friture Documentation

Release 0.19.0.8

Timothée Lecomte, James Collins

July 31, 2016

Friture is an application to visualize and analyze live audio data in real-time.

Friture displays audio data in several widgets, such as a scope, a spectrum analyzer, or a rolling 2D spectrogram.

This program can be useful to analyze and equalize the audio response of a hall, or for educational purposes, etc.

The name *Friture* is a french word for *frying*, also used for *noise* in a sound.

Source Code

Manual

Contents:

CHAPTER

ONE

INSTRUCTIONS TO GET FRITURE RUNNING

Most of Friture code-base is written in Python. As such, it is interpreted and does not need compilation, apart from the special cases below.

Dependencies

- Python3 (3.4 or later recommended)
- PyQt5
- PyAudio
- PyOpenGL
- PyOpenGL-accelerate
- Numpy
- Scipy
- Cython
- docutils
- psutil

UI and resource files

If friture.ui or resource.qrc are changed, the corresponding python files need to be rebuilt:

```
pyuic4 ui/friture.ui --from-imports > friture/ui_friture.py
pyuic4 ui/settings.ui --from-imports > friture/ui_settings.py
pyrcc4 resources/friture.qrc -o friture/friture_rc.py
```

Cython extension

Friture uses Cython extensions where computing is a bottleneck. These extensions can be built automatically with:

```
python setup.py build_ext --inplace
```

On Windows, if you use mingw:

```
python setup.py build_ext --inplace --compiler=mingw32)
```

Alternatively, to build the 'exp_smoothing_conv' extension manually on Linux, you can do:

```
cython exp_smoothing_conv.pyx
```

then (all as one command)

gcc -shared -pthread -fPIC -fwrapv -O2 -Wall -fno-strict-aliasing -I/usr/include/python2.6 -o exp_smoothing_conv.so exp_smoothing_conv.c

Filters parameters

The filters parameters are precomputed in a file called 'generated_filters.py'. To rebuild this file, run the script named 'filter_design.py'.

Windows executable

To build an executable for Windows, with the python interpreter included, run:

```
python setup.py py2exe
```

You get a 'dist' directory with friture.exe and a collection of dependencies (dlls and compiled python extensions), that can be bundled together in an installer.

CHAPTER

TWO

FRITURE CHANGELOG

(full changelog can be found in the git repository)

HEAD New:

- (generator) Add ramps at start/stop to avoid undesirable bursts.

Change:

- (delay estimator) Increase the delay smoothing for more reliabilty

Bugfix:

- (levels) remove the bogus factor of 2 on the rms. Now the rms of a sine is 3dB below its peak, as expected.

Version 0.6 - 2012/06/07 New:

- (general) Track the number of Xruns. Display it in the about dialog, in the statistics tab.
- (delay estimator) Cross-correlation is time averaged for robustness, better significance estimation, and phase info taken from the weighted cross-correlation too.
- (delay estimator) Use generalized cross-correlation computation, with PHAT weighting. Now the estimation is much more robust in a reverberating environment.

Change:

- (general) Increase sample rate to 48000 Hz. This simplifies and improves the quality of the filters in the octave spectrum.
- (octave spectrum) Fix midband frequencies, so that they match the ISO bands of an equalizer

Version 0.5 - 2012/05/26 New:

- (delay estimator) The maximum reliably-estimated delay is now configurable. Just in case somebody needs more than one second
- (delay estimator) Replace the peak detection by cross-correlation The cross-correlation is computed on (largely) subsampled data. It not only provides the delay, but also the in-phase information, and a measure of the correlation quality (value of the normalized correlation coefficient).
- Display the level labels with a 250 ms period. This reduces the work that needs to be done (drawing text is costly), and makes the values much more readable.

Version 0.4 - 2012/05/18 New:

- First embryo of documentation
- Add PyPI requirements.txt file, so that friture dependencies be installed by 'pip'
- New text file for user-readable changelog

Change:

- (delay_estimator) Optimization

Bugfix:

- (generator) Test for zero mean before using it as a denominator
- (Pink noise generator) Fix the parameter k k, the number of white channels to sum, was dynamic, and this was causing changes of colors in the produced noise, changes that could be heard.
- (Generator) Fix exception in pink noise generator. Now returns empty array when asked for n=0 samples
- (Generator) prevent overflows at -1. Premultiplying by 0.99.
- (Generator) time array was not incremented properly.

