

## 嵌入式语音系统中多路通信接口的实现

付雯潇, 杜旭, 徐晶

(华中科技大学电子与信息工程系, 武汉 430074)

**摘要:** 随着语音业务越来越趋于多样化, 传统的 DSP 技术只能简单地对语音进行处理, 不能完全胜任业务的多样性。该文介绍了一种利用通信处理芯片 MPC860 和 DSP 芯片 TMS320VC5509 实现多路实时语音通信的方案, 并主要讨论了其关键模块 QMC 驱动。该方案已经得到了应用, 能够较好地满足多通道、多路共听的实时语音业务需求。

**关键词:** 多路语音服务; MPC860; QMC; TMS320VC5509

## Implementation of Multi-channel Communication Interface in Embedded Voice Systems

FU Wenxiao, DU Xu, XU Jing

(Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074)

**【Abstract】** As the voice services become more and more diversified, the traditional DSP techs cannot satisfy the complex service requirements because they can only deal with the voice data simply. This paper presents a solution based on the communication processor MPC860 and the DSP chip TMS320VC5509 to realize a multi-channel real time voice communication design, and mainly analyzes the QMC driver as the key module. This solution has now been used in practice, and can satisfy the requirements of multi-channel communication and multiple voices mix well.

**【Key words】** Multiple voices service; MPC860; QMC; TMS320VC5509

将 DSP 芯片应用于语音处理已经是非常普遍的形式, 而随着语音业务越来越趋于多样化, 传统的 DSP 技术只能简单地对语音进行处理, 不能完全胜任业务的多样性。在这种趋势下, 由于嵌入式操作系统具有占用空间小、执行效率高、方便进行个性化定制和软件要求固化存储等特点, 成为了复杂语音处理的优先选择。在我们设计的综合通信系统中, 需要在应用中支持多板卡, 多通道的语音业务, 并且要求多路语音共听, 这就需要实现数据帧的时分复用, 并要保证实时性和稳定性。

本文针对这个问题, 给出了一个利用摩托罗拉公司的嵌入式 CPU MPC860 的解决方案, 主要讨论了关键模块 QMC 驱动, 并分析了数据收发流程中的若干问题。

## 1 系统框架

在本设计中, 一个完整的通信系统由多块板卡组成, 其中每块板卡的核心模块是 DSP 芯片和嵌入式 CPU, 各板卡间通过 VOIP 服务程序, 经由以太网接口进行语音数据的交互。

为了满足语音处理的实时性要求, DSP 芯片选用 TI 公司的 TMS320VC5509, 它基于 TMS320C55x 系列 CPU 处理内核, 增强了 DSP 的运算能力, 而且性能更好, 运算速度可以达到 144Mips<sup>[2]</sup>。为了满足语音业务的丰富扩展, 特别是为了适应网络传输, 处理器选择 Motorola 公司的 MPC860, 这是一款将嵌入式 PowerPC 核心和高性能, 低功耗外围组件整合在一起的适用于高端通信的产品, 集成了网络接口, 支持多种总线规范扩展外部设备, 提供丰富的串行接口以配合外设数据的高速传输<sup>[3]</sup>。

系统架构如图 1 所示。每块板卡在 A/D, D/A 端有多路语音以话筒或者线路方式输入, 及输出到耳机、电台等设备,

耳机要求支持共听。A/D 送来的数据通过 DSP 的多通道串口, 经由 DMA 发送给 DSP 内部进行编码处理。

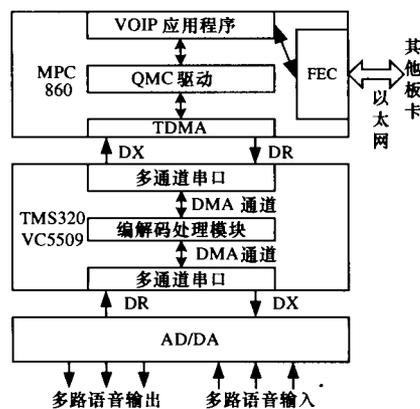


图 1 系统架构

由于语音信号的短时特性, 取每帧 16ms 的数据, 即  $16\,000 \times 0.016 = 256$  个样点。为了避免 DMA 读取数据过程中连续码流溢出, 设计缓冲区为双 Buffer 模式, 以乒乓方式工作, 并保证数据的处理时间小于一帧数据的采集时间 16ms。DMA 将这些数据以乒乓工作方式由多通道串口发送给 MPC860 的时分复用接口。对于解码流程, DMA 接收多通道串口的数据, DSP 将这些数据进行 CVSD 解码后, 由 DMA 经过另一多通道串口发送给 D/A。为了保证 DSP 与 MPC860 间数据的高速传输, 并能够灵活地控制通道个数, DSP 的

