The Software: CRISTAL

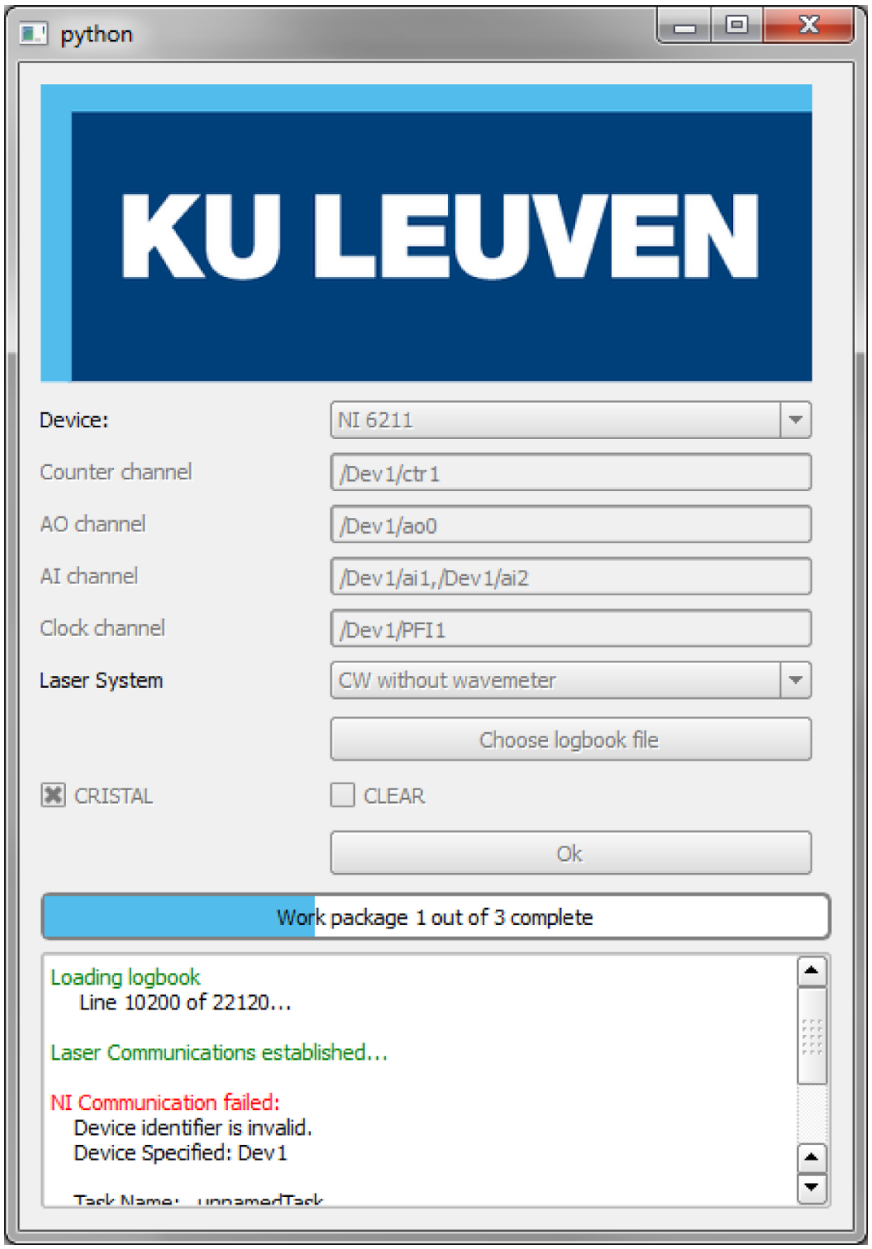
CRISTAL stands for CRIS laser Tuning, Acquisition and Logbook software. It takes care of laser scanning, acquisition of all of the information we typically want during CRIS operation, and ties both aspects together in an **interactive logbook**.

Starting CRISTAL

Proceed as follows:

1. Make sure you are working on PCCRIS6. Fasten your seatbelt as well.
2. Navigate to C:\CRISTALCLEAR. You should see at least two subfolders: Code and doc, as well as a shortcut to CRISTAL.pyc. Double click on the shortcut to CRISTAL.pyc. If you do not see said shortcut, navigate into the Code directory, and double-click on CRISTAL.pyc. If for some reason even that does not work, double-click on CRISTAL.py. A code editor called Sublimetext will open. Press Ctrl-B to run the python code.
3. A command prompt should appear. If at any point the program bugs out or throws errors, these errors will be printed in this command line interface. Let me know what the errors are if this happens!
4. A launcher screen with the KU Leuven logo will appear. Congratulations, you have started CRISTAL successfully! I will now discuss the launcher screen, which is used to further configure CRISTAL to your needs.

Launcher Screen



This screen is used to change the acquisition settings of CRISTAL.

1. The first combo box lets you choose which of the supported devices you are using. Currently that is either the *NI 6211* card, or *None*-in this case no actual data is gathered from any hardware, but dummy random data is generated. This can be useful when debugging.
2. If the NI card was chosen as device, the channels can be configured.

NOTE: the locations of the analog input channels should be separated by a comma, e.g. \Dev\ai1, \Dev\ai2, and in incrementing order.

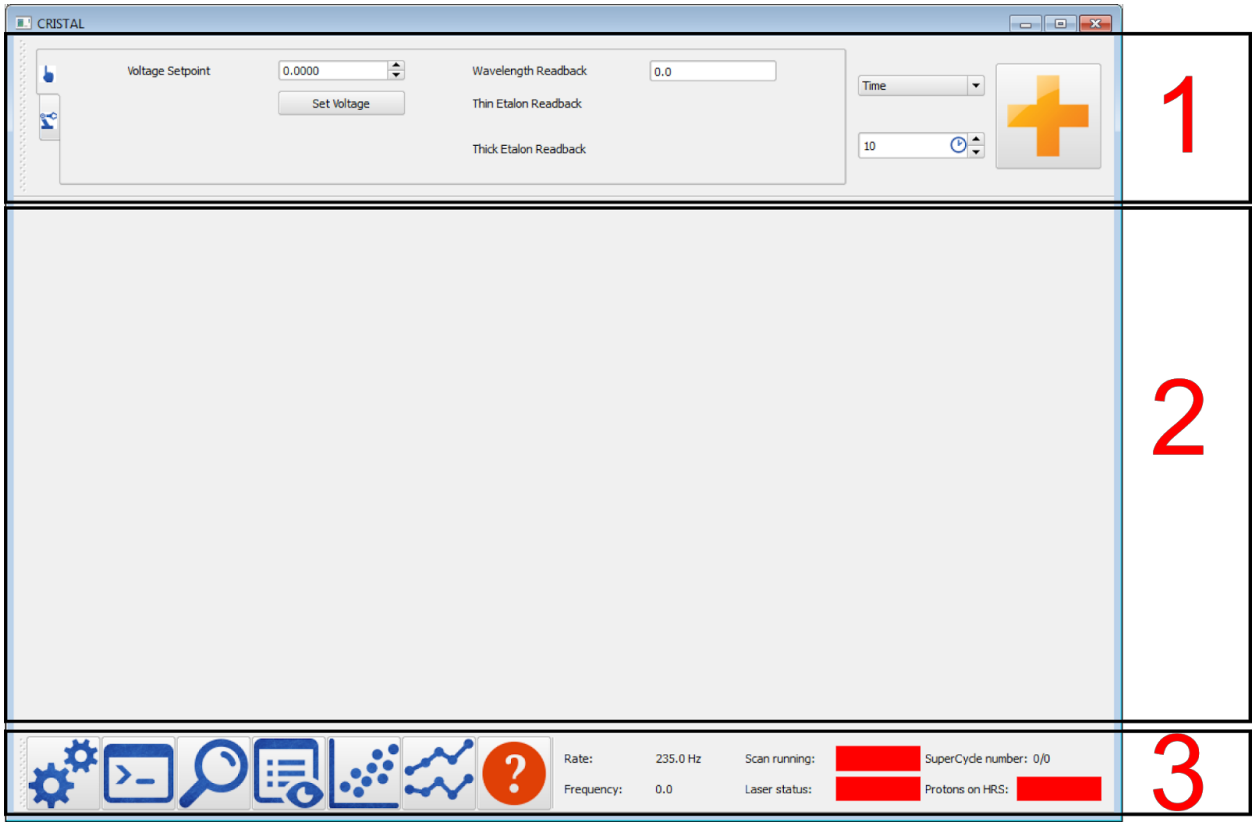
3. The laser system can also be chosen: the options are the cw laser (scanned using the analog output voltage), or the RILIS lasers. The ‘CW laser without wavemeter’ option only controls the laser, but does not read back anything from the wavemeter - useful when you don’t have a wavemeter, or you do not have labview installed on the machine.
4. The next button allows you to choose which logbook file you want to use, **You have to choose a logbook before you can proceed to launch CRISTAL!** If you pick the wrong one here, no worries, you can switch at any point during the program operation.
5. The two checkboxes allow you to choose if you want to use CRISTAL (DAQ), CLEAR (Analysis), or both.

So, in short: got to **C:\CRISTALCLEAR**, run **CRISTAL.pyc**. Complete the **info on the launcher** screen and pick a **logbook**. Wait a bit. Done.

