

Daniel James talks to the developers of FreeBoB - bringing FireWire audio to Linux

Hot on the wire



It makes a lot of sense to locate audio hardware outside the computer, in a so-called 'breakout box'. The output from analogue to digital converters should benefit from reduced electrical noise, and it also means that audio peripherals can be shared between machines. The same multichannel audio interface could be connected to a workstation in the studio one day, and on the next to a laptop for location recording, or a live performance. Early implementations of external audio interfaces used dedicated PCI cards or PCMCIA adaptors to connect to the host machine, but more recently the standardised I/O technologies of USB and FireWire, otherwise known as IEEE 1394 or iLink, have taken over.

which runs an embedded firmware known as BeBoB. This factor makes the creation of a generic 'class driver' possible, in much the same way that GPhoto2 supports most brands and models of digital camera on Linux without the user having to load individual drivers.

I'M FREE, BOB

Daniel Wagner is a Swiss software developer who has worked for BridgeCo since graduating from the Federal Institute of Technology in Zurich. As both a Linux user since the mid 90's and a guitar player, he decided to create FreeBoB, a free software stack which would support the BeBoB products that he works on in his day job. "BeBoB stands for BridgeCo

were looking for the next technology. The reason was that when FireWire was originally specified, audio manufacturers tried to build products and didn't manage it. Many millions of dollars were burned."

"When the Edirol FA-101 was finally released, it was clear that it was possible to create high quality standards-based devices with FireWire. The Edirol product was a door opener for BridgeCo, and maybe also for the technology. This year at the Musikmesse trade show in Frankfurt, you could see that FireWire as a technology is established. Of course, technical differences also exist between FireWire and USB 2.0. When FireWire was designed, the streaming part was always a key issue; therefore the bus

capture the communication between the main chips. That didn't go well; I destroyed my Isis, exit the ALSA driver."

"So I was looking for a replacement device. I was prepared to do the Linux support myself - I actually enjoy doing this kind of programming. You know that the device will (eventually) do what you want it to do, and that you'll be able to use it to its full potential. I specifically asked the vendors of candidate products if they supported Linux or were willing to help Linux driver development for their cards. I was in the middle of that process when Daniel sent out the mail requesting help with supporting the BridgeCo based FireWire products on Linux. Exactly the right question at the right time, I would say! I'm very happy with the support of

buffer. This can cause massive file system corruption, so you need a dedicated development machine. The kernel drivers for IEEE 1394 have volatile, undocumented API's, whereas the user space IEEE 1394 libraries have stable, documented APIs. And all other IEEE 1394 based device drivers, such as for video, are moving to user space, based on raw1394."

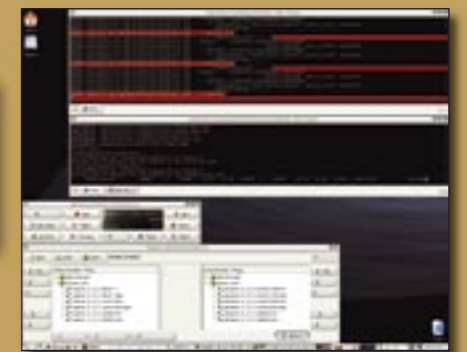
Palmer believes that FreeBoB could be part of the ALSA project at some point in the future. "For the moment we are still constructing the codebase. We chose Jack to do this because there was some code available for the IEC 61883 protocol that we could re-use for our tests. That is why it all went as fast as it did, for a two person after-hours project. Also, the user-space ALSA driver framework is rather

"BeBoB devices don't require that the capture stream is directed to the same machine as the playback stream is coming from. It would therefore be possible to direct the capture stream to machine A, and direct the playback stream to machine B. Setting up a link between A and B would close the loop - however, this could cause latency problems. Another possibility is to record the capture stream from a BeBoB device with two PCs, so each machine captures a different part of the stream. This would give both PCs direct access to all inputs on the BeBoB device. Or you could make all the PCs receive all the streams sent out by devices on the bus. I'm trying to design the streaming framework to allow for all these scenarios. To what extent they will be implemented remains a



The Edirol FA-101 proved that standards-based FireWire audio interfaces could work

The Jack patchbay shows the FireWire interface connected to the Linux system



Audio interface manufacturers at the higher end of the market are turning to FireWire

FireWire is a high-bandwidth bus for relatively short distances, with the 1394a or S400 standard allowing a theoretical maximum of 400Mb/s transfer over cables of a few metres in length. About 80% of the bandwidth is available for audio streams. The newer 1394b or S800 standard should double the bandwidth, and allow the use of optical cables up to around 100 metres long, but few audio products have implemented the revised specification so far. The IEC 61883 protocol is the published standard for streaming audio over the FireWire bus.

