

基于 SEP4020 的嵌入式 Linux 音频驱动程序设计

程杰, 凌明

(东南大学 国家专用集成电路系统工程技术研究中心, 江苏 南京 210096)

摘要: 论述了 Linux 操作系统中声卡驱动程序的设计方法, 主要介绍了基于 OSS 的声卡驱动设计原理以及 Linux 操作系统中声卡驱动程序的接口函数。针对具体硬件平台编写了相应的驱动程序, 并介绍了在 Linux 操作系统中使用声卡设备的几种常见方法。实现了 SEP4020 处理器在 Linux 平台的声卡驱动。

关键词: Linux 驱动程序; UDA1341; SEP4020

中图分类号: TP314

文献标识码: A

Embedded Linux sound driver based on SEP4020

CHENG Jie, LING Ming

(National Engineering Research Center for Application-Specific Integrated Circuit Systems, Southeast University, Nanjing 210096, China)

Abstract: In this paper, a method is provided to design a Linux sound driver based on OSS. The OSS theory in Linux has been introduced. To accomplish the sound driver, a few structures and functions should be filled by the designer. This paper also provides some ways to use the sound card in Linux. The SEP4020 microprocessor has been used in Linux sound driver in this paper.

Key words: Linux driver; UDA1341; SEP4020

目前, 手机、MID、MP3 等许多嵌入式设备都包含数字音频设备, 数字音频系统一般由处理器通过 IIS、AC97 等接口连接外部音频编解码器 (CODEC), 音频解码器实现声音的 AD 和 DA 转换。音频编解码是数字音频系统的核心, 衡量其性能的主要指标有采样频率和量化精度。在 Linux 系统中, 为了处理数字音频相关工作, 先后出现了 2 种音频设备框架: OSS 和 ALSA。ALSA 完全开放, 但是 OSS 更为成熟, 本文将基于 OSS 架构介绍数字音频设备和音频设备接口。

1 软硬件平台

1.1 SEP4020 嵌入式处理器

本文所述驱动程序基于 SEP4020 嵌入式微处理器, SEP4020 是由东南大学国家专用集成电路系统工程技术研究中心设计的一款处理器, 采用 0.18 μm 标准 CMOS 的工艺设计, 内嵌 ASIX CORE (32 位 RISC 内核, 兼容 ARM720T, 带 8 KB 指令数据 Cache 和全功能 MMU)。SEP4020 芯片中集成各种功能, 包括:

- (1) 8/16 bit SRAM/NOR FLASH 接口, 16 bit SDRAM 接口;
- (2) 硬件 NAND FLASH 控制器, 支持 NAND FLASH 自启动;

- (3) 10 M/100 M 自适应以太网 MAC, 支持 RMI 接口;
- (4) 64 KB 高速片上 SRAM;
- (5) 支持 IIS 音频接口;
- (6) 支持 MMC/SD 卡;
- (7) LCD 控制器, 支持 TFT 彩屏和 STN 黑白、灰度屏;
- (8) RTC, 支持日历功能/WatchDog, 支持后备电源;
- (9) 10 通道 TIMER, 支持捕获、外部时钟驱动和 MATCH OUT;
- (10) 4 通道 PWM, 支持高速 GPIO;
- (11) 4 通道 UART, 均支持红外;
- (12) USB1.1 Device, 全速 11 Mb/s;
- (13) 2 通道 SSI, 支持 SPI 和 Microwire 协议;
- (14) 2 通道 SmartCard 接口, 兼容 ISO7816 协议;
- (15) 支持最多 91 个 GPIO, 14 个外部中断;
- (16) 支持外部 DMA 传输;
- (17) 片上 DPLL, 支持 IDLE、SLOW、NORMAL、SLEEP 等多种功耗模式。

1.2 IIS 音频接口

本文主要使用了 SEP4020 处理器的 IIS 音频接口, IIS 接口 (Inter-IC Sound) 在 20 世纪 80 年代首先被 Philips 公司

软件天地 Software Technology

用于消费音频产品。IIS 总线只处理声音数据,其他信号(如控制信号)必须单独传输。为了使芯片的引出管脚尽可能少,IIS 只使用了 3 根串行总线,分别是提供分时复用功能的数据线、字段选择线(声道选择)、时钟信号线。SEP4020 的 IIS 控制器具有如下功能:

- (1)支持 MASTER 和 SLAVE 模式;
- (2)支持 TRANSMITTER 和 RECEIVER 功能;
- (3)支持 32、16、8 bit 字长;
- (4)支持立体声和单声道;
- (5)支持静音和停止播放;
- (6)数据高位(MSB)先出/先入;
- (7)接收发送共享 8×32 数据 FIFO。