Using CRISTAL

The CRISTAL UI has three main areas:

1. Top toolbar: the **scanner** widget. This toolbar allows you to change laser frequency (by defining a target wavelength or etalon setpoints for the RILIS laser, or by defining a voltage for the cw laser).
2. Middle area: a blank canvas on which all the fun and exciting widgets will appear once you click on the buttons in the...
3. Bottom toolbar: contains buttons to open and close the other widgets of CRISTAL. Also shows the current event rate on the MCP, laser wavelength, provides a boolean status indicator of the laser system, as well as information on the proton supercycle.

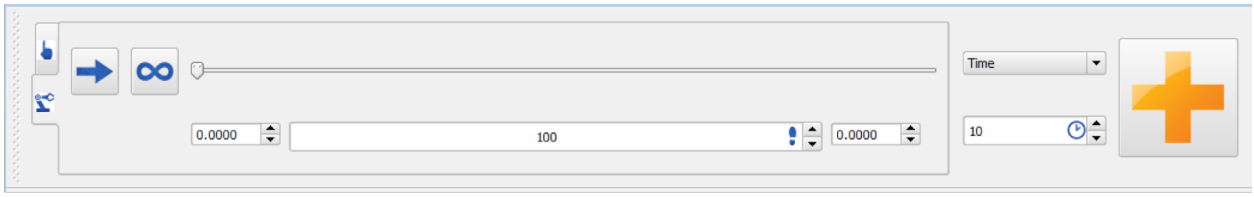


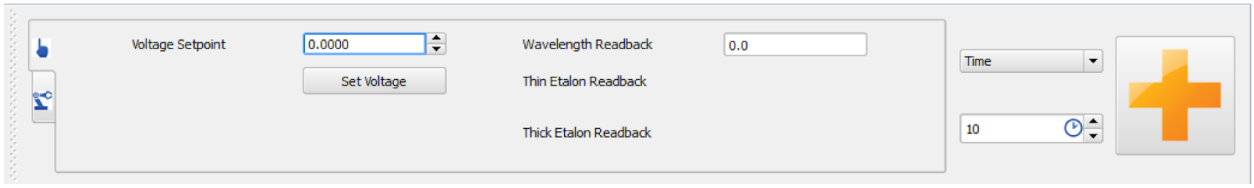
The toolbars can be dragged out of the UI or repositioned in the window.

All of the widgets can be dragged and dropped by clicking and dragging the blue label at the top of the widget to rearrange the interface. All the widgets created by the bottom toolbar can be opened and closed by toggling the buttons, except for the graph widgets (see the subsection below). We will go through the scanner widget and the widgets the bottom toolbar enables one by one.

Scanner Widget

The scanner widget is your way of telling the lasers what to do. On the right-hand side you can see a large button, initially with an orange plus-sign on it. This button handles the creation of a new scan, running of this new scan, and stopping of this scan.





Create a new Capture object in memory. None of the actual scan parameters have been initialized though, so you can still change the scan settings at this point (more info on the scan settings in a bit).

Starts the acquisition. Most of the interface is disabled, so no settings can be changed during a measurement. The measurement will continue until it reaches its predefined end, or...

… until you click the stop button. All measurements are then (gracefully) halted.

To the left of this “master button” you can find a dropdown menu and a spinbox with a clock icon in it. These two elements combined will tell the lasers how long they have to keep fixed at every wavelength step of the scan. The dropdown menu offers the following options:

1. Time: gather data for a time (defined in ms) per frequency step
2. Triggers: gather data for number of acquisition triggers per frequency step
3. Supercycle: gather data for number of supercycles per frequency step
4. Proton Pulse: gather data for number of proton impacts per frequency step

The time/number of triggers/supercycles/proton pulses is defined by the spinbox.

NOTE: Only the ‘Time’ mode has been extensively tested so far. ‘Triggers’ should also work fine. The other two, however, rely on correct interpretation of supercycle data - not guaranteed!

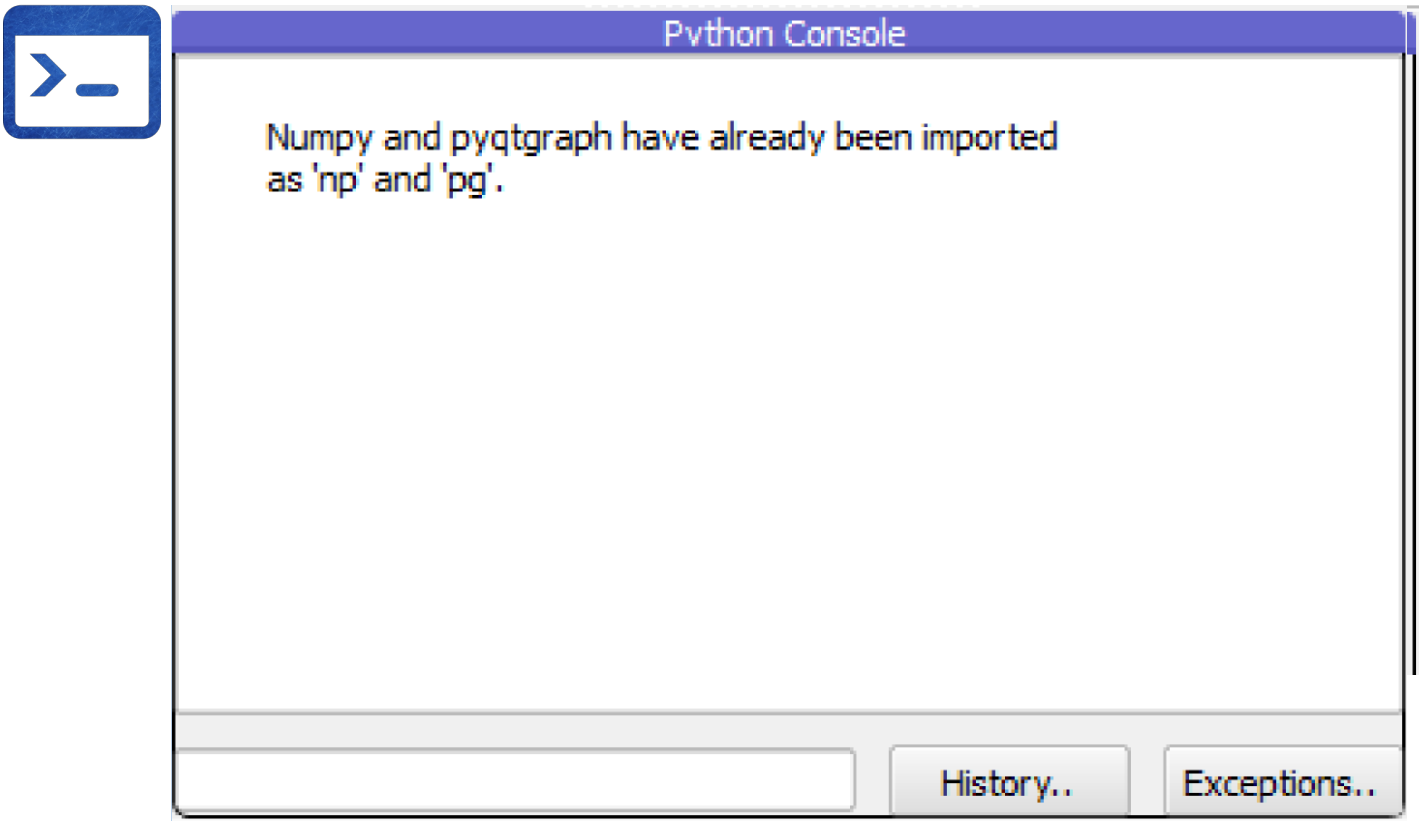
The biggest part of the scanner widget is occupied by a tabbed interface.

One of the tabs can be used to freely change scan parameters like the voltage or etalon setpoints. You actually have to press the `Confirm' button to apply your changes.

The other tab contains a widget that helps you create a scan region CRISTAL then scans for you. On the left are two buttons. The first controls if the scan only scans upwards in the scan parameter, or if it zigzags up and down. The other (with the infinity sign) can be toggled to

