



A practical guide to using FFADO

<http://www.ffado.org>

Linux.conf.au 2008 Multimedia miniconf

Jonathan Woithe

1 Tutorial outline

- The FFADO project
- FFADO technicalities
- Prerequisites
- Using FFADO
- Hints and tips
- Current status
- Helping FFADO
- Acknowledgements

2 The FFADO project

- FFADO: Free Firewire Audio Drivers for Linux
- Allows Linux software to use firewire audio devices via JACK
- Originally known as “FreeBob”, in reference to the “BeBob” platform used for some devices
- Renamed FFADO to be more vendor neutral after Freebob 1.0 release to reflect wider variety of devices being supported
- Hence the “first” FFADO release will be v2.0

3 FFADO technicalities

- Come to “FFADO: firewire audio for Linux” on Wednesday. :-)
- This talk is focused on the practical details of running FFADO

4 Prerequisites

- A modern JACK (0.112 seems to work well)
- User access to realtime (RT) scheduling (PAM, set_rlimits)
- libraw1394 (v1.2.1 or later)
- libiec61883 (v1.1.0 or later)
- dbus-1 (for mixer data transport)
- libxml++2 (perhaps to be removed in future)
- OPTIONAL: RT-patched kernel

To use the optional (but useful) device GUI mixer application:

- Qt (version 4.x preferred)
- Qt Python bindings (PyQt)
- dbus-python bindings
- SIP (if compiling PyQt locally)

4.1 Kernel firewire stack

- FFADO currently uses the “old” firewire stack (aka ieee1394)
- A future version of FFADO will also be supported under the “new” firewire stack (“juju”, aka firewire)
- New kernel firewire stack overcomes many operational and security shortcomings of old stack, but is still under development

4.2 RT-patched kernel

- Optional - needed only for certain setups
- For “high” latency configurations (50 ms buffers), “Low latency desktop” kernel is sufficient
- For low latency (less than 10 ms buffering), RT-patched kernel is currently needed
- Could be set to change - threaded IRQ kernel patches should address most of the issues with mainline
- General recommendation: start with standard kernel and run high latency settings

4.3 Realtime scheduling

- Needed to ensure FFADO is scheduled on time to deliver audio data to device
- Two approaches:
 - use PAM's `limits.conf` to grant elevated privileges to selected users. Eg:

```
@audio - rtprio 100
@audio - memlock 512000
```
 - use `set_rlimits` to elevate privileges to particular programs run by selected users. Download from http://www.physics.adelaide.edu.au/~jwoithe/set_rlimits-1.3.0.tgz
- Most systems run PAM, so just edit `limits.conf`
- `set_rlimits` can be used on non-PAM systems, or on PAM systems if finer control is needed

4.4 libraw1394

- Used to provide userspace with access to firewire bus
- Require version 1.3.0 or later
- Most distributions provide this as a package. Ensure the “devel” package (if provided) is also installed
- If compiling locally and using version 2.0 or later, ensure support for old kernel firewire stack is included
- Download from <http://www.linux1394.org>

4.5 libiec61883

- Provides low-level protocol support for some firewire audio devices
- Require version 1.1.0 or later
- Most distributions provide this library as a package. Ensure the “devel” package (if provided) is also installed
- Download source from <http://www.linux1394.org>
- To compile from source, libraw1394 must be installed first

4.6 **dbus-1**

- dbus is used to provide an interface to device mixers
- every modern distribution includes dbus
- source is available from <http://dbus.freedesktop.org>

4.7 **libxml++2**

- Used to parse configuration files and private cache files
- May be replaced in later FFADO versions
- For now version 2.6.13 or later is required
- Available as a package in most distributions

4.8 JACK

- Download from <http://www.jackaudio.org>
- Both JACK1 and JACK2 (aka JACKmp) are supported by FFADO backend
- FFADO must be compiled and installed before compiling JACK
- For JACK, versions 0.112 and 0.115.7 are known to be good. Generally latest SVN is ok.
- For JACK2, svn revision 3136 or later should be used.

