

## An ALSA Audio System Based On the Linux System

Hai-Xia YU<sup>1, 2,a</sup>, Shao-Juan XU<sup>1</sup>, Cai-Kui FU<sup>2</sup>

1. City Institute, Dalian University of Technology, Dalian 116024, China;

2. School of Information Science and Technology, Dalian Maritime University, Dalian 116023, China

<sup>a</sup>email: yhx\_dl@163.com

**Keywords:** Linux; ALSA; Audio

**Abstract.** Audio system has become an indispensable part in many products. On Linux 2.6 kernel audio driver is taken as a bridge to connect the audio hardware with ALSA audio subsystem. This paper described interface of ALSA audio subsystem and audio driver programming essentials in detail. Specially, PC audio driver architecture provided even better ALSA support has been widely used in audio system.

### Introduction

Linux is one of the most excellent open source system software, it is widely popular because of its complete open characteristic and fine performance, its continually improved functions and good flexibility make it be the preferred program in the embedded area. As coding with development of application, Linux issues is kernel input sub-system, it makes most benefits for driver design. Audio system has become an indispensable part of many products. Audio driver module plays a very important role in the audio system<sup>[1]</sup>. Audio hardware can generate and capture the sound of computer system. In the Linux, people also can enjoy the audio like Windows system. Audio as one of the parts consisted of PC and its embedded products, and can be used for chatting on a laptop, phone call, listening to the MP3, playing multimedia, experiencing the system language instructions etc. There are two architectures in the Linux sound card driver. One is OSS (Open Sound System), and the other is ALSA (Advanced Linux Sound Architecture). Not only they defines the audio application programming interface specification, more importantly, but also defines Audio Driver Module Architecture. OSS is Linux 2.4 Core. The OSS drive frame contains character equipment interface-DSP and the mixer, in the user space programming completely uses the file operation. Audio driver module architecture is used by default, as some open source and other factors. In Linux2.6 kernel, along with its advantages, ALSA has replaced OSS architecture and become the default audio driver module architecture.

Based on the detailed research on Linux 2.6 kernel in ALSA audio subsystem, this paper described interface of ALSA audio subsystem, and the devising conception and audio driver programming essentials in detail. This paper also presents and illustrates designing the ALSA audio driver, each module detailed design scheme and implementation key points. For specific requirements and application scenario, the overall design of software is given in this paper. Finally, this scheme is verified on function and performance. The results show that the designs of the ALSA audio subsystem have better performance. It can satisfy the requirement of general application. Specially, PC audio driver architecture provided even better ALSA support has been widely used in audio system.

### The Audio Architecture in Linux

A typical circuit of digital audio system composes: embedded microcontroller controller/DSP with PCM, IIS or AC97 audio interface, Audio codec, input and output interface. Figure 1 shows the audio connection of PC-compatible system. With an external encoder the audio controller on the Southbridge can achieve docking with analog audio circuits. By the AC97 interfaces PC can connect external audio codec with the controller. The sound can be realized A/D and D/A

conversion. The analog signal amplification function and power amplifier can be completed.

A typical circuit of digital audio system is mainly composed of three parts:: embedded microcontroller controller/DSP with PCM, IIS or AC97 audio interface, Audio codec, input and output interface. Figure 1 shows the audio connection of PC-compatible system. With an external encoder the audio controller on the Southbridge can achieve docking with analog audio circuits. By the AC97 interfaces PC can connect external audio codec with the controller. The sound can be realized A/D and D/A conversion. The analog signal amplification function and power amplifier can be completed.

Audio codec converts the digital audio signal into an analog sound signal which required by sound speakers. While audio codec records through a microphone, it will perform the opposite circuit process. Other common audio inputs and outputs can be connected with audio codec including headphone, microphone, earphone, audio input and output line. Audio codec also provides audio mixing functions. It will mix all audio inputs and outputs, and control the volume of audio signals.

By using PCM (Pulse Code Modulation) technique digital audio data is able to get a bit-rate sample from analog sound signals. The task of encoder is to support PCM bit rate sampling and record the audio, and play the sample audios in different PCM bit rates at the same time.