1.3 UDA1341 音频编解码芯片

SEP4020 开发板使用的音频编解码芯片为 NXP 公司的 UDA1341。UDA1341 支持 IIS 总线数据格式,采用位流转换技术进行信号处理,具有可编程增益放大器(PGA)和数字自动增益控制器(AGC)。UDA1341 对外提供 2 组音频信号输入接口,每组包括左右 2 个声道。由于 IIS 总线只处理音频数据,因此 UDA1341 还内置用于传输控制信号的 L3 总线接口。L3 接口相当于混音器控制接口,可以控制输入/输出音频信号的低音及音量大小等。

1.4 Linux 嵌入式操作系统

软件平台采用 Linux,版本号为 2.6.16。Linux 是当今最流行的操作系统之一,由于其源码开放性,现代操作系统设计的新思想和新技术能够不断用于其中,是一个非常好的学习平台。其次,Linux 操作系统占用资源较少,对处理器要求低,可运行于大多数含 MMU 的处理器上,特别适合嵌入式领域。SEP4020 处理器运行频率为 88 MHz,拥有 MMU 单元,因此选择 Linux 操作系统作为其软件平台。此外,Linux 平台具有完善的音频设备框架,基于该设备框架编写相应的驱动程序可以大幅简化开发难度,缩短开发时间。

2 Linux OSS 音频设备驱动

OSS (Open Sound System) 是 Unix 平台上一个统一的音频接口。过去,每个 Unix 厂商都会提供一个自己专有的 API 用来处理音频。这就意味着为一种 Unix 平台编写的音频处理应用程序,在移植到另外一种 Unix 平台上时,必须要重新编写。OSS 出现以后情况就大不一样了,只要音频处理应用程序按照 OSS 的 API 来编写,那么在移植到另外一个平台时,只需要重新编译即可。因此,OSS 提供了源代码级的可移植性。由于 Linux 对 Unix 有着良好的兼容性,因此只需很少的改变,就可以使基于 OSS 的程序在 Linux 下正常运行,Linux 内核也对 OSS 架构提供了完善的支持。

2.1 OSS 驱动组成

OSS 标准有 2 个基本音频设备: DPS (数字信号处理器)和 Mixer (混音器)。

DSP 也称为编解码器,可实现录音和放音的功能,其对应的设备文件是/dev/dsp 或者/dev/sound/dsp。向该设备写数据即意味着激活 CODEC 上的 D/A 转换器进行播放,而向该设备读数据则意味着激活声卡上的 A/D 转换器进行录音。DSP 的指标主要有:采样速率(电话为 8kHz,CD 为 44.1 kHz)、通道数目(单声道、立体声)、量化精度(8 bit、16 bit)。

Mixer 设备用来控制多个输入、输出的音量,也控制输入(microphone、line-in、CD)之间的切换。

2.2 DSP 和 Mixer 设备函数接口

DSP 设备接口函数中比较重要的有 read ()、write () 和 ioctl () 等。

write () 函数的作用是从用户空间复制音频数据到内核空间缓冲区并最终发送到音频控制器。在数据从缓冲区复制到音频设备的过程中,通常会使用 DMA。在放音时,驱动设置完 DMA 控制器的源地址和目的地址,DMA 控制器会自动将数据发送到 CODEC 的 FIFO 中,直到发完设定数据再通知上层。

read () 函数的作用是从音频控制器中获取录音数据到缓冲区并复制到用户空间。

ioctl () 函数用来设定采样速率、通道数、量化精度、DMA 缓冲区大小等参数。

mixer 设备主要通过 ioctl () 函数来实现不同的功能。

2.3 OSS 用户空间编程

OSS 的层次结构非常简单,用户通过调用 API 函数访问 OSS 驱动。开发 OSS 应用程序的一般流程是:

- (1)在应用程序中包含 OSS 结构的头文件 #include <linux/soundcard.h>;
- (2)打开设备文件,返回文件描述符;
- (3)使用 ioctl () 函数设置设备参数;
- (4)使用 read () 函数录音或使用 write () 函数放音;
- (5)关闭打开的设备,结束应用程序。

3 SEP4020+UDA1341 OSS 驱动设计

3.1 硬件接口描述

如图 1 所示,SEP4020 芯片与 UDA1341 相连,UDA1341 的 L3 总线用来配置其自身的参数,可以用来控制输入/输出音频信号的音量大小、增益、低音等。IIS_SCK 为 UDA1341 接口的时钟信号;GPIO_PG11 用作 GPIO 来控制双向模拟开关 4066,将处理器 IIS 接口数据信号 IIS_SD 连接在 UDA1341 的数据输入或者输出信号上,从而进行录音/放音的数据切换。SEP4020 与 UDA1341 的详细接口描述如表 1 所示。

