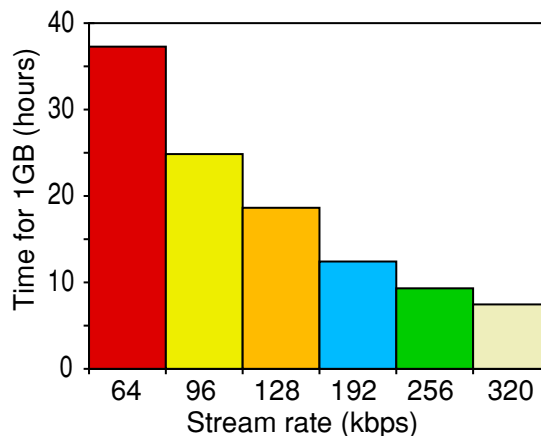


Sound 25 - Radio! Radio!

In my previous article I introduced the !RadioTimes application. Since then there have been some developments that extend its use. When I first started recording streams I wanted to save them directly to a USB memory stick. Alas the USB filer on my Iyonix was far too slow to cope reliably even with 128kbps! That was why I was recording to RAMDisc. The good news is that the problem is cured by using Fat32Fs. This dramatically increases the speed that data can be written to USB devices. As you might expect, it works best with devices formatted to FAT32. But I was surprised to find that it also increases the speed with FAT16 devices as well! So I now keep a copy of !RadioTimes on a 4GB USB FAT32 memory stick, and make recordings to that when I want them.



The diagram shows how many hours of internet radio you can listen to (or record) before you have downloaded 1GB of data. Note that 1GB = 1024 x 1024 x 1024 bytes, and that 1kb = 1000 bits, **not** 1024 bytes! This will give you a guide to how much listening you can do per month if your broadband link has a ‘data cap’.

The second advance is that there is now a way to decode aac/aac+ sound files on an Iyonix. This is because Christopher Martin has updated !FFMPEG so that it can now understand aac. !RadioTimes can be used to record aac/aac+ internet radio streams. If you obtain an up to date version of !FFMPEG you can now convert the results into lpcm wav files and play them with !DigitalCD. For example, if you have an aac file called “fred/aac” in a “Sound” directory you can carry out the conversion by typing the command

```
FFmpeg:ffmpeg -i ADFS::HardDisc4.$.Sound.fred/aac -f wav
ADFS::HardDisc4.$.Sound.output/wav
```

into a TaskObey or Obey file. (Note the command is all one line even if it looks like more than one here in the magazine.) For this to work as a TaskObey you will first have to increase the ‘next’ wimplot to 8MB or more as ffmpeg now requires that much memory to operate.

Alas, there is still one limitation. At present !FFMPEG interprets aac+ files plain aac format. You will get sound, but the sampling rate will be reduced (e.g. 22.05ksamples/sec instead of 44.1 for CD rate data). It is possible to force the output sampling rate to be 44.1 by using the “-ar” option, but that just provides simple interpolation so doesn’t sound any better. This limitation is because the ‘+’ part of the data isn’t being used as, currently, FFMPEG can’t decipher it.

Whilst discussing !FFMPEG it is worth also saying that it can now convert ac3 (Dolby) stereo format into lpcm. So if you wish you can now generate an output lpcm wav file from the ac3 soundtrack of a VOB file on a home recorded video DVD. In fact you can now also transcode ac3 directly into mp3 if you wish. To do this, you can use a command like

```
*FFmpeg:ffmpeg -t 120 -i ADFS::HardDisc4.$.Sound.Trovatore/AC3 -f mp3 -ab 96000 ADFS::HardDisc4.$.S.Sound.Trove/mp3
```

in a TaskObey or Obey file. Again, this should all be one line for the command. Note that in this case I have used '-t 120' to tell !FFMPEG to just convert the first 120 seconds as the above was just a trial. The '-ab 96000' sets the output file bitrate to use for the mp3 at 96kbps.

One warning about direct ac3 to mp3 conversions. They do avoid you having to generate large intermediate wav files, but they are veeery sloooooow on an Iyunix. So I suggest you do try an experiment first using the '-t' option to see how long a minute or two of sound takes to convert. Then use this to estimate how much time a longer file would take, and decide if that is OK.

Alas, the BBC still wrap their aac+ streams up inside a Flash iPlayer. So without Flash, Iyunix users are prevented from being able to record or listen to the BBC aac+ streams. For that reason I decided to take a bold step and try using a different OS to see how I got on...

I decided to install a modern version of linux on an ancient laptop computer I have. (I may say more about this in another article.) Having installed this, I then added the software to allow me to play various types of sound file, including mp3 and aac/aac+ (mp4). I also installed Flash so I could access the BBC streams via the iPlayer. This all worked nicely and I found I could connect and enjoy both the live and 'listen again'. I went to the /iplayer/labs page and signed up for the experimental version of the iPlayer which allowed me to get the aac/aac+ streams. This is worth doing as the sound quality is excellent.

My old laptop does have a coax spdif (IEC958) digital output. So I am also able to record the net broadcasts via a digital connection to an external recorder. This made it easy to make test recordings I could analyse to check if the system had any technical problems. The main snag was that the fans in the laptop and its power supply were too noisy to allow me to listen properly. I therefore decided to obtain a new computer that was mechanically silent. When it arrived I was pleased to find that it is, indeed, totally silent. No fans, and a solid state hard disc.

The new machine is clearly faster and more powerful. It has an optical spdif output, so I could also connect it to an external recorder. Alas, when I did some test recordings I found that the output was much more distorted than my ancient laptop. At present, for loud sounds the new computer outputs spdif waveforms that have around 10,000 times more distortion than my laptop for the same source material!

I am still investigating this and experimenting to see if the problem can be cured without having to get a different soundcard. For now, the bottom line is that if you are using a computer to listen to net radio and relying on it to decode and play the sounds, then there may be problems which are degrading the sound quality. All being well, I will say more in an article that looks at using linux for this.

Jim Lesurf
1100 words
7th May 2009