
Dev1/ctr
Topo channel:
Dev1/ai0
Sync trigger channel:
/Dev1/PF
Fit Settings
Do fit?
Plot init fit?
Fitting function
Single lor
Reference Frequency
2.87e+0:
Fit results
PL = 107.98 kCts/s
Position = 2836.51 MHz
B = 1.20 mT
Raw_data
Pymofit
Frequency (GHz)
PL counter settings
PL counter
DAQ0D : DAQmx_PLcounter
DAQ type:
DAQ0D
Detector:
DAQmx_PLcounter
Do Bkg
Take Bkg
PL counter PL
PL (kcts/s)
Samples (kS)
1.110e+02

Review of modules: Newport picomotor 8742



Review of modules: Newport picomotor 8742

`pymodaq_plugins_newport`

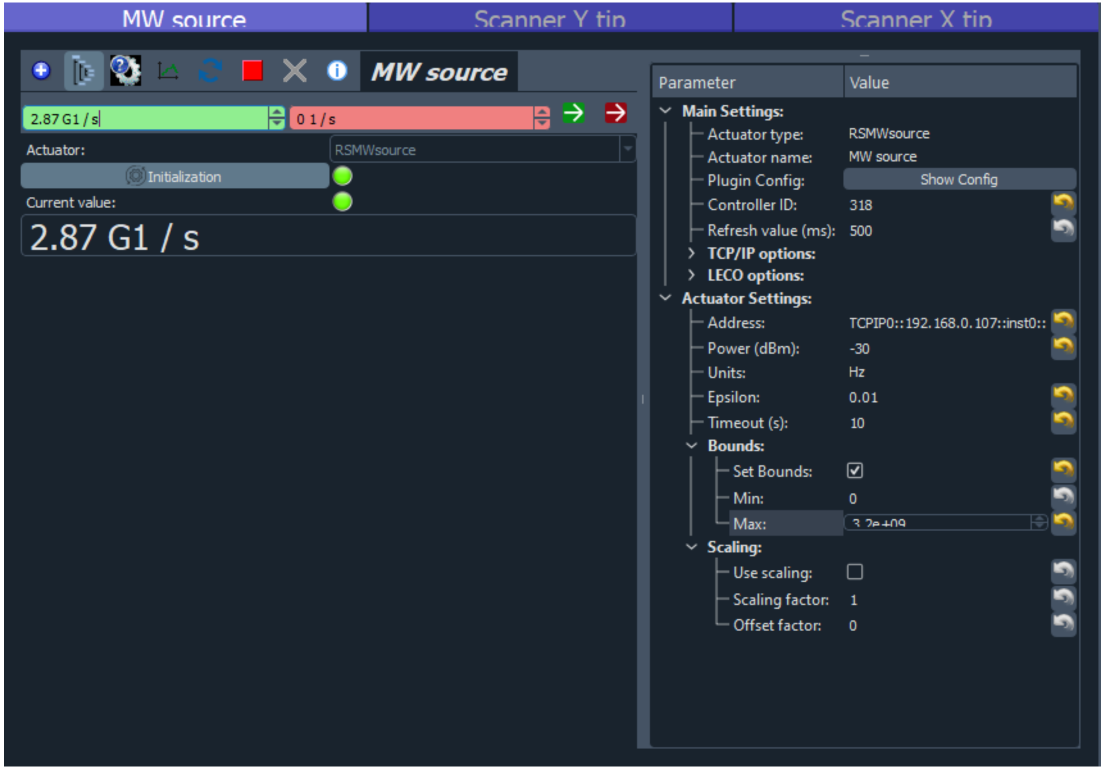


`DAQ_Move_Newport_Picomotor8742`

- We use the pylablib driver
- There are 4 axis on the controller, we have to use the Master/Slave mechanism
- Issue: the driver opens the communication when you init the wrapper, which created troubles with the old template
- Solved with the latest version of the template
- **Tip: check the current template version when you have an issue with your modules!**

