

基于 SIP 协议的嵌入式语音网关设计与实现

Design and Implementation of Embedded Voice Gateway Based on SIP

肖勇军 李海标 杨文 杜明 (桂林电子科技大学 杭电工程学院 广西 桂林 541004)

摘要: 基于嵌入式 ARM CPU S3C2440 设计了硬件平台,进行了音频驱动的开发。采用优秀的开源 SIP 协议栈—Osip2,eXosip 协议栈,并完成了 SIP 协议栈与语音编解码库在 Linux 操作系统上的移植,实现实时通话任务。测试表明,该设计能够按照 SIP 协议规范进行会话管理,通话过程中语音质量良好。

关键词: SIP VoIP ARM 嵌入式系统

互联网的快速发展,IP 网络已由当初的数据应用向多媒体转变,基于 IP 网络的语音通信技术成为研究的热点。与传统的电话网络相比较,使用网络来进行语音的传输,有更多的优点:比传统电话网络使用更为有效;可以在传输语音信息的同时,传输其他信息,便于更多服务的集成;各种设备费用比传统电话网络的电路交换设备要低得多。近年来,随着系统软件、IP 网络技术、语音压缩编码算法的发展,使 VoIP(Voice over Internet Protocol)技术获得了空前的应用,国际国内的各大主流运营商都已部署了软交换,这为嵌入式语音网关的发展提供了难得的机遇。

SIP^[1](Session Initiation Protocol, 会话初始化协议)是由 IETF(Internet 工程任务组)提出的一种用于 VoIP 的应用层控制信令协议。因其具有简单、灵活的技术特点,SIP 协议正在逐步取代原有 H.323 协议成为 VoIP 的标准协议。SIP 的功能扩展性以及网络伸缩性好,为开发各种增值业务和会议呼叫提供了很大的方便。因此,SIP 协议近年来得到了极大的关注与发展,具备了较大的研究意义。

1 语音网关的硬件实现及其处理流程

1.1 系统硬件电路设计

采用三星公司的 32 位高性能嵌入式微处理器 S3C2440 为核心部件构建语音网关的硬件平台。系统组成框图如图 1。S3C2440^[2]微处理器是一款由 Samsung 半导体公司推出的以手持设备为主而设计

的芯片,其特点是低功耗,高速的处理计算能力,内置 32bit ARM920T 内核,标称工作频率为 400MHz,最高可达 533MHz,适合各种控制应用。

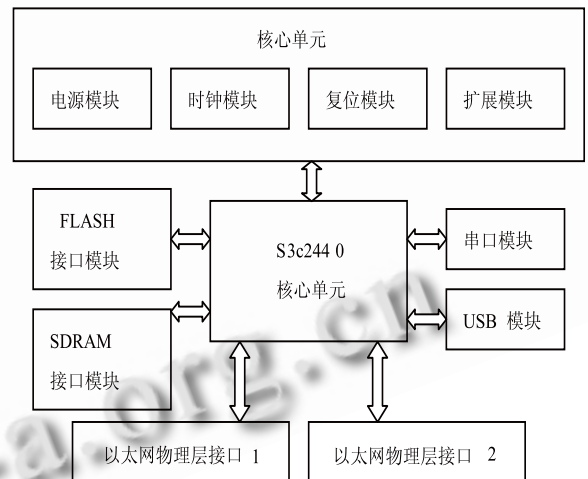


图 1 嵌入式语音网关系统框图

语音信号处理流程:模拟音频信号送入 SLIC 芯片,进行信号放大、去干扰的前置处理,然后送入 CODEC 芯片进行 A/D 转换,并被转换为 PCM 编码的音频信号,再进行语音的压缩编码,编码格式可以选用 G.729、G.726 等,即转换成 RTP 包格式的 Payload(有效数据净荷)。信号送入 CPU,通过 CPU 上运行的协议栈对 Payload 进行封装、打包,最后通过以太网交换芯片传输到目的网络。语音网关对从以太网收到的语音包则采用相反的处理流程。

