

Pymodaq for scanning NV center microscopy

S2QT team, Laboratoire Charles Coulomb, Montpellier

Elias Sfeir, Roméo Beignon, Elijah Wane, Aurore Finco

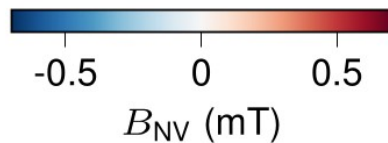
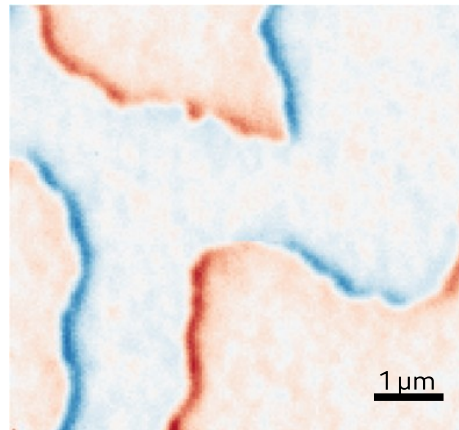


UNIVERSITÉ DE
MONTPELLIER

Pymoday, October 22nd 2024, Lyon

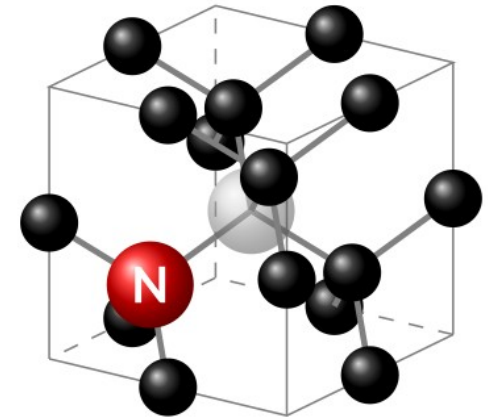
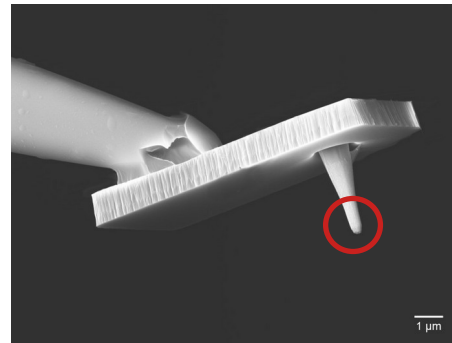
Scanning NV center microscopy

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



Our B field sensor

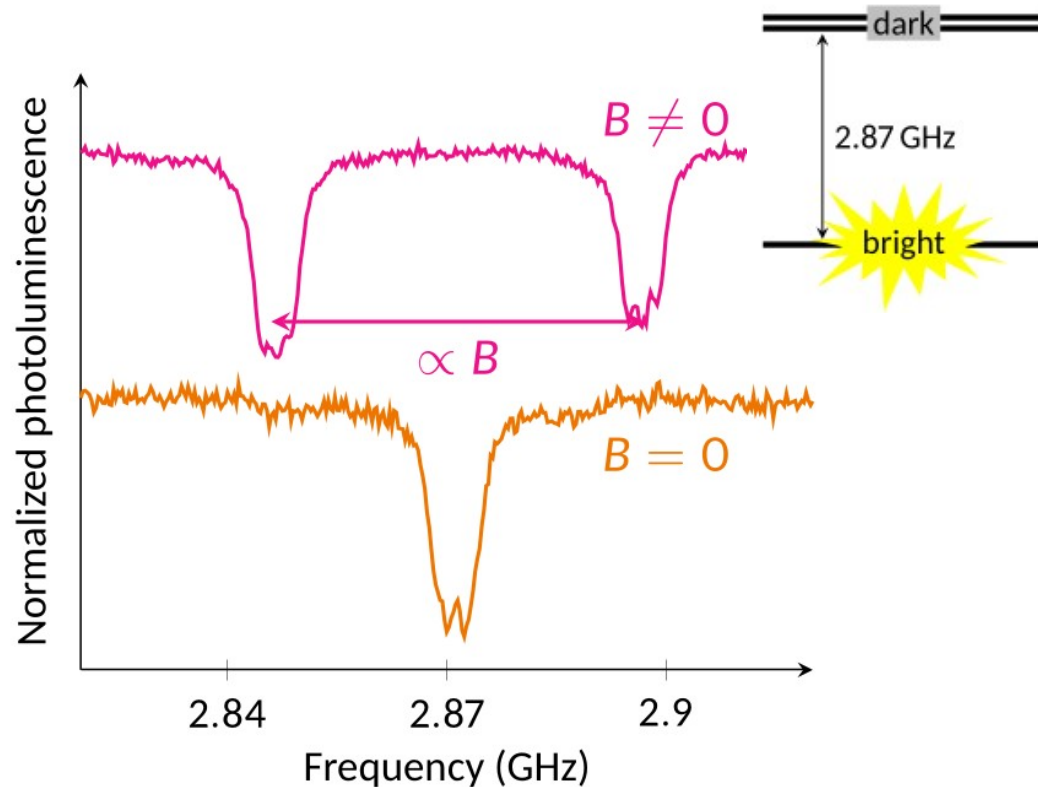
The nitrogen vacancy (NV) defect in diamond



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

Basic principle of the measurement

Optically detected magnetic resonance



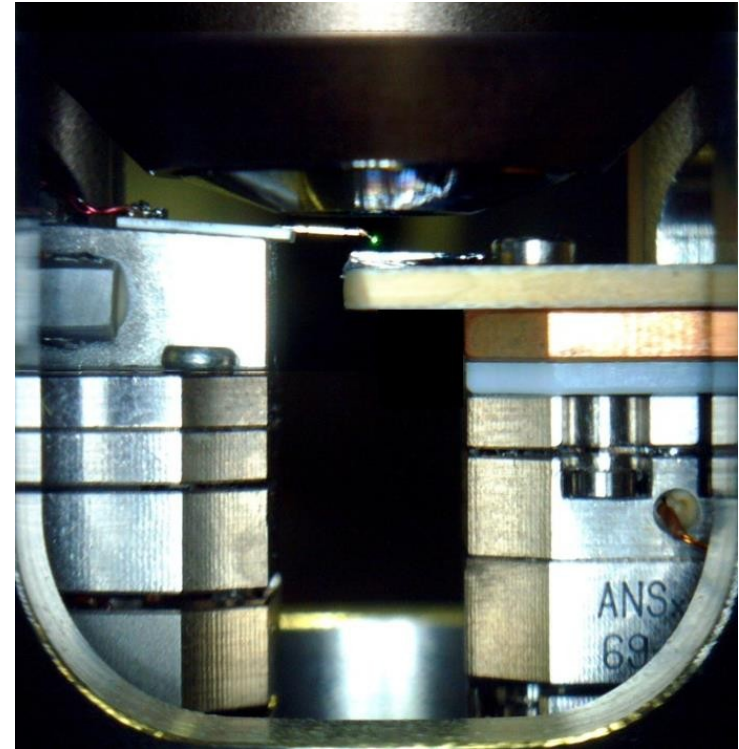
At each pixel of the map :