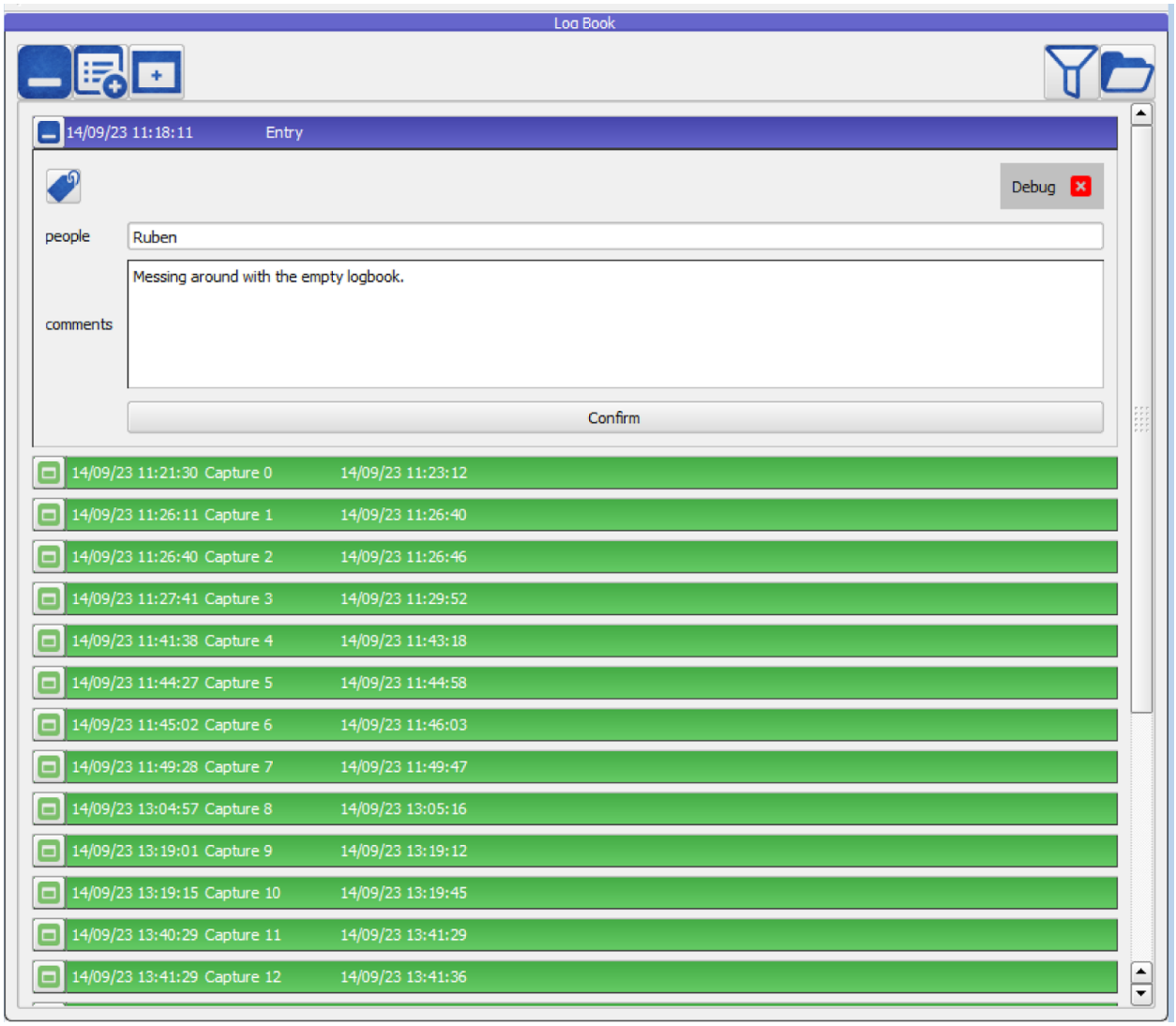
have the software repeat the scan across your region forever (until you untoggle this button or just stop the scan). The biggest part of this second tab is occupied by a slider which visualises the progress through the scan region. Underneath it are three spinboxes: one for the starting point of the scan, the second for the number of steps CRISTAL should scan, and the third fixes the endpoint of the scan.

Console widget



The console widget provides a Python console with the numpy and pyqtgraph libraries already loaded in the np and pg namespaces. Go crazy! Divide 1/3 and find out that in Python 2.7 this still equals 0!

Logbook widget



The **logbook widget** provides you with a logbook interface. Every time a new capture is initialized, a timestamped entry is added to the logbook, waiting for you to fill in the blanks.

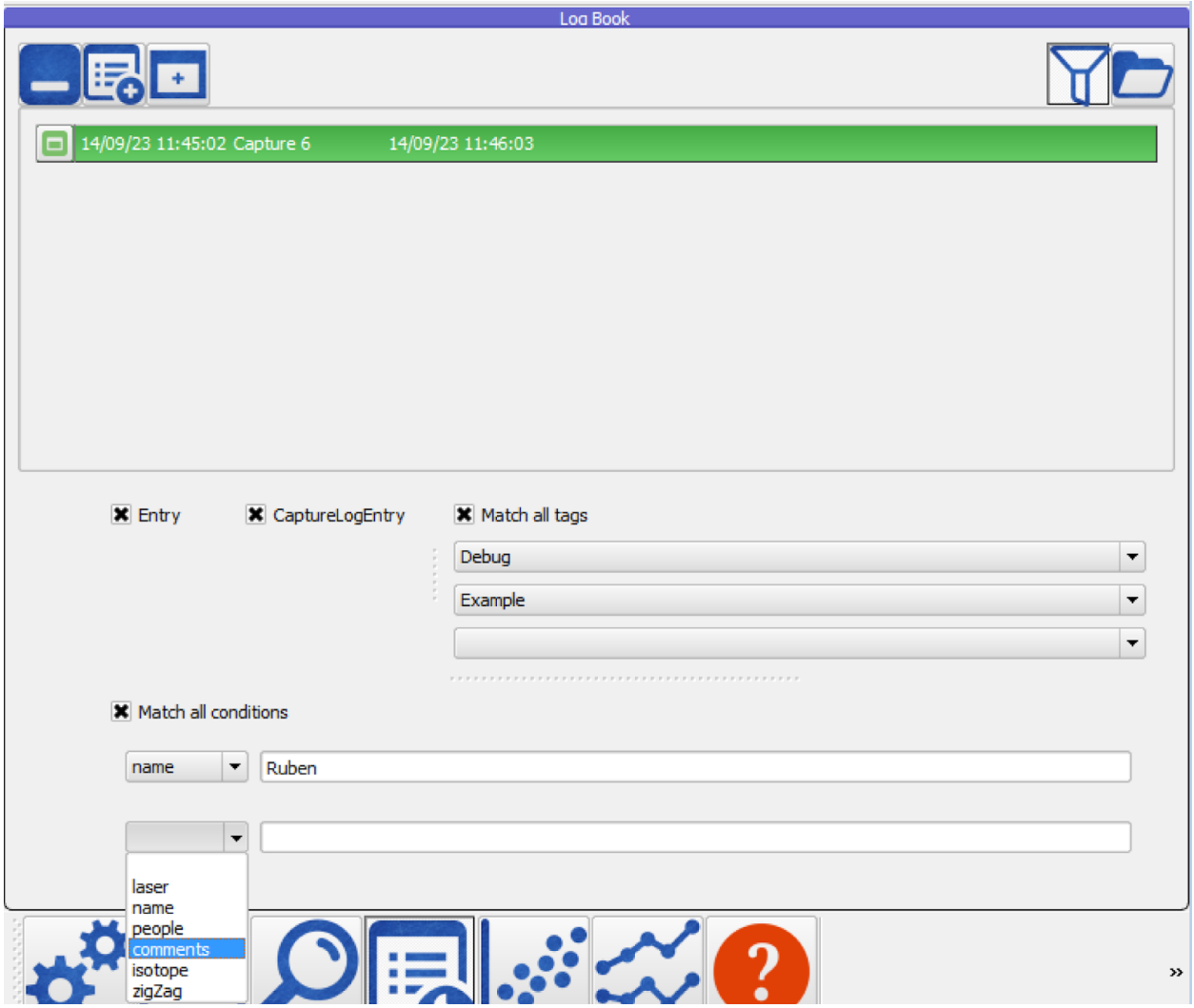
Each logbook entry can be shown or collapsed by clicking on the icon on the left of its name.

You can collapse all open entries by clicking on the top-left button.

Next to the collapse-all button you will find another button, which you can click to add an entry to the log that is not explicitly tied to a specific scan. Entries like this are colored blue rather than the green reserved for the capture entries.

The square with a plus sign in it allows you to define a certain property that you would like to always see written down in the log. In all following entries a textbox will appear, reserved for just this parameter.

The logbook can be filtered and searched through by clicking the funnel icon on the top right. Entries can be filtered on their type, their tags, and the contents of the various text fields you filled in.



You can open a different logbook by clicking on the folder icon in the top right. The logbook you choose will be used to write the new information to.

