IyoScope

In my previous articles in the 'Sound' series I have tended to concentrate on being able to process, analyse, and edit sounds that have already been recorded and stored as LPCM files. However it is also sometimes useful to be able to monitor in real time so you can see what is happening and make adjustments. Recently I decided to build myself a small 'Headphone DAC'. The initial requirement arose because I wanted to be able to hear sound files more clearly when editing them.

In practice a decent CD player makes an excellent source for test audio waveforms. So using !WAV_Gen and a CD player (Quad 67 in my case) I had a flexible 'waveform generator' which also provided SPDIF output to drive the Headphone DAC I was developing. (If you want to know more about the Headphone DAC I developed look at http://www.audiomisc.co.uk/HFN/HeadphoneDAC/HeadDAC.html) But what I lacked was a simple 'oscilloscope' and analyser so I could experiment and see what I was doing. Hence !IyoScope was born! As usual, I've supplied Jim with a copy and you can use it yourself if you have an Iyonix.

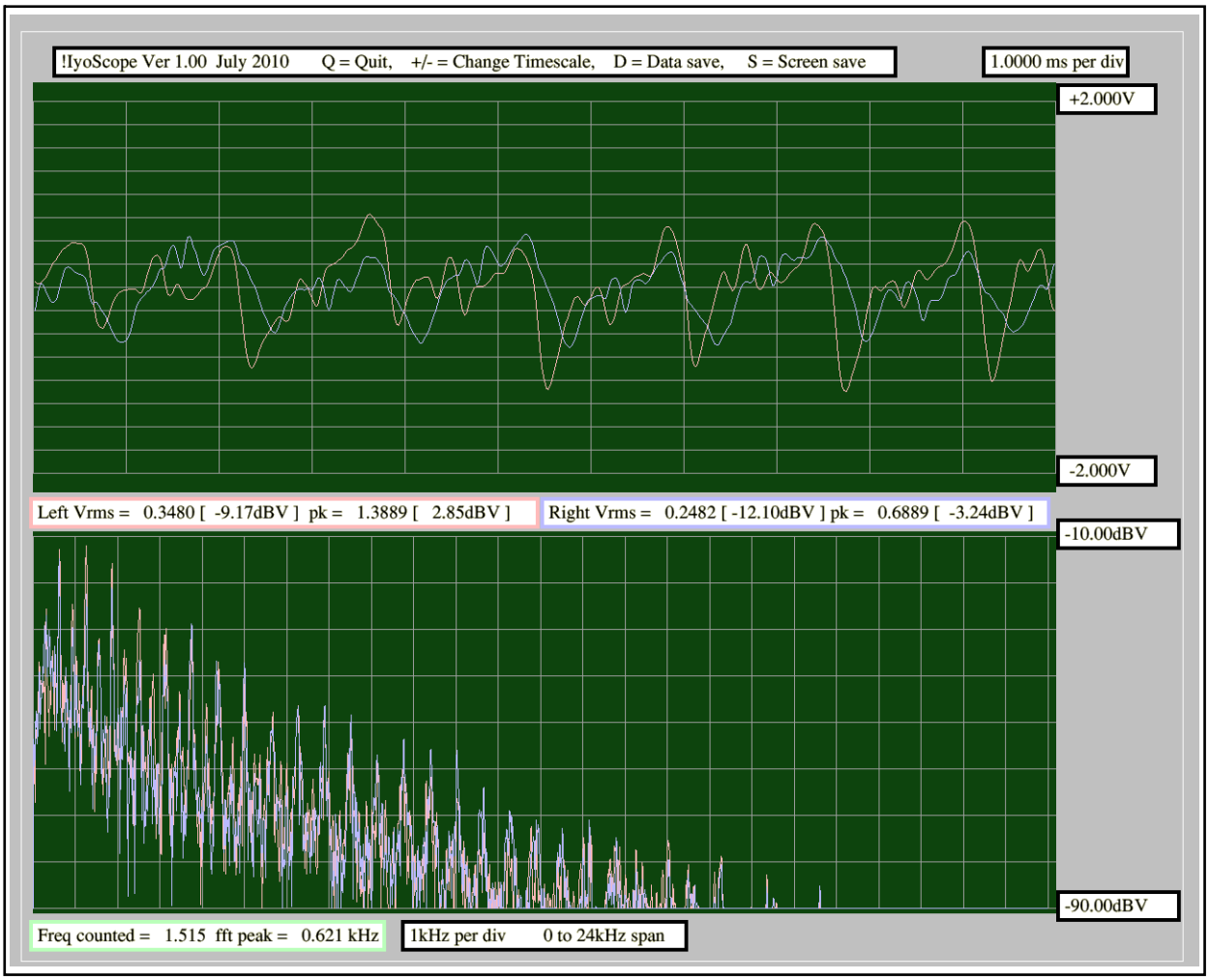

Figure 1

To use !IyoScope you should ideally set your display to a 1280 x 1024 pixel screenmode – preferably with 32k or 16m colours. (Larger modes will work OK, but the IyoScope will be limited to the bottom-left corner of the screen.) You then connect the audio input you want to

observe to the 'Line In' stereo jack socket on the back of the Iyonix. Then run !IyoScope in the usual way. . Detailed instructions on using !IyoScope are, as usual, provided in a !Help file inside the application's directory. But for basic use it should simply work.

Figure 1 shows the kind of results you will get when playing music. At the top you will see an 'oscilloscope screen' that shows how the waveform varies with time. At the bottom you will see a display of the spectrum of the waveform, obtained by FFT. By pressing various keys you can alter the scales of the display and save data. Figure 1 was obtained by pressing 's' whilst !IyoScope was running. That saved a sprite of the !IyoScope display to RAMDisc. Note that to save screens or data you need to have a RAMDisc with adequate amounts of free space. Data saves only require a small amount of space. But in 16m colour mode each saved screen sprite requires over 5·25MB.