**作者简介:** 付雯潇(1982-), 男, 硕士生, 主研方向: 嵌入式系统, 网络通信技术; 杜旭, 副教授; 徐晶, 博士

**收稿日期:** 2006-01-26 E-mail: fuwenxiao213@163.com

通道串口 McBSP0 与 MPC860 的接口采用时分复用的串行方式。该接口受控于 MPC860 内部的通信处理模块(CPM), 它含有一个 32b 的 RISC, 专门负责串行通信控制器(SCC)和串行管理控制器(SMC)的工作, 这些串行控制器可以以多种不同的协议传送数据。

各板卡上的语音数据通过 MPC860 提供的 FEC(快速以太网接口), 经由以太网和其他板卡进行通信。

## 2 QMC 驱动设计

由于本系统中各板卡需要满足不同的通道数和共听要求, 上层接口又是相对固定的 VOIP 程序, 如何在 DSP 和 VOIP 间高效兼容地传输数据成为了一个关键问题。一般的 Linux 内核并不提供这样的接口程序, 因此每块板卡必须根据需求, 设计一个基于时分复用的 QMC 驱动, 其原理如图 2。

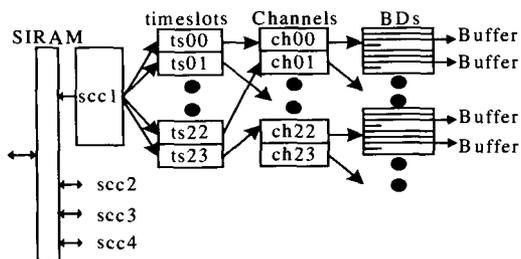


图 2 QMC 模式的工作原理

多通道控制器 (QUICC Multi Channel Controller, QMCC) 是为了高效收发 T1 帧而设计的, 它可以把 T1 帧的数据分发到多达 64 个通道的数据缓冲区里面。该模式下把每一帧的数据划分成若干个时隙(Timeslots), 每时隙 8 b。在时隙分配表里规定每时隙数据从属于某个通道, 每个通道都有一组特定的缓冲区描述符和相应的缓冲区。一个时隙的数据在收发时被放置在相应的缓冲区里面, MPC860 就可把分散的数据聚集在一起发送, 或把聚集在一起的数据分发到各自相应的缓冲区里面, 而不需要额外的处理来区分各种各样的数据流<sup>[4]</sup>。

QMC 驱动提供了在 MPC860 与 DSP 之间交换语音数据包与控制数据的能力, 主要完成多通道语音与控制数据的时分复用以及语音数据包的组包与解包功能。

### 2.1 驱动-DSP 帧格式定义

MPC860 以时分复用方式收发 DSP 中 McBSP0 的串行数据, 时钟信号和帧同步信号由 DSP 产生。以两路普通语音通道为例, 设置 2 个 channel, 每个 channel 的字长是 16b, 分别对应于 1 个语音数据通道, 也就是 1 个外部的实际语音接口(譬如一套耳机、话筒)。其收发帧格式如图 3 所示。



图 3 两路普通语音帧的格式

在有共听应用需求时, 为了指示相应的共听通道是否含有语音数据, 引入了语音指示控制通道。由于共听需求只存在于 MPC860 的发送方向中, 因此对其收发有不同的帧格式。

#### (1)共听方式下 MPC860 的发送

此处以 4 路共听需求为例, 即系统各板卡中 4 部话筒的声音都需要在本板卡的一路耳机中被听到。针对需求, 定义数据帧为 6 个 channel, 并且为了满足 DSP 处理的一致性, 每个 channel 的字长都是 16b, 见图 4。

第 1、第 2 个 channel 是语音指示控制字, 定义如下: 第 1 个 channel(共听指示)的 bit 15~bit 8: 耳机通道第 2 路是否

含有语音数据, 0x00 表示有, 0xff 表示没有; bit 7~bit0: 耳机通道第 1 路是否含有语音数据, 0x00 表示有, 0xff 表示没有; 第 2 个 channel(共听指示)指示第 3、第 4 路是否含有语音数据, 定义同上。第 3~6 个 channel 对应于耳机通道的共听输入。DSP 接收到帧后, 会跟据共听指示对后面的语音数据进行相应的检测, 解码处理。

