基于嵌入式 Linux 的音频系统①

Audio System Based on Embedded Linux

詹群峰　李灵杰　叶利福　陈文芗　(厦门大学 物理与机电工程学院 福建 厦门 361005)

摘　要： 利用 SamSung 公司的 S3C2410 微处理器和 Philips 公司的 UDA1341 音频芯片设计了一种应用于 GPS 车载导航的嵌入式音频系统。该嵌入式音频系统硬件部分采用基于 IIS 总线的音频系统体系结构；软件上,以嵌入式 Linux 操作系统作为平台,实现对 UDA1341 的驱动。

关键词： S3C2410 UDA1341 DMA Linux IIS 总线

1　引言

随着人们生活水平的提高，汽车的使用越来越普遍，而 GPS 车载导航系统的市场也已进入迅速发展时期。由于车载系统的嵌入式硬件资源一般很有限，所以在设计音频系统过程中如何避免声音失真及抖动，如何保证声音流畅也越来越重要。本文利用飞利浦公司的 UDA1341 音频编解码器设计一种嵌入式音频系统，用于 GPS 车载系统的行车播报及预警，包括硬件和软件设计两部分，其中主要讨论了程序中对于缓存的优化设计。

在嵌入式操作系统方面，采用了近几年不断兴起的嵌入式 linux 操作系统，由于其开放源代码，成本低，便于裁剪，以及丰富的软件支持[1]，在各种嵌入式 Linux OS 迅速发展的状况下，Linux OS 所占领的市场份额正逐渐扩大。

2　系统硬件设计方案

2.1 硬件概述

系统选用基于 32 位 ARM920T 内核的 S3C2410 作为嵌入式微处理器[2], S3C2410 内置 IIS 总线接口，有 4 个 DMA 通道，为了实现音频数据的全双工传输，本文使用 S3C2410 的 DMA 通道 1 和通道 2，接收数据选择通道 1，发送数据选择通道 2。S3C2410 的 DMA 控制器没有内置的 DMA 存储区域，因而程序中需为音频设备分配 DMA 缓存区，通过 DMA 直接将需要回放或者录制的数据放在内存的 DMA 缓存区中。

数字音频编解码芯片采用 UDA1341，UDA1341 支持 IIS 总线数据格式，是一款低功耗、全双工音频编码解码专用芯片，集 ADC 和 DAC 于一体，完全集成了模拟前端，包括可编程增益控制(PGA)和一个数字自动增益控制(AGC)，出色的数字信号处理(DSP)特性以及低功耗的优点使其可以广泛应用于立体声的磁盘系统和便携式的各种设备中。

UDA1341TS 和 S3C2410 之间通过 IIS 总线进行数据交互，其音频系统逻辑图及在总系统中的结构如图 1 所示。

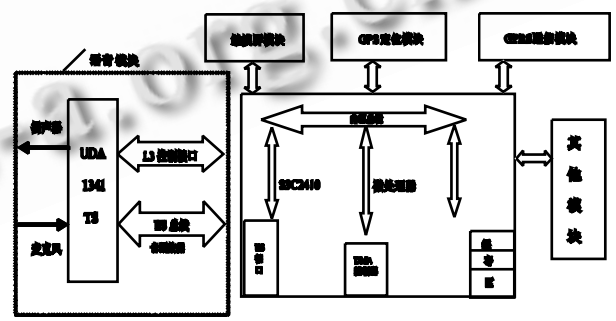

图 1 基于 IIS 总线的音频系统逻辑图(word2003 软件)

IIS(Inter-IC-Sound)总线是 PHILIPS 公司提出的串行数字音频总线协议。它是一种面向多媒体的音频总线，专用于音频设备之间的数据传输，为数字立体声提供序列的连接至标准编解码器。IIS 总线使用了 4