Review of modules: Rohde Schwarz SMB100A

`pymodaq_plugins_rohdeschwarz`

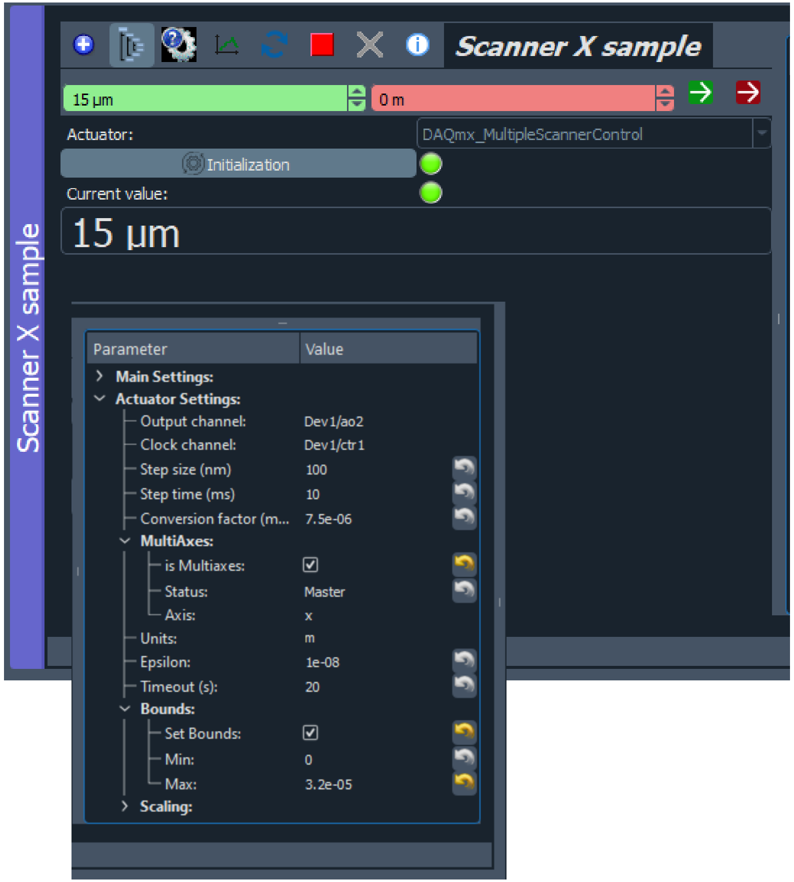


`DAQ_Move_RSMWsource`

- Hardware communication file adapted from qudi
- We choose that the controlled parameter is the frequency, and you set the power in the settings tree, but you could write a plugin with reverse config
- The unit is weird!
- Main issues come from the device itself, which is not so simple to control for the ODMR later