3.2 注册驱动及初始化硬件接口

在 UDA1341 OSS 驱动的模块加载函数中,将完成以下工作,初始化代码参见图 2。

- (1)初始化 3 根 GPIO 口,模拟 UDA1341 的控制总线——L3 总线;

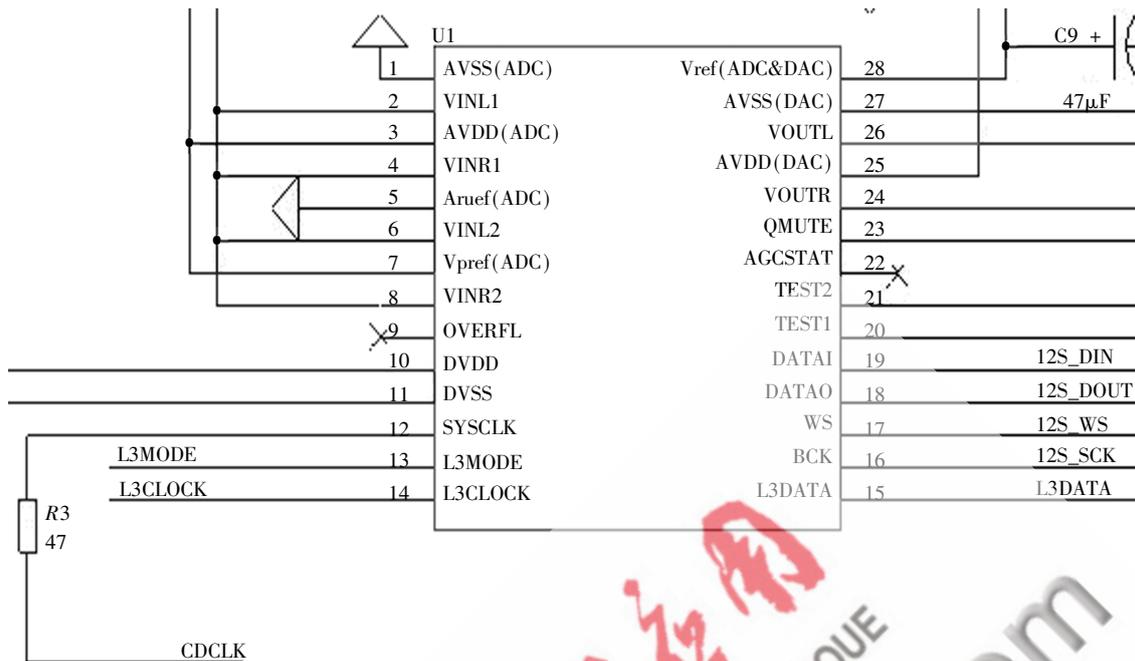


图 1 SEP4020 与 UDA1341 接口连线图

UDA1341 OSS 驱动初始化函数代码清单如图 2 所示。

```
static int __init sep4020_uda1341_init(void)
{
    printk("sep4020_uda1341_init! ");
    init_gpio();
    init_pwm();
    init_sep4020_iis_bus();
    init_uda1341();
    audio_init_dma(&output_stream, "UDA1341 out");
    audio_dev_dsp = register_sound_dsp (&sep4020_audio_fops,
-1);
    audio_dev_mixer = register_sound_mixer (&sep4020_mixer_fops, -1);
}

```

图 2 UDA1341 OSS 驱动初始化函数代码清单

表 1 IIS 音频接口关键信号描述

名称	信号定义
I2S_SCK	系统时钟
I2S_SD	I2S 数据信号
I2S_WS	I2S 字选择信号
SSL_CS1	CODEC 的 L3MODE 信号
SSL_TXD	CODEC 的 L3 总线数据信号 L3DATA
SSL_CLK	CODEC 的 L3 总线时钟信号 L3CLOCK
PWM3	CODEC 的系统时钟 SYSCLK
nCSC	音频数据流向控制信号

- (2)初始化 IIS 的控制口，配置成 IIS 模式；
- (3)初始化 PWM，提供 CDCLOCK 信号；
- (4)调用 init_sep4020_iis_bus () 函数，初始化 IIS 寄存器；

- (5)调用 init_uda1341 () 函数，初始化 UDA1341 芯片；
- (6)调用 audio_init_dma () 函数，初始化 DMA 控制器；
- (7)注册 DSP 和 Mixer2 个设备。