- 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

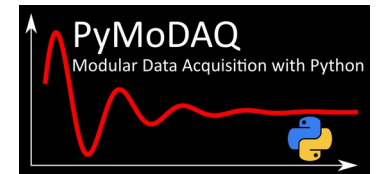


Qnami ProteusQ



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

→ They need a control software!



Outline

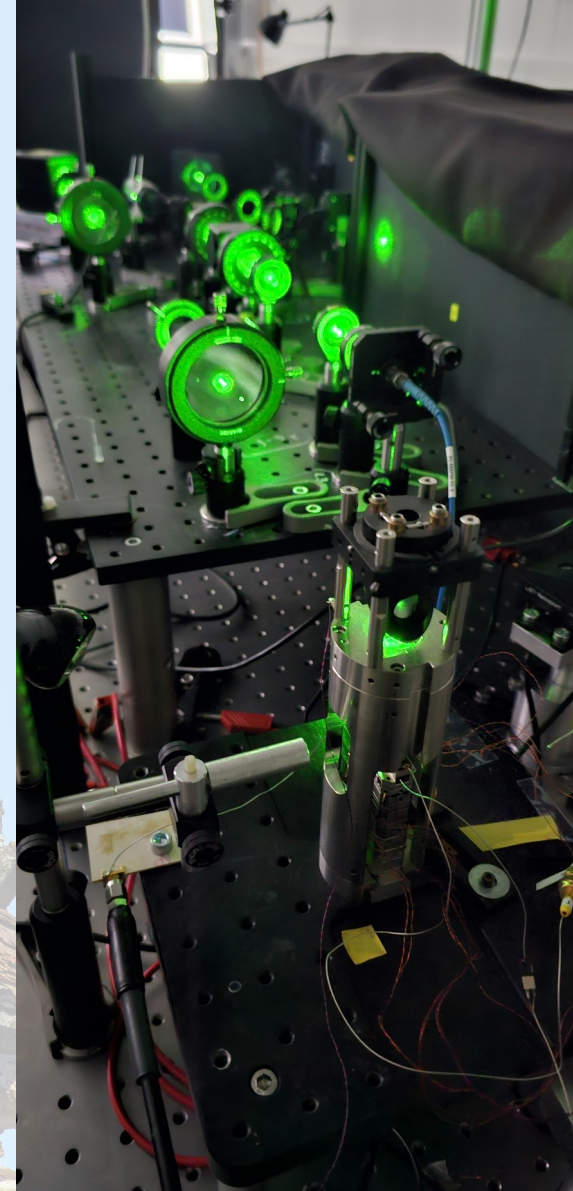
1. Our experimental setup

2. Welcome to our dashboard

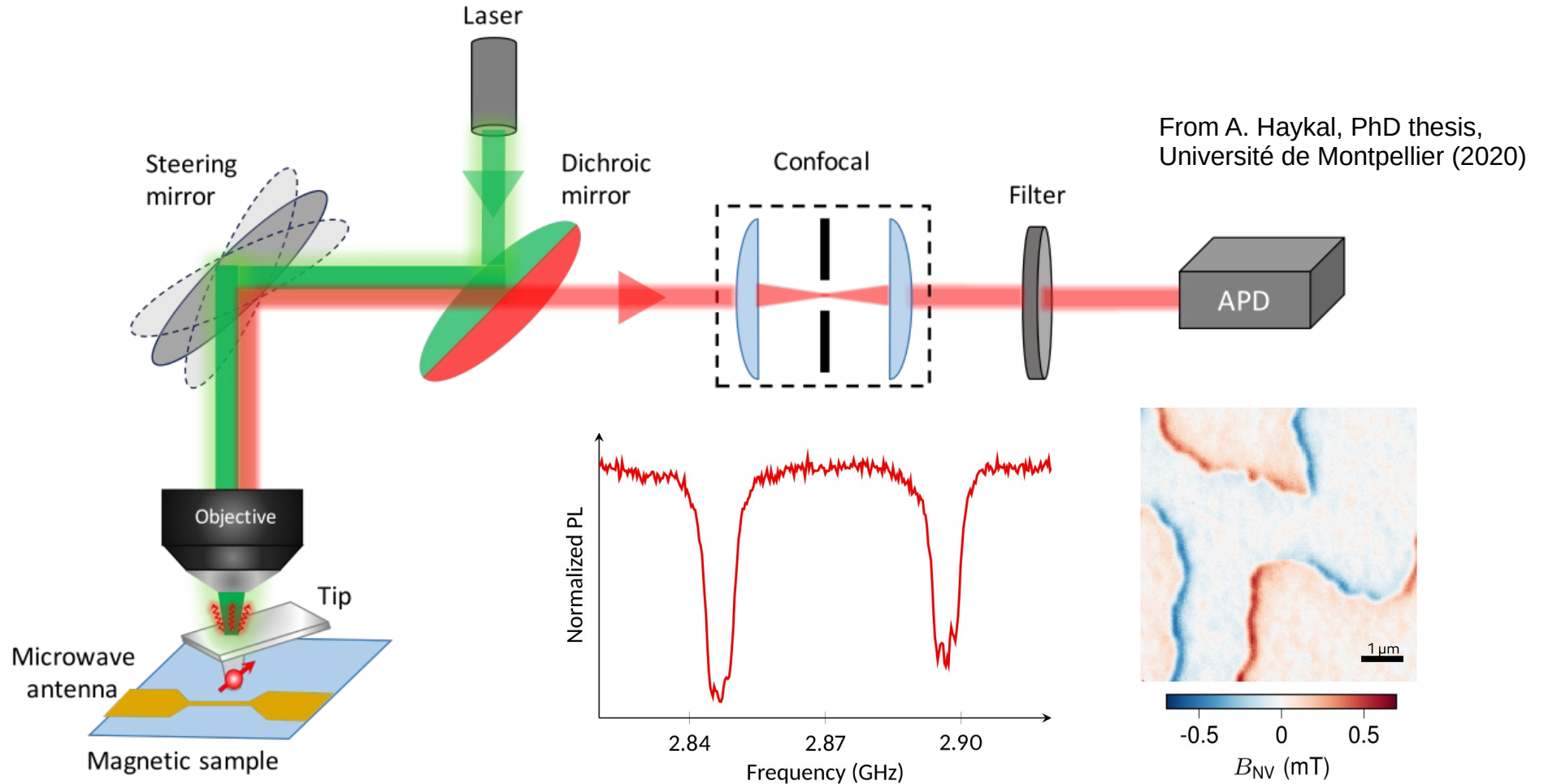
3. Review of our PyMoDAQ modules

- 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



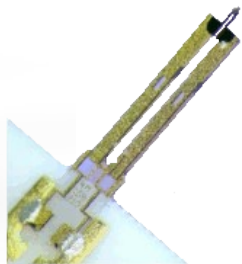
Our experimental setup

Actuators
Detectors
Can do both

Atomic force microscope



Attocube
scanners and
steppers



Akiyama
probe

Optical confocal microscope

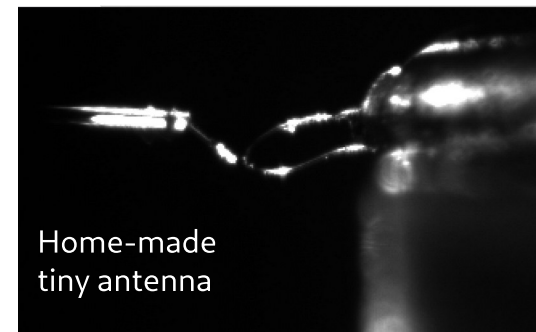