① 基金项目:福建省科技项目(2007H0036)
收稿时间:2009-01-12

条串行总线：两条提供分时复用功能的数据线、字段选择和时钟信号线，并且提供三种工作模式[3]，如表 1 所示。

表 1 IIS 总线结构图

模式	特点
正常传输模式	由 IISCON 寄存器对 FIFO 进行控制，CPU 通过轮询方式访问 FIFO 寄存器，以完成对 FIFO 缓存传输或接收。
DMA 模式	由 DMA 控制器对 FIFO 进行控制,当 FIFO 满时，DMA 控制器对 FIFO 中的数据进行处理，DMA 模式选择由 IISCON 寄存器的第 4 和第 5 位控制。
传输/接收模式	即全双工模式，IIS 数据线同时接收和发送音频数据,DMA 服务请求由 FIFO 只读寄存器自动完成。

2.2 硬件电路设计

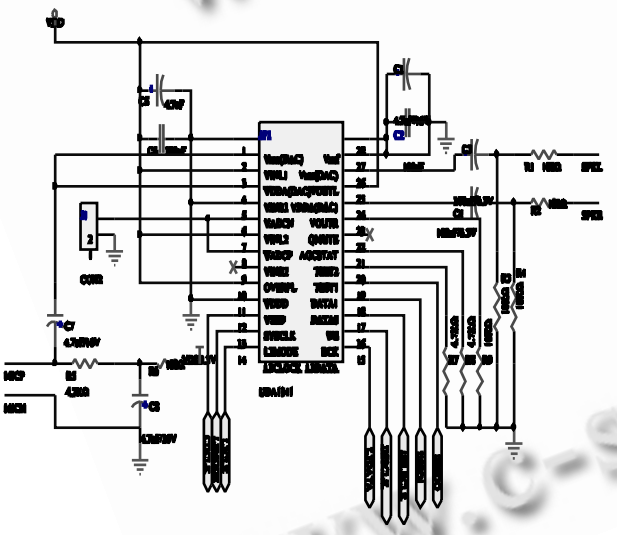

图 2 UDA1341 芯片接口电路图(protel99 软件绘制)

如图 2 所示，S3C2410 的 IIS 总线信号与 UDA1341TS 的 IIS 信号直接相连,UDA1341TS 内置有用于传输控制信号的 L3 总线接口，可以控制输入、输出音频信号的低音及音量大小等，其引脚 L3MODE、L3DATA 和 L3CLOCK 分别连接 S3C2410 的 GPB2、GPB3 和 GPB4 通用数据引脚，利用这 3 个 I/O 口模拟 L3 总线的全部时序和协议。L3 总线的时钟不是连续时钟，它只在数据线上有数据时才发出 8 个周期的时钟信号，其它时钟时始终保持高电平。MIC 的偏置电压从 3.3V 上面直接分压取得。UDA1341 的 test 脚不用，故接到低电平上。

3 音频设备的驱动程序设计

音频设备的驱动程序包含混音器和 DSP 设备驱动两部分。混音器设备用于音频的各输入、输出通道的选择和音量增益的控制。它通过读写 UDA1341TS 音频编解码器的相关寄存器实现控制功能。DSP 设备用于音频流数据的读、写，它通过将音频数据流读出、写入 DMA 缓冲区来实现音频的播放与录制功能[4]。这两个设备分别对应两个设备文件(/dev/dsp 和 /dev/mixer)。

3.1 音频设备的注册和卸载

音频设备驱动程序的注册是通过定义一个 struct device_driver 的结构体，对相应成员赋值后调用 driver_register() 函数来完成。s3c2410_uda1341_driver 结构体变量中有一个 probe()函数，它申明了驱动的初始化函数，并通过调用 init_uda1341()完成设备的初始化工作。包括：对相关寄存器初始化，完成对 UDA1341 音量、采样频率、L3 接口等的初始化，对控制器中的相关寄存器进行设置，对输入、输出缓存区数据结构进行初始化，设置 DMA 中断等。它在 driver_register()函数将驱动注册到内核后会被马上执行。