收稿时间:2008-12-10

1.2 语音电路设计

语音到电信号的转换采用市面上常见的驻极体话筒,性能可以达到要求且价格低廉。由于话筒输出的信号十分微弱,只有十几毫伏左右,要经过放大后才能送到 A/D 转换器进行采样。因此设计了语音信号放大电路,实现了将微弱的语音信号放大到 A/D 转换器能够采样的电压范围。语音输出电路采用 LM386 集成功率放大器,对 D/A 转换器输出的信号进行功率放大,推动扬声器发出声音。语音电路原理图如图 2 所示。

一般的运算放大器电路需要使用双电源供电,这就增加了电源电路的复杂性。因此选用可以使用单电源供电的运算放大器 LM324,并设计实现了单电源供电语音放大电路。为了降低输出阻抗,在语音信号进入 A/D 转换器之前和 D/A 转换器输出信号进入功放之前分别加一级射随电路。

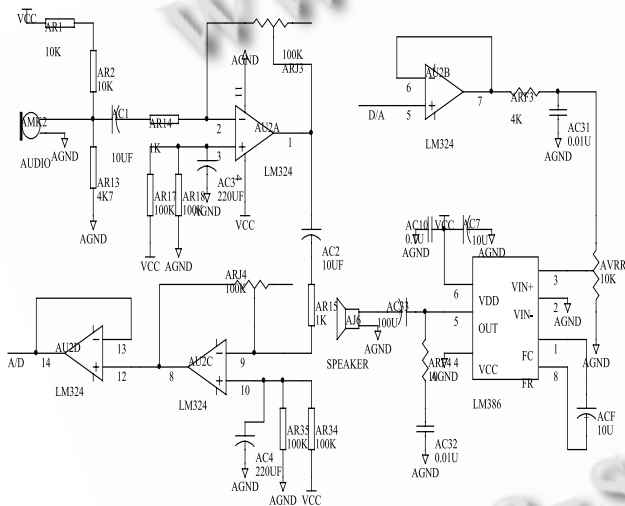


图 2 语音电路

2 语音网关模块软件设计

2.1 音频接口驱动程序实现

UDA1341^[3]是 Philips 公司的一款经济型音频 CODEC,片内集成了立体声 ADC、DAC 以及模拟输入通道,其中模拟输入通道包括可编程增益放大器(PGA)、数字自动增益控制(AGC)、数字声音处理(DSP)等。用于实现模拟音频信号的采集和数字音频信号的模拟输出,并通过 IIS 数字音频接口,实现音频信号的数字化处理。为了能录音和播放 WAV 文件,首先要对硬件 UDA1341 进行初始化,其实现函数如下^[4]:

```
rGPBDAT=rGPBDAT&~(L3M|L3C|L3D)|(L3M|
```

L3C);

```
rGPBUP= rGPBUP & ~(0x7<<2) |(0x7<<2);
rGPBCON=rGPBCON&~(0x3f<<4)|(0x15<<4);
```

```
_WrL3Addr(0x14+2); //STATUS(000101xx+10)
```

```
_WrL3Data(0x60,0); //Status 0,Reset,256fs,IIS-bus,
```

```
_WrL3Addr(0x14+2); //STATUS(000101xx+10)
```

```
_WrL3Data(0x20,0); //Status 0,Noreset, 256fs,IIS-bus
```

```
_WrL3Addr(0x14+2); //STATUS(000101xx+10)
```

```
_WrL3Data(0x81,0); // 1,0,0,0, 0,0,01
```

接着对 IIS 控制器进行初始化,其关键代码分别如下:

```
rIISPSR = (11<<5) + 11; //Prescaler A,B=11
<- FCLK 135.4752MHz(1:2:4) , fs = 44100
```

```
rIISCON = (1<<5) + (1<<2) + (1<<1);
```

```
rIISMOD = (0<<8) + (2<<6) + (0<<5) + (0<<4) + (1<<3) + (0<<2) + (1<<0);
```

```
rIISFCON = (1<<15) + (1<<13); //Tx DMA,Tx FIFO --> start piling
```

初始化完成后,就可以启动 IIS,通过 DMA 方式播放语音数据了。

2.2 SIP 协议栈的选取与移植

目前的开源 SIP 协议栈主要有 OPAL,VOCAL,sipX,ReSIPProcate,Osip2 我们选择了 Osip2 和 eXosip 作为方案的 SIP 协议栈来完成应用程序的开发,采用 oRTP 协议栈来实现实时传输协议 RTP。Osip2,eXosip 和 oRTP 协议栈都是用标准 C 实现的协议栈,移植的主要工作就是对其进行交叉编译,生成能够在 ARM 环境里运行的二进制文件。

