The PDAudio Project.

Erik de Castro Lopo

<erikd@mega-nerd.com>

LCA 2004 - Audio Mini Conference
Adelaide

January 12, 2004

How it Came About

- Len Moskowitz (of Core Sound LLC in the US) emailed the LAD list in late 2002 looking for an audio coder.

- He had already contracted Elan Digital systems in the UK to design the PDAudioCF card.
  `http://www.elandigitalsystems.com/index.php`

- ALSA driver written by Jaroslav Kysela (based in Solvak Republic) of the ALSA Project.

- Me here in Australia doing the user space application.
- Communication via email, IRC and SSH.
- Result:
  Hardware : `http://www.core-sound.com/`
  Software : `http://www.mega-nerd.com/PDAudio/`

About the iPAQ

- Model I was using is iPAQ H3765.
  - 206 MHz StrongARM CPU
  - 240 * 320 pixel screen
  - 64M RAM
  - 32M ROM
  - 16 bit, stereo audio out only (up to 44k1)
  - Microsoft Pocket PC OS

- There are quite a few sub-families in the iPAQ range. Familiar runs on the some but not all.

- If you are interested in running Linux on an iPAQ check the web page for supported models: `http://familiar.handhelds.org/`

PDAudio Compact Flash Card

- PDAudioCF card was designed by Elan Digital Systems in the UK.
- Standard Compact Flash interface.
- Supports Mono and Stereo.
- 24 bits per sample.
- Sample rates of 44.1kHz, 48kHz 88.2kHz 96kHz (later versions might also do 176.4kHz and 192kHz).

- Accepts S/PDIF (Sony / Philips Digital Connect Format) digital audio with a single connector which accepts both optical and coaxial S/PDIF.

- No analogue input!!!! You need a separate digital-to-analogue converter. See for instance http://www.core-sound.com/

- PDAudioCF can be used with a laptop via a Compact FLASH to PCMCIA converter.

- Also need a BackPack expansion unit to get two CF slots (one for the Flash disk and the other for the PDAudioCF card).

Courtesy Len Moskowitz (Core Sound)

Familiar Linux.

- This is the Linux distribution we are running on the iPAQ.
  `http://familiar.handhelds.org/`
- Uses GNU Palmtop Environment (GPE).
  `http://gpe.handhelds.org/`
- Use IPKG as a package format.

Development System

- Cross compilers (for i386 Linux) are available from the Familiar web site and install in /usr/local/arm-linux.

- Also need to be able to link to the ARM Linux libraries on the development machine.

- There are Debian packages available containing the necessary ARM library binaries but they are not always up-to-date.

- Also possible to download the ARM library packages, rip them apart and install them where the cross tools can find them.

- Either way, setting up a working cross compiler tool chain is a pain.

- It is especially hard to track a constantly changing target.

- Really should try native compiling.

The Application

- Written in C using the GTK+ 2 libraries.
- Attempt to keep the application as lightweight as possible.
- It uses the ALSA drivers just like any other audio application (JACK was not considered an option for this).
- One set of source code can compile for both the iPAQ and a standard Linux desktop or laptop systems making debugging far easier. Big plus.

Complications

- The 200MHz StrongARM is a little under powered.
- For desktop and laptop systems, need to use the `mlockall()` system call to lock the application in memory and prevent it from being paged out.
- Might need to use `sched_set_scheduler()` which requires root privileges so needs to be a setuid application.
- GTK+ won't run as setuid.

- GUI and audio parts of the program were put in separate processes and the GUI process drops privileges before starting calling first GTK function.

- Audio process has multiple threads.

Hacking libsndfile

- Early on in the development process, libsndfile was used for file I/O.
- libsndfile contains methods for reading/writing float and double but the iPAQ doesn't have an FPU.
- Hacked together a Lite version of libsndfile without floating point operations, fewer file types and no compressed formats other than u-law, A-law and GSM 6.10.

- Lite version is generated from the standard version by a Python script which uses markers in the source code and other rules to 'edit' the code.
- This was fine for reading files during playback. Not fine for writing files to disk on the 200Mhz StrongARM.

Custom File Writing

- Since Len Moskowitz only needed to be able to write 16 and 24 bit PCM WAV files, wrote a real lightweight WAV writer.
- Standard `write ()` function call was causing problems (probably related to disk buffering).
- Uses `mmap()`. Mmap on write a little tricky. Need to map a region, and `write()` to each page of the region before accessing the memory.
- Turned out it was better to use `pwrite()`.

GTK GUI

- Pretty standard apart from the custom meter widget.
- Metering was a pain.
- Looked at Steve Harris' Meterbridge code but it was all floating point.
- Looked at about 2 dozen different ways of doing metering but ended up doing an integer version of Steve Harris'.

- Updating is critical.
- Metering that looks great on my dual 450Mhz P3 desktop, killed the audio on the iPAQ.
- Slowing down metering till it allows reliable audio makes for crap looking meters.

