

IN3OTD's web site

...under perpetual construction.



Home

Electronics

**Amateur
radio**

**From the
bench...**

Programs

Resistor
calculator

Single-layer
coils
inductance

One-turn loop
inductance

Inductor self-
capacitance
estimation

Variometer
design

Inductance of
arbitrary coils
configurations

Bandpass
filters design

glfer

glfer is a GTK+ application with a waterfall spectrum display and slow CW (QRSS) / dual-frequency CW (DFCW) transmitter control interface. It's composed of two main parts:

- the spectrogram window, where you can see the spectrum of the received signal vs. time
- the transmission functions, to emit cw characters at a slow but precisely controlled speed, using the QRSS (slow CW) or DFCW (Dual Frequency CW) modes

The signal to be analyzed can be acquired in real-time from the sound card or can be read from a WAV file.

Starting from release 0.3, it is possible to select a spectral estimator among several different types:

- the "classical" periodogram, which is obtained as the modulus square of the discrete Fourier transform, after tapering the data with a "window function" selectable among the Hanning, Blackman, Gaussian, Welch, Bartlett, Rectangular, Hamming and Kaiser types,
 - the multitaper method, which is a weighted combination of periodograms computed with different windows, all belonging to the same family and having certain peculiar properties [1].
- This method requires more CPU power than the first one, due to the fact that several FFTs are performed on the same block of data, using different windows. The resulting spectrum is similar to a classical

QRSS and
DFCW with
Linux

Diode mixers
analysis

Creating disks
for rotary
encoders

Pictures

periodogram, but with much less variance (i.e. less variation in the background noise [speckle]). Performances are also similar to the periodogram, maybe it makes detection of QRSS signals a little easier, but this doesn't mean they are always more readable.

- a (so called) "high performance" ARMA model, which assumes a certain model for the input data (in this case it assumes white noise plus sinusoids) and tries to extract the parameters of this model (sinusoids frequency and strength) from the data [\[2\]](#). The number of sinusoids is estimated automatically from the samples autocorrelation.

Unfortunately this spectral estimator performs poorly when the noise spectrum is not flat (as it is usually in the RX audio, due to the IF filters) and when the signal-to-noise ratio (SNR) is low. On the other hand it provides a very good visual SNR with signals not buried in the noise... I will try different related methods in the future, in the hope of finding one really useful for digging weak signals out of noise.

- a new LMP estimator (details to follow)

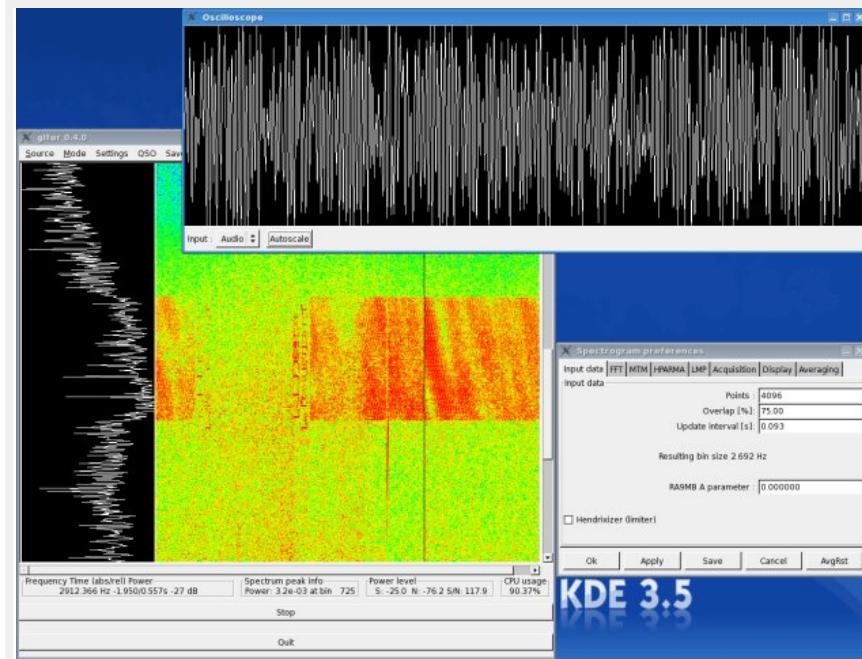
The spectrogram window includes also an Automatic Gain Control, which allows the output spectrogram to cover the entire palette to ensure the maximum visual contrast. In addition to this, there is also an automatic limiter which can be enabled to try to enhance the signal detection (based on an idea of RA9MB).

When using FFTW (see below) the number of points per FFT is not restricted to powers of two and is anyway user selectable up to about 65536 points. This limitation is mainly due to the maximum window size and could be (easily, or at least I think so) removed in the future. The sound card sampling rate is selectable to values depending on the capability of your particular hardware. For easier reading of the signals frequencies, you can automatically add an offset, with the usual convention of negative offsets meaning LSB reception.

The transmitter can be controlled using the serial or the parallel port; the pins used for PTT and key are the same as in [Rik Strobbe QRS](#), i.e. RTS = key and DTR = PTT/DFCW for the serial port, DB0 = key, DB1 = PTT/DFCW for the parallel port.

A more detailed description is in the `README` file included in the glfer source distribution.

Here is a screenshot of `glfer` (click on the picture for a larger image [350 kB]):



`glfer` is written using the GTK+ library (version > 2.0) for the graphical user interfaces, so please make sure that it is already installed on your system before trying to compile the program; the configuration script will anyway warn you if it will not be able to find the GTK+ libraries. You can download the GTK+ library at the [GTK+ home page](#)

You may also use the FFTW library for greater speed and flexibility in FFT calculations; FFTW, by Matteo Frigo and Steven G. Johnson, is available at the [FFTW home page](#); if FFTW is not available, `glfer` will use its own internal FFT routine.

You can here [download the source code of glfer](#) [167 kB], release **0.4.2**; this release features a TX beacon mode, a news spectral estimator and several averaging modes for the spectrogram (added by Edouard Griffiths, F4EXB).

Version 0.4.2 allows to select any device in `/dev` as the control device for the TX.

To compile `glfer`, after extracting the tar archive, you should just need to type:

```
./configure  
make
```

and at the end of the compilation you should have the executable file, named `glfer`, in the current directory, together with `mixer`, a small utility to set the input/output levels of the sound card.

Debian users should find a package for `glfer` at
<http://packages.debian.org>

openSUSE users can find a `glfer` package in the [hamradio repositories for openSUSE](#) 10.2, 10.3, 11.0 (and also 11.1, when this release will be published).

If you like `glfer` and would like to see some more features, or would like to tell your opinion about the program, please [drop me a note](#).

References:

- [1] David J. Thomson, "**Spectrum Estimation and Harmonic Analysis**," *Proc. IEEE*, vol.70, no. 9, Sep. 1982.
- [2] James A. Cadzow, "**Spectral Estimation: An Overdetermined Rational Model Equation Approach**," *Proc. IEEE*, vol.70, no. 9, Sep. 1982.

© 2000-2015 Claudio Girardi, in3otd@qsl.net

Created: Tue Dec 11 21:43:17 CET 2001

Last modified: Fri May 15 19:54:25 CEST 2015