ISSN No. 0976-5697

Volume 4, No. 6, May 2013 (Special Issue)

International Journal of Advanced Research in Computer Science

REVIEW ARTICAL

Available Online at www.ijarcs.info

Developing Linux Device Drivers For Audio Codec

Miss.Sapana S. Kukade[1]

[1]final Year M.E.Digital Electronics Engg, SSGMCE, Shegaon, India

Skukade06@gmail.com

Abstract: For coding and decoding or compression and decompression of digital data according to a given audio file format or streaming media audio format, one computer program is there in software called as "audio codec". This can effectively reduce the storage space and the bandwidth required for transmission of the stored audio file. Most codecs are implemented as libraries which interface to one or more multimedia players. codec implementations for various platforms and environments has inspired us to develop a tool that makes it easy to implement multi-codec applications like MP3 Players, IP phones, call center monitoring, or audio players/recorders. In hardware, "audio codec" refers to a single device that encodes analog audio as digital signals and decodes digital back into analog. For the linux operating system we are developing a device driver for the application of “audio codec”.

Keywords: - Audio codec, ALSA, Linux, Driver, sound card. Audio

I. INTRODUCTION

Main objective is where we write device drivers in the operating system, it’s important to make the distinction between “user space” and “kernel space”.

A. Kernel space:

a. Kernel is a main program of Linux system, it controls H/W,CPU, Memory, Hard disc, etc
b. Provides the user a simple and uniform programming interface.
c. It forms a bridge or interface between the end-user/programmer and the hardware.

B. User space:

a. End-user programs, like the UNIX shell or other GUI based applications, are part of the user space.
b. This interact with the system's hardware through the kernel supported functions.

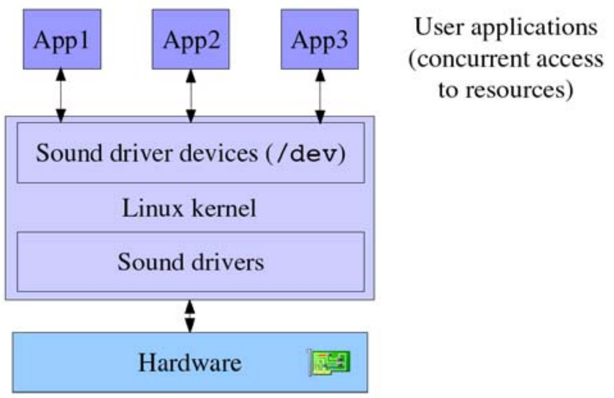

Figure: 1

We are developing a linux device driver for the 24 bit stereo audio codec, it is low power high quality codec for portable and general purpose audio application. Also there is precision of 24bit stereo audio ADC and DAC. This integrates a broad range of additional functions to simplify implementation of complete audio system. A fractional PLL is available to accurately generate any audio sample rate for the codec using commonly available system clock from 8MHZ through 33MHZ. application. In computing, a device driver or software driver is a computer program allowing higher-level computer programs to interact with a hardware device. A driver typically communicates with the device through the computer bus or communications subsystem to which the hardware connects. Device drivers, particularly on modern Linux platforms, can run in kernel-mode in user-mode. The primary benefit of running a driver in user mode is improved stability, since a poorly written user mode device driver cannot crash the system by overwriting kernel memory. On the other hand, user/kernel-mode transitions usually impose a considerable performance overhead, thereby prohibiting user mode-drivers for low latency and high throughput requirements.

This 24 bit stereo audio codec operates with analog supply voltage from 2.5vto 3.6v. In this on chip DSP include 5 band equalizer,3-D audio enhancer and mixed signal automatic level control for the microphone or line input through ADC and digital limitar/ dynamic range compressor function for the playback path.