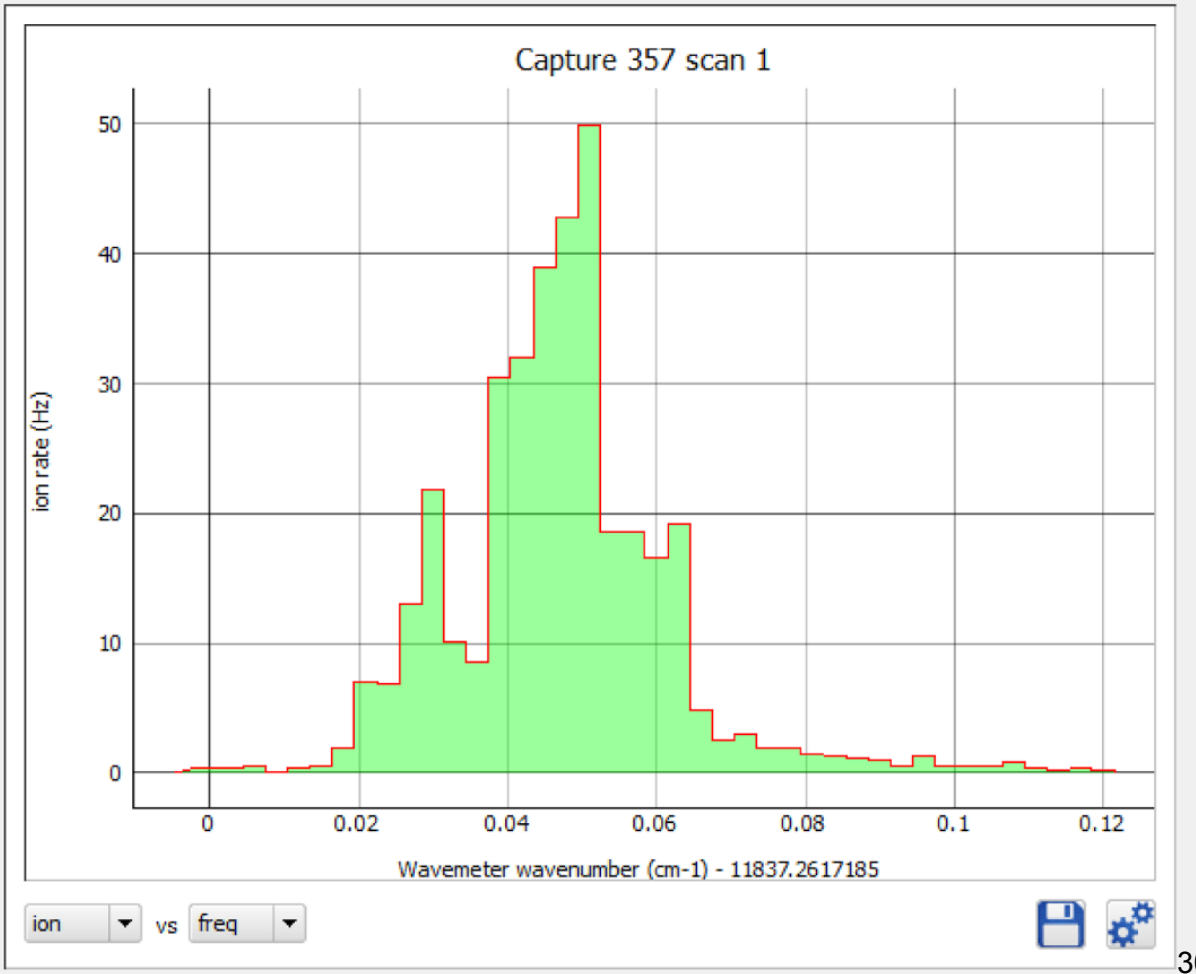
This is a typical log entry for a capture:

14/08/10 01:55:55 Capture 357 14/08/10 02:22:53

Right

people	Ruben, Thomas, Ivan
comments	Scan of the right peak of 205 Fr.
isotope	205 Fr
ISCOOL	40024 V
Laser Delay	92.320 us
ISCOOL delay	20162.5 us
Beam gate	Open

Confirm

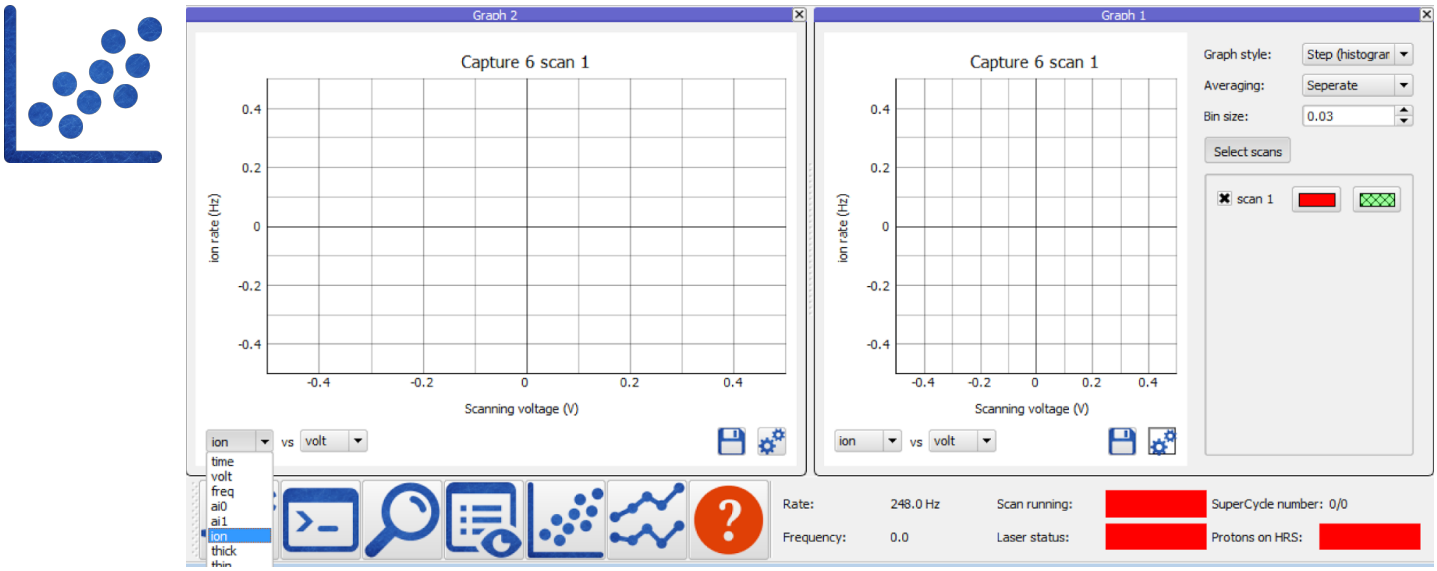


Each entry contains several text boxes where you write relevant information. You can always change previously made entries by clicking the `edit'-button and then confirming your changes. You can instantly see that this is a scan of 205 Fr, in particular the rightmost multiplet (note the tag). The ISCOOL voltage is easily spotted, as is the delay of the YAG laser relative to the RILIS master trigger and the ISCOOL delay. For this measurement, we can immediately see the ISOLDE beam gate was open.

The capture entries also have a graph button on the left which you can use to show the data from earlier measurements.

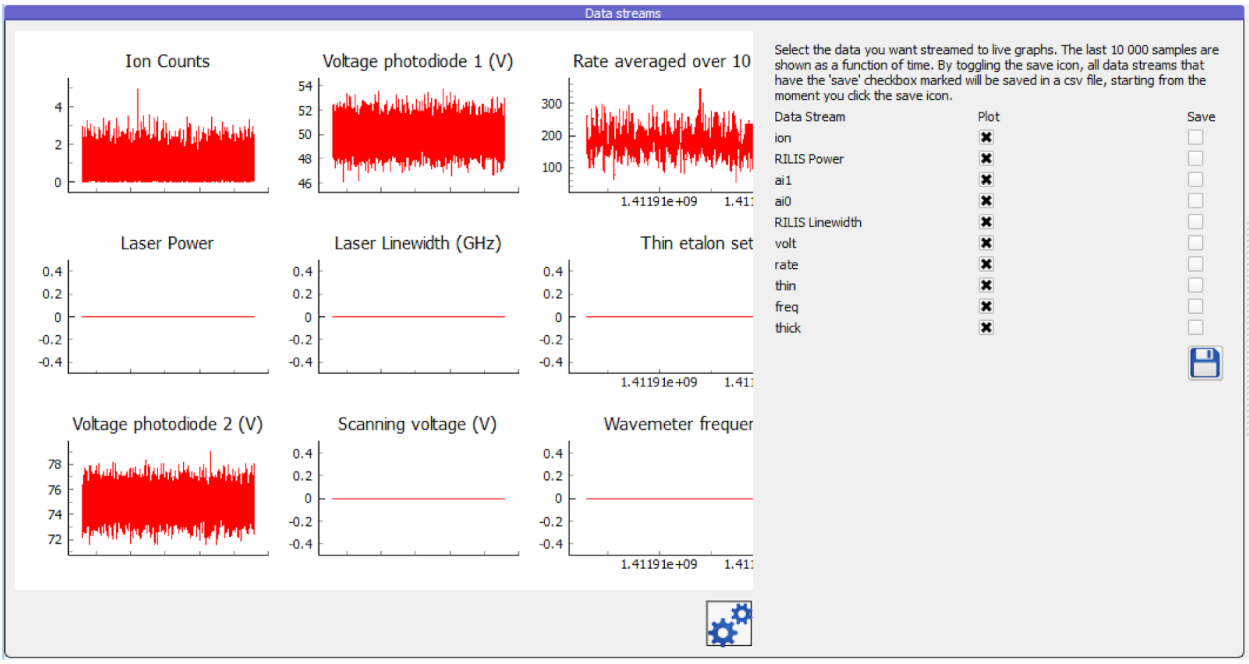
In addition, you can associate certain “tags” to entries as well. Think of these tags as ‘boolean properties”; in the above example, the entry has the ‘debug’ and ‘example’ tags. Practical examples of tags are e.g. ‘right multiplet’, ‘isomer hunt’ or ‘Empty’. Tags are useful when you want to filter through or search in the logbook.