Review of modules: NIDAQmx controlled scanners

`pymodaq_plugins_daqmx`

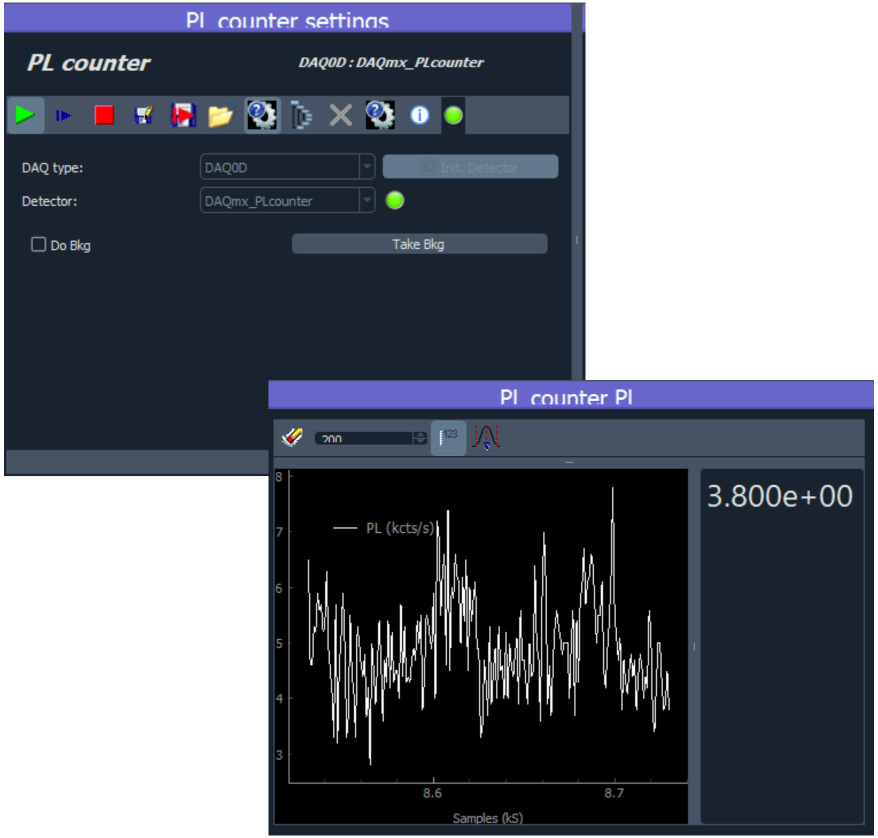


`DAQ_Move_DAQmx_MultipleScannerControl`

- Sends analog voltages to any scanners (here Attocube scanners and Newport steering mirror)
- Troubles arrive when we start using the DAQ_scan (cf last year's talk) but the plugin now works (almost) properly
- If you plan to use several scanners together in a scan (x and y for example), they should share a controller (use Master/Slave) for proper timing
- We could get rid of the knm! :)

Review of modules: NIDAQmx based photon counter

pymodaq_plugins_daqmx



DAQ_0DViewer_DAQmx_PLcounter

- Counts the voltage pulses sent by the APD
- Naive counting with simple daqmx functions did not work properly, the number of photons were not correct
- We copied what was done in qudi, using a SemiPeriod Counter Input channel and summing the counts during both halves of the period
- We had to add a clock for the timing

Review of modules: Our custom ODMR detector

pymodaq_plugins_s2qt_odmr

DAQ_1DViewer_ODMR

- Our controller consists in 3 DAQmx objects (counter, clock, analog input) and a MWsource.
- Multiple output:
  - ODMR spectrum (1D)
  - Topography = position of the z scanner (0D)
  - B value, iso-B data (0D)

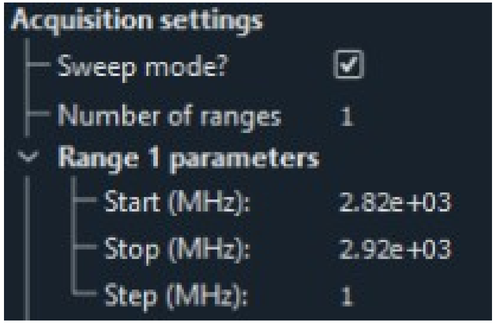
Working principle

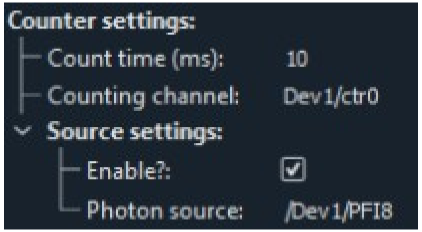
i. Define a list of frequencies and an acquisition time.

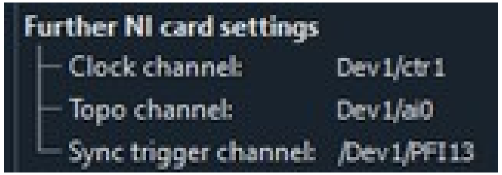
ii. Send the list to the MW source, and set up a clock with the proper timing

iii. Define in the NI card a task to read the PL, read the topo and trigger the MW source following this clock

iv. Send the data to the viewer.