ALSA

- Read the data as ints to get 24 bit data.
- Use the memmap functions rather than read/write for better performance.
- ALSA is still a pain to compile and install, especially when cross compiling.
- Is this API ever going to stabilize?

Audio

- Three threads:
  - Audio in thread
  - Audio out thread
  - File I/O thread.
- Two circular audio buffers: one for audio in and one for audio out.
- On playback audio out buffer is shared between the file I/O thread and the audio out thread.

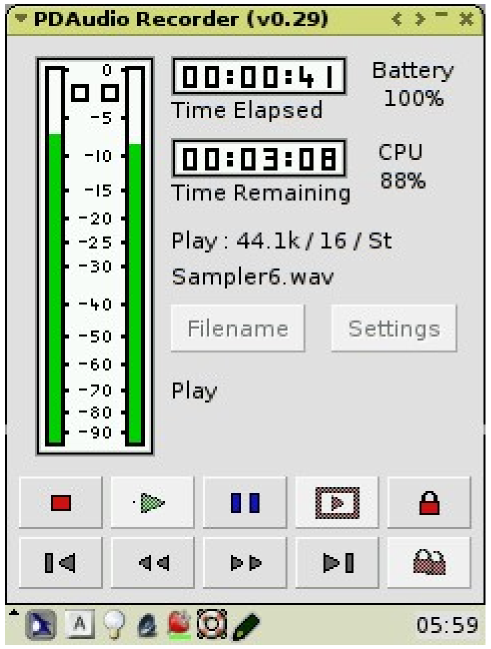
- On record (with monitoring), audio in, audio out and file I/O share a buffer. Audio out thread does the conversion from int short required by the audio out device.

- Synchronisation of threads is tricky. Mutexes are the standard solution but are too much overhead in this situation.

- Instead, take care about how the buffers are updated. Only modify the the read and write pointers into the audio buffers in one thread.

- Tried things like `sched_set_scheduler()` to improve reliability. Didn't work because it seems to cause more problems than it solved.

Screenshots


PDAudio Recorder (v0.29)

00:00:41 Time Elapsed

Battery 100%

00:03:08 Time Remaining

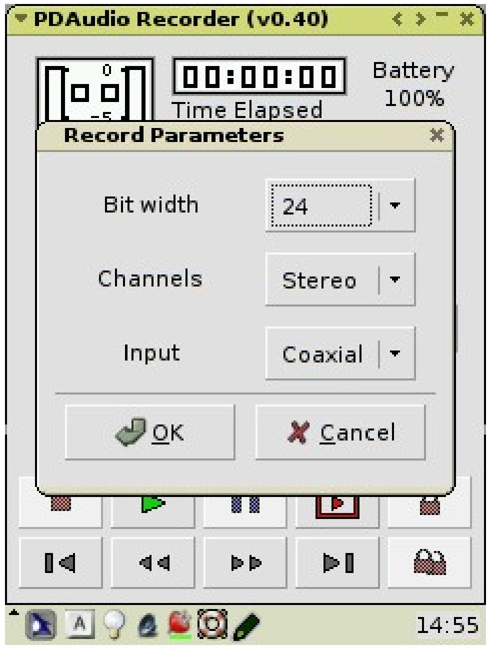
CPU 88%

Play : 44.1k / 16 / St

Sampler6.wav

Play

05:59


PDAudio Recorder (v0.40)

00:00:00 Time Elapsed

Battery 100%

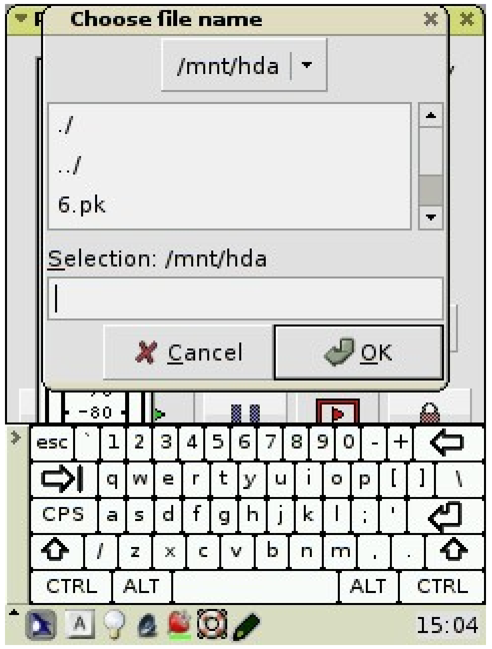
Record Parameters

Bit width: 24

Channels: Stereo

Input: Coaxial

14:55


Choose file name

/mnt/hda

./

../

6.pk

Selection: /mnt/hda

15:04

Cancel
OK
esc ` 1 2 3 4 5 6 7 8 9 0 - +
q w e r t y u i o p [ ] \
CPS a s d f g h j k l ; '
/ z x c v b n m , .
CTRL ALT ALT CTRL

Courtesy Len Moskowitz (Core Sound)

TODO

- Make everything work.
- May need to move to 400Mhz iPAQ.
- Splitting files across multiple files when recording.
- Release.