Version 0.3 - 2012/04/29

- (delay estimator) also display the delay in meters
- (py2exe) Exclude more dll from the package
- Increment version
- Add version and release date in the about page.
- Properly delete widgets when closed or changed.
- (py2exe) do not exclude subprocess; subprocess is now required by psutils
- (py2exe) do not ignore difflib; difflib is now required for numpy
- Do not redraw the delay label if the text has not changed.
- Disable the linear screen interpolation in timeplot. This was bogus signal-wise, and practically prevented from displaying some data correctly.
- Introduce a variable for the time window in the scope.
- Remove the minus peak in the burst generator.
- Remove the bogus time inversion in the scope.
- Fix message for delay estimator in single channel.
- Fix detection of peaks in the delay estimator.
- Add a new widget to measure delays. Detect peaks in two channels and measure the time difference between the two.
- Enlarge the trigger window (and remove leftover debug statement)
- Change scope to trigger on center of window, improve triggering.
- Fix buffer growth in ringbuffer.py
- Allow negative times in the scope horizontal axis
- Fix indentation
- Immprove the burst generator
- Make the computation cleaner, and double its amplitude.
- Move the norm computation (after fft) to own function so that it appears on the profile directly
- Move the sweep generator in a separate class
- Pass 0 to cpu_percent to make it non-blocking
- Fix indentation

- Add more comments about instructions for PyPI
- Add some usage comments to setup.py, and add a little text to the description

Version 0.2 (Windows 2011/10/31 - all platforms PyPI 2012/01/04)

- Add Numpy include to Cython extension building
- Fix audiobackend reference for the generator when launched in the central widget
- Increase pink noise fidelity
- Cleanup generator import (avoid scipy import)
- Fix py2exe inclusion of OpenGL packages
- Fix integer comparison in pink noise code
- Plug the generator dock into the dock widgets
- Generator dock
- Add output devices listing in audiobackend class
- Update properly the histogram spectrum coordinate transform when the scale changes
- Avoid calling the filters with empty input arrays
- Fix use of the decimate function in the no-initial-conditions case
- Define a separate decimate function
- Enable the filter initial conditions propagation for correct filtering
- Ring buffer size can grow (fix issues with long delays)
- Fix qwt/opengl/numpy widget for arrays proper initialization (actually fix multiple simultaneous FFT widgets)
- Add a spectrum GUI setting for single/dual channels
- Introduce a delay setting that acts one the spectrum (for now)
- Save and restore the input device/channel/single//duo configuration
- Simplify baseline computation (and remove substraction warning
- Make the about dialog a child of the mainwindow so that it is closed automatically when the mainwindow is closed.
- Some preliminary code for baseline setting
- Some preliminary code for nicer shading
- Fix qwt/opengl/numpy plot for linear scale
- Add legend to the scope when two channels are used
- Two channels mode for the level meter
- Two channels for the scope
- Make the spectrum draw the power difference when two channels are used
- New feature : two-channel difference as input
- Fix background gradient in spectrum
- Enable the qwt/opengl/numpy spectrum plot instead of the original curved one
- Add cumulative time profiling stats

- Put the individual icon files in the repository

Windows Version 2012/05/26

- Qt plugin for svg icons (qsvg4.dll) now depends on QtXml, thatpy2exe failed to include automatically

Windows version 2012/05/17

- Fix histplot transforms initializations (removes errors on startup) and make transformation more performant
- Use array operations for the canvas transformation
- Fix add-dock toolbar icon, which was badly rendered on windows
- Fix dock settings icon, which was badly rendered on windows
- Convert to MUI2, fix encoding, improve uninstaller
- Nice microphone photo for the splash screen - license is cc-by-2.0
- Remove non-updated statistics
- The current working directory no longer needs to be set at the end of install
- generated_filters.pkl is no longer to be installed
- Store the filter params in a true python file, simplifies the import
- Fix vu-meters that decrease indefinitely