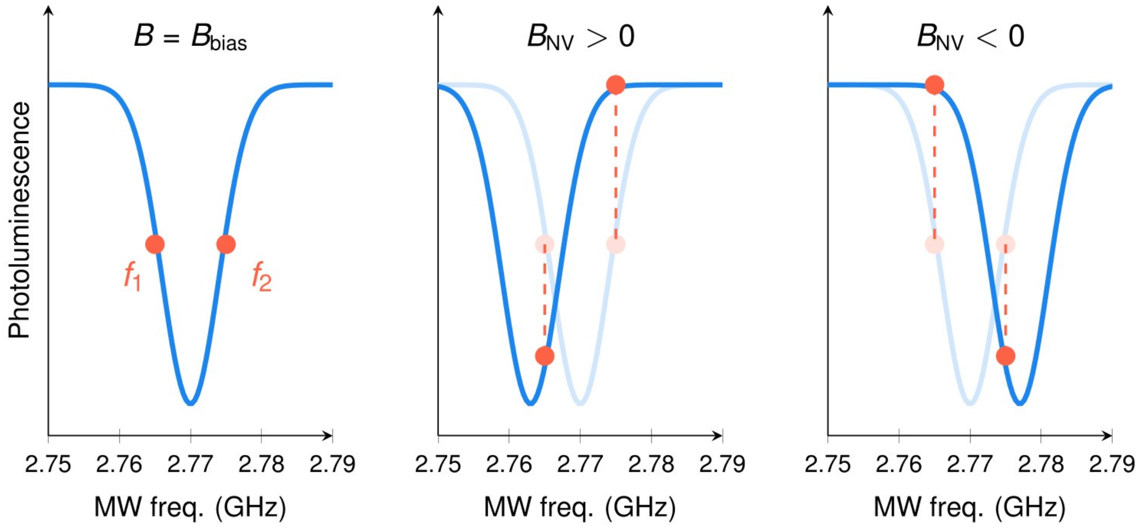


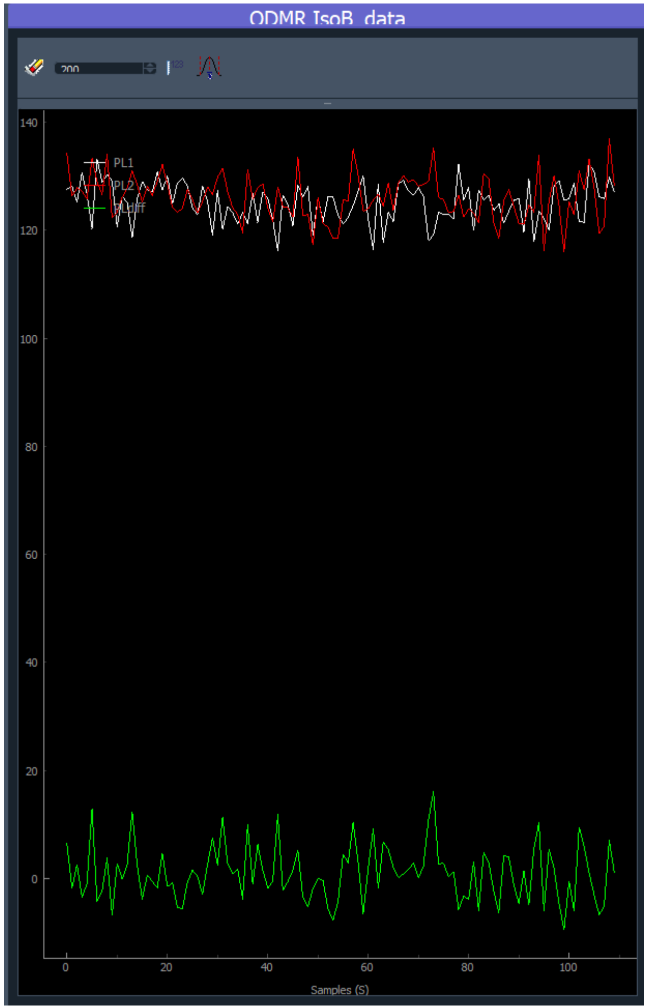
Fitting

- We need to fit our spectra to optimize our measurements (contrast, linewidth)
- Several functions available: lorentzian and gaussian fits, single or double.
- We want to fit the averaged signal! No easy access to it from inside the DAQ_viewer plugin
  → dirty workaround for now, ideas?