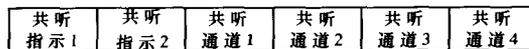


图 4 4 路共听帧发送方向的格式

#### (2)共听方式下 MPC860 的接收

DSP 处理之后的数据经 McBSP0 的 DX 引脚串行发送给 MPC860T 的 SCC1。时钟信号和帧同步信号由 DSP 产生。为了保持帧格式一致和收发时钟的统一, MPC860T 的接收方向也设置 6 个 channel, 每个 channel 的字长是 16b, 其中第 1 个 channel(也可设置为其他 channel)是耳机通道的语音数据, 其余 5 个 channel 保留, 可以通过配置时隙分配表不予接收。



图 5 4 路共听帧接收方向的格式

同样地, 如果板卡上语音通道更多, 或者多路通道同时要求共听, 也可以按照上面的形式定义多 channel 的复合帧格式, 加上相应的控制字来满足需求。因为 AD/DA 芯片每路的采样频率固定, 所以通道数增多时每帧变长, 而 DSP 的帧同步信号并不改变, 只是实际数据传送速率增加, 从而保证了各路语音的同步和实时性。

### 2.2 驱动-VOIP 接口定义

QMC 驱动根据不同的外部设备需求注册不同数量的通道字符设备, 驱动主设备号定为一个空闲数值, 本系统采用 253; 次设备号即为通道号, 例如两路通道就注册字符设备 /dev/vchan1 和 /dev/vchan2。每一个通道字符设备提供一套标准的文件操作函数。而驱动与 VOIP 程序交互的语音包是如下定义的:

```
发送包: typedef struct transmitpacket{
char SID;
char DID;
char Code;
char Mode;
unsigned int size;
char outpacket[MAX_PAYLOAD_SIZE];
}DSP_transmitpacket_T;
```

以上结构中, 包括了数据包的来源 SID, 目的地 DID, 在驱动中指示收发的具体通道; Code 和 Mode 是 DSP 的解码方法和工作模式的指示, 该两个字段暂时保留, 需要时可以控制改变 DSP 的模式; size 和 outpacket 是包的个数和指向数据包的指针数组。VOIP 会将数据封装成这样的架构, 通过 write 函数传递给驱动程序。

```
接收包: typedef struct receivepacket{
unsigned int size;
char inpacket [MAX_PAYLOAD_SIZE];
}DSP_receivepacket_T;
```

在接收数据结构中, size 为驱动记载该数据包语音数据大小变量, 并不传送给 VOIP, 传送给 VOIP 的为 inpacket 数组里面的纯语音数据。送给 VOIP 的数据包大小, VOIP 由 read()系统调用的返回值获得, 这要求 dspc\_read 函数返回正

确的语音数据包大小。

这样的收发接口可以和 VOIP 程序正常通信, 并且有灵活性, 便于适应具体板卡和扩展需求。

### 2.3 驱动子模块设计

#### (1) 通道初始化

通道初始化是在打开设备时完成的, 驱动判断如果是第 1 次启用该语音通道, 而且是第 1 次启用 QMC 模式, 将进行 QMC 模式的初始化配置, 包括设置工作模式、分配缓冲区和缓冲区描述符、分配时隙、分配引脚等。否则将只简单地进行模块使用计数。本系统中选择了 SCC1 作为 TDM 接口, 引脚配置如下。

```
PC4 RSYNC
PA7 RXC
PA8 RXD
PC5 TSYNC
PA5 TXC
PA9 TXD
```

#### (2) 通道中断处理过程

数据收发的核心是中断处理函数。QMC 中的所有逻辑通道共同分享 SCC1 的中断, SCC1 维持有一个全局的中断表, 表中每一项里面都含有逻辑通道号和中断源, 可以被驱动识别以处理不同情况下出现的事件。中断发生时, 驱动从 SCCE 寄存器里面判断全局中断源, 如果中断是由逻辑通道所引起, 中断服务程序便查找全局中断表以找到该逻辑通道, 并从中断条目里面检查中断信息以确定是什么类型的中断, 再调用相应的收发和错误处理函数。