Version 0.1 on PyPI - 2011/02/02

- Fixes for non-py2exe setup
- Improve setup.py for distribution, with classifiers, startup script and manifest
- Allow friture.py to be executed as a child of another script, such as lsprofcalltree
- Fix off-by-one pixel for the level meter
- Remove bogus styling of the level meters
- Let python manage the spectrogram painter object lifetime
- Ordering for object creation fixed
- Move friture.py to subdir
- Move scripts to subdir, remove lsprofcalltree from friture.py
- Move Cython extension to subsubdir
- Move source files to subdir

Windows version 2011/01/29

- Fix uninstaller, improve installer info
- Use packaging date for the installer version
- Fix for the installer move
- Move some source files to subdir
- Move .ui files to subdir
- Move ui*.py and resource py file to friture subdir
- Move the installer and Microsoft redist pack to a subdirectory

- Update install instructions
- Move test files to sandbox subdir
- Move image files and resource file to a subdirectory
- Move installation details to a separate file
- Move profiling info to separate file
- New colors for FFT spectrum, less agressive
- New style for octave spectrum's peaks
- Add 'central dock' label to clarify what is the central widget

Windows version 2011/01/09

- Stop py2exe from complaining about msvcp90.dll
- Add manual bounds ckecking in exponential smoother
- Add code to include numpy headers when building the cython extension on Windows
- Major cleanup and optimisation in the histogram plot
- Better peak meter, remove integer conversions
- Add exponential smoothing (0.125 s) to the RMS level widget
- Cleanup and optimization in octavespectrum update
- Fix bug : could not add any dock when all of them were removed
- Major performance improvements in histplot by caching and avoiding QwtInterval
- Specific import from PyQt4.QtGui to improve startup time
- Add Cython machinery to setup.py
- Make the exponential smoothing smarter, and implement it as a cython module (dramatic speed improvement)

Windows version 2010/11/08

- (NSIS installer) Fix Working Directory when starting Friture at the end of the installation

Windows version 2010/11/04

- Optionally remove settings when uninstalling (NSIS)
- Improve dock deletion, and dock index selection
- Use a real exponential moving average at the output of the octave filters
- Fix settings initialization for the 2D spectrogram, fft and octave spectrum (Note: could be done slightly smarter)

Windows version 2010/10/25

- Clean and fix errors about bar widths in the octave spectrum
- Fix minimumSizeHint computation for octave, time and spectrum plots from a missing replot() at init end
- Add tooltips to the control bar
- Update splash screen with logo
- Tweak FFT spectrum default settings
- Tweak defautl settings for the octave spectrum
- Fix default settings for the 2D spectrogram (log frequency scale)

- Remove levels and scope widgets from the default set of widgets shown on the first launch
- Improve the profiler info
- Add an option to set the reponse time in the octave spectrum.

Windows version 2010/09/29

- Check if the bar pixmap needs to be updated only once per timer update.
- Use an opaque color for the FFT spectrum plot brushes, disables the stroke, and draw the grid on top.
- More generic NSIS script, more robust ot my own mistakes !
- Even more precise import for lighter py2exe distribution
- Tweak About message, add email address, remove name.
- Fill the peak curve in the FFT spectrum
- Import from scipy.signal.sigtools and scipy.signal.filter_design instead of scipy.signal
- Import lfilter from scipy.signal.sigtools instead of scipy.signal
- Hyperlink for the homepage address in the about widget

Windows version 2010/07/14

- Exclude unused modules from py2exe
- Include the filters file in nsis distribution
- Separate filter design and use in two different source files.
- Import more selectively. Hopefully helps py2exe
- Include the filters coeff file in py2exe packaging
- Add X-grid to the octave spectrum
- Tweak widget names
- Display a colorbar next to the spectrogram plot
- Fill under the spectrum curve, makes it a little more readable
- Recreate ring buffers when necessary only
- Do not call len() multiple times
- Avoid checking for proper ringbuffer at each timer tick.
- Use pre-generated filters coeff instead of recomputing them at each change.
- Disable filter conditions propagation for now.
- Introduce peaks in the octave spectrum.
- Tweaks to the decimating filters analysis.
- Go for order-3 decimating filter
- Go for 50 dB stopbands
- Better analysis of the filter bank frequency response
- Go for a non-decimating filter bank since the frequency output of the decimating one in close to catastrophic...
- Introduce zi/zf to the filtering without decimation
- Code for visual analysis of the filters frequency response