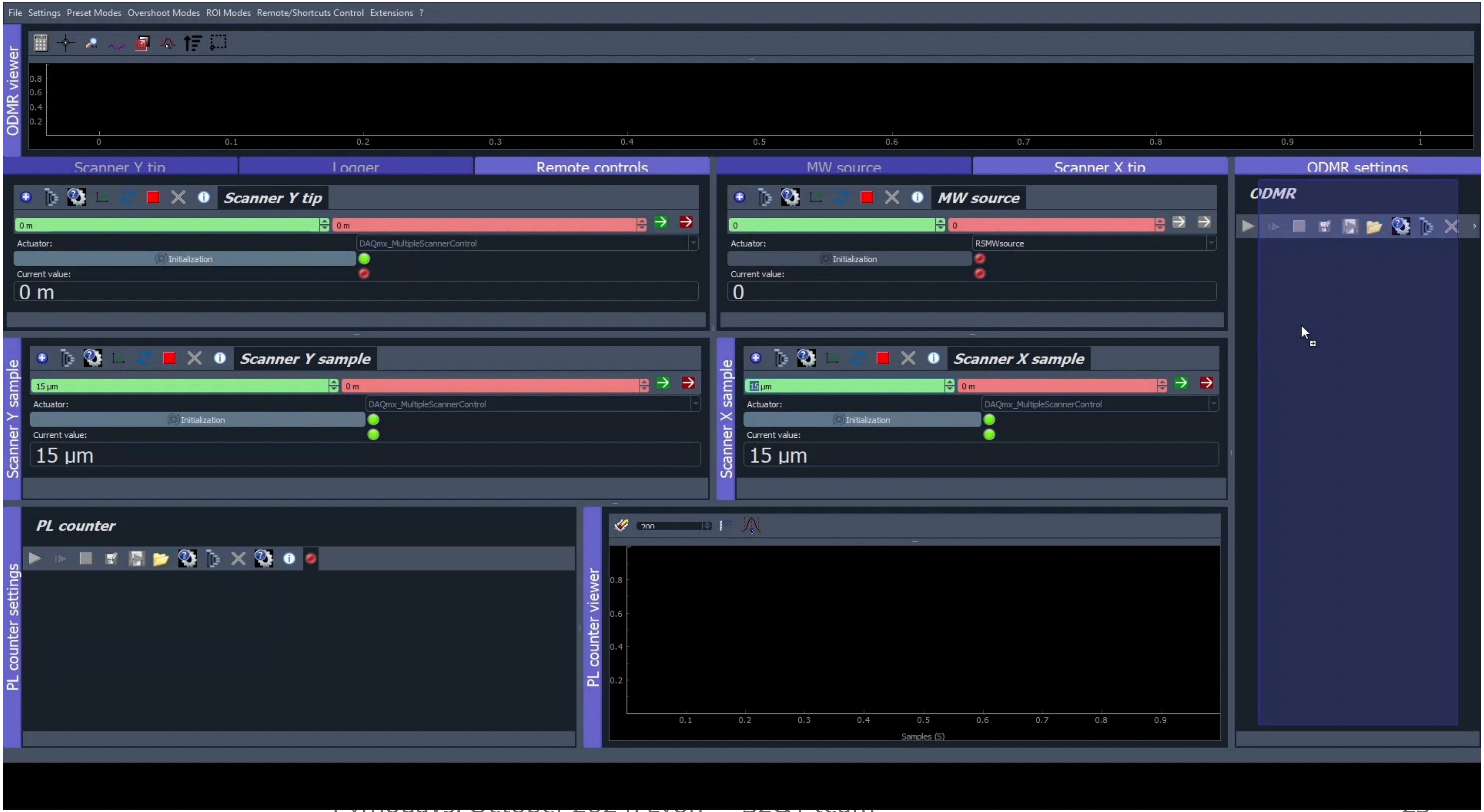
Iso-B mode

Measure the signal only at 2 MW frequencies
to go faster, but get a qualitative map





Demonstration!



File Settings Preset Modes Overshoot Modes ROI Modes Remote/Shortcuts Control Extensions ?
ODMR viewer
Scanner Y tip
Logger
Remote controls
MW source
Scanner X tip
ODMR settings
Scanner Y tip
Actuator:
DAQmx_MultipleScannerControl
Initialization
Current value:
0 m
MW source
RSMWsource
0
ODMR
Scanner Y sample
Scanner Y sample
15 µm
Scanner X sample
Scanner X sample
PL counter settings
PL counter
PL counter viewer
Samples (S)

The future

- Use the PID extension to readjust the position of the beam on the NV center
- Create a Pulser extension to generate sequences of laser and MW pulses allowing us to control the spin of the NV defect
- Update and improve our data analysis codes with the help of the PyMoDAQ data structure (hello PyMoDAQ v5!)

Our repositories

- https://github.com/Montpellier-S2QT/pymodaq_plugins_s2qt_odmr
- https://github.com/Montpellier-S2QT/pymodaq_plugins_newport
- https://github.com/Montpellier-S2QT/pymodaq_plugins_rohdeschwarz
- https://github.com/Montpellier-S2QT/pymodaq_plugins_daqmx