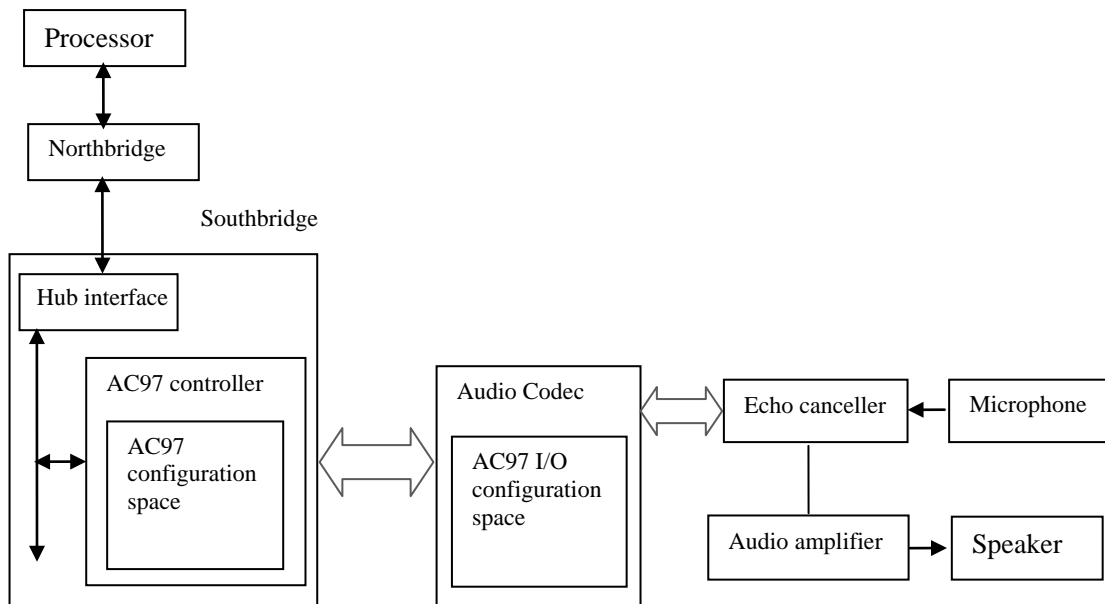


Fig.1. Linux PC Audio Architecture

### Proposed audio subsystem in Linux

OSS and ALSA are the most common two sets of standards. Linux 2.6 cores tend to abandon the OSS structure architecture, and default ALSA architecture. The ALSA actuation frame takes card and the module as a master line, in the user space programming does not use the document connection, but `alsa-lib`. Figure 2 shows a simplified diagram of ALSA sound drivers and user-space architectures<sup>[2]</sup>. Normally, we use the ALSA audio architecture. In the layer of kernel device drivers, the ALSA provides `ALSA-Driver`. And in the application layer, the ALSA provides `ALSA-Lib` for us. The application can use the API which provided by `ALSA-Lib` to control the bottom audio hardware. It is shown in Figure 3.