- Use log10 scale engine while the log2 engine still has references issues.
- Switch to 80dB stopband filters to make the octave analyzer nicer with single frequency excitation.
- Simplify histogram draw by removing the horizontal bar feature.
- Fix canvas height caching
- Implement ring buffer for filtering - seems like there's an issue with the filters stability
- Move the ring buffer class to its own file for reuse
- Move octave filtering to a class
- Add audiobuffer method to get the new points only
- Fixed and improvements to the histplot pixmap cache
- Cache bar in a pixmap.
- Spacebar is a shortcut for start/stop
- Move from 7 octave to 8 octave

Windows version 2010/06/29

- Allow 1/3, 1/6, 1/12 and 1/24 octave filters, with proper weightings.
- Implement IIR+decimation instead of IIR only. Will be especially useful for more-than-one bands per octave.
- Make bands number automatically computed from the bands per octave choice.
- Add Hann window to the fft code for better frequency resolution.
- Implement an octave-band filter widget.
- Finish and cleanup implementation of an octave (or fraction of octave) filter bank
- Move scope trigger to the left edge
- Move the log to a dedicated tab in the about dialog
- Make the statistics widget part of the about window instead of a dock widget.
- Start/Stop for the spectrogram timer.
- Spectrum scale defaults to log10.
- Spectrogram default to log10 scale.
- Move About code to source file instead of ui file, and prepare for tabing.
- Fix start/stop button text.
- Fix the start/stop icon state.
- Fix minimum size of the level widget.
- Display the weighting state in the spectrum y-axis title.

Windows version 2010/06/16

- Add human middle-ear weightings A, B and C to the spectrum and spectrogram settings.
- Put common code for the docks control bar in a separate file.
- Allow to select the channel for any device stream with more than one channel.
- Bugfix : actionStart belongs to self.ui

- Reorder icons in the toolbar
- Pass the logger to widget classes
- Monitor the global CPU usage in the statistics widget. This may deserve its own widget, with something like QProgressBar. The feature is based on psutil, which becomes a new dependency for friture !
- Set size policies of scope, spectrum, spectrogram so they can be shrinked
- Move ui code from Ui_xyz.py to xyz.py directly

Windows version 2010/06/06

- Fix to the vcredist section in the installer script.
- Lighter gradients in the widgets.
- Tell the user that the levels and scope widgets have no settings yet.
- Put the VC++2008 redistribuable libs in the tree.
- Install VC redistribuable libraries.

Windows version 2010/06/04

- Draw a white background on the plots mouse trackers.
- New log icon with an exclamation mark
- New icon to add a new dock
- Use About Friture instead of just About for the About button
- Use the friture icon for the About button
- Comment out the toolbar styling for now, not that beautiful
- At first launch, do not display the log and stat docks.
- At first start the central widget is a spectrogram.
- Default set of docks at first start.
- Use smaller fonts for the plot axis titles.
- Remove the docks separator styling, not very nice.
- Update the installer file list.
- Automatically distribute the Qt svg plugin.
- Exclude powrprof.dll in py2exe distribution.
- Put the settings icon on the left.
- Do not fail when ctypes cannot find the dll
- Comment out the stylesheets for now, and use stock dock widgets instead of using a custom titlebar
- Set the separator style to black 1 pixel.
- Apply some styling to the toolbars so that they look more custom.
- Slightly change the plots background gradients to nicely merge with the grid.
- Set central layout margins to zero.
- Pass the logger to the spectrum, scope, levels widgets. Do the same for all widgets in the central widget too.
- Add a dummy scope and levels settings dialog.

