ROACH User Manual

Tutorial

Make a script for starting python

Log into the roach machine with:

ssh -Y myusername@164.54.85.124

To start the roach software in a Linux box at APS you must run ipython, the Enthought Distribution supplied by APS. In your home linux directory make a script like this using the vi editor.

vi runipython

To enter lines into the file, hit *i*

Copy this line into the editor terminal:

/APSshare/epd/rh6-x86_64/bin/ipython -pylab

Hit this key sequence to save, quit.

ESC

:w

:q

Make the script executable with

chmod +x runipython

Starting ROACH software in python

Start python

./runipython

In the python screen, cd to your install of the roach software. For now we assume it is in

ROACH/projcts. In the python terminal type the following lines:

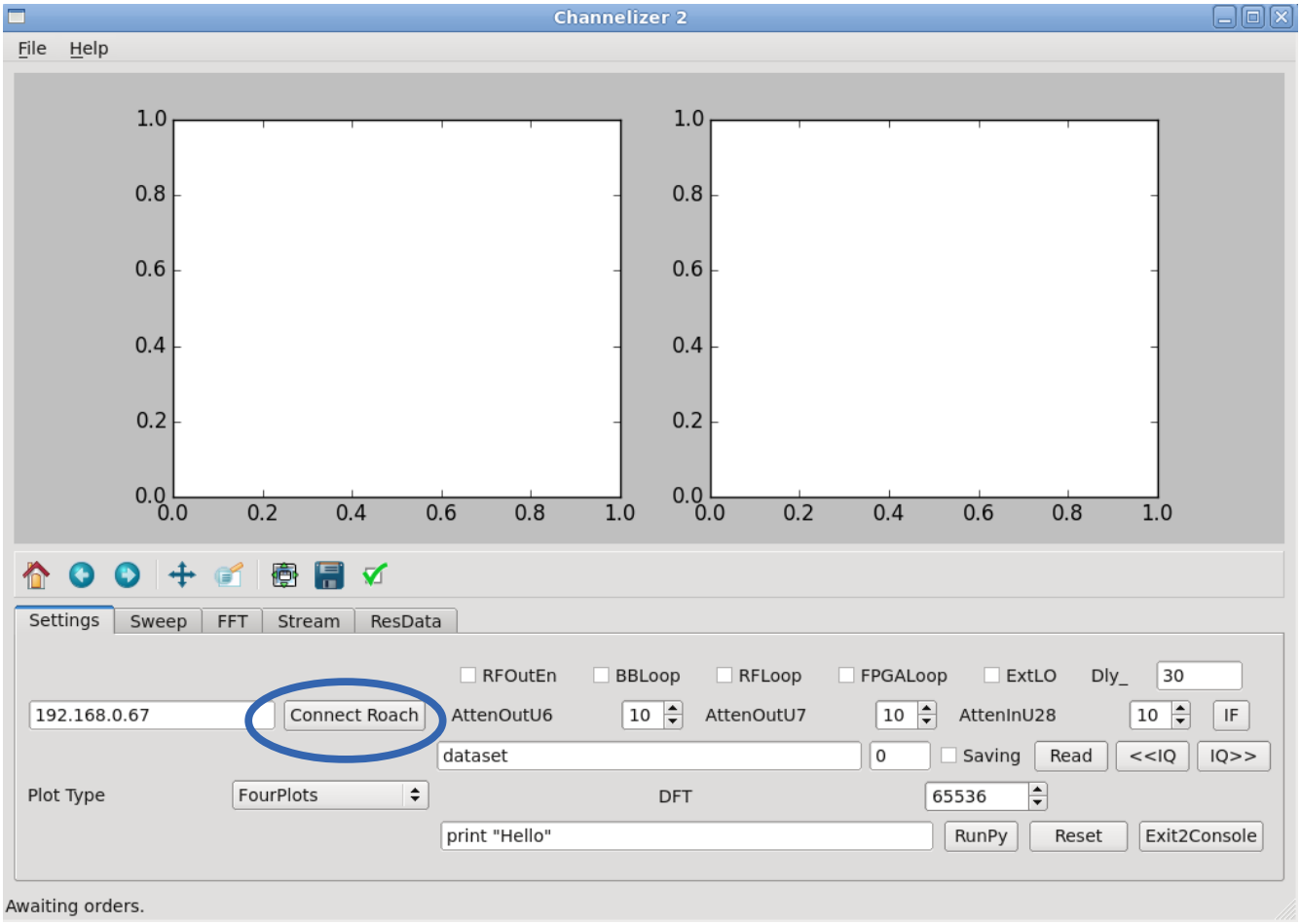
cd ROACH

cd projcts

execfile('natAnalGui.py')

main()

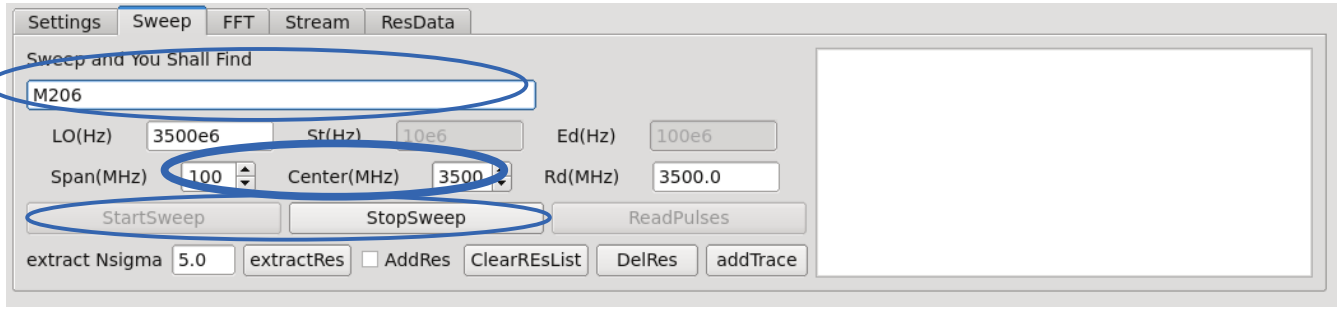
You should see this window open:



To get started, hit *Connect Roach* button. You may see some errors on the terminal, ignore them. They will get fixed eventually. Once you see *connction established* on the bottom of the window, you can search for resonators.

Finding and making a list of resonators

Hit the *Sweep* Tab. You will see this:



Type in the name of your device, above it is M206. This will keep your data associated with your device name.

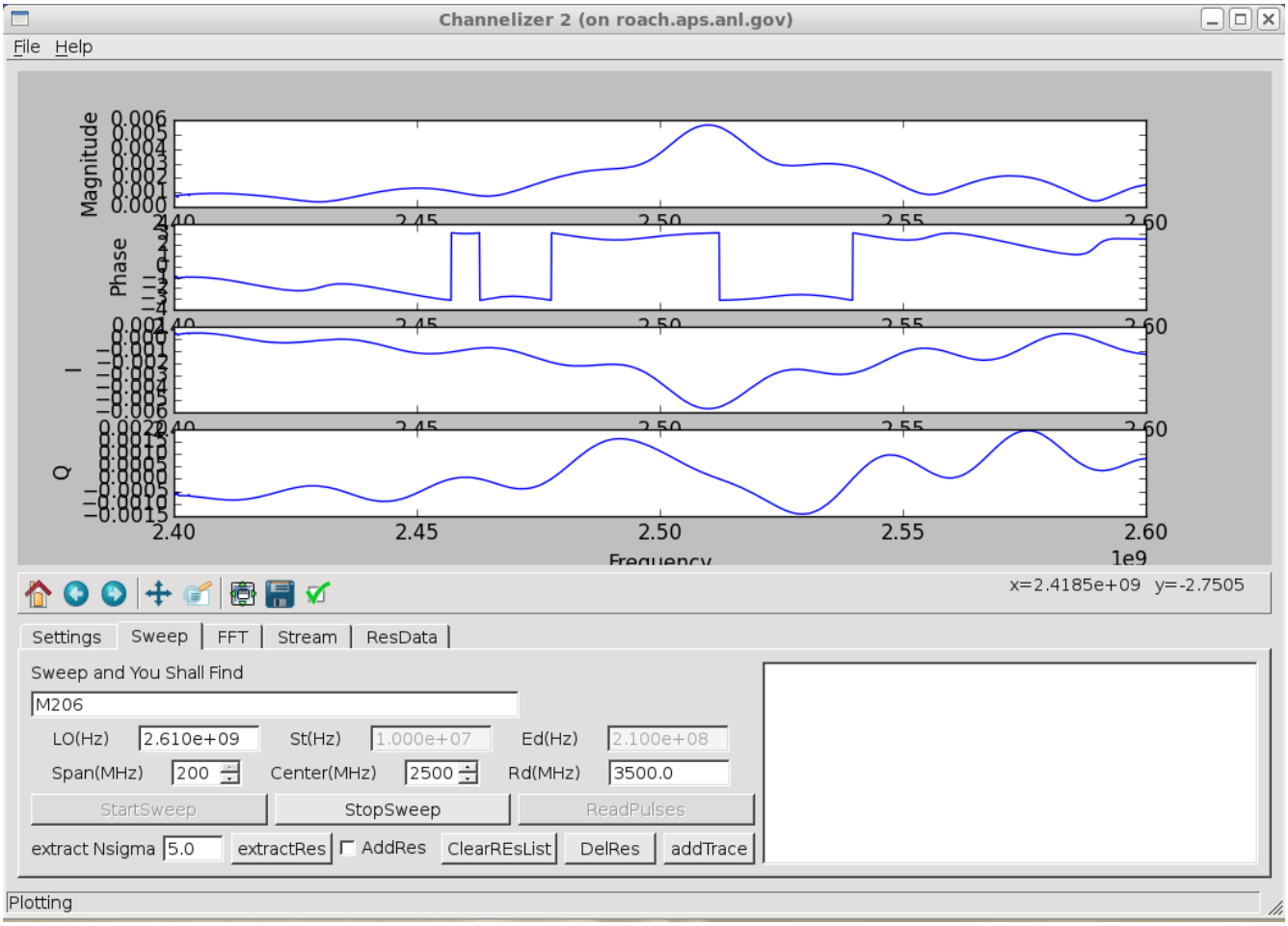
We will now find resonators on the device.