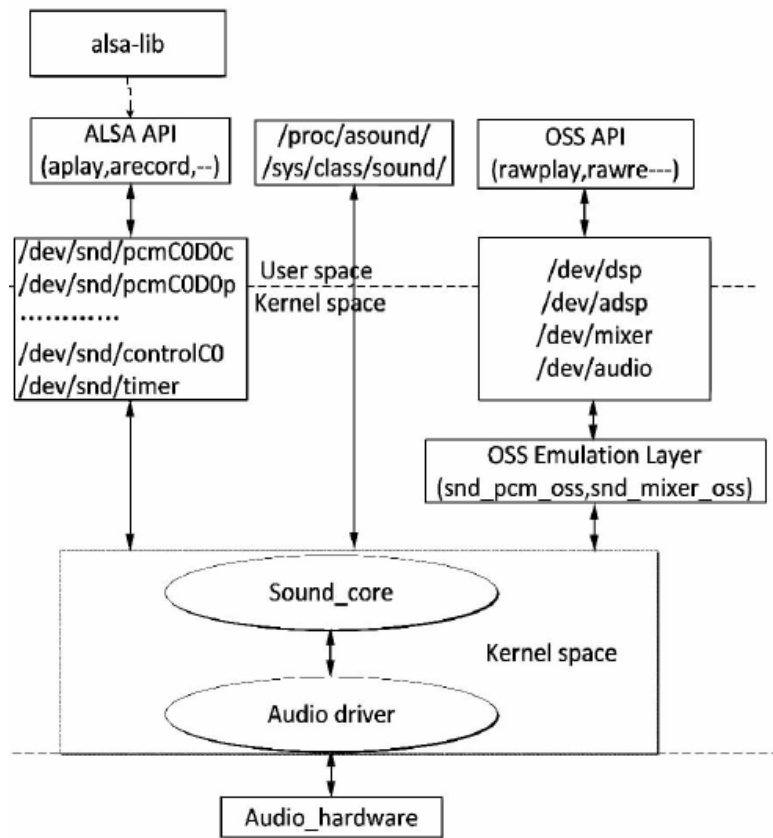


Fig.2. ALSA sound drivers structure diagram and user space system

Sound core is the core of code. It consists of routines and structures which can be used by the other layers of Linux<sup>[3]</sup>. The sound core has isolated to some extent, each part of sound subsystem are isolated from each other. Meanwhile, sound subsystem provides API of ALSA for application. For example, ALSA core creates and manages `/dev/snd/*` device nodes, `/dev/snd/controlC0` is a control node. `/dev/snd.pcmC0D0p` is playing device.

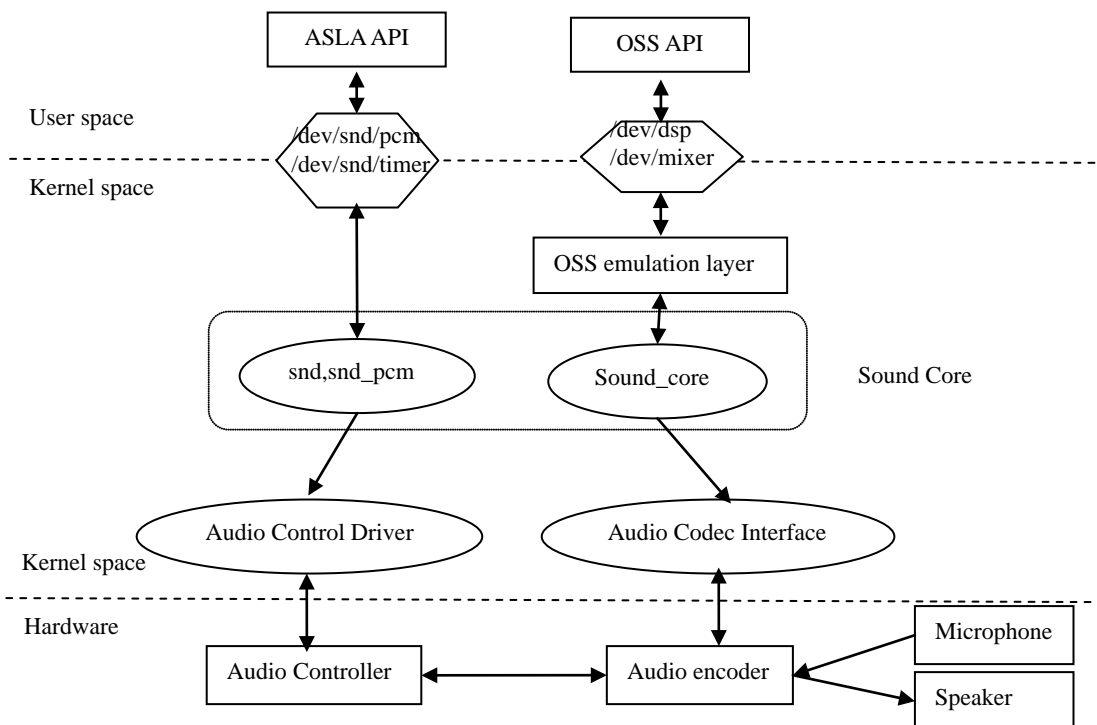


Fig.3. Linux Sound Subsystem

## The Validation of the Proposed Audio System

The audio system is an important part of PC systems, audio driver as a bridge to connect the audio hardware and operating system, plays an important role in the audio system. The digital audio signal taking the key components of multimedia is not lack of the Linux device driver.