Graph widgets



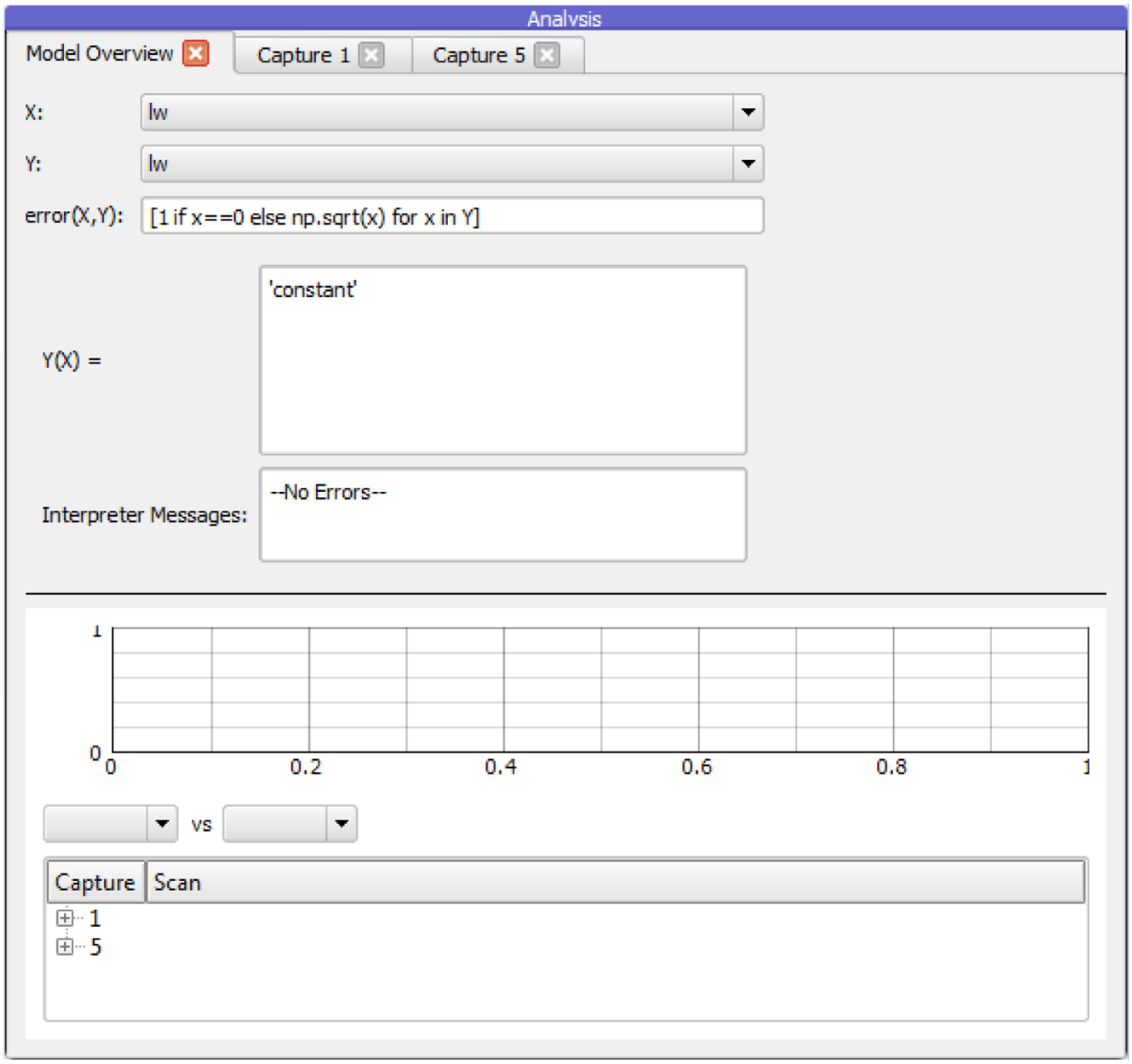
Every time you click on the scatterplot icon, a graph canvas is added to the UI. By using the combo boxes at the bottom left of the graph, you can choose what you want to plot (e.g. the ion counts as function of the wavelength, or the diode signal as function of time). By clicking the gears icon on the bottom right, you get access to some options. You can choose whether you want to see the raw data or rather the binned version, if you want to see all the scans within one capture, or rather their average. You can also hide or show individual scans, and change their color. These graphs can be closed by clicking the x button in the top-right.

Datastreams widget



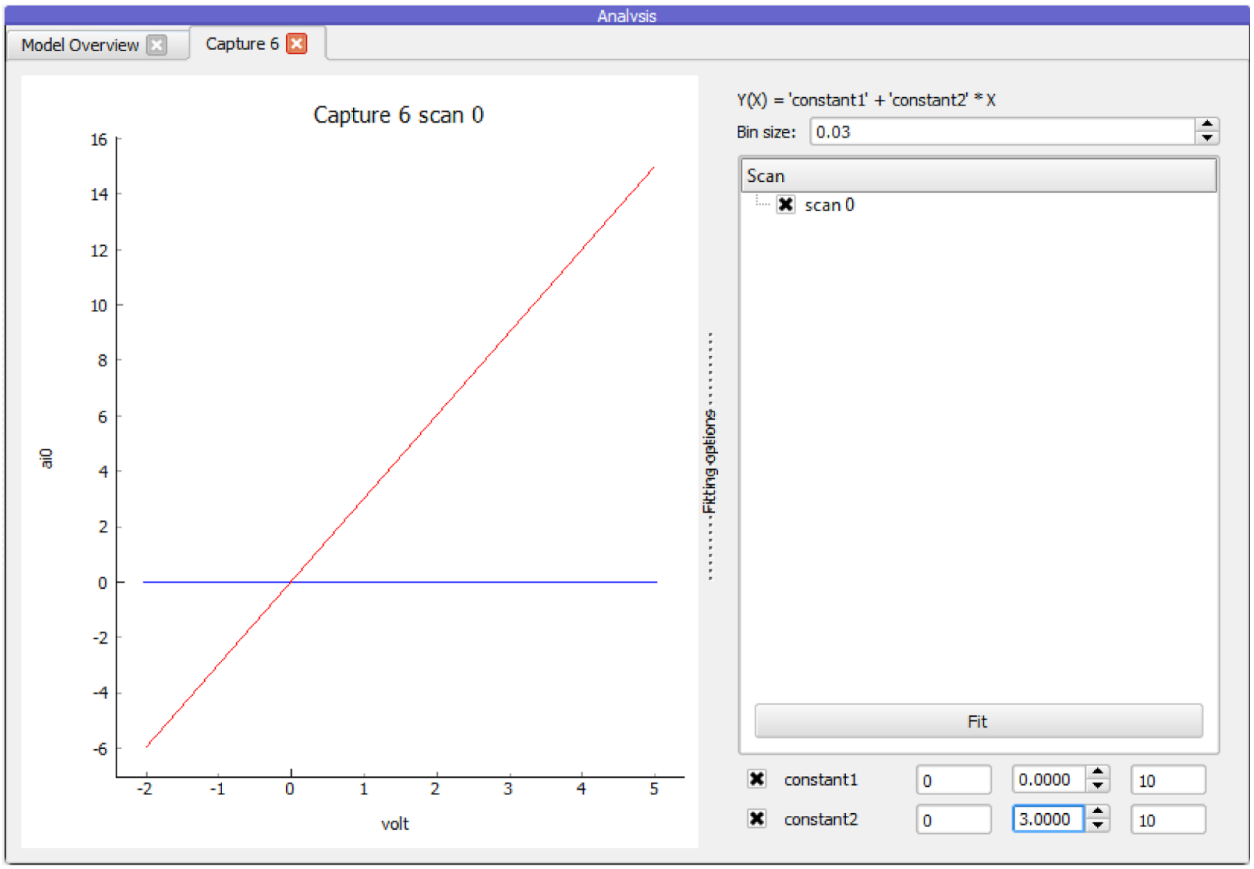
The line plot icon opens up a widget that shows you all of the raw signals the DAQ monitors as a function of time. The stream only goes 10000 samples back in time. By clicking the gears icon in the bottom right, you can show or hide specific data streams, as well as access a way to write the required streams to a file.

Analysis widget



CRISTAL's latest and most experimental feature. Expect this to change significantly in the future. **NOTE: THE CURRENT VERSION LOOKS NOTHING LIKE THIS - WILL UPDATE SOON.**

With the analysis widget you can analyse data. By clicking and dragging the green labels from the logbook entries onto this widget, a capture is added to a new tab of this widget. In the model overview panel, you can select the x and y data you want to plot and analyse, as well as define a suitable error function on this data. You can type in a fitting function as well. When doing this, use capital X and Y to refer to the data, and type the name of fit parameters between single quotes ('). On the bottom of the model overview you can plot the results of the fit (errorbar plots of the parameters), as well as find a table for all of the captures you added (per scan).



In the panel for a capture you will see plots of the data, seperated per scan. The checkboxes on the right enable or disable plotting and fitting of scans within a capture. Bottom-right provides text boxes and a spinbox to define the initial guess and boundaries on each fit parameter. The (binned) data is plotted in blue, the fit function in red.

HELP PANIC BUTTON AAH

Clicking this opens a pdf with all the information you found here as well. Hope that's sufficient help...