Set Span to 200MHz

Set Center Freq to 2500MHz

Hit *Start Sweep*

You will see plots of the resonators if they exist:



The plots above from top to bottom are

Amplitude in rms normalized to 1,0, Phase in radians, I, Q normalized to 1.0, rms. The x axis is freqiency in Hz.

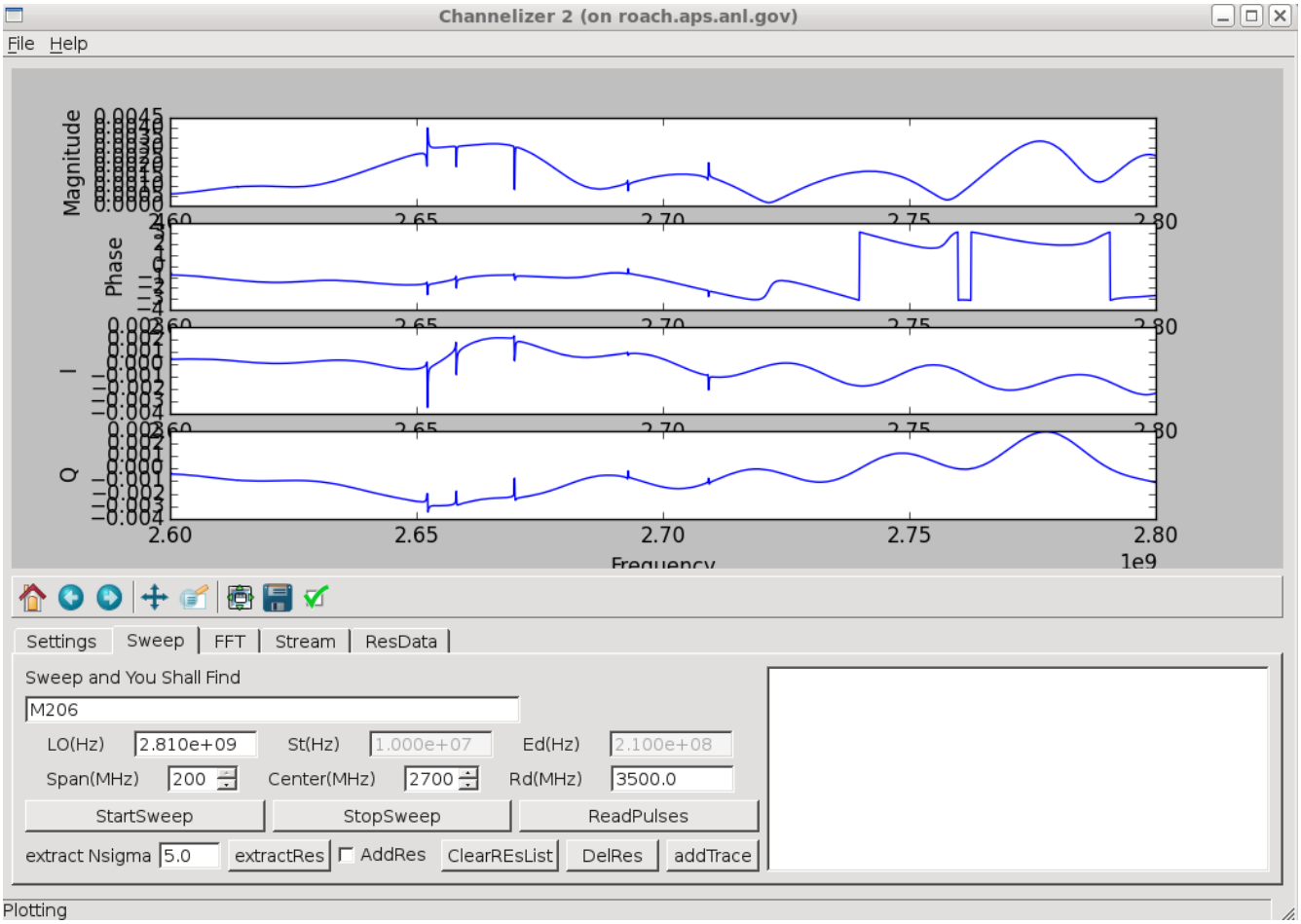
If you see no resonators,

Hit *Stop Sweep*

Increment Center(MHz) by hitting up arrow. Set to 2700MHz

Hit *Start Sweep*

Notice we see resonators below:



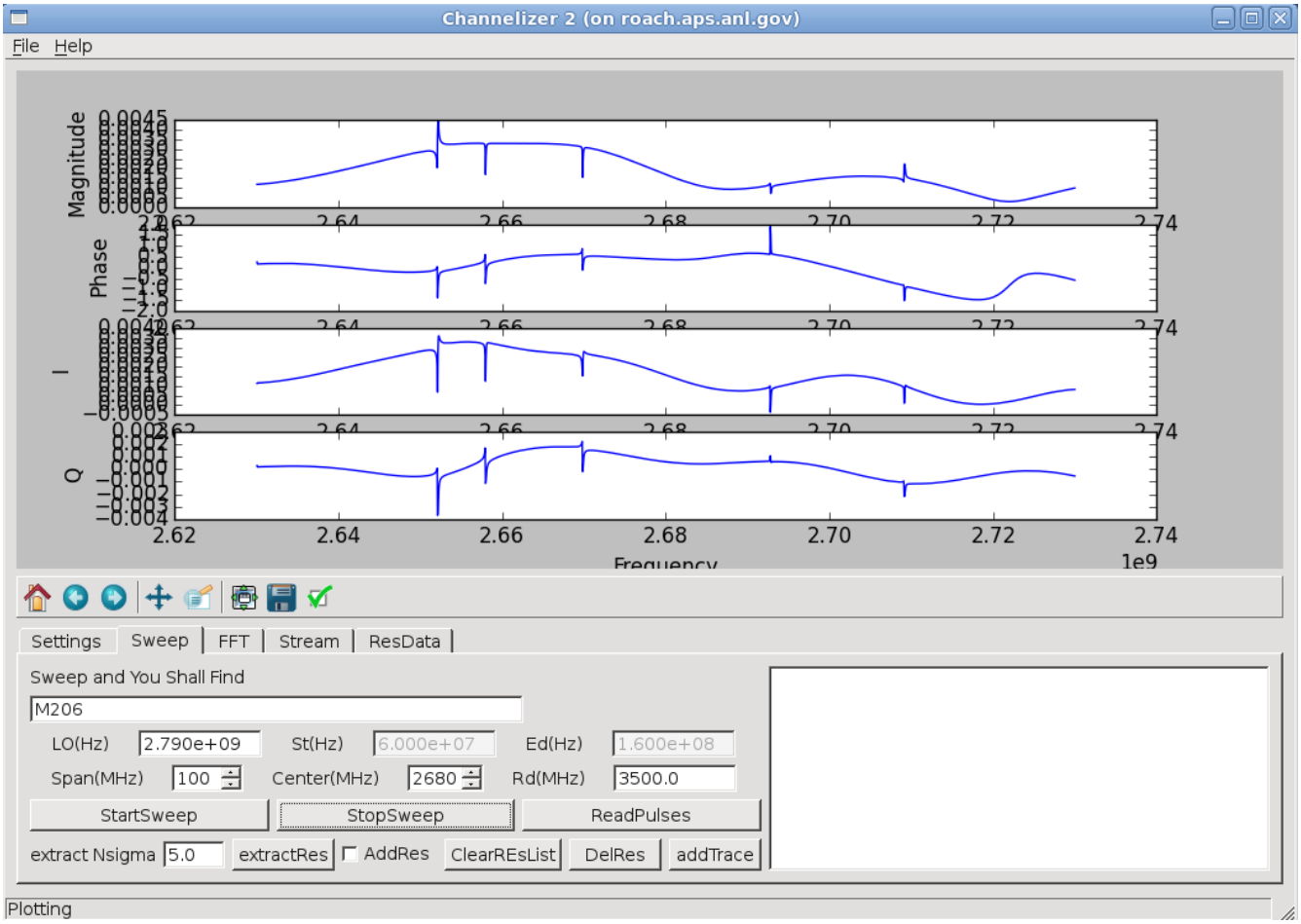
Let is ZOOM in.