Until this year, there was almost no support at all for using FireWire audio devices with Linux. Fortunately, many of the devices on the market are based around the same BridgeCo chipset,

Enhanced Breakout Box. BridgeCo sells both the chip and the software for this kind of application. I'm a member of the core software developer team which is very tidily coupled with the ASIC group. We are writing and maintaining the operating system, originally based on the ThreadX RTOS, and provide most of the libraries for application developers."

It does appear that audio interface manufacturers at the higher end of the market are turning to FireWire instead of USB 2.0 - making support all the more important for Linux audio users. Wagner is sure that the hardware OEMs have now made their decision. "There was a lot a fuss around USB 2.0 one year ago. People had the impression that FireWire never really established itself, and therefore they

has some real-time properties which USB 2.0 lacks, making the latter difficult to use for serious audio products."

The other developer on FreeBoB, Pieter Palmers, is a Belgian electronic engineer who also plays the saxophone and piano. He's currently working towards a PhD on the CMOS integration of high speed analogue to digital converters at the Catholic University of Leuven. A Linux user for around five years, he began audio hacking experiments by writing an ALSA driver for the Guillemot Maxi Studio Isis soundcard. "There was no support whatsoever from Guillemot, and I had to reverse engineer everything. It took me two years to get to a point where playback would work. I decided to attach a logic analyser to the PCB in order to

BridgeCo. I have all the information I need, and the hardware support is also very good. At one point my test device broke, and BridgeCo fixed it for me."

FLOATING IN USERSPACE

The FreeBoB software works in userspace rather than the kernel, and is not yet part of the ALSA driver project. FreeBoB is currently designed to work directly with the JACK audio server, as Wagner explains: "The main reason is that this is not just a simple driver. It involves many parts of existing drivers and programs that work together. Firstly, the Linux 1394 stack already existed, with a nice userland interface. Secondly, there was JACK, with lots of cool applications, such as Ardour. Thirdly, I don't add anything to the kernel if there is a clean way around it. While I enjoy hacking the kernel, I also enjoy using powerful libraries and tools which are only available in userland."

Naturally, Palmers agrees. "There is no reason why it would have to be in the kernel, but there are several reasons why doing it in user space is better. Kernel programming isn't fun. If you make a mistake, most of the time you have to reboot your system. Sometimes, it's even worse: imagine a mistake that makes the driver write to a memory location that is the hard disc write

new, and there isn't much code to see how it should be used. I'm now rewriting our current streaming code into a library that can be used for both ALSA and Jack drivers, and others. I think that eventually this will go into libiec61883, but for the moment it isn't yet clear how this should be done."

THE FIREWIRE AUDIO NETWORK

High-speed multichannel audio interconnects with low latency allow the creation of audio networks. While the mLAN networking system developed by Yamaha also uses a FireWire infrastructure, BeBoB devices are different in that they are designed to be used in a host-based context, with the PC or laptop at the centre of the system. Palmers is enthusiastic about the possibilities: "The system we're working on has some really juicy stuff in its design. The multiple device capabilities we'll be supporting are very cool - it will be possible to use several BeBoB based devices on the bus as one big virtual device, without the need for external sync, such as wordclock. On top of that, it will be possible to introduce Linux based PCs as 'processing nodes' executing software synths, samplers or LADSPA effects. These nodes would appear to the master node as part of a single sound device, accessible through ALSA or Jack."

matter of what is useful."

Linux users can help the FreeBoB project by testing the software releases, or just informing the developers that they are using the code - the interest creating additional leverage when it comes to discussions with the hardware vendors. Wagner points out that the project could also benefit from additional developers: "We need someone with good knowledge of ALSA. At this point the prototype driver works, so interested hackers can buy a device without the fear that it will never work. I heard from several people that they were interested to help out, but wanted to wait for some positive results before they got involved. I understand that perfectly, because these devices can cost a lot of money, but the time is right to go and buy one and start to hack. There is a lot to do, but we have gained enough information and confidence to write the 'real' driver. We also have to pester manufacturers to update their firmware, because only the newest BeBoB versions support multi-device setups."

Key Links

FreeBoB homepage
freebob.sourceforge.net

FireWire on Linux
www.linux1394.org

Jack audio server
jackit.sourceforge.net

ALSA project
www.alsa-project.org