Single photon
counter (APD)



Acousto-optic
modulator

Microwave excitation



Home-made
tiny antenna



Attocube ASC500 controller



NIDAQmx
card



Steering
mirror



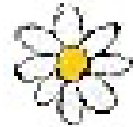
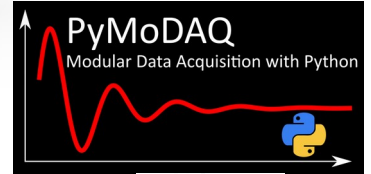
Rohde Schwarz SMB100A
MW source



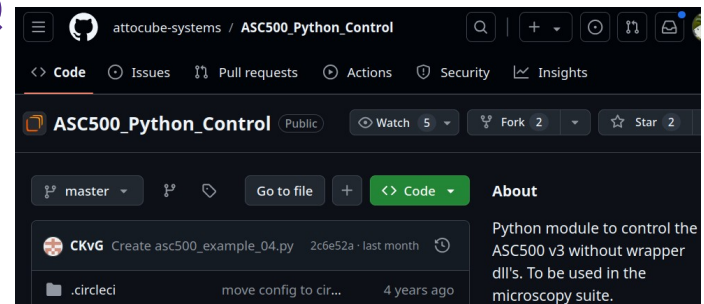
Picomotor
motion
controller

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

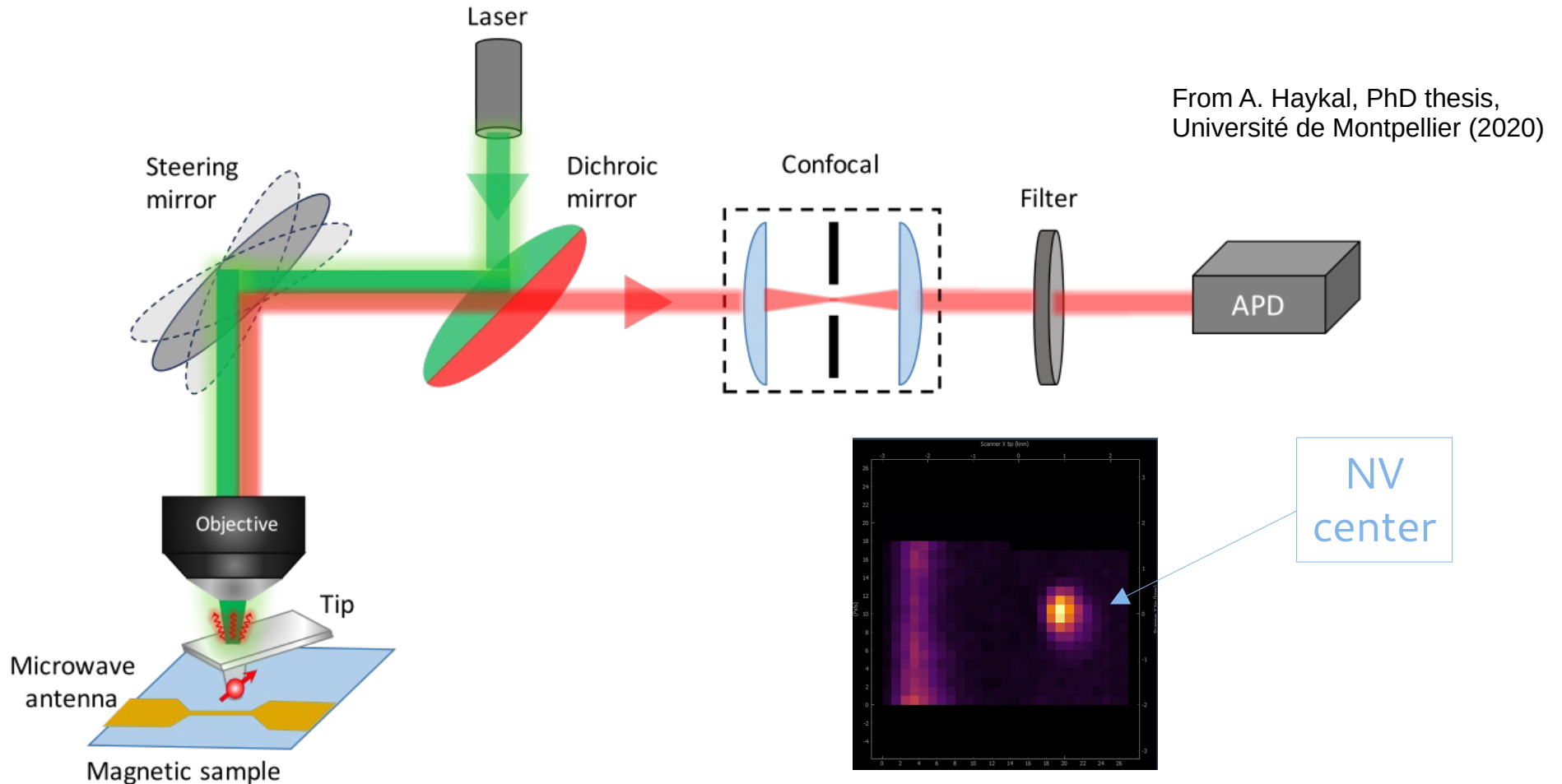


Daisy : Attocube control software for the AFM

The screenshot displays the Daisy AFM control software interface, titled "AFM - Daisy @ ASC500 SPM Controller V2". The interface is divided into several functional areas:

- Top Panel:** Includes a menu bar (File, Window, Displays, Server, Settings, Help) and a toolbar with icons for scan, zoom, and other functions.
- Left Panel (Scanner Ctrl):** Contains controls for the scanner, including "RT" and "LT" indicators, "Output Active" status, and "Output Limits". It also features a "Scanner Ctrl" section with play/pause buttons and a grid for setting scan parameters.
- Top-Right Panel (Z Control):** Shows "Z Control" settings, including "Loop On" status, "Retract" button, "Input signal" (HF 1 Ampl), "Z Out" (305 pm), "Z limit min" (0 nm), "Z limit max" (2.8 μ m), "P" (7.016 m), "I" (7.016 Hz), "P/I const" checkbox, "Setpoint" (270.5 mV), "Inv. Polarity" checkbox, "Slope Comp." and "Setpoint Mod." buttons, and "Clipping" and "Adjusting" indicators. A vertical scale on the right ranges from 0 to 1800.
- Middle-Right Panel (Frame/Line Views):** Displays two "Frame View" windows. The left window shows "Z out inv" and "Underground Filter" (Off). The right window shows "Counter" and "Underground Filter" (Off). Both windows have a "300%" zoom level and a "-1V Auto 1V" control.
- Bottom-Left Panel (Excitation and Q Control):** Contains "Excitation" settings (Aexc: 96.4357 mV, Frequency: 30.223661 kHz) and "Q Control" settings (Enable, Phase: 0 deg, Feedback: 0).
- Bottom-Right Panel (Resonance and Frequency Analysis):** Shows "Resonance" and "Line View" plots. The "Resonance" plot displays a sharp peak at approximately 33 kHz. The "Line View" plot shows a similar peak. Both plots have a "HF 1 Ampl" input and a "Start" button.

The confocal microscope



The confocal microscope control and detection

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



Steering mirror

Generate laser pulses to manipulate the NV defect



Acousto-optic modulator

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

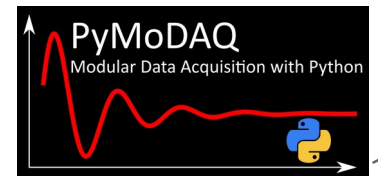


Single photon counter (APD)

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



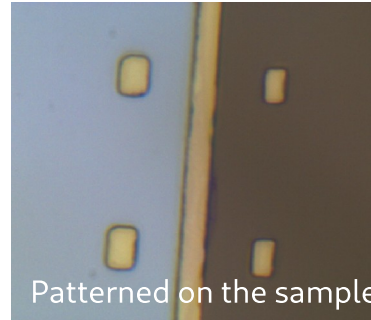
NIDAQmx card



The microwave excitation

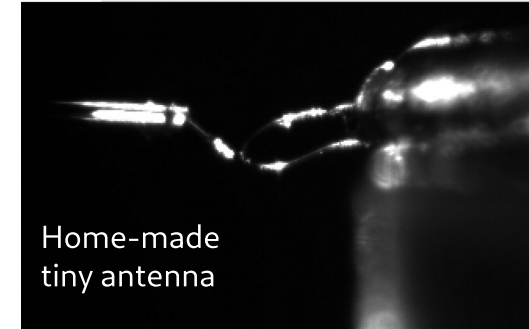


Rohde Schwarz SMB100A
MW source



Patterned on the sample

or



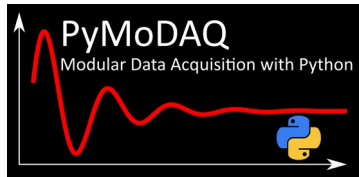
Home-made
tiny antenna



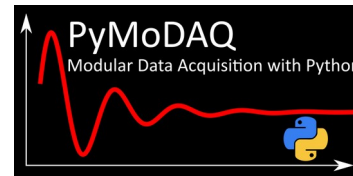
Picomotor
motion
controller



Trigger to synchronize
with photon detection



NIDAQmx
card



Positioning with low risk
to break the precious tip

Welcome to our dashboard!