Hit *Stop Sweep*

Set *Span* to 100MHz

Set *Center* to 2680MHz

Hit *Start Sweep*

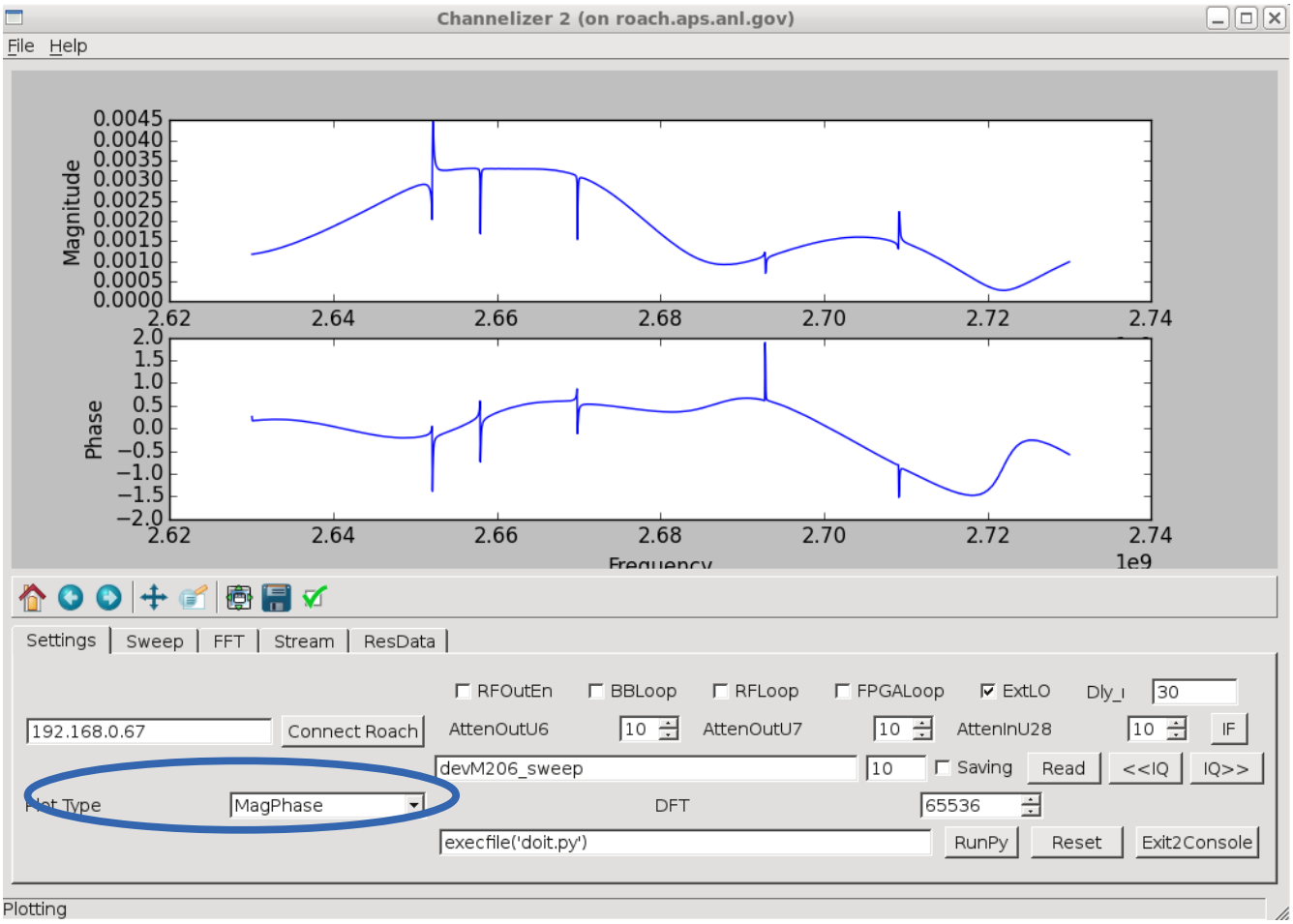


Now we will make a list of resonators.

Hit *Settings* Tab.

Under *Plot Type*, select *MagPhase*

You should see:

