INSIGHT OF AN AUDIO DRIVER BASED ON ALSA

Chandrasekar R
Agenda

- Basics of Audio
- Audio Subsystem Overview
- Audio software architecture
- Android Audio System
Basics of Audio

- Analog Audio Data vs Digital Audio Data
- Conversion from Analog Audio Data to Digital Audio Data
  - Data at each discrete time is called Sample.
- Number of samples per second
  - Sampling Rate, Frame rate (FS)
- Number of bits per sample
  - Bit-Length Count (BLC)
- Channels
  - Mono, stereo, 5.1 channel etc
Audio Sub-system comprises of following blocks

- I2S Interface
- Audio Subsystem Clock
- DMA
- I2C
- Codec (Analog to digital and Digital to analog converter)
I2S is a 3-line serial bus-interface
- Word select line – LRCLK line
- Clock line – SCLK line, bit-clock line
- Data line – One line for two time-multiplexed channels

- 8/16/24 bits per sample
- Master/slave mode
I2S Data Formats

Foundry Design Services (FDS) SW

SAMSUNG
Clock configuration is the impairment element in audio
- Based on the sampling rate appropriate root clock need to be configured
- The same clock need to be passed on to the codec if it takes from the external source
- Some codec generates the clock, in that case i2s block will take the root clock from the external audio codec
3 DMA channels normally used for mobile SoC

- TX Primary DMA
- RX DMA
- TX-Secondary DMA
Audio codec chip is controlled through I2C interface
Audio codec driver registers itself as a client device to the I2C
Codec driver used regmap to access the code registers through I2C
I2C Layers

User-space

Program

Driver

Client

I2C-dev

I2C-core

Algorithm

Adapter

Adapter-specific code

Kernel

Hardware

Adapter-hardware

I2C-device

I2C-device
CPU copies PCM data from memory in user space to DMA buffers

ADMA channels copy from DMA buffer to I2S FIFO

From I2S FIFO data is on I2S BUS
Data Flow in Audio Sub-System

Data flow from memory to CODEC
Control Flow in Audio Driver

- HAL
- ALSA LIB
- ALSA CORE
- SOUND CARD DRIVER
  - I2S
  - MIXER
  - CODEC
  - DMA
  - LPASS
Each Interface functionality uses H/W as separate entity
A separate device for each functionality is created
For Example:
Like pcmC0D0c and pcmc0d0p –
c0 – card0
D0- device 0
P for playback and c for capture

After probe separate PCM DEVICES registered for each DAI-LINK
Mixer is a hardware device mixes the digital audio data from different sources and send backs to them based on the selection.
Kcontrols in Mixer

- Mixer driver exposes some controls for upper application layer though Kcontrols
- These controls are used to change SFRs of mixer which is done through regmap

```
kctl->get(kctl, control);
```
Mixer_ctl_get_value() is called which triggers SNDRV_CTL_IOCTL_ELEM_READ operation. Here Kcontrol list is checked and corresponding function is executed (get/put).

Kcontrol Flow
Power ON/ OFF sequence of respective paths gets when playback or capture stream opened

- **Playback Paths**
  - Speaker On /Off
  - Headphone On /Off
  - Earpiece(receiver) On /Off

- **Capture paths**
  - MIC1 On /Off
  - MIC2 On /Off

- **Gain controls**
  - DAC, ADC Digital gain control
  - MIC1, MIC2, analog gain control
  - SPK, EP, HP analog gain controls

- **Mixer path selection**
  - MIC input data mixing can be selected through ADC mixer Controls
MOBILE USECASE of AUDIO DRIVER
Android Audio System

Phone Application
/Media player / Recorder Applications

Application Framework
Media player / Recorder / Audio Manager

Native Framework
Audio Track / Audio Recorder / Audio Flinger / Audio Mixer / Audio Policy Service

HAL
Audio Hardware Interface / Audio Policy Manager

Kernel
ALSA / Audio Driver
Voice Call Path

Foundry Design Services (FDS) SW

Modem

AP

Native Framework

HAL

Kernel Driver

Codec
Any Questions?
THANK YOU