- Fix spectrogram initial settings.
- Implement the spectrum y range settings.
- Fix spectrum initial settings.
- Remove previous static scope, spectrum and spectrogram widgets
- Save and restore central widget state
- Remove the settings buttons from the spectrum and spectrogram widgets
- Now the central widget is very close to the dock widget in feature/aspect
- Move spectrum stylesheet from mainwindow ui file to spectrum code file.
- Move scope stylesheet from the global ui file to the scope file
- Add a settings icon to the dock.
- Add an icon to the “dock” button.
- Dock icon based on svg objects, no contour or path.
- Custom icon for dock/undock.
- Use an icon for the dock close button.
- Add (modified) about svg.
- Restore state before restoring geometry.
- Save and restore the window geometry.
- Save and restore dock states.
- Add the possibility to choose the dock widget type on creation.
- Save and restore docks existence and positions.
- Assign a fixed height to the dock control widgets.
- Add a dock button to the floating dock.
- Stylesheet the dock control widgets.
- Add another special control widget to a floating dock.
- Add a custom toolbar as the dockwidget titlebar.
- Enable the spectrogram timer in a dock
- Enable the code to select a widget in a dock.
- Put the Dock code in a separate file
- Connect the new dock widget to the timer, pass it the audiobuffer.
- Directly connect the widgets to the display timer.
- Pass the audiobuffer in separate functions instead of passing it when asking for the widget update.
- Connect the statistics update function directly to the timer tick
- Move the buffer update to a specific function, directly connnected to the timer tick
- Actually add a dock and a widget (needs to be connected to the timer tick still)
- Move settings for spectrogram and spectrum to separate classes and window
- Move audio backend code to a separate file

- Move audio initialization to a separate method
- Use isVisible() functions instead of visibilityChanged signals.
- Refactor the frequency range code
- Move the spectrogram code and ui in separate files
- Move spectrum to separate code and ui files
- Use numpy fancy indexing to compute the peak falloff
- Move scope widget to separate .py and .ui files
- Fix for levels statistics
- Move the level widget code in a separate source file, and in a separate ui file.
- Make the log text selectable
- Scroll the log widget so that the last log line is visible
- Set the log and statistics widget backgrounds as white
- Add a line number to the log messages
- Remove log base 2 frequency scale
- Move audio ring code to a separate file.
- Source file encoding fix
- Factorize the stream opening function.
- Comment functions as slots or methods. Will help me to separate methods to their own appropriate module if applicable.
- New Logger class, built to handle log messages from all the program classes.
- Put “Settings” dialog code in a separate file.
- Put the “About” dialog code in a separate file.
- New titlebar widget mockup. Plan: one dockwidget = one visualiation placeholder.

Windows version 2010/03/20

- Remove dash from Windows installer filename.
- Automatically use the date as the Windows installer name.
- Add the icons to the source versioning, and fix setup.py
- Add icon to the Windows exe.
- Workaround to properly display the icon in Windows 7 taskbar.

Windows version 2010/03/17

- Special case for stderr on Windows, logging to a file.
- Add custom icon for the About icon in the toolbar
- Add an icon in the About dialog
- Add an about dialog.
- Update window icon
- Move some messages to the new logging window.

- Add logging window
- Cleanup stylesheet
- Display text labels in the toolbar, in a white font.
- Use solid lines for plot grids
- Disable latency logging by commenting out (forgot to disable the file creation, caused permission problems on Windows).
- Update NSIS script with proper license text, and installer name that include the version number.
- Update levels icons with gradient and white offset.

Windows version 2010/03/16

- Update statistics icon.
- Add a grid to the spectrum
- Add a grid the scope
- Disable latency logging facility for now.
- Add small radius to the border corners of levels, scope and spectrum.
- Add stylesheets for scope, spectrum and levels to make them a little fancier, and learn how to use CSS.

Windows version 2010/03/15 (executable & installer not publicly released)

- NSIS script to generate a install exe from a py2exe distribution.
- Special case for IOError on stream read. To be more specialized to the input overflow case.
- Add frame corders to all qdockwidgets, so that they are more easily moved.
- Add a “cool modern” linear gradient as a toolbar background.
- Put the statistics in a QScrollArea to gain some space.
- Change settings dialog title and icon.
- Update window icon.
- Add icon to the settings menu
- Update start/stop icons

Windows version 2010/03/12 (no installer, executable not publicly released)

- Add some excludes to setup.py:py2exe
- Comment out cochlear code since it is unused for now, and causes py2exe to include a lot of scipy, if not all.