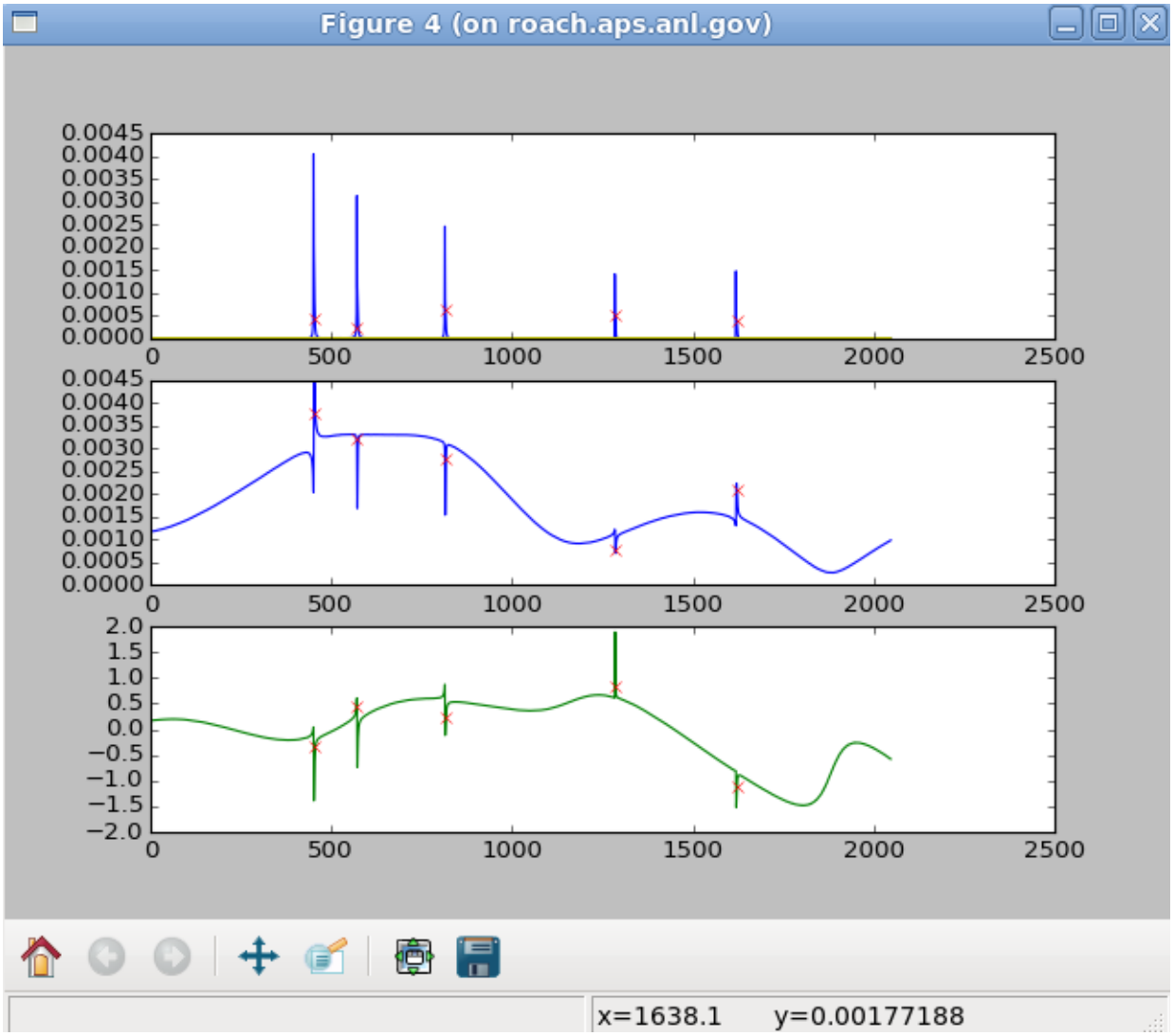


Hit *Sweep* Tab

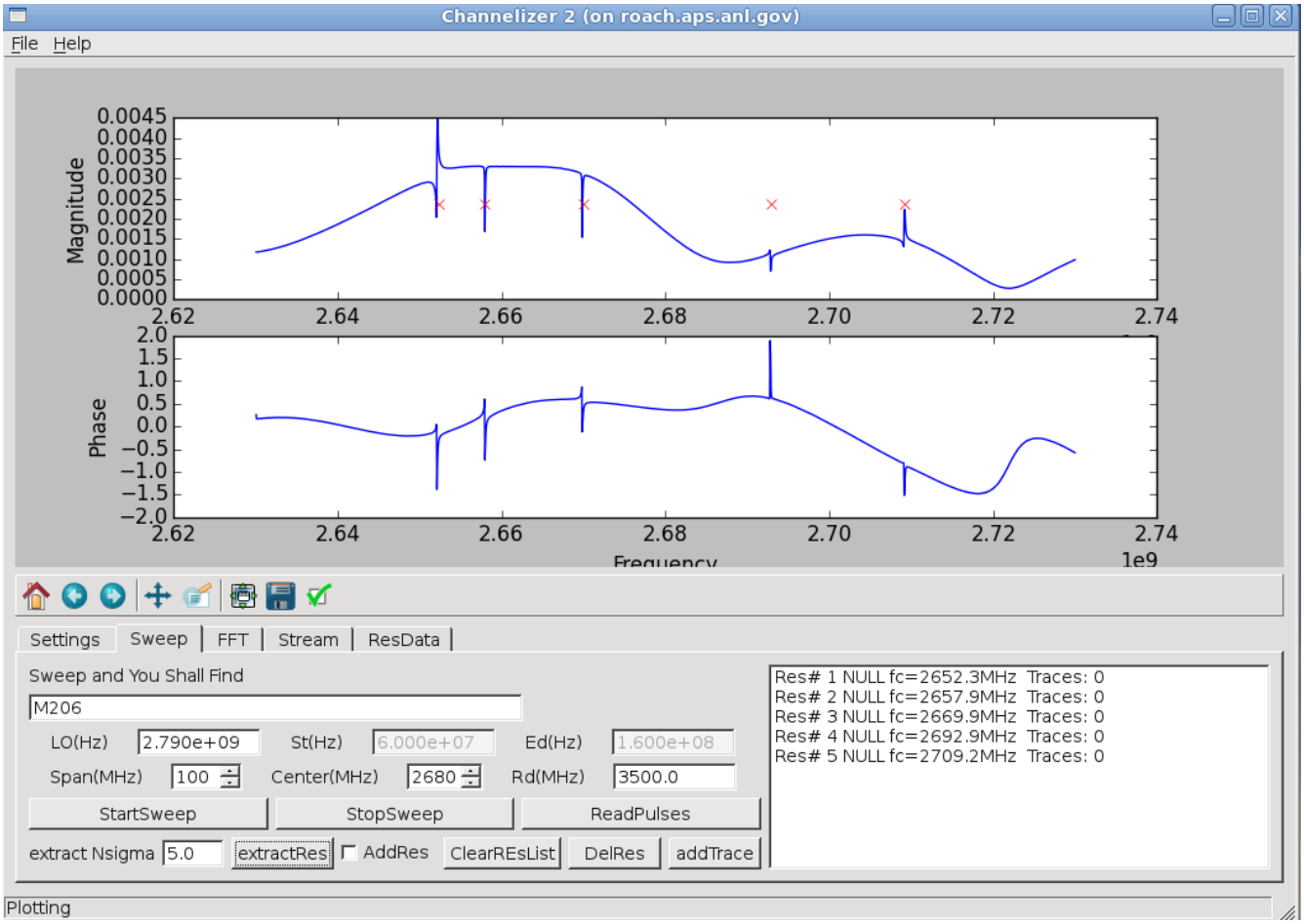
Hit *StopSweep*

Hit *extractRes*

A window comes up:



This shows that we found resonators correctly. The main window will look like this:



Note that resonators have red *x* marking them on the plot. Also we see a list of resonators to the right.

Notice that the names are NULL. This is a little bug...Under "*Sweep and you shall Find"* retype the name M206. Then the list updates. Sorry about the bug.

If you have problems finding resonators, with *'x'* being in wrong places, you can delete resonators from the list, clear the list, and add resonators manually.

Clear the List:

Hit *ClearREsList*

To Delete a resonator:

In the list, select the one you wish to delete

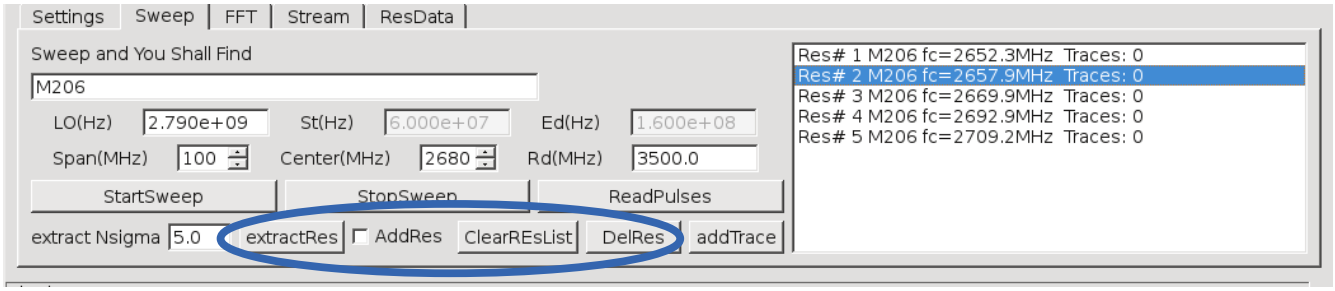
Hit *DelRes*

To add resonator manually

Check *AddRes*

Click in the TOP plot, the magnitude, on the resonator you wish to add

Selected resonator BELOW



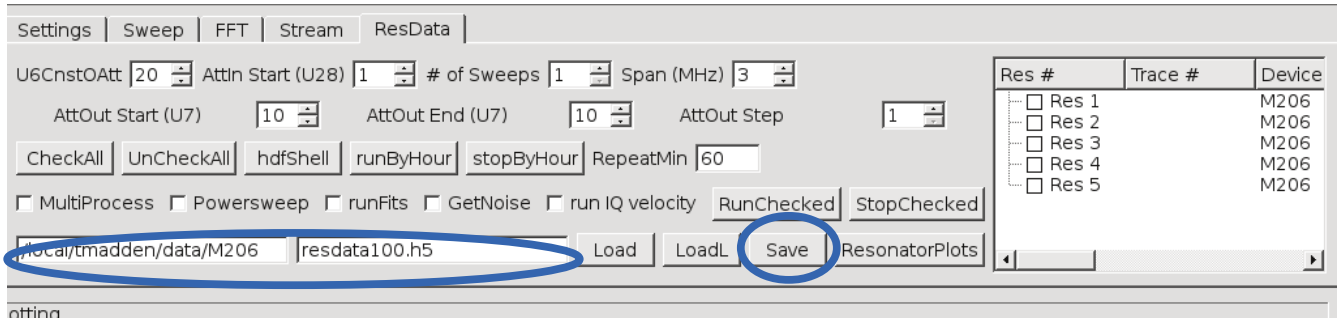
Now we have a list of resonators.

Let is save the list before we lose them!!!

Hit *ResData* Tab

Type in a path and name of a file and hit *Save*

You may see a “cannot stat” error message in your terminal, Ignore it. It is trying to make backups of your files. Once you save several times, you will have several backups in the ROACH/projcts directory where you have yoor ROACH software.