Key Features of the Audio Codec:-

a. DAC: 94dB SNR and -84dB THD (“A” weighted)
b. ADC: 90dB SNR and -80dB THD (“A” weighted)
c. Integrated programmable microphone amplifier
d. Integrated line input and line output
e. On-chip PLL
f. Integrated DSP with specific functions:
    a) 5-band equalizer
    b) 3-D audio enhancement
    c) Input automatic level control (ALC/AGC)/limiter
    d) Output dynamic-range-compressor/limiter
    e) Notch filter and high pass filter

(a). Standard audio interfaces: PCM and I2S

(b). Serial control interfaces with read/write capability
(c). Real time read back of signal level and DSP status
(d). Supports any sample rate from 8kHz to 24 Bit stereo audio codec 48kHz

II. BASIC STRUCTURE OF ALSA (ADVANCED LINUX SOUND ARCHITECTURE)

Advanced Linux Sound Architecture (known by the acronym ALSA) is a free and open source software framework providing an API for device drivers for sound cards. As such, it is a Linux kernel component.

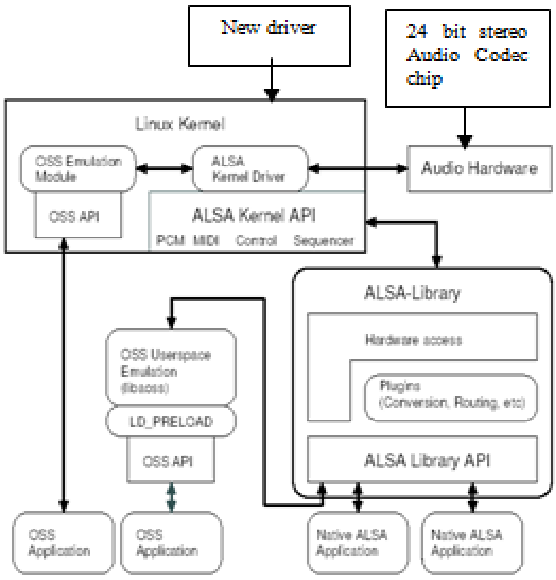

Figure: ALSA Basic Structure

In the basic structure of ALSA in place of audio hardware we are using our 24 bit low power stereo audio codec and in the linux kernel space in place of ALSA kernel driver we are writing a driver for that audio codec.

The Advanced Linux Sound Architecture (ALSA) provides audio and MIDI functionality to the Linux operating system.

ALSA has the following significant features:

a. Efficient support for all types of audio interfaces, from consumer sound cards to professional multichannel audio interfaces.
b. Fully modularized sound drivers.
c. User space library (alsa-lib) to simplify application programming and provide higher level functionality.
d. SMP and thread-safe design.
e. Support for the older Open Sound System (OSS) API, providing binary compatibility for most OSS programs.

A. ALSA – Benefits:

From the 2.6.x kernels onward ALSA has been the default Linux sound architecture, replacing it's predecessor OSS.

a. Android uses 2.6 and higher Linux kernel. So, if you want your codec to work with any of the Android systems, it has to be ALSA compliant.
b. As ALSA is fully modularized, most of the sound system components are re-used. You needn't reinvent the wheel! This significantly shortens TIME TO MARKET your product and ensures HIGH QUALITY.
c. A huge code base(tools/applications) is already developed for ALSA. If your codec driver is ALSA compliant, all existing tools for media playback, recording and mixing will automatically work with your device.
d. ALSA is maintained by very active community. If your hardware has some latest and greatest features which are yet not supported by the framework, you needn't worry. There are high chances to get this new stuff supported by framework if it is interesting enough.
e. ALSA compliance will increase the acceptance and exposure of your device by multifolds, thus increasing it's foot print and revenue.

Fig. 1 depicts the basic structure of ALSA system and its data _ow. The ALSA system consists of ALSA kernel drivers and ALSA library. Unlike the OSS, ALSA-native applications are supposed to access only svia ALSA library, not directly communicating with the kernel drivers. The ALSA kernel drivers offer the access to each hardware component, such as PCM and MIDI, and are implemented to represent the hardware capabilities as much as possible. Meanwhile, the ALSA library complements the lack of function of the cards, and provides the common API for applications. With this system, the compatibility can be easily kept even if the kernel API is changed, because the ALSA library can absorb the possible internal changes and keep the external API consistent.