Scanner Y tin

0 m

Actuator: DAQmx_MultipleScannerControl

Current value: 0 m

MW source

0

Actuator: RSMWSource

Current value: 0

Scanner Y sample

15 μ m

Actuator: DAQmx_MultipleScannerControl

Current value: 15 μ m

Scanner X sample

15 μ m

Actuator: DAQmx_MultipleScannerControl

Current value: 15 μ m

PL counter settings

PL counter

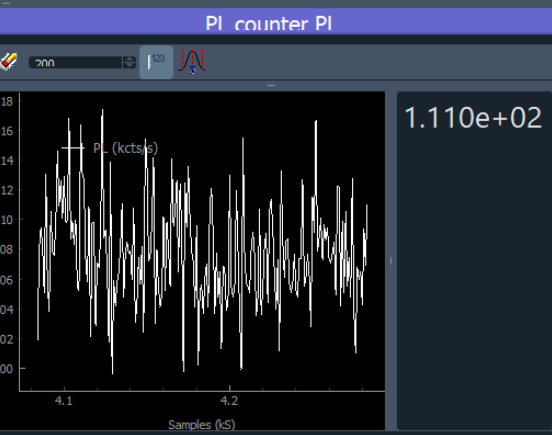
DAQ00 : DAQmx_PLcounter

DAQ type: DAQ00

Detector: DAQmx_PLcounter

Do Bkg

Take Bkg



ODMR settings

ODMR

DAQ ID : ODMR

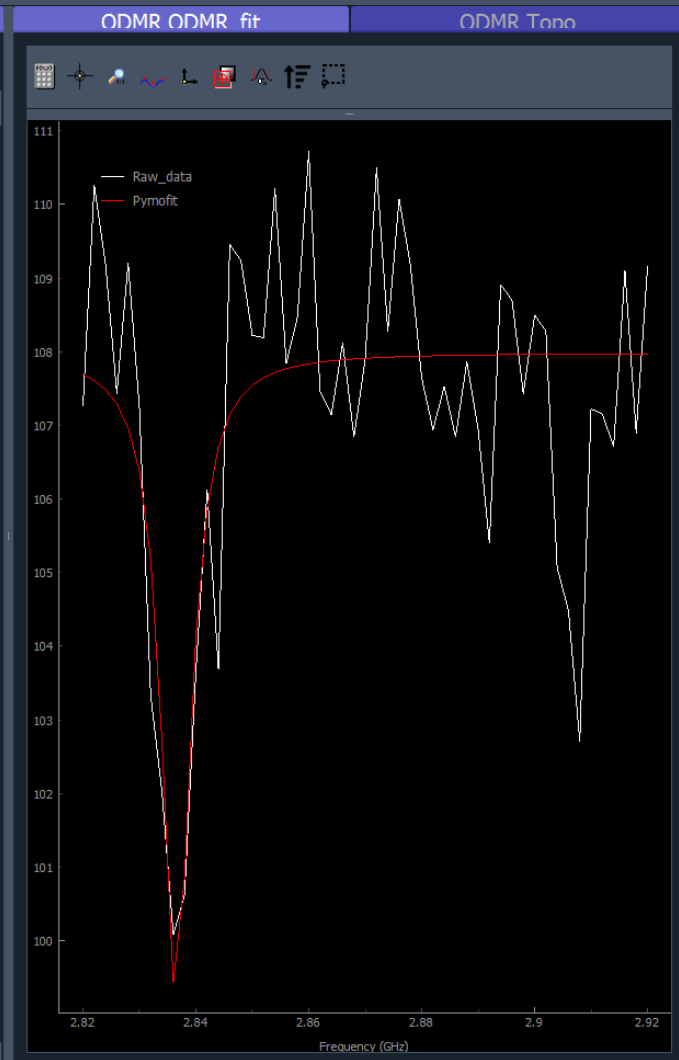
DAQ type: DAQ ID

Detector: ODMR

Do Bkg

Take Bkg

Parameter	Value
Main Settings:	
Detector Settings	
Controller Status:	Master
MW source settings	
Address:	TCP/IP0:
Power (dBm):	20
Counter settings:	
Count time (ms):	10
Counting channel:	Dev 1/ctrl
Source settings:	
Enable?:	<input checked="" type="checkbox"/>
Photon source:	/Dev 1/PF
Acquisition settings	
Sweep mode?:	<input checked="" type="checkbox"/>
Number of ranges:	1
Range 1 parameters	
Start (MHz):	2.82e+00
Stop (MHz):	2.86e+00
Step (MHz):	?
Dual iso-B mode?:	<input type="checkbox"/>
Further NI card settings	
Clock channel:	Dev 1/ctr
Topo channel:	Dev 1/ai0
Sync trigger channel:	/Dev 1/PF
Fit Settings	
Do fit?:	<input checked="" type="checkbox"/>
Plot init fit?:	<input type="checkbox"/>
Fitting function:	Single lon
Reference Frequency:	2.87e+00
Fit results	
PL =	107.98 kCts/s
Position =	2836.51 MHz
B =	1.20 mT



Review of modules: Newport picomotor 8742

The screenshot displays a software interface for controlling three Newport picomotors (Antenna X, Y, and Z). The interface is divided into three main panels, each corresponding to a motor. On the left, there is a sidebar with a parameter list and a log window.

Parameter List:

- Log level: DEBUG
- Loaded presets: single picomotor.xml
 - Preset file
 - Overshoot file
 - Layout file
 - ROI file
 - Remote file
- Actuators Init. (Antenna Y, Antenna Z, Antenna X) - all green
- Detectors Init. - all green

Log Window:

2024/10/17 11:54:52: Preset mode (single picomotor.xml)

Antenna X Panel:

- Actuator: Newport_Picomotor8742
- Current value: -29 k
- Abs. Value: 0
- Rel. Increment: 5 k
- Buttons: Find Home, Set Abs., Set Rel. (+), Set Rel. (-), Stop, Update Value

Antenna Y Panel:

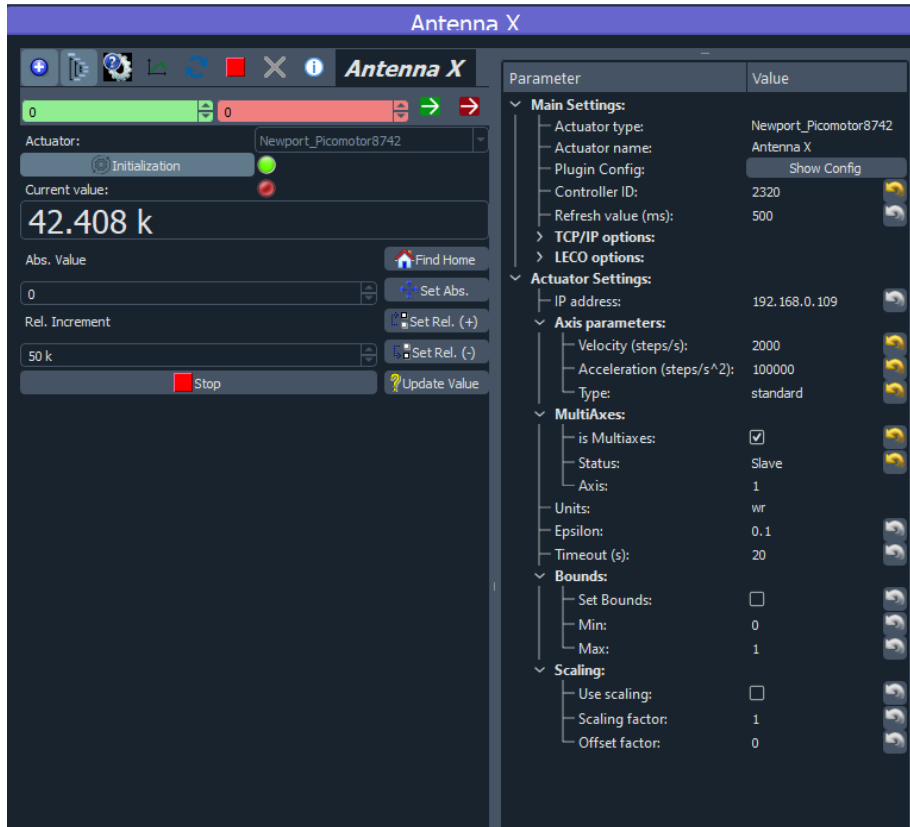
- Actuator: Newport_Picomotor8742
- Current value: -29 k
- Abs. Value: 0
- Rel. Increment: 10 k
- Buttons: Find Home, Set Abs., Set Rel. (+), Set Rel. (-), Stop, Update Value