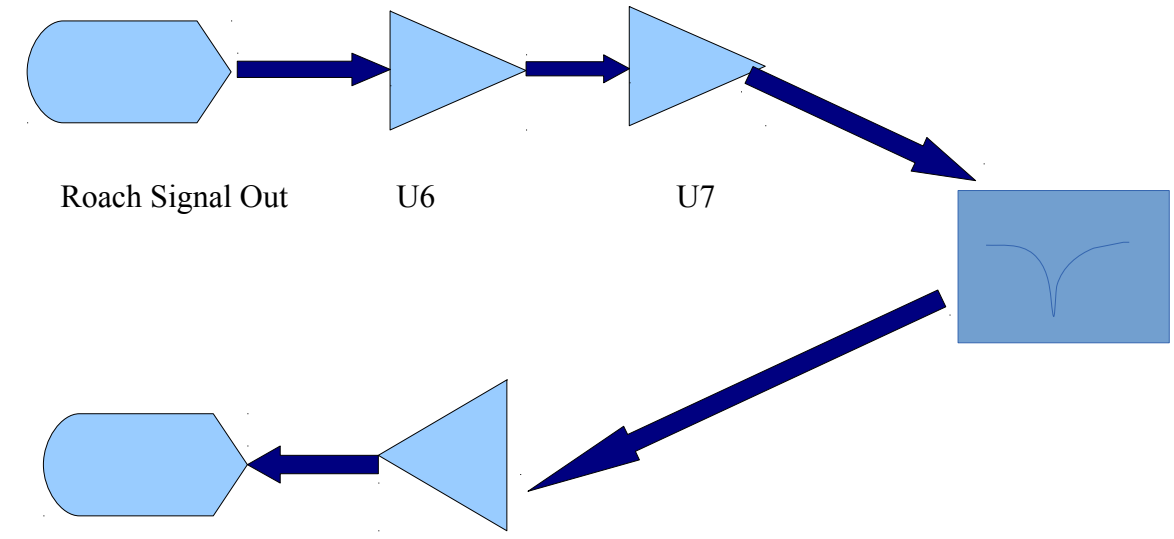


Resonator Power Sweep

Now you can do a power sweep. Follow these steps:

1. Set U6CnstOutAtt to some number of dB. You can leave it at 20. This atteunator will be held constant during the power sweep.
2. Set AttOutStart (U7) to a small atteniation, like 0Db.
3. Set AttOutEnd (U7) to a large attenuation like 20dB
4. Set AttOutStep to 2dB.
5. Set AttinStart (U28) to AttOutEnd(U7) + 1. IN this case set it to 21dB.

What we did is set up a power sweep. Attenuator U6 will be held constant at 20dB. Attenuator U7 will setp from 0dB to 20dB in 2dB steps. U28 will steo from 21dB to 1 dB in 2dB steps. Here is a diagrab of how these atteniators are connected:

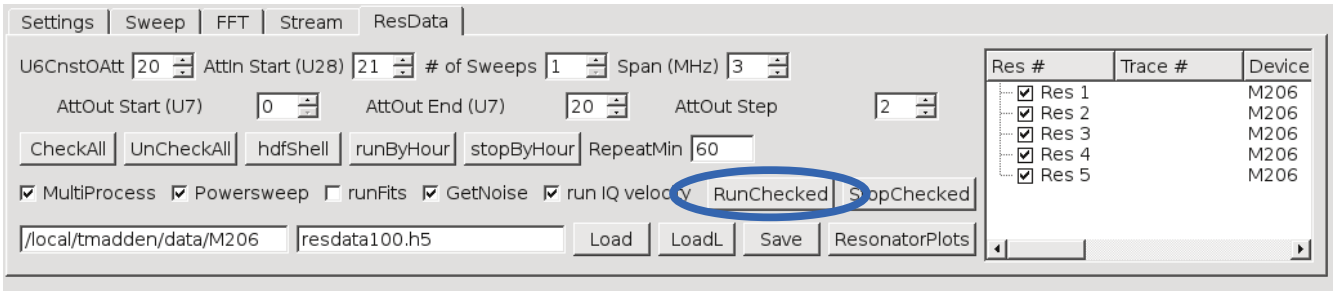


Now we will start the power sweep:

1. In the list on the right, select boxes of resonators you wish to sweep. OR- Hit *Check AlII* button.
2. Check Multiprocess

3. Check Power Sweep
4. Check RunFits if you wish. It is slow, so you may skip it. You can run them later.
5. Check GetNoise. You will then get some noise data on each resonator. You can skip this, as it takes time. It is not the best mesaurement of noise anyway. There is abetter way to be shown later.
6. Check runIQVelocity. This is essential of you get noise data.

Here is our setup:



Now hit

RunChecked

to start the power sweep.

It takes some time. Do something else.

When done, hit *Save*

The system makes backups of your sweep data called *ROACH/projcts/powersweep_backup.h5*

Also, yoru file will be saved automatically from time to time during the sweep. If the software crashes you can retrieve data from your file later, or from backup files called:

ROACH/projcts/powersweep_backup.h5

ROACH/projcts/backup_N.h5

The smallest N is the newest backup.

Again, *Roach/projcts* could be a different directory. It is the home directory where your roach softwarwe are installed.

Once you get yoru data, see the section below on **Hdf Data Analysis.**

Streaming Noise and Pulse data

Noise Data

For best results, the resonators should have been power swepted at at least one attenuation and IQVelocity should have been run, to find the resonator centers. See above.

To stream noise data into a file, this is the best way to measure noise.

1. Hit *ResData* tab
2. Select resonators you wish to stream. For ROACH, the cleanest measurement is one res at a time because we have not developed high speed data transfer from ROACH. You can read out several resonators, but blocks of data are lost.
3. Hit *Stream* tab.
4. Hit *MKIDs->ROACH.* This gets the info from the selected resonators and sets up the ROACH board accordingly.
5. Type in a path/filename.
6. Type number of seconds you wish you stream data.
7. Make sure *PulseDet* is NOT checked.
8. Hit *StartStream*

You will end up with an hdf file on your computer with streamed data. See **HDF Data Analysis.**

Pulse Data

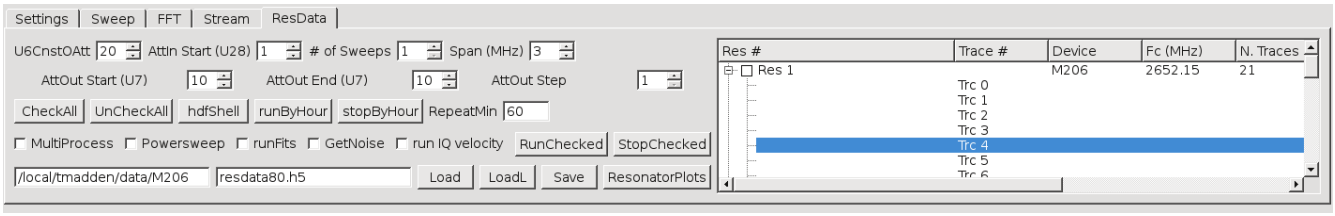
To collect pulse data:

1. Do steps 1-6 above under NOISE DATA.
2. Check *PulseDet*
3. Set PulseDet Nstd to 5.
4. Hit *MeasMeans.* This takes a short nosie measuremnt on resonators and gets mean values of noise and phase, magnitude values from each resonator.
5. Hit *ProgPulseDet.* This will program the pulse detector, and take a test measurement for .5 seconds to get pulse rate. AT *PulseDet Nstd* =5, you will see pulses.
6. Reduce the pulse rate by setting *PulseDet Nstd* to a larger number like 7. This is number of noise standard devuations.
7. Hit *ProgPulseDet* a few times to get count rate. A zero count rate may mean you never see pulses. A count rate of 80 pulses /sec is good. You will see something.
8. After settign up the pulse detector hit *StartStream*.

9. You should have an hdf file with pulses in it.

Making Plots in the GUI

Once you have done a power sweep you will see resonators with many traces:

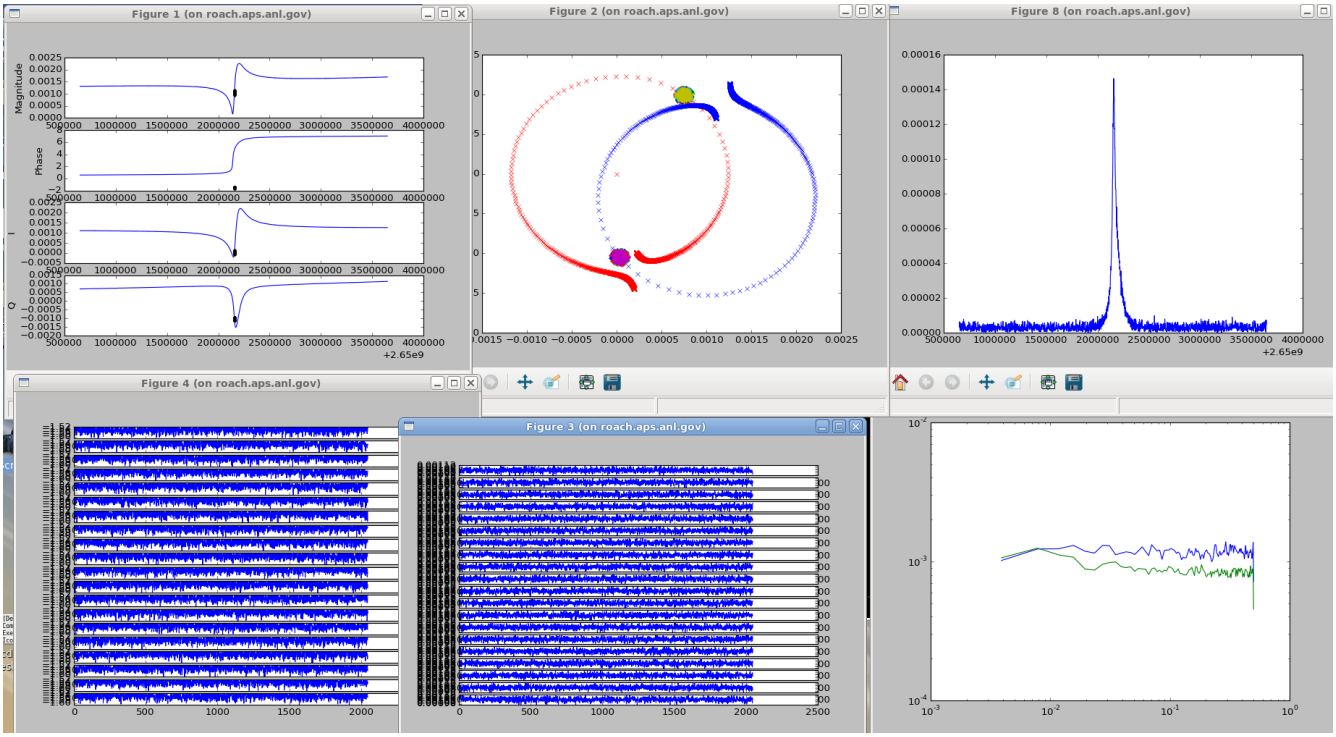


Select a trace as shown in the figure above.