在与注册函数相对应的设备卸载函数中，调用了函数 driver_unregister()，该函数的执行会导致 s3c2410_uda1341_driver 结构体中的 remove()函数的执行，remove 域指向 s3c2410_uda1341_remove()函数，在该函数中，调用 unregister_sound_dsp()函数卸载 DSP 设备，调用 unregister_sound_mixer()函数卸载混音器设备。卸载设备时还同时释放了驱动程序中使用过的各种资源，包括 DMA、设备中断等。

3.2 音频设备缓存区的设计

在音频设备的驱动程序设计中，DMA 缓存区的管理和数据传输是最主要的问题。常见的设计思路是使用一个缓存区，CPU 先对缓存区处理，然后挂起，音

频设备对缓存区操作，音频设备处理完后唤醒 CPU，如此循环。如果需要处理大量音频数据的音频设备驱动程序，可以使用双缓冲[5]。

由于在 GPS 车载导航系统中行程的播报以及警报提醒等对实时性有很高的要求，而且需要处理的数据量又很大，所以在 CPU 的负载比较大的情况下，当缓存设计相对较小时，应用程序处理缓存中的数据的时间就比较短，缓存可能会溢出，从而将这些数据转存到永久性存储设备上时可能会出现数据丢失的问题；而当缓存设计过大时，又容易出现延时[6]。所以必须合理分配资源，使用合适的算法，加快对音频数据的处理，并减少延时。故其关键在于缓存区块的设计。

为了同时解决延时和声音不失真的问题，驱动程序使用了环形、多段缓存机制。在这种机制下，将缓存区分割成若干个相同大小的块，并使用算法实现环形缓冲。块的大小和个数使用 ioctl 系统调用来调整，这样对较大的缓存区的操作就转变为对较小的缓冲段的操作，从而可以在不增加对缓存区操作时间的情况下增加缓存区的大小。

音频设备的 DMA 缓存区的初始化过程核心代码如下：

```
static int audio_setup_buf(audio_stream_t* s)
{
    ……
/*使用 for 语句对音频缓冲区片进行循环操作，一次性分配大小为 “s->nbfrags×s->fragsize。若不成功，将段个数逐渐减少 */
  for (frag = 0; frag < s->nbfrags; frag++)
  {audio_buf_t *b = &s->buffers[frag];
    // 定义 audio_buf_t 结构指针变量
    if (!dmasize)
      {dmasize = (s->nbfrags - frag) * s->fragsize;
       do
      {dmabuf = consistent_alloc(GFP_KERNEL|GFP_DMA, dmasize, &dmaphys),
       if (!dmabuf)
          dmasize -= s->fragsize;
} while (!dmabuf && dmasize);
   /*内存空间的虚拟起始地址 dmabuf 返回的值为 0，若内存申请失败，dmasize 减去一个缓冲区片*/
      if (!dmabuf)
          goto err;
          b->master = dmasize;
          memzero(dmabuf, dmasize);
      }
      b->start = dmabuf;/*一段内存分配成功，初始化该 audio_buf_t 结构体*/
      b->dma_addr = dmaphys;
      sema_init(&b->sem, 1); //初始化信号量为 1
      DPRINTK("buf %d: start %p dma %d\n", frag,
b->start, b->dma_addr);
      dmabuf += s->fragsize;
      dmaphys += s->fragsize;
      dmasize -= s->fragsize;
     }
     s->buf_idx = 0;
     s->buf = &s->buffers[0];
     return 0;
err:
     ……
}
```

在音频驱动程序中，有两个很重要的缓存区数据结构体 audio_stream_t 和 audio_buf_t，audio_buf_t 中包括了内存的虚拟起始地址和物理起始地址，内存大小等等，audio_stream_t 包括了音频缓冲区片大小、数量，DMA 通道等等，它是用于管理多缓冲区内存的结构体，为音频流数据组成了一个环形缓冲区，简单来说，我们用 audio_buf_t 来管理一段内存，而用 audio_stream_t 来管理多个 audio_buf_t 控制下的内存。程序中还使用了 consistent_alloc 函数来分配内存空间。