Antenna Z Panel:

- Actuator: Newport_Picomotor8742
- Current value: 4 k
- Abs. Value: 0
- Rel. Increment: 1 k
- Buttons: Find Home, Set Abs., Set Rel. (+), Set Rel. (-), Stop, Update Value

Review of modules: Newport picomotor 8742

pymodaq_plugins_newport

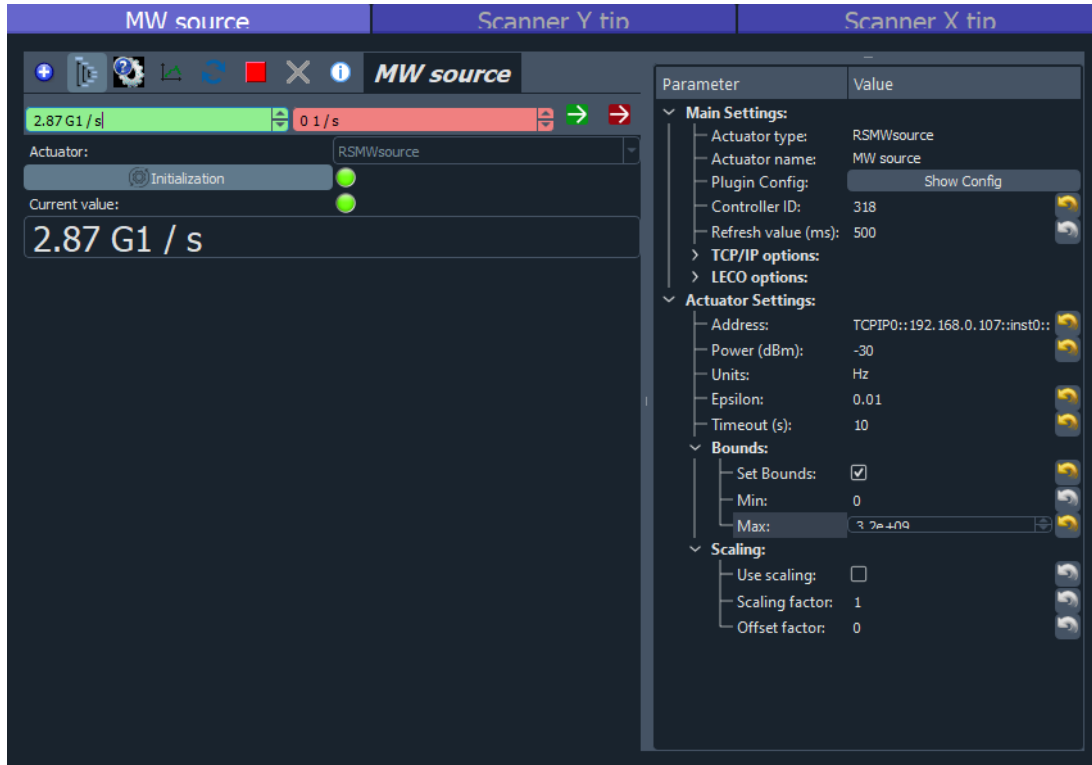


DAQ_Move_Newport_Picomotor8742

- We use the pyblablib 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`

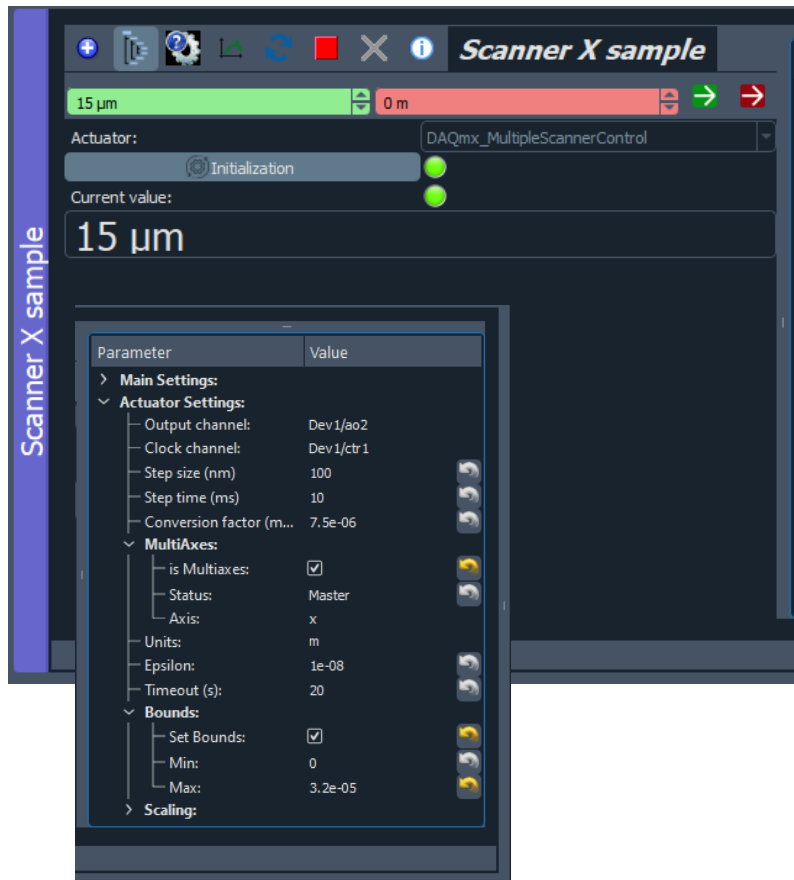


DAQ_Move_RSMWsource