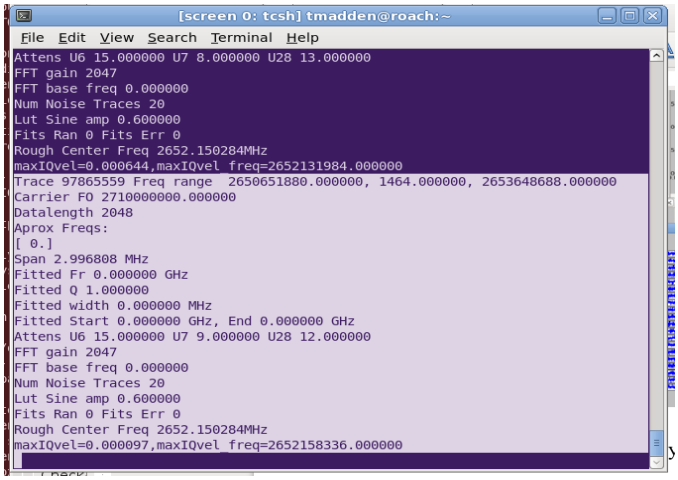
Hit *ResonatorPlots*. You will see the following plots come up:

1. Mag, Phase, I, Q vs. Frequency. Sweep with noise included.
2. I vs Q raw (blue)and transrotated (red)
3. IQ velocity vs. Frequency. (sweep data)
4. Noise magnitude and phase plots.
5. Power spectrum of noise.

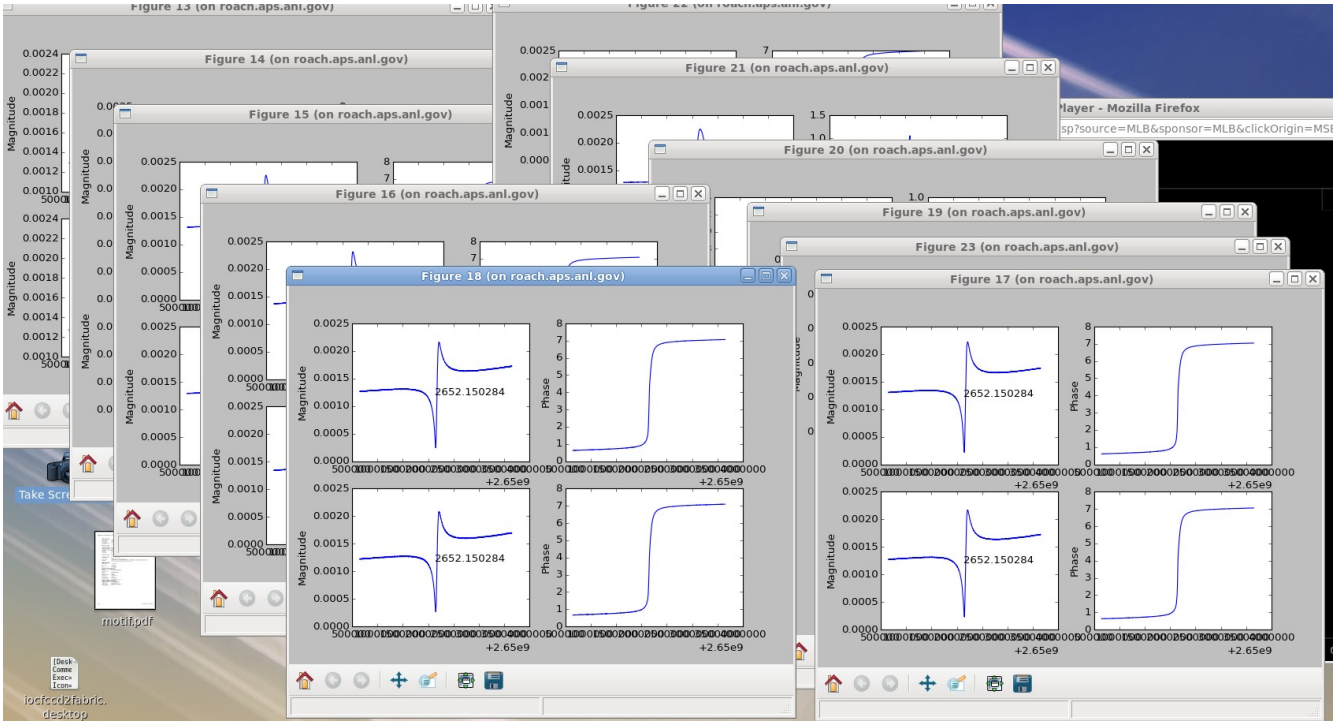
Plots are shown below.



Also when you hit *ResonatorPlots,* you will get info on the terminal on the traces



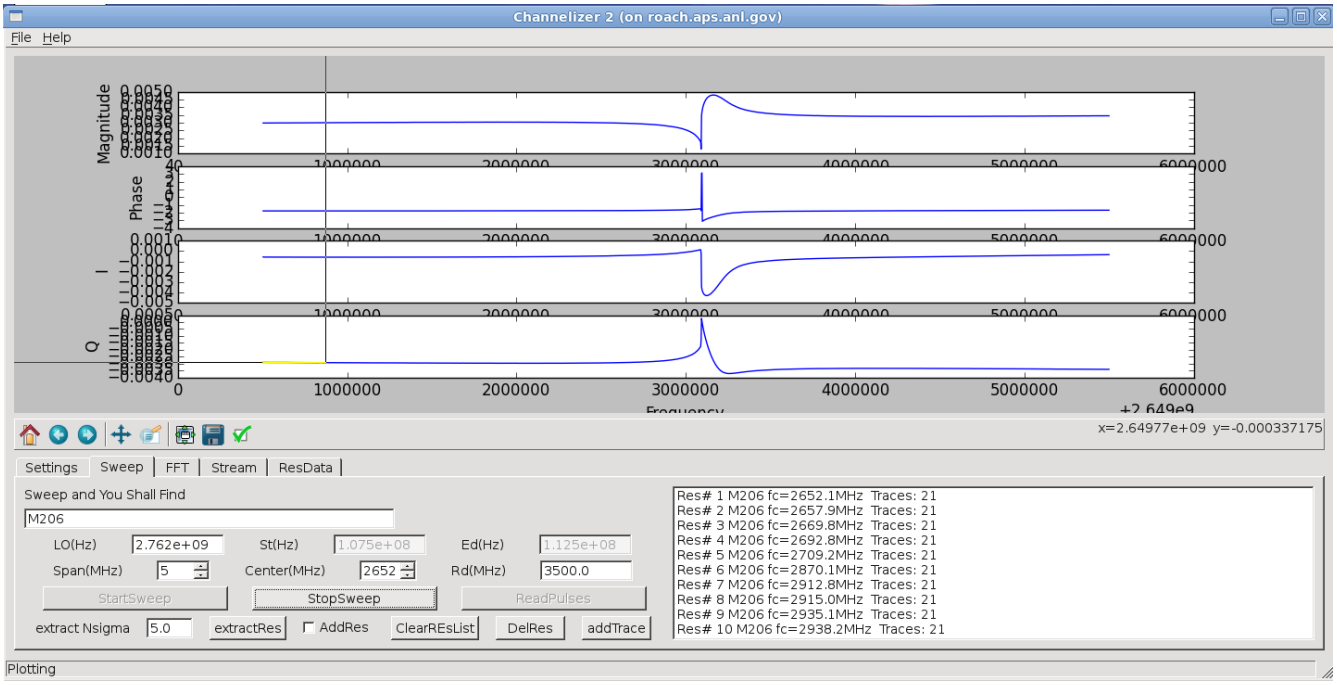
If you select the MKID, and hit *ResonatorPlots* you will get plots of all sweeps.