以下为编译步骤:

```
(1) 建立安装树,并设定路径 mkdir/arm_sip
export ARM_INSTALL_TREE=/arm_sip
```

```
(2) 编译 osip2./configure--prefix=/usr--host=arm-linux--with-gnu-ld--disable-static
Make
```

```
make install DESTDIR=$ARM_INSTALL_TREE
```

```
(3) 编译 eXosip./configure --prefix=/usr
--host=arm-linux--with-gnu-ld--disable-stat
ic
--disable-glib--with-osip=$ARM_INSTALL_TR
EE/usr
Make
make install DESTDIR=$ARM_INSTALL_TREE
```

```
(4) 编译 oRTP./configure--prefix=/usr--
host=arm-linux--with-gnu-ld--disable-static
Make
make install DESTDIR=$ARM_INSTALL_TREE
```

最后将安装树目录内生成的库文件放到目标板根文件系统的/usr目录下。

2.3 语音编解码库的选择与移植

SIP 协议中并未规定所使用的编解码算法,这里我们选用语音压缩编解码的开源软件包 HawkVoice。它支持 Linux 和 Windows 两个操作系统平台,具有很好的可移植性。它内含许多免费的语音编解码器,例如 G.711 u-law、Intel/DVI ADPCM、GSM、LPC 和 CELP 等。本文使用 HawkVoice 软件包里面的 Intel/DVI ADPCM 语音编解码器。它与标准的 ADPCM 编解码不同的地方是编码器输入和解码器输出直接就是 16 位的均匀 PCM 信号,跳过了 A-law 或 u-law PCM 与 16 位均匀 PCM 之间的转换。软件包提供了简单易用的编程接口,编解码由以下两个函数实现: adpcm_coder(short *indata, unsigned char *outdata, int len, struct adpcm_state *state)和 adpcm_decoder(unsigned char *indata, short *outdata, int len, struct adpcm_state *state)。编码函数 adpcm_coder 的参数 indata 是要编码的音频数据存放的缓冲区的指针, outdata 是编码以后的数据存放的缓冲区的指针, len 是 indata 指针指向的缓冲区中数据的个数。state 是一个 adpcm_state 类型的结构体的指针, adpcm_state 结构体保存预测值和预测系数的信息,以备编码和解码函数使用。解码函数 adpcm_decoder 的参数 indata 是要进行解码的数据存放的缓冲区的指针, outdata 是解码以后的音频数据存放的缓冲区的指针,其余的三个参数与编码函数的对应参数具有相同的含义。设置好缓冲区、填好相应的参数,就可以调用编解码函数实现音频编解码。

3 实验测试

3.1 SIP 协议测试

```
5: 27: 34.62679 IP 202.193.74.217.sip } ray.sip:SIP.length:836
0x0000: 4200 0560 0000 4002 4011 2901 cac8 3c58 E...@.)...<X
0x0010: cac8 3ca3 13c4 13c4 034c f948 494c 5649 ..<.....L.MIWI
0x0020: 5445 2073 6970 3a32 3032 2e32 3030 2e30 TE.sip:202.193.74.
0x0030: 302e 1023 520f 380f 6521 5e23 300d 0e24 217.SIP/2.0.Via:.
0x0040: 3642 3c21 5124 51e6 621e 32e1 100e 1620 SIP/2.0/UDP.20
0x0050: 3140
```

图 3 SIP 协议测试结果部分截图

测试表明本语音网关 SIP 信令流程符合 RFC3261 规范流程,双方通话时的 RTP 流正常,RTP 包中封装的语音编码格式也正常。通话过程中语音质量良好。

3.2 音频编解码的闭环测试

为验证音频编解码器的功能,做了如下测试^[5]:嵌入式系统采集单一频率正弦信号(600Hz)进行实时编解码并将解码以后的音频信号回放,利用 PC 机声卡对该信号进行采集,然后在 PC 机上对信号进行分析并和编解码前的信号对比就可以验证编解码器的功能。音频测试系统框图如图 4 所示。

