Linux Audio for Smartphones

System integration basics
• Android audio stack overview
• Modern smartphone audio subsystems
• Traditional solutions
• ASoC – the Linux solution
  • Design overview
  • Brief introduction to chip drivers
• Walk through of system audio driver construction
• Debugging tips
• Future directions
Android audio stack

Applications

MediaPlayer
MediaRecorder

AudioFlinger

libaudio
Modern smartphone audio subsystems

- HDMI
- CPU
- DSP
- Mixing
- Earpiece
- Speaker
- Headset
- Dock
- Cellular modem
- Bluetooth
Traditional driver model

Memory ➔ Processing ➔ Analogue
Traditional driver model

• **Monolithic driver for each card**
  - No structure for managing off-CPU hardware
  - Very little reuse

• **Tight coupling between application and kernel code**
  - Per-use case register settings
  - Detailed register level knowledge of components

• **Time consuming**
• **ASoC embedded audio framework**
  • Merged since 2.6.21, April 2007
  • Provides standard ALSA interface to applications
• **Reusable drivers for each chip**
• **Minimal per-system drivers**
• **Use case configuration done by userspace**
• **Automatic and transparent power management**
• **More reuse, less coupling**
Dynamic Audio Power Management

- Looks for audio paths connecting inputs to outputs
- Powers only components in an active path
- Automatically activates DACs and ADCs
• **Four classes of control**
  • Audio processing controls (eg, volume, effects)
  • Audio routing controls (DAPM controls and routes)
  • Power controls (DAPM widgets, bias)
  • Stream control (Digital audio streaming)
• **Mostly direct mapping into register map**
  • `SOC_DOUBLE_R_TLV("DAC1 Volume", WM8994_DAC1_LEFT_VOLUME, WM8994_DAC1_RIGHT_VOLUME, 1, 96, 0, digital_tlv)`,
Driver integration walkthrough

- S3C6410/WM0010
- WM8915 CODEC
  - “Baseband”
  - WM9081 Amplifier
  - Headset
  - Speaker
  - Mics
  - Subwoofer

http://opensource.wolfsonmicro.com/content/speyside-audio
• **AudioPolicyManager and AudioHardware**
  - platform/hardware/alsa_sound – Generic ALSA, asound.conf, LGPL
  - devices/samsung/crespo – Nexus S, hard coded, Apache licensed

• **Getting use cases**
  - Devices specified when streams are opened
  - `setMode()`

• **Applying use cases**
  - Run external utilities
  - Use `asound.conf`
  - Call raw ALSA control APIs
  - Apply settings with ALSA UCM
  - Using common base use cases helps
• **Data in debugfs**
  - `CONFIG_DEBUG_FS`
  - `mount -t debugfs /dev/null /debug`

• **codec_reg – Register map**

• **dapm_pop_time – log sequences**

• **dapm directory**
  - **SPKL**: Off in 0 out 1
  - in "DAC2L" "DAC2L"
  - out "static" "SPKL PGA"

• **Tools:**
  - `git://git.opensource.wolfsonmicro.com/asoc-tools.git`
• Audio stuck – check clocking
• Silent audio – check volumes and mutes
• Use bypass paths to bisect
• Turn volumes up to maximum
• Make sure machine drivers check error codes
• Check kernel logs for errors
• 2.6.38 and later support trace points
  • http://www.sirena.org.uk/log/2011/01/22/tracing-asoc-with-trace-points/
Future work

- Nicer handling of digital basebands
- Resolve headset detection API compatibility
- Greater use of DSP
  - Enhanced features – ambient noise cancellation, beam forming, offloaded decompression, speaker compensation
  - Even more dynamic reconfiguration of the audio subsystem
- Coefficient management and in-system calibration
- Use case development and management
  - Media controller API
  - User interfaces for configuration development
- ASoC conference, 4th-5th May
  - http://www.slimlogic.co.uk/?p=268