4.9 dbus-python

- The included optional mixer GUI is written in python
- dbus-python is required to gain access to mixer data provided by FFADO driver
- Source code is at <http://dbus.freedesktop.org/releases/dbus-python/>

4.10 Qt

- Used by the optional mixer GUI application
- Qt4 is preferred
- A Qt3-based mixer is included but it is officially deprecated and no longer kept up to date
- Qt4 framework provides more flexible environment. For example, mixers can appear or disappear dynamically as devices are added or removed from the system.

4.11 SIP

- Needed only if compiling PyQt locally
- Download from <http://www.riverbankcomputing.co.uk/sip/index.php>

4.12 PyQt (Qt Python bindings)

- Used by the optional mixer GUI application
- Available in many distributions
- Source is at <http://www.riverbankcomputing.co.uk/pyqt/>
- Ensure PyQt version matches the Qt version in use (3.x for Qt3, 4.x for Qt4)

5 Using FFADO

5.1 Interfacing to audio systems

- FFADO provides a JACK backend driver
- Both JACK and JACKmp (aka JACK2) are supported
- Userspace ALSA driver is planned

5.2 Running (via JACK)

- OPTIONAL: boost priority of firewire IRQ handler:

```
chrt -f -p 72 'pidof "IRQ 21"'
```

Not as effective as it could be due to scheduling prioritisation issues within the kernel firewire stack (eg: each traffic type has own softirq tasklet but all run at same priority).

- Run JACK “normally” but use “firewire” driver:

```
set_rlimits jackd -R -P60 -dfirewire -r 44100 -p1024 -n4
```

- Audio I/Os now accessible via JACK in the usual way

6 Hints and tips

- Running audio over firewire needs “good” firewire chipset in PC (applicable to all OSES)
 - Anything by Texas Instruments is good
 - All but the most recent VIA chipsets can give trouble
 - Ricoh chipsets are patchy
 - Possible issues with some SiS chipsets
- Start with conservative (higher latency) jack setups (eg: `-n 4 -p 1024`)
- User running jackd must be able to access `/dev/raw1394` for read-write. Perhaps use `video` group for this.
- Binary Nvidia and ATI video drivers have caused problems on some systems - especially if IRQ is shared by video and `ieee1394`
- USB action can upset FFADO due to kernel scheduling issues. Should be fixed with threaded IRQs in the kernel.

7 Current status

- Currently running “release candidates” for FFADO 2.0
- Release date: Early 2009
- FFADO 2.0 will support:
 - Interfaces based on DM1000 chip: Focusrite Sapphire, Edirol FA-101 & FA-66
 - Some Echo Audiofire devices
 - Some MOTU devices (Traveler, 828Mk2, 896HD, Ultralite)

8 Helping FFADO

- Purchase devices from FFADO-friendly vendors and tell them their support of FFADO/Linux is why you're purchasing their interface
- Avoid devices from hostile vendors and tell them why you're not buying their devices
- Download beta releases or subversion snapshots and test them
- Donate/lend devices to FFADO developers

9 Acknowledgements

- Fellow primary developers (Daniel Wagner, Pieter Palmers)
- Companies actively supporting FFADO: Echo, ESI, Focusrite, Terratec (now Musonic), Mackie
- Our users, for continued testing and bug reports
- The trademarks of companies referred to throughout this presentation are acknowledged

10 Links

- FFADO project: <http://www.ffado.org>
- JACK: <http://www.jackaudio.org>
- Set_rlimits:
http://www.physics.adelaide.edu.au/~jwoithe/set_rlimits-1.3.0.tgz
- libraw1394, libiec61883: <http://www.linux1394.org>
- dbus-python: <http://dbus.freedesktop.org/releases/dbus-python/>
- SIP: <http://www.riverbankcomputing.co.uk/sip/index.php>
- PyQt: <http://www.riverbankcomputing.co.uk/pyqt/>

Contacting me:

- jwoithe@physics.adelaide.edu.au