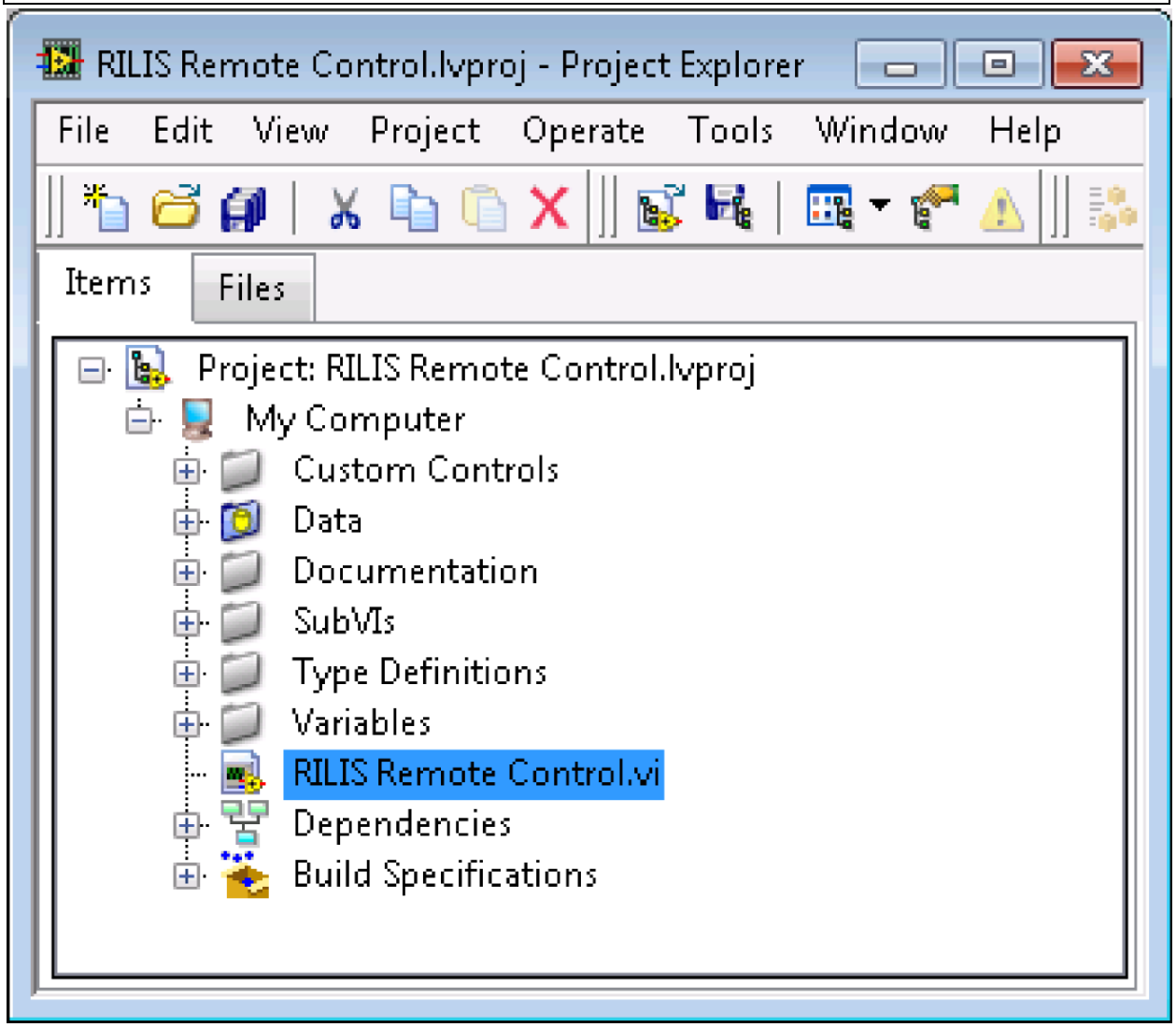
CRISTAL Code Structure

Dependencies: Required Software, drivers and Python Libraries

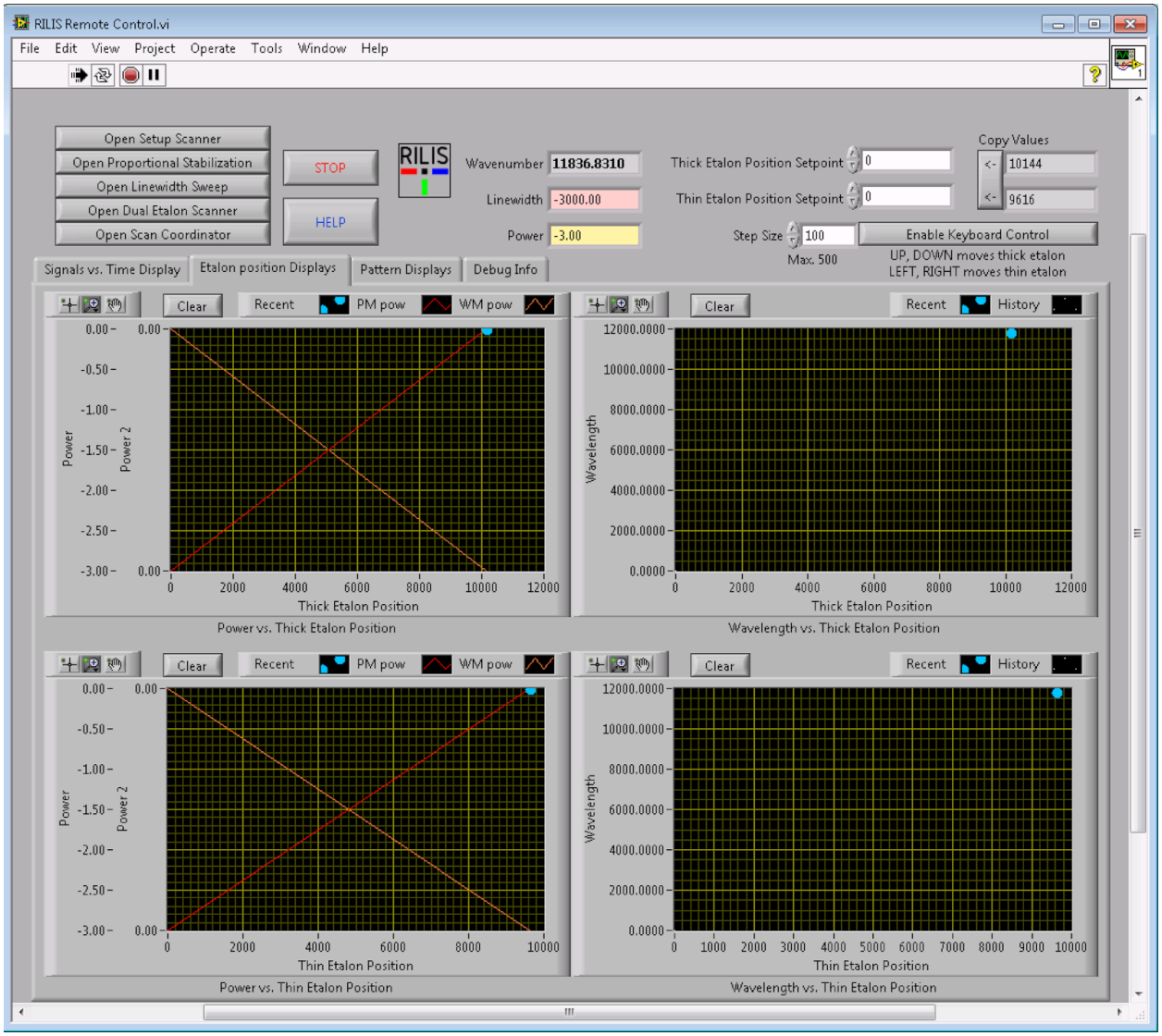
Communication with RILIS lasers

A big topic that deserves a section of its own. To scan the RILIS lasers, the RILIS Remote Control.vi is required. This VI can be found in the folder **C:\CRISTALCLEAR\CODE\RILIS Remote Control**.