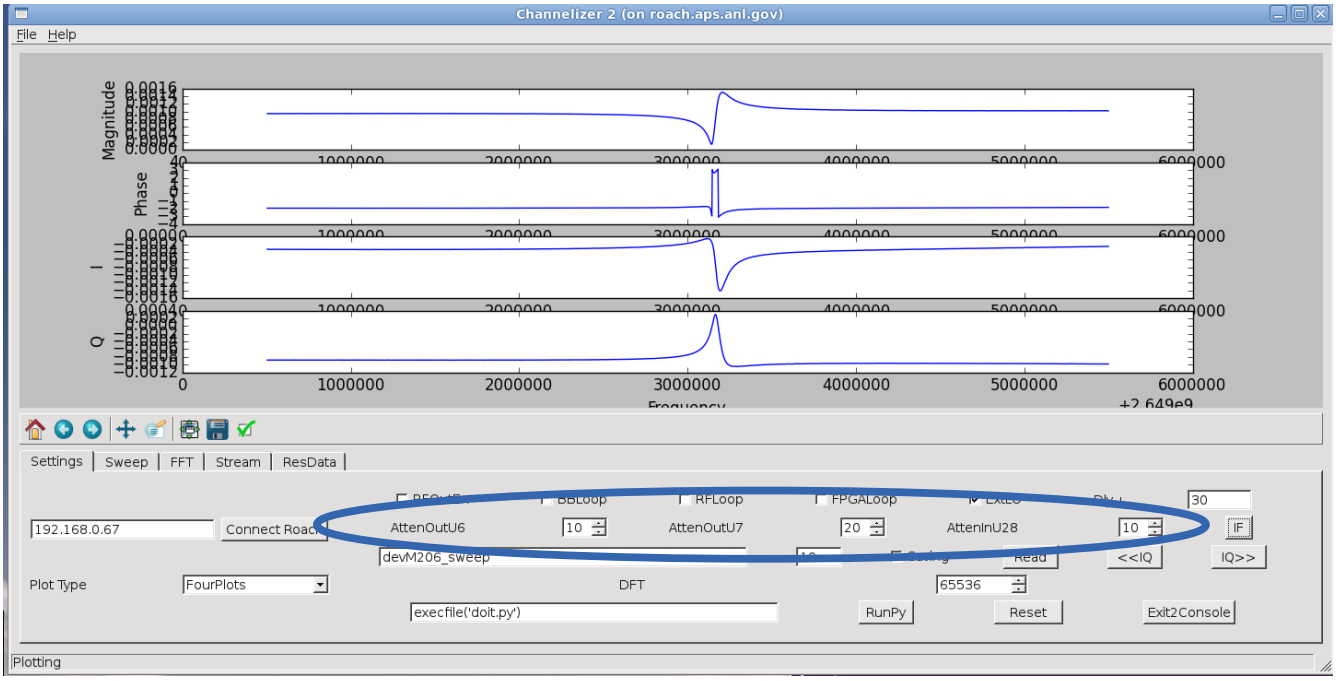
All sweeps of MKID 0

Sweeping and Live Plots

We can set up the system to sweep over and over a resonator like a network analyzer by setting the Span and Center Frequency on the *Sweep* Tab. See below.



Note that the resonator has too much power being sourced. Simply goto the Settings Screen and set the attenuators.



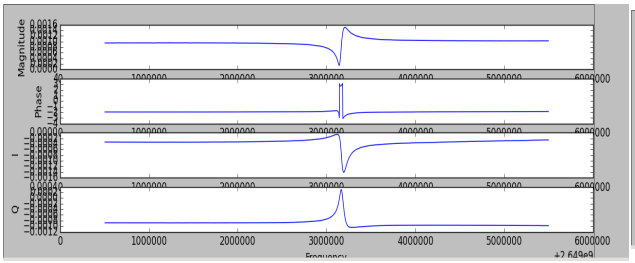
Addionally it is possible to loop back the system by clicking

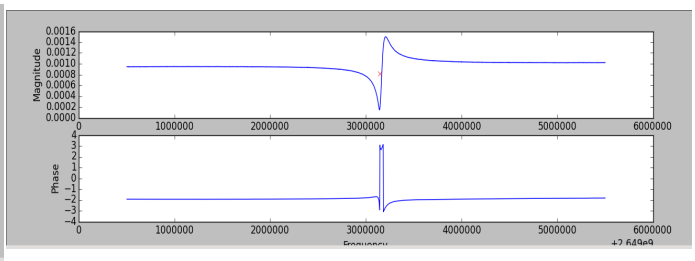
- RF Loop- Loops back the RF signal at GHz size of Mixer.

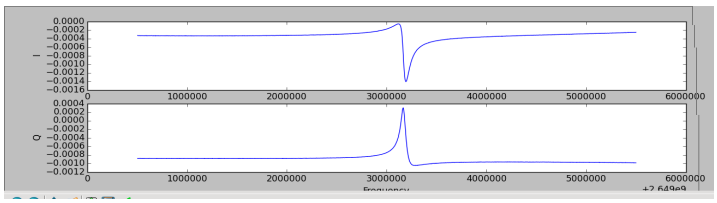
- BB Loop- loops back Baseband, leaving mixer out of circuit.
- FPGALoop- digital loop back inside the FPGA. No analog circuitry involved awt all.

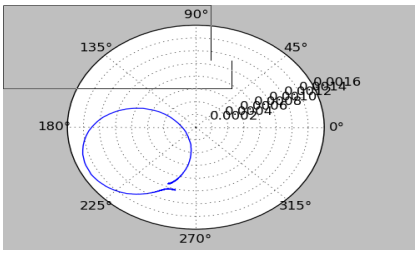
There are several plots that can be dieplayed live:

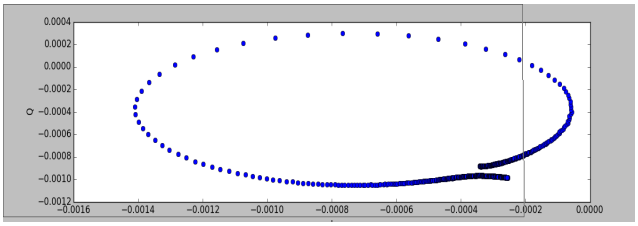
- four Plots- mag, phase, I, Q vs frequency for sweeps.
- Mag, Phase sweep
- I and Q sweep
- Polar
- I vs Q











There are more plots, but they do not work with the Sweeping network analuzer firmware.

HDF Data Analysis

In ipython:
hdfshell()

Or if you are in the roach gui you can do this following:

1. On *ResData* tab, hit the *hdfShell* button
2. *OR*- Exit the ROACH gui to get python, by hitting the X at top of window.
3. *OR*- On *Settings* tab, hit *Exit2Console*. You will then have access to python.
4. In python type *hdfshell()*

open /dir/dir/myhdf.h5

Opens the hdf file.

close

closes the hdf file

q

Exits hdfshell() and saves hdffile.

help
man

print help screen

Navigating HDF5 files

This starts a shell for navigating hdf files.
Use linux commands to navigate.

Opening hdf file
ls -rlt
Lists files on disk.

cd /somedir/somdir etc
cds in linux to dir.

open backup_1.h5
opens hdf file backup_1.h5

List contents
ls
ls -ds
will print dataset valies
ls dirname
will print contents in that group

cd groupnema
will set pwd to group

pwd
shows current group

rm groupordataset
moves group or data to the a group called /trash

mv oldname newname
renames or moves data in the hdf5 file

cat mydateset
more dataset
These commands show what is in a dataset.

Searching HDF files

Use linux style find command:

find Finds everything at current dir.

find . Same as above

find Device_M206 finds everthing under dir Device_M206

find Device_M206/Resonator_1 -name fbase

Finds everything under Device_M206/Resonator_1 with fbase in the name.

find Device_M206/Resonator_1 -name fbase -ds

Finds everything under Device_M206/Resonator_1 with fbase in the name, prints dataset values

find Device_M206/Resonator_1 -name time -attr -ds

Finds everyting in Device_M206/Resonator_1 with time in the name. Also searches attributes. Prints valuies

find Device_M206/Resonator_1 -name timefl -ds -out createtimefl

Finds everyting in Device_M206/Resonator_1 with time in the name. Prints valuies. These values will be stored into a new data set at current working dir, called createtimefl. You can use an absolute path for -out such as -out /mydir/mydir2/mydata

Plotting Datasets

plot xdata ydata

plot ydata

plots dataset, 4k points

plot -cs b xdata ydata

plots x versys y data, 4k points

plot -cs rx ydata

Use python colorspec rx and plot ydata

plot -ts ydata

plot ydata events according to timestamp. Useful for pulses

plot -l 40000 ydata

plot 40000points of ydata

clf

clears plot

Making IQ Plots

Because data may not be in the correct form for an IQ plot, the pliq command exists. Example, if we have mag and phase data, we must first convert to rectangular coordinates before making an IQ pliot. Als, if data is stored in a multi-dimentional array, there is no easy way to extract the slabs of data from the dataset. For example, often resonator sweeps are stored as iddata[2][L} for L point sweeps.

pliq -iq iqdata

pliq -mag magnitudedata -ph phasedata

pliq -i idata -q qdata

For i nd q data in separate

pliq -c Chan+00000

For a Chan_xxxxx directory with magnitude and phase data.