(1)The compositions of audio driver

The audio driver is mainly made of three parts: the audio player processing routine, the playback processing routine and the mixer controlling function<sup>[4]</sup>. ALSA sound is comprised of many sound card drivers. It also provides an API library called liba sound. Application developers should not use liba sound kernel ALSA interface. Figure 4 is shown a typical audio application flow chart .

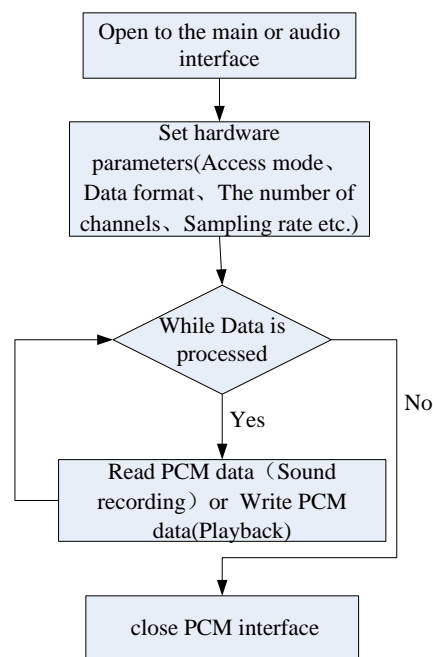


Fig.4. Audio application flow

(2)The statement of sound card example

ALSA audio driver completed the processing of audio data, including register release and PCM sound card interface function, etc. For audio driver, the "card" records must be assigned.

snd\_card\_new creates sound card records , including ID and name<sup>[5]</sup>.

The examples of struct snd\_card characterizes the sound card.

```
struct snd_card{
    int number,
    char id[16],
    struct module *module,
    void *pre_data,
    /*.....*/
    struct list_head controls,
    struct device *dev,,
    /*.....*/}

struct snd_card *card=snd_card_new(-1,id[dev->id,THIS_MODULE,0]);
```

For example, snd\_card\_new is used to create sound card. The first parameter is the card index which is used to distinguish other sound card in the multi-card system. The second parameter id returns backto the id field of snd\_card structure. The third parameter is the owner module. The last parameter is the size of private data, usually is used to store data with chips.

snd\_card\_free is used in the example of releasing sound coming from sound card subsystem.

(3)The examples of creasing PCM (Pulse Code Modulation)

snd\_pcm\_new() is used to create PCM example connecting with the examples of creating sound card.

```
int snd_pcm_new(struct snd_card *card,char *id,int device,
               int playback_count,int capture_count,
               struct snd_pcm **pcm);
```

The first parameter card is the example of creating sound card. The second parameter device shows one is the current created PCM. The first PCM device starts from 0. The third parameter playback\_count indicates how many playback sub streams of this PCM are captured. The fourth parameter indicates t how many capture sub stream of this PCM are captured.

(4)The setting of PCM example

```
void snd_pcm_set_ops(struct snd_pcm *pcm,
                    int direction,
                    struct snd_pcm_ops *ops);
```

snd\_pcm\_set\_ops is used to Bind play operation with PCM instance created, the created PCM example, set the PCM example, and define the PCM operation interface in the structure of struct snd\_pcm\_ops.

(5)Binding mixer device

Using the snd\_ctl\_add() to connect the sound card with one mixing control association unit, it can be used for volume controlling.

```
snd_ctl_add(card,snd_ctl_newl(&mycard_playback_vol,&myctl_private))
```

snd\_ctl\_newl() uses the structure of snd\_kcontrol\_new as the first parameter, and sends back a pointer which point to snd\_kcontrol structure.