初始化 PWM 给 CDCLK 提供信号时，由于 UDA1341 芯片本身要配一个分频系数 (256、384、512)，这里使用了 256，所以 PWM 频率要配成接近 44.1 kHz×256，最终配出的 CDCLK 和采样率如下：

$$88 \text{ MHz}/(4 \times 2) = 11 \text{ MHz} \quad (\text{PWM4_DIV_V} = 0 \times 4)$$

$$11 \text{ MHz}/256 = 42.96 \text{ kHz}$$

初始化 IIS 时也要配置一个分频系数，即 SCK 时钟分

《微型机与应用》2009 年第 20 期

频参数，参考 SEP4020 数据手册，计算得到 88 MHz 主频时配为 0×1F 最接近 44.1 kHz。

初始化 UDA1341 芯片时首先完成 2 个函数，uda1341_l3_address () 和 uda1341_l3_data ()。SEP4020 使用 GPIO 用口线模拟，初始化参数中，重音、清晰度随便配即可，分频比配置为 256 fs；SEP4020 的工作模式必须配置成 IIS 方式，否则是杂音；音效可以尝试替换各种参数，以获得最佳效果。

3.3 DSP 接口音频数据传输

OSS 声卡驱动中，DSP 接口的读写函数是核心，编写 OSS 声卡驱动程序需要完整的最关键的函数是 read () 和 write ()。

以 write () 函数为例, 在函数现实中使用了 Ring Buffer(环形缓冲区) 技术。具体实现方法为: 初始化设备时开辟 8 个环形缓冲区 (缓冲区数量可自由配置), 从内存中申请了 8 块独立空间。当驱动调用 write () 函数时, 系统以此使用这 8 个缓冲区, 并不断地填充数据进入缓冲区。当每一个缓冲区填充完毕, 就将其排列进 DMA 队列, 然后继续填充下一个缓冲区。如果所有缓冲区都进入 DMA 队列, 则进程进入挂起状态, 等它出现空闲缓冲区。

同时, write () 函数负责配置 SEP4020 的硬件 DMA, 每次 DMA 传输前需要清空相应通道上的 DMA 传输完成中断寄存器。每次传输的数据大小为 8 196 B, DMA 的 burst 长度配置为 4, 源端地址配置为缓冲区地址, 目的地址配置为 IIS 硬件寄存器 FIFO 地址。特别需要注意的是由于 Linux 操作系统采用虚拟内存管理, 所有经过 CPU 处理的数据 (包括寄存器地址) 都必须使用虚地址。由于 DMA 操作不经过 CPU 处理, 因此对其配置时, DMA 源和目的地址需要填入物理地址。具体配置代码见代码清单 2, 更详细的说明请参考 SEP4020 用户手册^[4]。

本文介绍了针对 SEP4020 处理器在 Linux 操作系统下声卡驱动的开发流程, 叙述了 Linux OSS 层、SEP4020 的 IIS 控制器和 UDA1341 音频解码芯片的基本原理。本驱动的开发主要为 Linux OSS 层和 IIS-UDA1341 硬件编写接口函数和数据结构, 为底层硬件和上层软件提供了连接的桥梁。在 Linux OSS 基础上, 用户可以开发丰富多彩的应用程序, 满足客户提出的多媒体音频需求, 为嵌入式 Linux 的应用开辟了更广阔的道路。

IIS-DMA 寄存器配置代码清单如图 3 所示。

```
/* 清除要用的 dma 通道 (dma1) 上的传输完成中断寄存器 */
*(volatile unsigned long*)(DMAC_INTTCLEAR_V) = 0x2;

*(volatile unsigned long*)(DMAC_INTTCLEAR_V) = 0x0;
*(volatile unsigned long*)DMAC_C1CONTROL_V =
((AUDIO_FRAGSIZE > 2) << 14) + (1 << 12) + (2 << 9) +
(2 << 6) + (3 << 3) + 3;
*(volatile unsigned long*)DMAC_C1SRCADDR_V = b->
dma_addr;
*(volatile unsigned long*)DMAC_C1DESTADDR_V =
I2S_DATA;
*(volatile unsigned long*)DMAC_C1CONFIGURATION_V
= 0x200b; //iis
```

图 3 IIS-DMA 寄存器配置代码清单

参考文献

- [1] 冯国进. 嵌入式 Linux 驱动程序设计从入门到精通 [M]. 北京: 清华大学出版社, 2008.
- [2] 韦东山. 嵌入式 Linux 应用开发完全手册 [M]. 北京: 人民邮电出版社, 2008.
- [3] 宋宝华. Linux 设备驱动开发详解 [M]. 北京: 人民邮电出版社, 2008.
- [4] SEP4020 用户手册 v1.7.1[Z]. 南京博芯电子有限公司, 2009.
- [5] Superlp.s3c2410-uda1341.c[Z]. 三星电子有限公司, 2008.

(收稿日期: 2009-08-13)