First ever Windows executable (not publicly released) 2010/03/11

- Workaround a bug in scipy imports that appears when using py2exe.
- Add code for cochlear processing. Not enabled yet, may not be the best idea after all.
- Add a file with an implementation of the gammatone filter bank in numpy (currently only half-converted from Matlab)
- Add code for logarithmic scope, not enabled yet. A better solution could be to dynamically adjust the scale, using the same kind of algorithm as for the peak computations.
- Finally fix off by one color computation.
- Ask Qt not to update the canvas background on each paint event.

- Use astype instead of digitize, which is mush slower because it can do so much more. Nest step could be to use a function instead of a precomputed palette.

2009/11

- Small simplificaton in UI layout code
- Use numpy clip function
- Add profiling instructions at the top of friture.py
- Add a script to convert the pstats output to a graphviz dot file, which can in turn be converted to a png
- Replace a boolean mask with a simple substraction
- Cache the decimation computation to avoid a log2.
- Replace uses of .max() with [-1] where possible.
- Do the dB conversion once instead of twice for level widgets
- Remove a useless sqrt since we do a log10 just after.
- Cache the canvas object for all plots, since it appears to be a slow call in qwt.
- Add a window icon, currently very basic
- Simplify peak computation
- Introduce one more cached array for peak computation
- Decimate as much as possible before doing the fft
- Remove log10 usage from peaks computation in the spectrum
- Just import what is needed in audioproc.py
- Cache the frequency scale in audioproc
- Move the spectrum processing to its function.
- Move scope processing ot its function.
- Move level processing to its function
- Remove the use of the status bar to gain screen spece. Use the plot picker instead.
- Move the frequency scale definition to the processing function. This is in preparation to the use of decimation to decrease computing time for fft of small upper range.
- Compute the peaks directly on the interpolated spectrum. Fewer peaks to compute when the fft size is larger than the number of pixels in the plot, which is a common case.
- Move the computing of peaks to a specific function. That way it appears independently in the profile.
- Use command-lien arguments to select the profiling tool. Default to no profiling.
- Bug fix : stream is in int16 format, not int32 ! This fixes the frequency scale factor of 2 that was observed.
- Use SAMPLING_RATE instead of hardcoded number.
- Use rfft since the input is real.
- Cleanup spectrum code.
- Add basic trigger capability to the scope widget
- Move the default position for the statistics widget to the left.
- Replace the settings button with an action in the toolbar.

- Move settings to a separate dialog. Greatly cleans up the interface !

2009/10

- Disable antialiasing fro scope and spectrum. Dramatically improves the performance !
- Intelligently (i.e. with interpolation) limit the number of points to be drawn on the plot to the canvas width (in pixel units). Avoids the spectrum painting time to skyrocket for high fft sizes
- Add statistics field about spectrogram timer period
- Change statistics description to be more meaningfull
- Add timing statistics about computation and display
- Add file for log2 scale engine
- Disable plots decimation since it raises issues with the spectrum plot and its peaks
- Add time information for each timer events
- Factorization of log10
- SYnc to qsynthmeter.cpp rev 306, adds gradient look
- Simplify fft size save/restore
- Fix frequency scale saving
- Fix the frequencies labels formatting
- Fix the frequency label formatting
- Save/restore color ranges
- Save/restore time range.
- Save/restore frequencies minimum and maximum
- Save/restore the frequency scale
- Save and restore the fft size when closing/starting Friture.
- Move state save/restore to specific functions
- Add base 2 logarithmic frequency scale for the spectrogram. Warning: the frequency scales drawn on the side of the graph are wrong ! I am waiting for a bug fix in PyQwt before implementing them.
- Change the frequency range for the spectrogram too.
- Add two spinBoxes for frequency range selection. Only works for the spectrum for now.
- Uncheck actionStart when changing input device
- Major timer redesign. Now we have two timers: one for 1D and 2D widgets (levels, scope, spectrum, statistics), and one for 3D widgets (spectrogram). Using an audio buffer avoids latency issues. This audio buffer behaves like the spectrogram pixmap. It is a circular buffer, with data written twice.
- Add statistics about FFT period.
- Log to a file the time delay between spectrogram updates. Add a script to analyze it, called hist.py
- Add new file “hellogl.py” to experiment with opengl drawing.
- Save and restore dock and toolbar states on exit and restart, respectively.
- Allow to modify the time range of the spectrogram.
- Fix the logic of lost chunks in the timer slot.