- 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

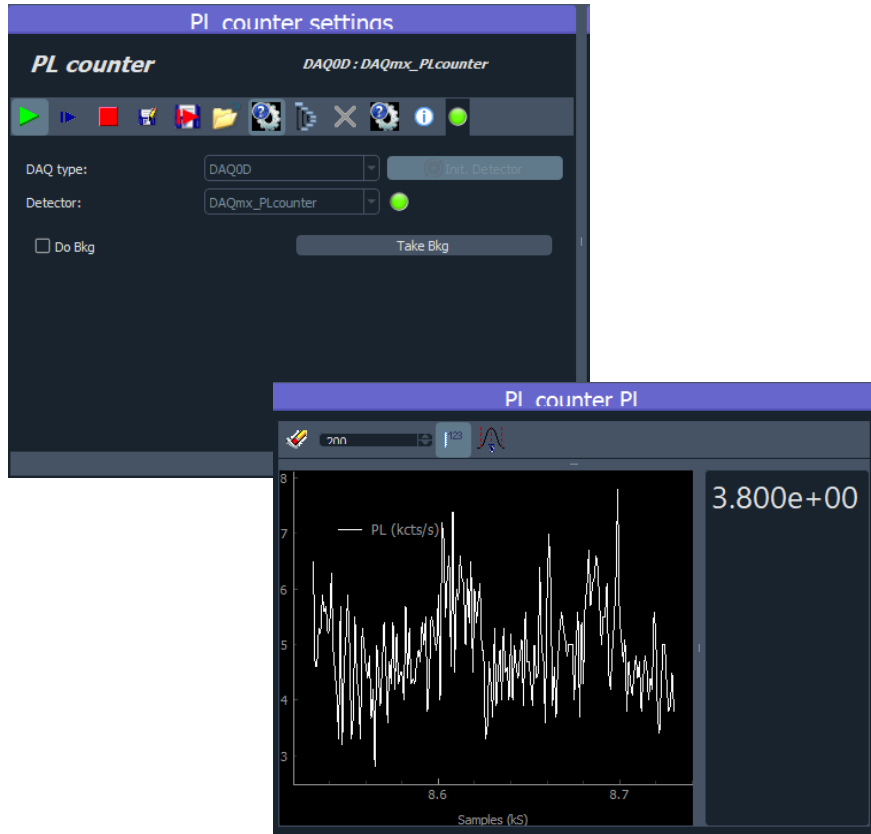


DAQ_Move_DAQmx_MultipleScannerControl

- 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

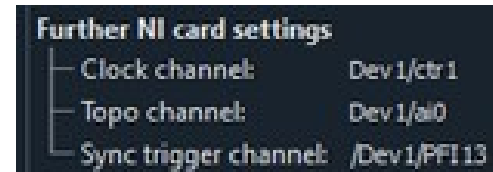
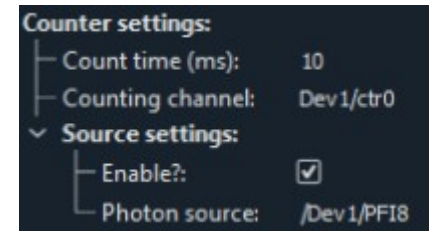
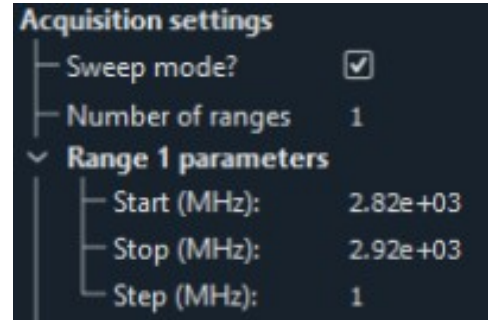
DAQ_1DViewer_ODMR

- 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)

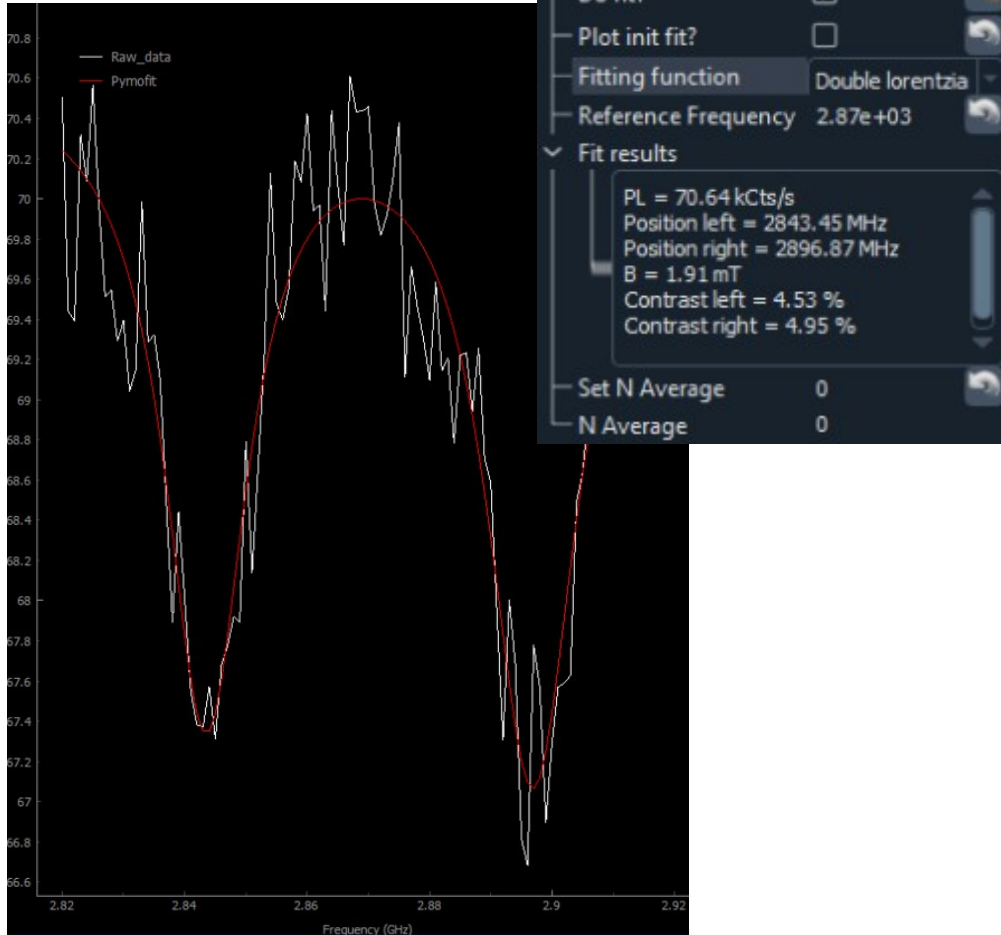


Working principle

- 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.