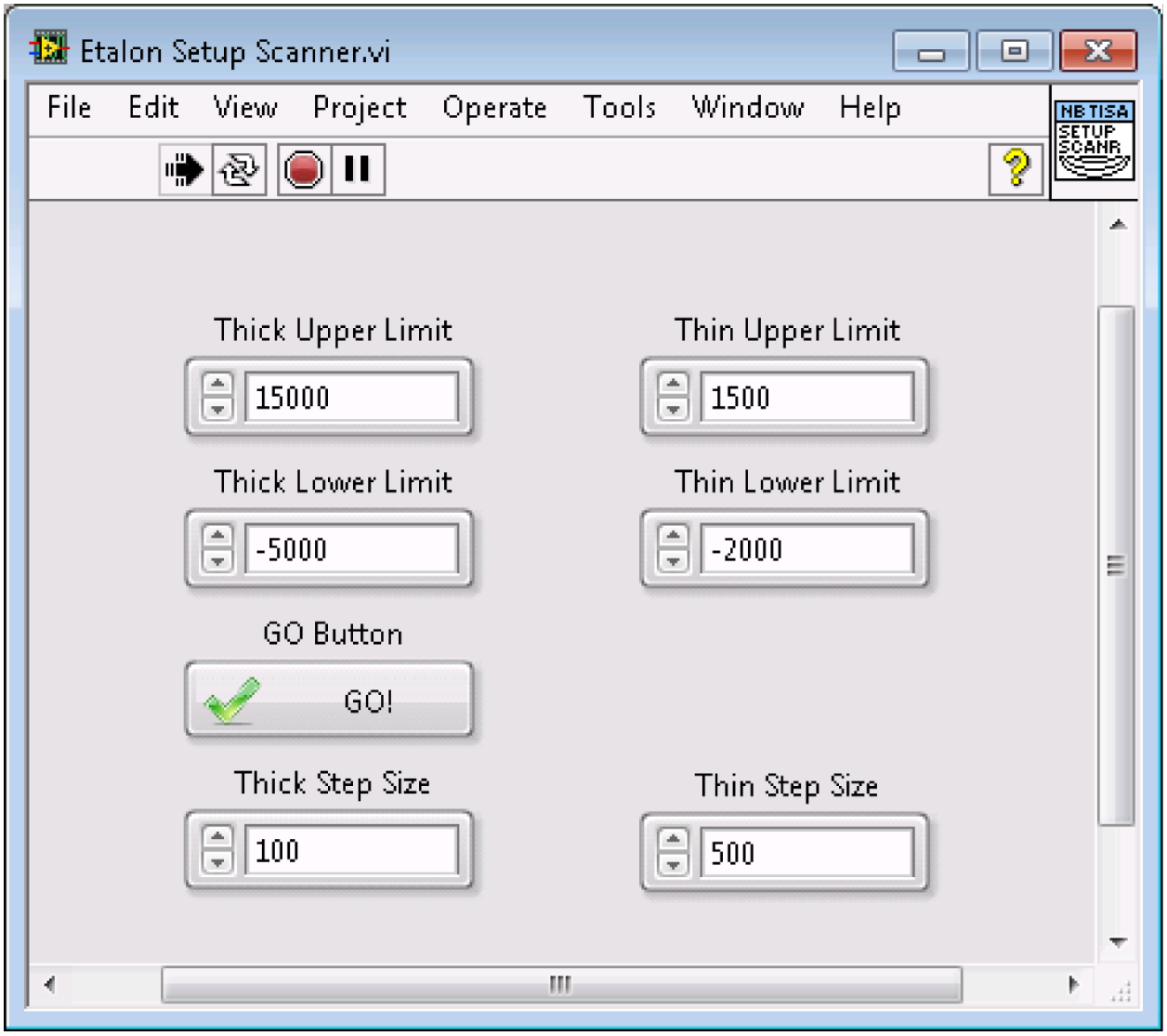
NOTE: Always start the labview PROJECT, and run the VI from there. This ensures all of the shared variables are properly hosted.



The second tab of the remote control VI shows four graphs. The two graphs on the right are most interesting: they show the laser wavelength as function of the etalon positions.

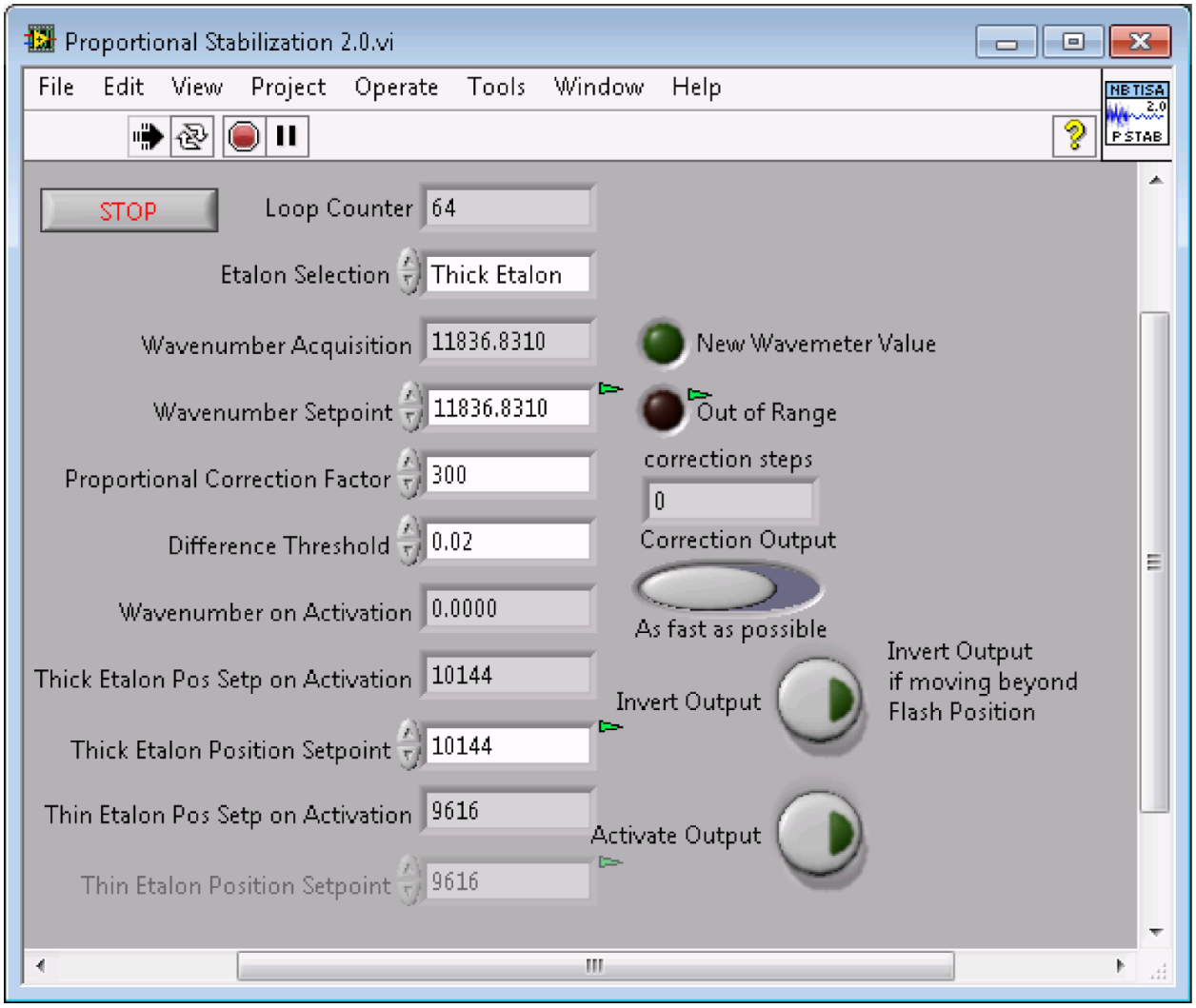


You can get a map of this etalon-wavelength phase space by opening the Setup Scanner (button on the top left).



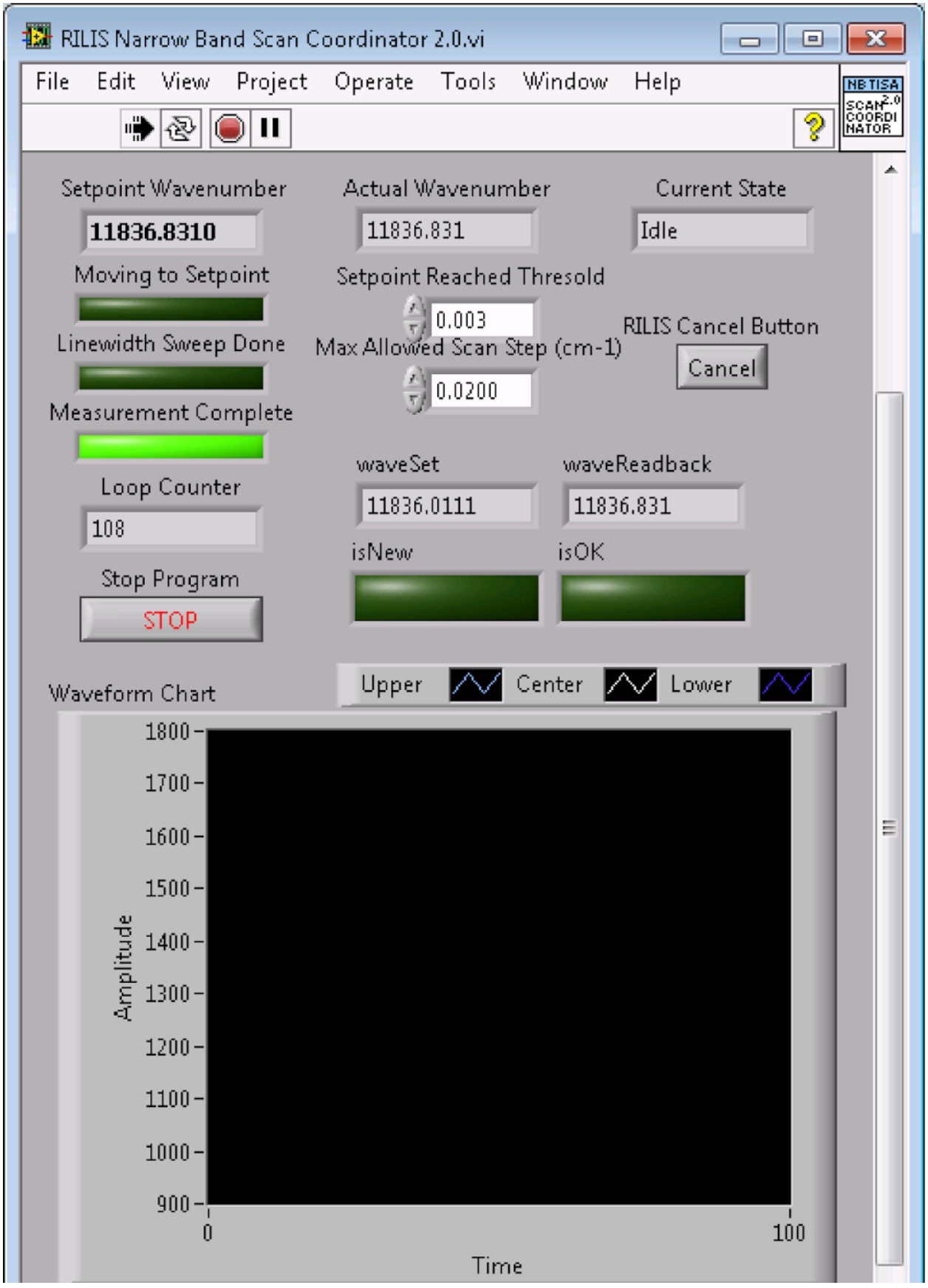
This VI simply scans across a range for the thin and thick etalons, and record the wavelengths. Ralf should have determined the optimal thin and thick etalon limits. If not, you will have to manually figure out what those limits are. You can change the etalon positions yourself using the two spinboxes on the top-right of the main control VI.