- Make the time range a parameter in the CanvasScaledSpectrogram class.
- Add a flag to indicate that we are processing the last chunk of the timer fire, so that if we are not we do not bother updating the screen.
- Replace while loop with a for loop, allows to know where we are in the process.
- Store the number of available chunks in a variable instead of calling the interface multiple times.
- Add a statistics entry about mean number of chunks by timer fire, in the last 1000 timer fires.
- Add a latency measurement to the statistics.
- Add icon for the statistics button in the toolbar. Now we have icons for all buttons.

2009/09

- Add profiling option with console output instead of a file for kcachegrind.
- Add an icon for the spectrum action.
- Do not update the spectrum when the widget is not visible.
- Do not compute and update the levels when the widget is not visible.
- Do not try to display the statistics when the widget is not visible.
- Do not update the scope widget when it is not visible.
- Add svg icons for levels and scope widget.
- Add an icon for the Stop button.
- New resource svg for the Start button. Also rename the resource file to resource_rc.py since it's what Qt Designer asks for.
- Use QDockWidget::toggleViewAction instead of our own actions with signals and slots.
- Fix FFT size in UI
- Add a __del__ method for explicit deletion of the spectrogram painter and its pixmap. Avoids a segfault.
- Add a setup.py file with basic commands for py2exe. Allows building an exe for Windows.
- Add resource.py to the versioned-controled files.
- More fine tuning to the level widget layout. Move it to the left.
- Fix Level widget layout.
- Fix minimum size for the qsynthmeter widget.
- Small adjustment to level widget layout. Needs more work.
- Remove frame around spectrogram. Fix statusbar.
- Add tollbar buttons to toggle the display of individual widgets.
- Move the Start/Stop button to a new toolbar. Move the scope, spectrum, level and statistics widgets to dock widgets.
- Dynamically choose the maximum number of chunks to read by timer tick based on the latency reported by portaudio.
- Handle up to three chunks in a row instead of discarding the additional ones.
- Be a little more verbose for statistics, to be clearer.

2009/08

- Add a link to the wiki in README
- Rework the layout of the level meter a little bit.
- Add peaks to the spectrum widget
- Derive qsynthMeter from QFrame and give it a raised look
- Int conversion cleanup
- Use default for size hints of levels.
- Change the way audio levels are displayed : put them closer to the level widget.
- Remove the display of the max of the current audio data.
- Fix the time axis on the spectrogram widget. Now set to 10 seconds for every choice of the FFT window length.

2009/07

- Add a spacer in the first horizontal layout to give a more natural size to the buttons.
- Use only integers in frequency scales.
- Move buttons at the top of the layout for better alignment of the widgets.
- Use custom labels using 'k' for thousands in frequency scales widgets.
- Switch from seconds to milliseconds in the signal widget.
- Add pointer info in the status bar when clicking in the signal or spectrum widgets.
- Add pointer info in the status bar when clicking on the rolling spectrogram widget.

2009/06

- Add statistics about useless timer events
- Bugfix : forgot to reset the needfullreplot switch
- Display info about lost chunks of data
- Fix logarithmic scale in the spectrum plot
- Fix use of timer and Start button when changing device
- Factor device test code
- Print an error message when selecting a non-working input device
- Non-fatal error on device change
- Iterate over available devices for a working one On startup, first try the default system device, and then the other
- Bugfix for delays If the program is too slow to process all the chunks, discard those in excess. In fact, it is not that it is too slow, it is probably that there is a small mismatch between the timer period and the audio buffer period.
- Major timer-based refactor All sub-processes have been removed and replaced with a timer in the main process. This timer fires every 20 ms or so and retrieves audio data. This removes a lot of complexity coming from synchronization issues. Additionnally a level-meter widget is introduced, and the plot widgets have been optimized so that only the needed part is redrawn (namely axes are not redrawn).

2009/01

- Add a copy of the GPLv3.
- Update License statements to GPL version 3.
- Add proper encoding statements, and License statements.