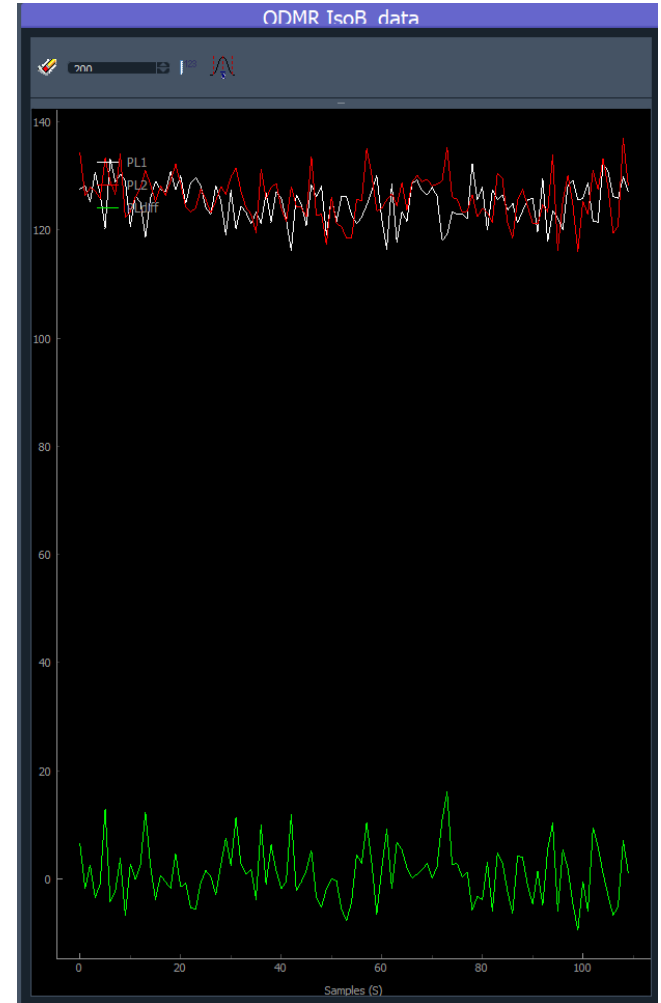
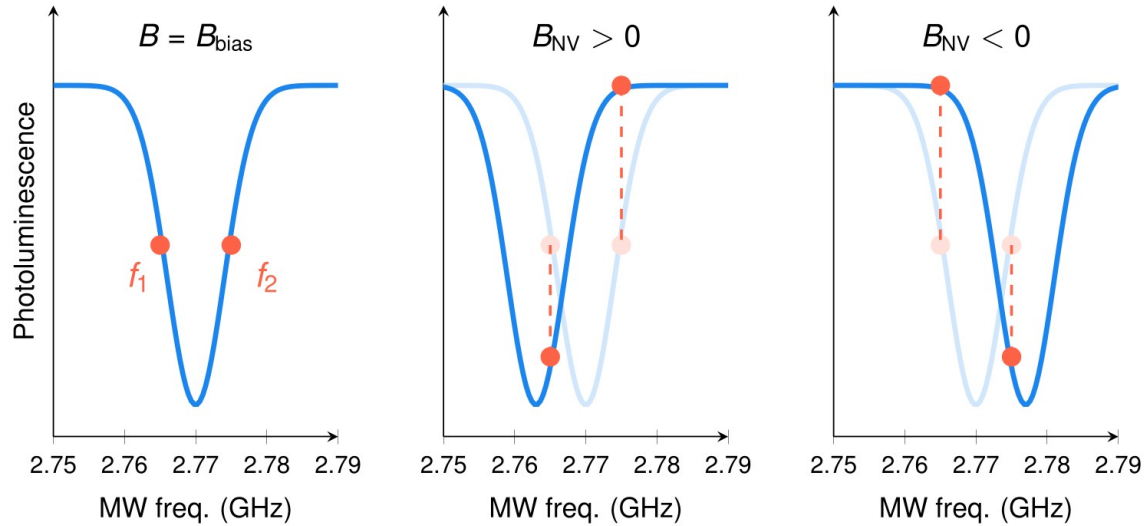
Fitting



- 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?

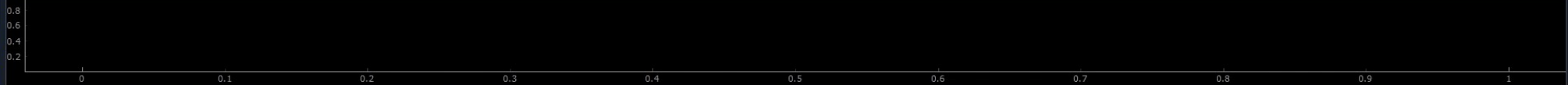
Iso-B mode

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



Demonstration!

ODMR viewer



Scanner Y tip

Loader

Remote controls

MW source

Scanner X tip

ODMR settings

Scanner Y tip

0 m

Actuator: DAQmx_MultipleScannerControl

Initialization

Current value: 0 m

MW source

0

Actuator: RSMWsource

Initialization

Current value: 0

ODMR

Navigation icons: play, stop, refresh, search, help, close

Scanner Y sample

15 μ m

Actuator: DAQmx_MultipleScannerControl

Initialization

Current value: 15 μ m

Scanner X sample

15 μ m

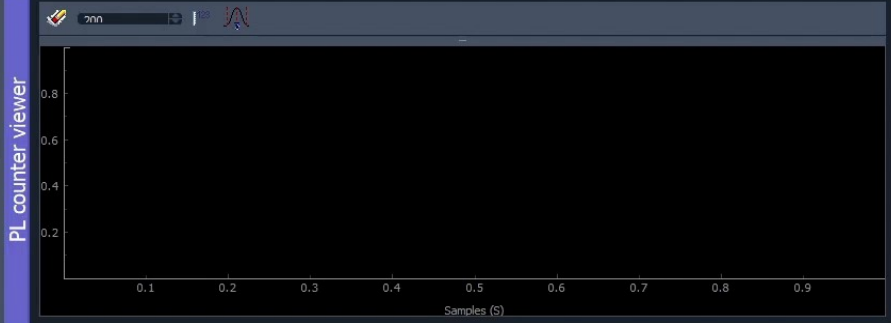
Actuator: DAQmx_MultipleScannerControl

Initialization

Current value: 15 μ m

PL counter settings

Navigation icons: play, stop, refresh, search, help, close



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



Pymodays, October 2024, Lyon — S2QT team