The following components are supported by ALSA

a) PCM interface: Managing digital audio capture and playback.
b) Controls interface: General purpose facility for managing registers of sound cards and querying available devices.
c) Raw MIDI interface: Access to a MIDI bus on a sound card. Works directly with the MIDI events. Protocol and timing management up to the programmer.
d) Mixer interface: Controls the devices on sound cards that route signals and control volume levels. Built on top of the control interface
e) Timer interface: Access to timing hardware on sound cards, used for synchronizing sound events.
f) Sequencer interface: A higher level interface for MIDI programming and sound synthesis than the raw MIDI interface. Handles much of the MIDI protocol and timing.
g) Hardware dependent Device

Some commands Used in OSS(open sound system) Devices:-

/dev/dsp:- D/A and A/D converter device, access, to generate audio or to read audio input.
/dev/mixer:- Mixer control (mainly for controlling volume)
/dev/audio:- Sun compatible digital audio (.au file format)
/dev/sequencer:-Audio sequencer (MIDI)
/dev/sequencer2:- Alternate sequencer device

CONFERENCE PAPER
"A National Level Conference on Recent Trends in Information Technology and Technical Symposium" On 09th March 2013
Organized by
Dept. of IT, Jawaharlal Darda Inst. Of Eng. & Tech., Yavatmal (MS), India

To create the device files:

sudo mknod /dev/dsp c 14 3
sudo mknod /dev/mixer c 14 0

The major and minor numbers for these devices are defined in Documentation/devices.txt in the kernel sources.

/proc/asound/version:- ALSA version
/proc/asound/cards:- List of available sound cards
0 [I82801DBICH4] : ICH4- Intel82801DB-ICH4
Intel 82801DB-ICH4 with
STAC9750/51 at 0xf4fff800, irq 5
1 [Modem]: ICH-MODEM – Intel 82801DB-ICH4 Modem
Intel 82801DB-ICH4 Modem at 0xb400, irq 5
/proc/asound/devices:- List of card devices
/proc/asound/card<i>/id:- Card identifier
/proc/asound/card<i>/pcm[c|p]<j>/info:- Information about a capture (c) or playback (p) PCM device.

For Creating Device File:

```
# Sound devices, group under /dev/snd
KERNEL=="controlC[09]*",    NAME="snd/%k"
KERNEL=="hwC[D09]*",        NAME="snd/%k"
KERNEL=="midiC[D09]*",      NAME="snd/%k"
KERNEL=="pcmC[D09cp]*",     NAME="snd/%k"
KERNEL=="seq",              NAME="snd/%k"
KERNEL=="timer",            NAME="snd/%k"
```

ALSA are the default audio drivers when you install ubuntu Linux on your system. The ALSA drivers control the sound input and output on the entire system, but may cause hang ups and audio freezes. Because of these conflicts, you may want to install new audio drivers that interact with your sound card and applications in different ways.

III. SCOPE FOR WORK

There are some Characteristics for this low power 24 bit stereo Audio Codec

a) Reduce the storage space and the bandwidth required for transmission of the stored audio file.
b) Implemented as libraries which interface to one or more multimedia players.
c) In hardware, "audio codec" refers to a single device that encodes analog audio as digital signals and decodes digital back into analog. In other words, it contains both an Analog-to-digital converter (ADC) and Digital-to-analog converter (DAC) running off the same clock.
d) This is used in sound cards that support both audio in and out, for instance.

A. Audio File formats Include:-

a. Uncompressed audio formats:

a) Store more or less any combination of sampling rates or bitrates. This makes them suitable file formats for storing and archiving an original recording.
b) Such as WAV, AIFF(audio interchange file format) header-less PCM;

b. Lossless compressed audio formats:

a) Stores data in less space by eliminating unnecessary data.
b) Enable the original uncompressed data to be recreated exactly.
c) Reduce processing time while maintaining compression ratio good
d) FLAC(Free lossless audio codec), WavPack

c. Lossy compressed audio formats:

a) Enables even greater reductions in file size by removing some of the data.
b) Greater compression
c) Compression is measured in Bit rate, lower the rate smaller the file.
d) MP3(For transfer and Playback of Music)

Phase 1: Driver Development

(a). Developing fully ALSA (ASoC) compliant audio codec driver for 24 bit stereo audio codec
(b). Testing this driver thoroughly to ensure high quality

Phase 2: Patch Posting and acceptance by alsa-devel mailing list

a) Submitting this driver to the open source community (alsa-devel mailing list)
b) Getting this driver accepted by community so that they become part of mainline Linux kernel.