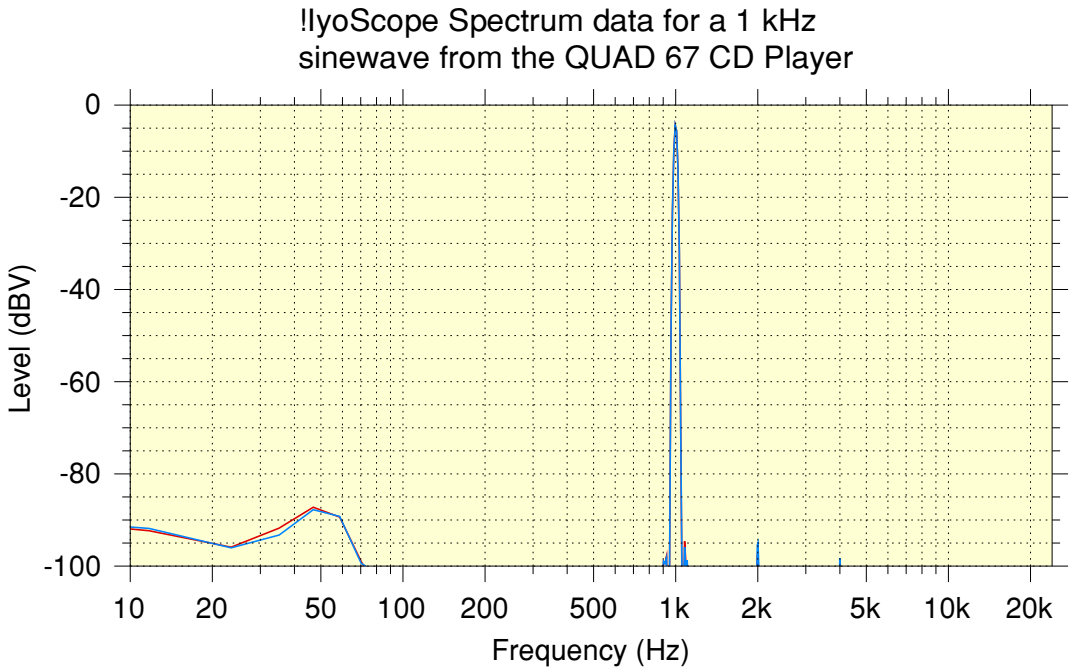

Figure 2

By pressing 'd' you can save data from !IyoScope. This allows you to do further analysis or use the data with another application. Figure 2 shows the contents of a 'Spectra' file saved by pressing 'd' whilst the input was a 1 kHz sinusoid. Note that the distortion and noise level are quite low. This indicates not only that the CD player works nicely. It also shows that the Iyonix input also works well. As usual for me, Figure 2 was plotted using !Tau, but you can load the saved data into other applications as you prefer. As with a conventional oscilloscope the display regularly updates and you can save data or screens repeatedly – provided you have sufficient RAMDisc space!

By default !IyoScope takes input from the 'Line In' and samples at a rate of 48000 samples per second per channel. This input has a fixed gain which I calibrated for my Iyonix so the results would be in Volts. There are some basic limitations and problems with the Iyonix hardware. Fuller details are in the !Help file, but the two most significant points are worth mentioning here.

Firstly the 'Line In' input seems to saturate and distort if the input rises above about +/-2Volts. This is a hardware limit and occurs *before* digital-to-analogue conversion. Because of the internal scaling of the hardware this occurs at a level *below* the maximum 16-bit sample integer magnitude. Hence although !IyoScope reads the data in 16 bit LPCM you can't use the entire nominal range of integer values that implies. For that reason the largest vertical scale !IyoScope displays is +/-2 Volts and you may see clipping distortion if the input is too large.

Secondly, !IyoScope takes 'grabs' of 4096 samples per channel at 48000 samples per second. This limits the time range of each grab and the resolution of the spectrum. The Iyonix hardware does provide a range of other sample rates, and info on how to use them is in the !Help file *BUT* in general I advise *against* using other rates! This is again because of a hardware problem with the Iyonix. The analogue inputs are sampled by hardware that insists on always sampling at 48000 samples per second regardless of what sample rate you ask it to provide. That means that if you choose a rate less than 48000 some samples are 'lost', usually in an irregular manner. The result is a distorted waveform and spectrum. For this reason the !IyoScope display and saved data assumes a 48000 rate is being used.

There are, however, two special exceptions to this rule. One is if you change the input from 'Line In' to 'CD'. The CD source *if digital* is at 44100 samples sec and bypasses the ADC hardware. So it may work well with a 44100 rate. But I've not tested this as yet, so beware! The other is that one of the rates provided by the hardware is 8000 samples/sec. This is 1/6th of 48000, so the discarded samples are in a regular pattern of "discard 5, take 1, discard 5, take 1,..." This regularly undersamples so avoids distortion *if* the input contains no components above 4kHz. Hence if you want to monitor inputs where you know there is nothing above 4kHz you can change the sample rate to 8000 samples per second and get decent results. However if you do, note that !IyoScope still displays and saves the data assuming it was taken at 48000 samples per second, so any frequency or time values will be out by a factor of 6!

As usual I may modify or develop !IyoScope, and it includes the 'C' source code which you are also welcome to use or modify as you wish. Overall, despite the limitations of the Iyonix sound hardware I am quite impressed by what it can do at the 48000 sample/sec rate. The noise and distortion are lower than I'd expected and it gives quite useful results.

1000 Words
Jim Lesurf
18th Aug 2010