- Add details in TODO.
- Add items and details in TODO.
- Add a TODO file.
- “Simplify” locking.
- Move profiling code to be all conditional.

commit 812b5b8050a921267c4844457d70d7a2a853c6e5 Author: Timothée Lecomte <timothee.lecomte@lpa.ens.fr> Date: Fri Jan 9 21:47:07 2009 +0100

first commit

CHAPTER

THREE

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007 Copyright © 2007 Free Software Foundation, Inc <http://fsf.org>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: **(1)** assert copyright on the software, and **(2)** offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that

patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that **(1)** displays an appropriate copyright notice, and **(2)** tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that **(a)** is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and **(b)** serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code

for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- **a)** The work must carry prominent notices stating that you modified it, and giving a relevant date.
- **b)** The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

- **c)** You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- **d)** If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- **a)** Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- **b)** Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either **(1)** a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or **(2)** access to copy the Corresponding Source from a network server at no charge.
- **c)** Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- **d)** Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- **e)** Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either **(1)** a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or **(2)** anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has

substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- **a)** Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- **b)** Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- **c)** Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- **d)** Limiting the use for publicity purposes of names of licensors or authors of the material; or
- **e)** Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- **f)** Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but

permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated **(a)** provisionally, unless and until the copyright holder explicitly and finally terminates your license, and **(b)** permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s “contributor version”.

A contributor’s “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor’s essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either **(1)** cause the Corresponding Source to be so available, or **(2)** arrange to deprive yourself of the benefit of the patent license for this particular work, or **(3)** arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient’s use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license **(a)** in connection with copies of the covered work conveyed by you (or copies made from those copies), or **(b)** primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others’ Freedom

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil

liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> <one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>
>
> This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

> <program> Copyright (C) <year> <name of author> This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands *show w* and *show c* should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

CHAPTER

FOUR

SOME DETAILS ABOUT PROFILING POSSIBILITIES

First option (cProfile and gprof2dot)

python -m cProfile -o output.pstats ./friture.py scripts/gprof2dot.py -f pstats output.pstats -n 0.1 -e 0.02| dot -Tpng -o output2.png

Second option (cProfile and pstats)

./friture.py –python

Third option (cProfile, convert to kcachegrind)

python scripts/lsprofcalltree.py friture.py kcachegrind

Fourth option (sysprof system-wide profiler on Linux)

> For sysprof, either use “sudo m-a a-i sysprof-module” to build the module for your current kernel, on Debian-like distributions, or use a development version of sysprof (>=1.11) and a recent kernel (>=2.6.31) that has built-in support, with in-kernel tracing as an addition.

sysprof ./gprof2dot.py -f sysprof sysprof_profile_kernel| dot -Tpng -o output_sysprof_kernel.png

TODO : write a converter from sysprof to callgrind (similar to lsprofcalltree)

CHAPTER

FIVE

TODO

Improvements :

- Code : continue profiling (why do python and Xorg take so much CPU while profiling shows that most of the time is pent being idle ?)
- Code : comments in the code
- Code/Maths : constant-Q-transform for efficient and precise logarithmic spectrogram
- Code/Maths : **or** adaptive frequency resolution in the fft spectrum and spectrogram
- Code : replace QwtColorMap with QLinearGradient to remove a dependancy on Qwt
- Code/Graphics : use the nicer histogram from octave spectrum fot the fft spectrum

Features :

- Code : overlapping fft windows for smoother spectrograms
- Code/GUI : new widget for long run level measurements

Bugfixes :

- code : unwanted delay between input and display

  Seems related to pulseaudio:

  http://forum.skype.com/index.php?s=7609fb1fac9ee65573e0ceb92562c481&showtopic=237601&st=0&p=1059071&#
  https://bugzilla.redhat.com/show_bug.cgi?id=444388
- code : when there is no input, audio data goes to 0., and log spectrum goes to -Inf, which seems to slow down computations enormously.
- code : the device info from portaudio reports a low input latency (max 46 ms), but after opening the stream, it is reported to be 139 ms. Why ?

CHAPTER

SIX

INDICES AND TABLES

- genindex
- modindex
- search