上述对音频设备缓存区的优化设计，可以同时解

决在音频数据量较大时易产生声音延时以及声音失真的问题，对于声音的流畅播放起到了关键的作用，保证了及时的行程播报和提前预警。

3.3 音频应用程序的设计

本系统中的音频设备主要用来播报行程、行车状态和警报提醒，先将需回放的音频文件在 PC 机上录制下来，存放为.wav 的文件格式，然后下载到嵌入式设备上，在需要时进行回放。音频设备的打开和关闭分别由 open()和 close()函数来完成，在此不赘述。程序运行流程如图 3 所示。

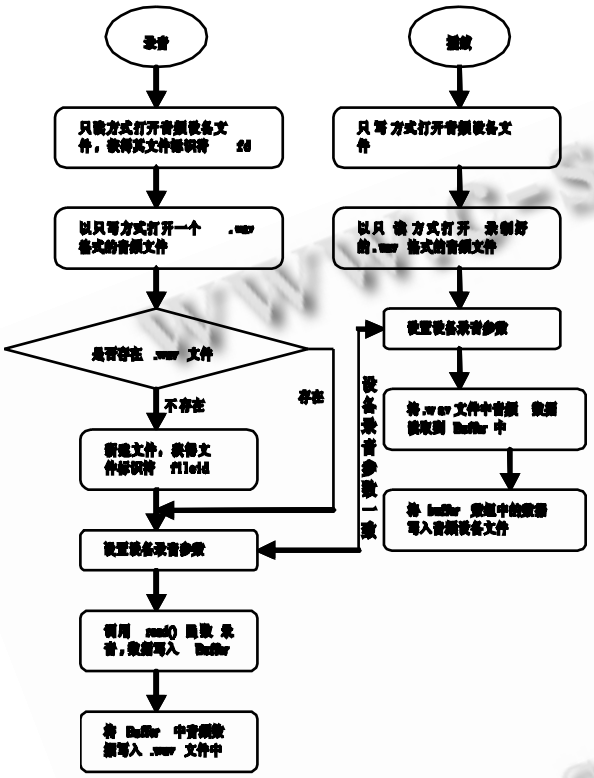

图 3 音频系统应用程序流程图

4 结语

本文介绍了在嵌入式 Linux 系统下基于 IIS 总线的音频驱动，实现音频的打开，关闭，播放和录音。具体讲述了 Samsung 公司 S3C2410 微处理器和 Philips 公司 UDA1341 芯片的硬件连接的实现及嵌入式 Linux 系统下的音频驱动程序。该系统已经应用在 GPS 车载导航系统中，可以顺利地进行行车播报及警报提醒。

参考文献

1 张杰,曹巍华,等.基于 S3C2410 的 Linux 移植.微机发展, 2005,6:142－146.

2 S3C2410 RISC Microprocessor Datasheet. Samsung Semiconductor, Inc, 2002.

3 徐睿,李斐,王申康.基于 IIS 总线的嵌入式音频系统设计.计算机应用, 2004,4:7－9.

4 李俊编.嵌入式 Linux 设备驱动开发详解.北京:人民邮电出版社, 2006:243－248.

5 于明,范书瑞,曾祥烨.ARM9 嵌入式系统设计与开发教程.北京:电子工业出版社, 2006:258－279.

6 杨华,陈明义,等.基于嵌入式语音通信系统的研究.衡阳师范学院学报,2005,12:29－31.

7 秦贵和,徐云鹏,洪宇,李宝玲.基于 ARMLinux 的嵌入式音频系统设计.计算机工程与设计, 2007,6:2611－2613.