Phase 3: Support and Maintenance (Optional)

a) Updating 24 bit stereo Codec driver for new Kernel Releases
b) Development of Codec driver for other codec chips
c) Bug Fixes and Enhancements for all developed drivers
d) Technical Support to codec and their end customers.

B. Development Platform:

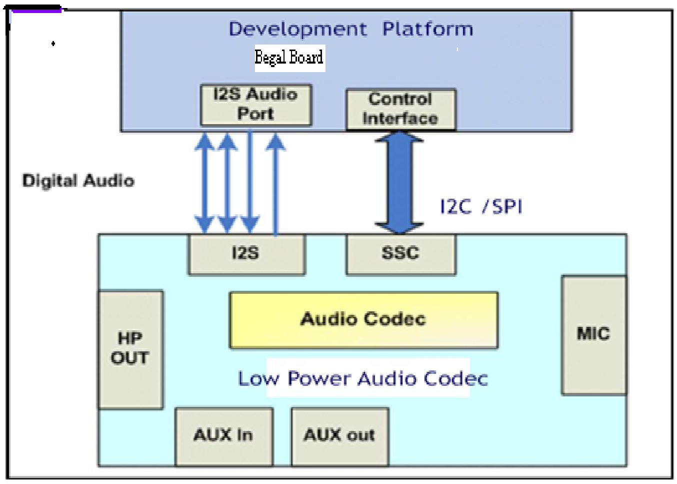

Figure: 2

a. One Beagle Board:- It has all required hardware and software components required for audio codec driver development, e.g.

a) Hardware – I2C, SPI, I2S connections are exposed via headers

b) Software – Board support is available in Linux kernel version 3.2

And also for “audio codec” one low power audio codec evolution board is used

CONFERENCE PAPER
“A National Level Conference on Recent Trends in Information Technology and Technical Symposium” On 09th March 2013
Organized by
Dept. of IT, Jawaharlal Darda Inst. Of Eng. & Tech., Yavatmal (MS), India

IV. APPLICATION

Device drivers, particularly on modern Linux platforms, can run in kernel-mode in user-mode. The primary benefit of running a driver in user mode is improved stability, since a poorly written user mode device driver cannot crash the system by overwriting kernel memory. On the other hand, user/kernel-mode transitions usually impose a considerable performance overhead, thereby prohibiting user mode-drivers for low latency and high throughput requirements. Application such as MP3 Players, IP phones, call center monitoring, or audio players/recorders.

V. CONCLUSION

In this paper we are developed a driver for the low power audio codec for the linux operating system and audio sample rate for the codec using commonly available system clock from 8MHZ through 33MHZ application. In the modern smart phones and also in android phones we are use this kind driver for the fast coding and decoding of our data. Also various types of iphones and for small application such as aMP3 player include such kind of audio codec. We are having various application for our daily electronics gadgets.

VI. REFERENCES

[1]. Sound Systems on Linux: From the Past To the Future Takashi Iwai <tiwai@suse.de>SuSE Linux AG, Nuremberg, Germany Linux 2003 Conference, Edinburgh, Scotland

[2]. //www.kernal.org/

[3]. Linux Audio Developers (LAD) mailing-listhomepage: http://www.linuxdj.com/audio/lad/

[4]. N.S. Jayant, Peter Noll, "Digital Coding of Waveforms - Principles and Applications to Speech and Video", Prentice Hall.

[5]. Robert Love, preemptive patchset: http://www.tech9.net/rml/linux/

[6]. Advanced Television Systems Committee, "Digital Audio Coimpression (AC-3) Standard", Doc. N52, Nov, 92.

[7]. H. G. Musmann, "Genesis of the MP3 audio coding standard," IEEE Trans. Consumer Electron., vol. 52, no. 3, pp. 1043–1049, Aug. 2006.

[8]. S. Dimitrov and S. Serafin. An analog I/O interface board for Audio Arduino open soundcard system. In Proceedings of the 2011 Sound and Music Computing Conference, 2011.

[9]. K. Brandenburg, "MP3 and AAC explained," in Proc. 17th Int. AES Conf., Florence, Italy, 1999, pp. 99–100.