

DM6446 的音频编解码及播放实现

王磊¹ 冯立杰²

(1、武警工程学院研究生管理大队 2、武警工程学院通信工程系,陕西 西安 710086)

摘要: DM6446 核心处理器集成了 ARM 内核与 C64x+ DSP 内核,提高了系统集成度和整个系统的运算速度,更加适合实时状态下的音频信号编解码处理并实现语音播放。本文主要介绍了音频处理应用程序的构成,着重讲述了音频编解码及播放过程。

关键词: 达芬奇技术;DM6446;音频编解码

1 基于 DaVinci 技术的音频编解码综述:

达芬奇(DaVinci)技术是一种专门针对数字音视频应用、基于信号处理的解决方案,能为音视频设备制造商提供集成处理器、软件、工具等支持,以简化设计进程,加速产品创新。

由于大量的音视频编解码工作需要一个强大的 DSP 处理器作为支撑,所以在本文提出的解决方案中采用了 TMS320C64x+ DSP 处理器,它是 TMS320C6000 系列 DSP 处理器中性能最为优异的定点 DSP 核。因为其基于第二代先进超长指令字(Very-Long-Instruction-Word,VLIW)结构,能够很好满足现场音频编解码处理和播放的需求。

2 OSS(开放源代码软件)驱动简介

OSS 为多种 Unix 系统(或 Unix 兼容的操作系统)提供声卡和其他声音设备的驱动。使用 OSS 应用程序接口(API)的应用程序在移植到不同的 Unix 系统(或与 Unix 兼容的系统)之前只需要重新编译,而不必重写代码。OSS 支持的主要设备文件中有些是和声卡拥有的设备如数字声音设备(Digitized Voice Device)、混合设备(mixer)、合成器(synthesizer)等相对应。

2.1 /dev/mixer

混合器设备文件和声卡中的混合器相对应,OSS 驱动支持在一个系统中存在多个混合器,不同的混合器被命名为/dev/mixer0、/dev/mixer1 等,而设备文件/dev/mixer 是某个设备文件的符号链接(通常是第一个混合器,/dev/mixer0)。

2.2 /dev/sndstat

该设备文件不同于其他声音设备,它产生可被读懂的诊断信息。此设备打印出所有被 OSS 检测到的端口和设备信息。执行命令“cat /dev/sndstat”将会显示设备配置的有用信息。

2.3 /dev/dsp 和 /dev/audio

这些设备是数字音频应用的主要设备。任何写入这个设备文件的数据都会通过声卡中的 DAC/PCM DSP 设备播放出来。而从此设备文件中读出的数据就是从当前输入设备(默认输入是麦克风)录下的音频数据。

/dev/dsp 和 /dev/audio 设备非常相似。不同在于,/dev/audio 在默认情况下使用对数 μ 律编码,而 /dev/dsp 使用 8 位无符号线性编码。 μ 律编码会把采集的 12 位或 16 位数据转换成 8 位数据。在经过数据转换后,/dev/dsp 和 /dev/audio 对数据的处理是相同的。与设备 /dev/mixer 相似,OSS 同样支持在一个系统中存在多个 dsp 和 audio 设备。

3 OSS 应用编程设计

下面主要讲解程序中关于 OSS 驱动控制编程方法。

打开设备文件时使用标志位 O_WRONLY 来指示该设备以只写的方式打开。除了标志位 O_WRONLY 外,设备还支持以 O_RDONLY(只

读)和 O_RDWR(可读可写)方式打开。如下图所示为打开音频设备代码:

```
#define SOUND_DEVICE "/dev/dsp"
soundFd = (SOUND_DEVICE, O_WRONLY);
```

设定的参数有数据格式、声道格式和采样率。

3.1 采样率配置:

采样率是音频播放的重要参数。OSS 的 API 允许设置采样率为 1Hz 至 2GHz 中的任何一个频率,但是实际应用中只有有限的几个采样率被用到。下图是配置音频采样率的代码示意:

```
int sampleRate 8000;
ioctl(soundFd, SNDCTL_DSP_SPEED, &sampleRate);
```

SNDCTL_DSP_SPEED 用于指示 ioctl 函数对设备文件/dev/dsp 的采样率进行配置。

3.2 声道格式设置:

程序中将声道格式设置为双声道立体声。OSS 中默认模式为单声道,而最多支持 16 个声道。声道格式配置代码示意如下图所示:

```
int channels = 2;
ioctl(soundFd, SNDCTL_DSP_CHANNELS, &channels);
```

SNDCTL_DSP_CHANNELS 指示函数 ioctl 对设备文件/dev/dsp 的声道数进行配置。

3.3 采集与播放设置:

采集与播放我们通过 read 与 write 函数完成。函数中第一个参数表示设备标识,第二个参数表示缓存地址,第三个地址表示地址长度。如下图所示:

```
write(soundFd, decodebuffer, BUF_SIZE);
read(soundFd, encodebuffer, BUF_SIZE);
```

4 语音编解码实现

语音模块需要先通过 OSS 驱动对语音信号进行采集,之后放到共享缓存中交给 DSP 做语音压缩,收到的数据解压后在通过 OSS 驱动播放出来。

4.1 语音编码

语音编码模块收到了采集进来语音数据后,送到编码缓存中对其进行编码,再在送到传输模块将编码后的数据发送出去,处理流程如下图所示:①语音模块首先设置两个全局函数——编码缓存和解码缓存,使其可以被传输模块

块直接使用,作为语音模块与传输模块之间的数据交流通道;②使用 Codec Engine 的 Engine_open()来创建音频编码算法引擎,返回一个句柄,所有使用相同 Engine 的模块线程都需要单独的句柄,来确定线程的安全;③使用 speechEncodeAlgCreate()创建编码算法,使用 SPHENC_create()里的静态参数来创建“g.711enc”语音编码器;④使用 Memory_contigAlloc()函数为编码缓存与原始音频数据缓存分配一段连续的内存空间;⑤使用 SPHENC_process()函数调用 G.711 算法对数据进行编码,而后送到传输模块,完成语音编码任务。

4.2 语音解码



语音模块收到传输模块传来的编码数据后,先将其解码而后播放出来。解码流程如下图所示:①语音模块首先设置两个全局函数——编码缓存和解码缓存,使其可以被传输模块直接使用,作为语音模块与传输模块之间的数据交流通道;②使用 Codec Engine 的 Engine_open()来创建音频编码算法引擎,返回一个句柄,所有使用相同 Engine 的模块线程都需要单独的句柄,来确定线程的安全;③使用 speechDecodeAlgCreate()创建解码算法,这包括:a.使用 SPHDEC_create()里的静态参数来创建“g.711dec”语音解码器;b.使用 SPHDEC_control()和 XDM_GETSTATUS 来设置动态的语音解码参数,查询编解码缓冲区大小;④使用 Memory_contigAlloc()函数为解码缓存分配一段连续的内存空间;⑤使用 SPHDEC_process()函数调用 G.711 算法对数据进行解码,完成解码任务,而后播放。

小结. DM6446 平台应用达芬奇双核技术,能够快速地完成对实时传输要求很高的音频数据处理任务,本设计可以较有效的解决语音抖动和播放延迟的问题,实现音频编解码和播放,为在该平台上开发网络语音通话服务奠定基础,也为在该平台上进行其他多媒体应用提供了参考。

参考文献

- [1] 彭启琮. 达芬奇技术——数字图像/视频信号处理新平台[M]. 北京: 电子工业出版社, 2008.
- [2] 白桦. 数字单兵头盔音视频处理与网络传输相关技术研究[J]. 2008.