Once you made this map, you can close the setup scanner and open the proportional stabilization VI using the corresponding button in the top-left.

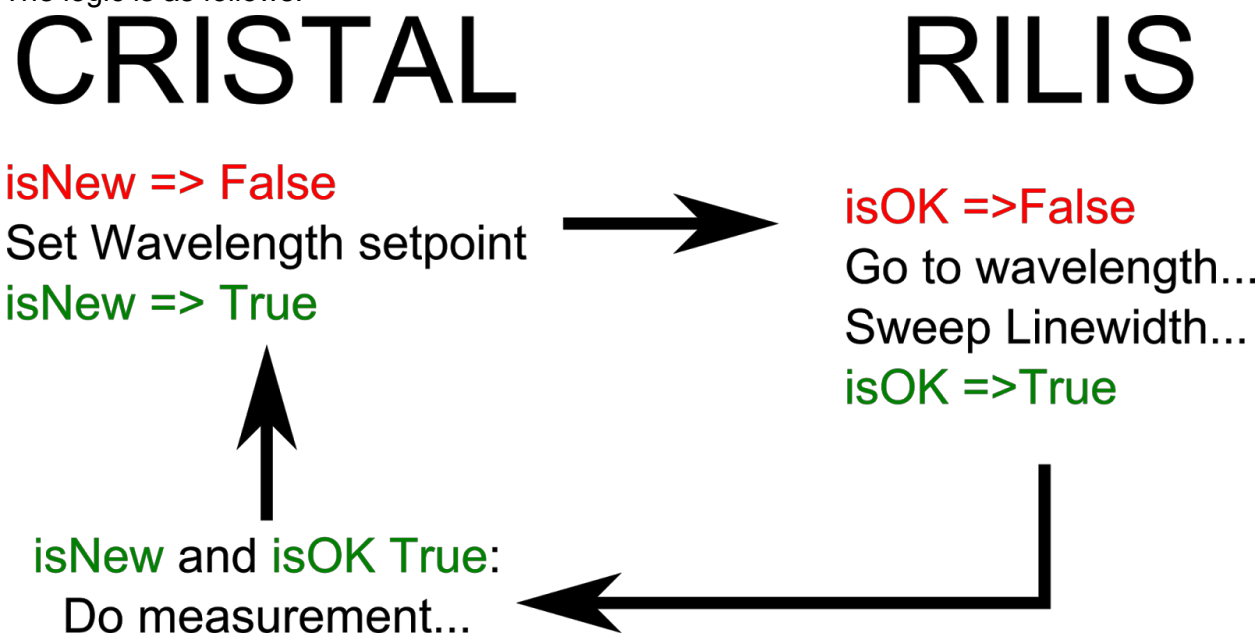


Just run this subVI, and activate the output using the button on the bottom-right. If you are on a downsloping curve of the etalon-wavelength map, you should invert the output. This subVI will make small adjustments on the thick etalon to keep the wavelength fixed. If the target wavelength is too far of the current one, the program will give flag Out of Range, and you will have to manually move the etalons until it is resolved, or slowly bring the setpoint down from the current wavelength to where you want it to be.

With just these VI running, you can fully control the RILIS laser. If you want to communicate with CRISTAL as well, you will have to run the Scan Coordinator VI (once again, button on the top-left).


RILIS Narrow Band Scan Coordinator 2.0.vi
File Edit View Project Operate Tools Window Help
NBTISA SCAN COORDI NATOR 2.0
Setpoint Wavenumber
11836.8310
Actual Wavenumber
11836.831
Current State
Idle
Moving to Setpoint
Setpoint Reached Thresold
0.003
RILIS Cancel Button
Cancel
Linewidth Sweep Done
Max Allowed Scan Step (cm-1)
0.0200
Measurement Complete
waveSet
11836.0111
waveReadback
11836.831
Loop Counter
108
isNew
isOK
Stop Program
STOP
Waveform Chart
Upper
Center
Lower
Amplitude
1800
1700
1600
1500
1400
1300
1200
1100
1000
900
0
100
Time

The logic is as follows:



Setting isNew to false trigger the VI to set isOK to false and change the wavelength to the new setpoint. Once this is done, isOK is set to True, which triggers CRISTAL to perform a measurement. Once this measurement is complete, the cycle starts over.

NOTE: There is no way to change the Scan coordinators mind: once it shows a setpoint on the GUI, it will not allow anything to happen until *that precise* wavelength is reached. So, make sure you enter the correct wavelength in CRISTAL! If the program is stuck on a wavelength which is out-of-range of it's automatic algorithms, you will have to manually set the laser by tweaking the etalons.

NOTE: It happens sometimes that the wavelength and etalon readbacks freeze when the laser wavelength is being changed. No worries, this is usually not a bad sign. I hope.

It can happen that there is a sync issue between CRISTAL and the labview code; somehow, both booleans are acting weirdly. There is a way to resolve this.

1. Close CRISTAL. Make sure the CRIS Relay VI is also closed.
2. Open a command prompt (Windows key + R, type cmd, press enter).
3. Run python (type 'python' and press enter). You started a python interpreter.
4. Type code below. It will start a communication process with the RILIS program, and manually overwrite the shared variables.

```
import OpenOPC
opc = OpenOPC.client()
```

```
opc.connect('National Instruments.Variable Engine.1')
opc.write(('CRIS local SVs for Python.PythonIsNew',False))
opc.write(('CRIS local SVs for Python.PythonIsOK',False))
```

5. Launch CRISTAL again and check. You might have to repeat the last two python statements, or play around with setting them True as well, until the sync issue is resolved. If it is, it is good practice to close the command prompt.

NOTE: Make sure the CRISTAL settings are set to control the RILIS laser rather than the cw system! If this is the case, the scanner widget should display values for the thin and thick etalon as well as a wavelength readback.

Troubleshooting

Golden rule of CRISTAL: When in doubt, open task manager, kill python threads, close the relay VI, try again.

I started the program as you explained, but no python window appears!
Check if you are sending the trigger into the NI card. Without this, the program can't start sampling from the card, which means it can't start initializing all the parameters.

I was not touching anything, I swear, but the UI got unresponsive!
If the UI got unresponsive, restart the program. If you can't even close it using the normal methods, just kill the python processes using the task manager. If you did manage to just close the window, be sure to kill possible ghost python threads anyway using the task manager. They tend to linger when errors are thrown.)

The command prompt spits out the following: The specified resource is reserved. The operation could not be completed as specified.
This means a task is still running on the device. To end the task and recover, open the NIDAQmx utility from Start => National Instruments => NiMAX. Select Devices and Interfaces => NI USB 6211 then select `Reset Device', and await confirmation.

The command line shows an error message!
Copy-paste-email-me! If it is an error relating to binning of data or plotting: I know these sometimes pop up; they're mostly harmless. Let me know anyway.

The UI is a bit sluggish.
Two options: the pc is bogging down due to god-knows-what-reason, or there are too many graphs being updated at the same time. Rebooting the computer can help with the former, just closing a few graphs with the latter (you data-junkie!).

I was playing around with the plotting options, and the graph just disappeared!
It could be that you set the bin width to be too small. Change to something more reasonable. Another possibility is that you changed what is put on the x- or y-axes, so that the graph is actually just out of view. If you hover over the graph canvas, you should see a small icon with an `A' in it appear in the bottom-right corner. Clicking this will attempt to auto-range x- and y-axes to bring the graph in view.

Also, make sure you didn't set the line color to white ;).

I attempted to plot the data from earlier measurements, but I don't see a graph/the UI gets unresponsive or slow.
If this happens, something went wrong during the data read-in from file. Not much I can do to help you if I'm not around. Just let me know (and consider not trying to look at that measurement again).

If the whole UI gets slow, just click on the graph icon again to hide the plot. If the UI got unresponsive, restart the program (be sure to kill possible ghost python threads using the task manager. They tend to linger when errors are thrown.)