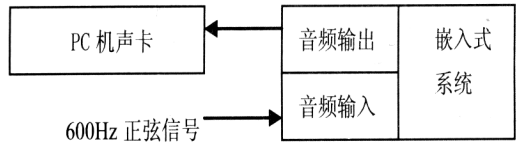
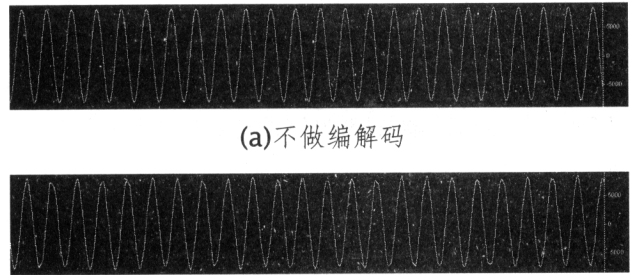


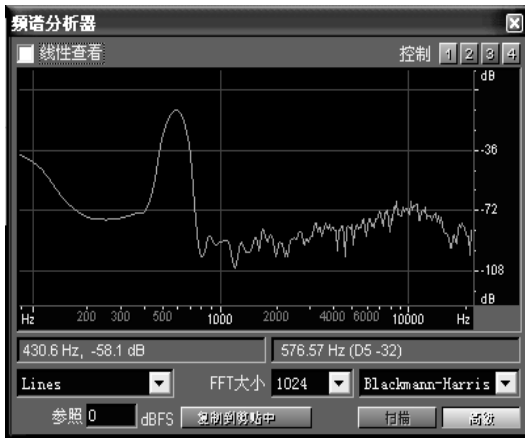
图 4 音频测试系统框图

图 5 是 PC 机通过音频处理软件利用声卡采集到的嵌入式系统输出的信号波形,其中(a)是嵌入式系统采集到信号以后直接回放的波形,(b)是嵌入式系统采集到信号然后做 ADPCM 编码、ADPCM 解码以后回放的波形。图 6 是图 5 中对应信号的频谱。

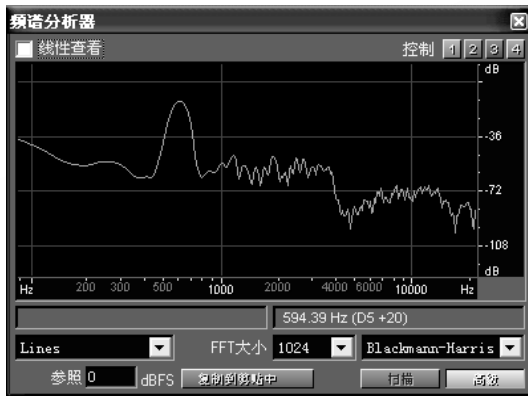


(b)做 ADPCM 编解码

图 5 嵌入式系统输出信号波形



(a)不做编解码



(b)做 ADPCM 编解码

图 6 嵌入式系统输出信号的频谱

由图 5 和图 6 可以看出,解码器能较好的恢复原始信号。

4 结束语

本系统采用 SIP 这种新的协议,设计了一个基于 ARM9 的嵌入式语音网关的方案,系统具有简单、灵活等优点,在功能性和增长潜力方面具有很强的优势。测试结果表明系统能够按照 SIP 协议规范进行通话,并能够在以太网上进行实时语音传输。通话过程中语音质量良好。

参考文献

- 1 SAMSUNG Electronics.S3C2410X 32-bit RISC micro-processor user's manual revision 1.2. 2003.
- 2 TMS320C6711 Datasheet.(2002-08). www.ti.com.
- 3 李钰含,秦飞,赵保军.嵌入式多媒体监测节点的网络适应性设计.计算机工程,2007,21:223 - 225.
- 4 于明,范书瑞,曾祥焯.ARM9 嵌入式系统设计与开发教程.北京:电子工业出版社,2008:260 - 266.
- 5 倪坤.嵌入式网络可视对讲系统的研究与设计[硕士学位论文].桂林:桂林电子科技大学,2007:40 - 44.
- 6 Rosenberg J, Schulzrinne H, Camarillo G, Johnston A, Peterson J, Sparks R, andley MH, Schooler. SIP: Session-Initiation Protocol, RFC3261, June2002.