#### (3) 通道数据接收过程

通道的接收过程是由收到一个 buf 的数据的中断事件激发的, 当一个接收缓冲描述符所对应的缓冲区接收到规定数据后, CPM 会产生中断要求驱动程序处理。如果有共听应用, 会根据接收到的数据的控制字节决定其所控制的两字节数据是否有效, 有效则复制到环形接收缓冲中去, 否则将丢弃掉该两字节的数据; 如果不支持共听, 则认为数据都有效。当数据处理完毕后, 将设置接收环形缓冲的状态位, 把数据封装成有效的 VOIP 数据包。然后唤醒睡眠在 PollRead 队列中

的进程, 通知 VOIP 进程进行读取。对于共听应用的需求, 当有语音传输时, 上层应用程序还需要根据语音有效指示控制字来判断某路共听通道是否存在有效数据。

#### (4) 通道数据发送过程

每一个语音通道都维持有一个传输环形缓冲 buffer(tx-ringbuf)用来缓冲额外的上层发送过来的 VOIP 数据包。如果要满足共听应用的需求, 上层应用程序需要语音有效指示字来指示哪一路共听通道有有效数据。我们对数据包每隔两个字节添加一个字节的语音指示字段, 组装成 DSP 所能识别的数据流格式, 保存在相应通道的传输缓冲区里面; 如没有共听需求, 则将数据直接保存在传输缓冲区中。通信处理模块(CPM)会自动完成缓冲区里数据的发送工作。发送完毕后产生中断, 中断处理函数区分中断源, 并且调用 2 次发送函数, 检查是否含有没有发送的 VOIP 数据包, 有的话进行 2 次传输, 同时还将唤醒睡眠在 PollWait 队列的睡眠等待进程。

### 3 结束语

本设计已经应用于我们的系统之中。在多通道语音服务的测试中语音质量稳定, 多路共听声音清晰, 没有串扰和杂音, 没有较大延迟。说明硬件设计正确完备, 驱动程序和 DSP 处理程序性能优良。

#### 参考文献

- 1 Gibson L, Howard D, Tyrrell A. A Parallel Processing System for Polyphonic Singing Synthesis[C]. Proc. of EUROMICRO Conference, 1999-09, 2: 70-74.
- 2 Texas Instrument Company. TMS320VC5509 Digital Signal Processor Data Manual[Z]. 2003-07.
- 3 Motorola Inc. MPC860T PowerQUICC™ Technical Summary[Z]. 2001-12.
- 4 Motorola Inc. QMC Supplement to MC68360 and MPC860 User's Manuals[Z]. 1997.
- 5 毛德操, 胡希明. Linux 内核源代码情景分析(下)[M]. 杭州: 浙江大学出版社, 2001.

(上接第 257 页)

### 5 结束语

煤矿事故的应急救援是一个大的系统工程。在“煤矿事故救援体系”的指导下, 提出了正交软件体系结构模块化的设计思想, 详细分析了 SMS 组件设计的技术路线, 并给出了以事故监控和事故救援管理为主要功能的煤矿救援指挥通信系统的设计与实现。在国内煤炭行业, 首次将组件式 GIS 和 SMS 技术结合起来, 为煤矿救援管理提供了一个实用、直观、便于操作的可视化界面, 从而为煤矿安全管理提供了新的模式, 有效提高了煤矿救援部门的工作效率和科学性, 尤其是在突发事件中能够大大提高煤矿救援决策的科学性和准确性, 有着广泛的应用性。

#### 参考文献

- 1 李学来, 胡敬东. 煤矿应急救援技术的研究及应用现状[J]. 煤炭工程, 2005, 52(4): 62-64.
- 2 孙长嵩, 毕力格, 王文婧等. 一个基于组件式 GIS 技术的矿井安全与救援信息系统[J]. 计算机应用, 2004, 23(1): 52-55.
- 3 王晓娟, 黄忠全, 张根保. 短消息系统设计与实现[J]. 重庆大学学报, 2004, 27(5): 96-98.
- 4 刘光, 刘小东. 地理信息系统二次开发实例教程——C#和 MapObjects 实现[M]. 北京: 清华大学出版社, 2004: 24-303.
- 5 张友生. 软件体系结构[M]. 北京: 清华大学出版社, 2004: 72-76.
- 6 刘桥喜, 毛善君, 马嵩乃等. 煤矿安全地理信息系统的设计与实现[J]. 测绘通报, 2004, (2): 60-62.