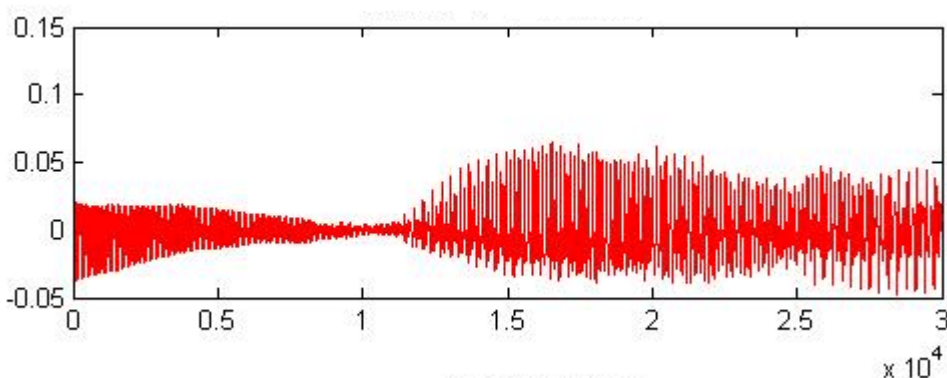
snd\_kcontrol structure is used to describe a control unit. Our driver uses it as the entry point for volume control. snd\_ctl\_add() register nd\_kcontrol unit of ALSA framework, a control function is called when the user application execute

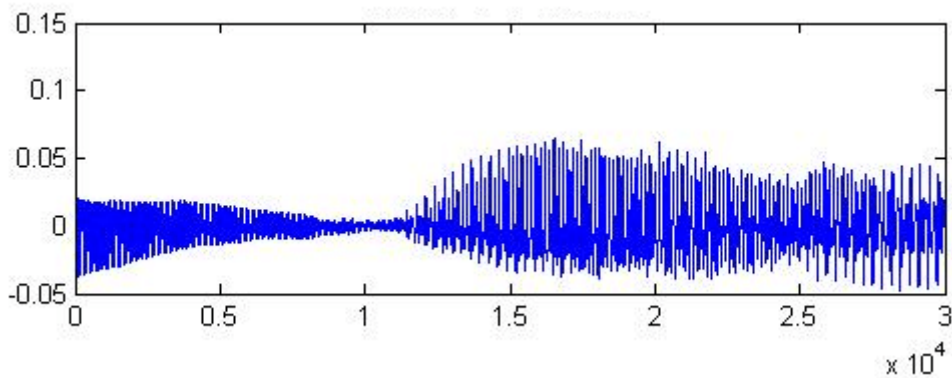
(6)Register sound card into ALSA framework

Device is taken as a file by Linux device driver<sup>[6]</sup>. Use the snd\_card\_register(card), and register the sound card into the ALSA frame.

## Experiment Results

This paper plays a recording on adopted in this system, the system plays sound samples compared with real music without distortion. Experiments show that the driver system runs steadily and achieve the desired results.





## Conclusion

This paper studies the framework of sound subsystem and the main described structure of sound subsystem in Linux system. It also studies the described structure of audio driver and analyzes the design features of audio driver.

## References

- [1] Yong Ding, Yu Zhou, On Linux 2.6 Based Embedded ASOC Audio Driver And Its Implementation, Computer Applications and Software, Vol. 27 No. 4, Apr. 2010, 267-270
- [2] Yuan-Zhi Zheng-Guo Qiang Sun, Analysis and verification of audio driver based on ASOC for embedded system, Information Technology, NO.29, 2013.
- [3] Yong-Ming Wei, Yue-Di, Shu-Yi Zhong translate. Linux device driver [M]. China Electric Power Press, 2006. 46-74.
- [4] Bao-Hua Song, Shao-Ran He, Hai-Bin Shi, Guo-Cheng Wu (translate). LINUX device driver development. Beijing, People Post Press, 2010.
- [5] Peng Zhou, Cheng Wang, etc, Design Of WM8976 Based On The ALSA Audio Driver, JOURNAL OF WUT (INFORMATION & MANAGEMENT ENGINEERING), Vol.33, No.4, Aug. 2011: 517-520.
- [6] CORBET J. Alessandro Rubini Linux Device Drivers [M]. Sebastopol: O'Reilly & Associates, 2005: 137-142.