Pliq -l 40000 -i idata -q qdata

Plot 40k points.

Pliq -cs rx -iq iqdata

plot iqdata is red x's

Nosie plots

welch /Chan_xxxx/phase

makes PSD plot of phase nosie. For looking at 1/f noise

Trans-Roatation of Resonator Sweep and Noise/Pulse data.

To rorate data based on a sweep:

transrot -r /sweeps/ResData_4 -c /Chan_00000

Take sweep data in ResData_4, use it to rorate the magnitude and phase in Chan_00000. You must use the correct resonator sweep that corresponds to the Chan. To find out, match the rf freqeucny or baseband frequwnecy of nosie and sweep data.

Datasets magnitude_tr and phase_tr are created in the Chan_xxxxx directory of transrotated data.

pltrot -r /sweeps/ResData_4 -c /Chan_00000

Plots resonator as a circle on IQ plane, plots noise cloud and pulses on the circle from Chan directory.

Making Notes

To store a note in hdf file:

note mynote.txt

You then type a note. Hit ctrl-D to end note. The note is now stored.

To see note:

cat mynote.txt

Extracting Data to Python

To extract a data set to pythomn as a variable,

topy /mygroup/mydataset mypyvarname

The dataset is created as a python variable as a global, and in namespace myvars.

That is a global called mypyvarname is created and myvars. Mypyvarname is created.

py print mypyvarname

py will run a python command. Py and follow withy any python command. Globals will be seen in the scope of the cammend. If you exit hdfshell() with ‘q’ then you are returned to python interpreter. The global py variabels will still exist.

Plotting data versus timestamps

Each resonator sweep has a timestamp when it was swept. The data sets are called

createtime

List of [year, month, day, hour, min, second]

createtimefl

floadint point number of seconds since epoch. See python time module for explanation. This is used as a timestamp when plotting data vs timestamp.

For some MKID, we can crearte a dataset of all sweep timestamps

find Device_M206/Resonator_9 -name createtimefl -ds -out Device_M206/Resonator_9/createtimefl

We find all create time stamps for that resonator for all sweep traces. These timestamps are put into dataset at

Device_M206/Resonator_9/createtimefl

You can plot this data:

plot Device_M206/Resonator_9/createtimefl

Or you can now get some other data point like the center freq of resonator, most useful is maxIQvel_freq. That is the res center freq based on max IQ velotucyu.

We make a new data set for centerfreqs.

find Device_M206/Resonator_9 -name maxIQvel_freq -ds -out Device_M206/Resonator_9/centerfreqs
Now we have a centerfreqs dataset w/ center freq for each trace.

To plot centerferq vs sweep time:

plot -cs x Device_M206/Resonator_9/createtimefl Device_M206/Resonator_9/centerfreq

Examing Resonator Sweep Data

Because taking sweeps and noise to characterize resonators creates much data, a special program called resView exists solely for dealing with ressonator sweep data.

To start resview, resonator data must be loaded into a data structure internal to the python interpreter. In hdfshell, there are two types of resonator data:

1)Single sweep data from streamed noise
    /sweeps/

2) Multi-Sweep data from sweeping resonators w/ power sweeps etc.
    /Deviuce_XXXX

If there is a sweeps directory,
cd sweeps
loadres
resview

    Each resonator sweep will be loaded as a seperate MKID, with one sweep each on loadres.

Resview starts the resonator viewer.

The structure of the sweeps director is as follows:

/sweeps/ResData_0

/sweeps/ResData_1

etc.

Each ResData_N directory is a sweep of a separate MKID. IN this type of data file, each MKID has a single sweep.

If there is a Device_XXXX directory,

cd to root directory above Device_XXXX dir.

loadres

resview

The structure of Device_XXXX is as follows:

Device_XXXX/Resonator_1

Device_XXXX/Resonator_2

etc

The Resonator_X directories coorespond to indivitual MKIDs. Each MKID may hagve manuy sweeps.

Device_XXXX/Resonator_1/ResData_0

Device_XXXX/Resonator_1/ResData_1

etc.

The ResData_N directories coorespond to each sweep trace for Resonaotr_1.

After calling loadres,

resview

The resonator viewer program is started. You will see a prompt, ResView>>.

q

Exit resview back to hdfShell

Resvuew Commands

Resview has the concept of current MKID and current trace. The idea is that we can step through MKIDs and traces and plot each one easily.

q

quit resview return to hdfshell

save

save hdf file.

help

man

List resview commands

list

lists MKIDs and all traces

<return>

Plot current trace

<space><return>

Increment to next trace and plot

+ <return>

Increment to next trace, plot, but do not clear plots

hold

nohold

Turn off automatic clear plots with hold. Nohold turns on auto clear of plots.

nmkid

goto Next MKID

st

Goto 1st MKID and 1st trace

plmkid

Plot all traces of this MKID. Need to call hold first so plot is not cleared after drawing.

fit

Run fits on cirrent trace

clf

Clear plot, replot current trace

-<return>

Goto previous trace, plot

pr

Plot all fitting curves for this trace

iqvel

For this MKID, use all traces to make IQVelocity plot versus atteniation. This is for finding preferrend attenuation for this MKID

prefat

Set preferred attenuation for this MKID. This is then stored in HDF File, and can be used later for taking data.

flatph

If phase is increasing too much w./ freqiencuy due to wrong time delay in data, this sets time delay

properly for thsi trace. Meks fitting easier

fitprefat

fit resonator trace that is at preferred atten for this MKID. Finds correct trace based on atten, and fits.

Fitprefatall

Fit all MKIDs for traces at preferred atten.

Findat

find trace bawsed on U7 atten

findat2

find trace for thsi MKID based on total atten, U6,7, and LUT amp.

plnoise

Turns on plotting of nosie traces

nonoise

Thrns off pliotting of noise

heory of Operation

The ROACH runs two firmware designs:

1. Network Analyzer- A network analuzer design that can sweep resonators and read one resonator at a time.

2 FFT Analuzer- A code that uses FFTs for reading many channels at a time. Up to 256 channels can be read out.

The software interface for the ROACH is a collection of python scripts. Here is a list of the python scripts:

1. t_brdconfig.py- functions for IF board control, testing functions, network connection to ROACH, roach register setup. Control of anritsu freq. Generator.

2. netAnaluyzer.py- control of network analyzer firmware. NetAnaluzer class is defined.

3. fftanalyzer.py, fftanalyzerd.py, fftanalyzeri.py. These are codes for control of fft analyzer FW. They all inherit from network analyzer class. These three class are a line if inheritence. Fftanaluyzeri inherits fftanaluzerd, which inherits fftanaluyzer, which inherits netanaluzer. There were several versions of the fft formware, and each class represents major versions. The i version is the current version.

4 fitters.py- Resonator trace data objects, MKID_list, the current liost of MKIDs. MKID opbjhects, HDF read/write functions. Fitter class, which has all fitting functions. Several plotting functions.

5 resciew,.py- HDF file tool. A linux-shell that runs in ipython for navigatin HDF files.

6 natAnaluzer.py- GUI for controlling the ROPACH.

7, roachMatlab.py- a debygging script for allowing all py scripts to talk to matlab simulator instead of ROACH board. This is for testing optython codes with FPGA sumulation.

8. controlScripts.py, Misc. Py scripts for controlling roach. Many degugging scripts. Scripts for muylti-processing of fits. Sets up fitting queeus for multi process.

9. roachpref.py=- preferences for roach gui. Edit this for your favorite gui setup.

10 doit.py- pyt anything you want in here. You can execute it from the gui.

11 mpfit.py- needed for fitters.py

Deprecated unused scripts:

dumpView.py

fshift.py

sim.py

test.py

debug.py

ljh.py

doit.py

ftAnalyzere.py

runFits.py

test2.py

theory of operation of netanaluyzer FW

Registers

Data format returned

theory of operation of FFT FW

Registers

Data format in Fft outptu memory

Streamed Data Readout

ROACH settings

RF Switch settings

OSC settings

anritsu control

report()

Roach board ethernet settings

Sending FW to the roach board. /boffiles

Running Fw on roach.

Ioreg interface on roach

logging into roach

Finding resonators

the Resonator List

Sweeping resonators

FFT Readout

Streaming data

Hdf files

Dump files

Debugging data- pckle

Useful Roach py scripts

Controlling network analuzer from python

Controlling FFTs from puython

Useful plots

Streams from python

Resonatopr objects and fits

useful plots