

文章编号: 1671- 251X(2011)01- 0044- 06

基于 SPI 协议的音频流解码系统的研究与设计

徐博， 郭秋敏

(煤炭科学研究院常州自动化研究院, 江苏 常州 213015)

摘要:介绍了 SPI 通信协议, 提出了一种采用 ARM 控制器 LPC2368 与 VS1003 语音处理芯片的音频流解码系统的设计方案, 详细介绍了系统架构、SCI 与 SDI 传输原理以及系统软件关键代码的设计。通过初始化调试与正弦波测试, 该系统能够较好地实现音频模拟信号的播放功能。

关键词:音频流解码; 数据传输; SPI 协议; LPC2368; VS1003; SCI 传输; SDI 传输

中图分类号: TD655 **文献标识码:**B

Research of Audio Codec System Based on SPI Protocol and Its Design

XU Bo, GUO Qiu-min

(Changzhou Automation Research Institute of CCRI., Changzhou 213015, China)

Abstract: The paper introduced SPI communication protocol, proposed a design scheme of audio-stream decoding system by use of ARM-controller LPC2368 and voice-processing chip VS1003, and described structure, transmission theories of SCI and SDI of the system and design of key codes of the system's software in details. The system can realize playing function of audio-analog signals perfectly by initialization debugging and sine-wave testing.

Key words: audio-stream decoding, data transmission, SPI protocol, LPC2368, VS1003, SCI transmission, SDI transmission

0 引言

所谓音频流解码系统, 就是以高性能和高运算速率的单片中央处理器为核心, 将打包过的 MP3/WMA/MIDI 等音频编码文件按照固定频率的时序对二进制代码进行传输和解码处理, 将数字信号源源不断地还原成模拟音频信号并通过模拟运放电路播出的集成电路系统。在该过程中, 二进制音频文件的处理和二进制代码的数据传输是 2 个最关键的环节。一般来说, 用于二进制音频文件代码处理的单片中央处理器是 DSP 芯片, 芬兰 VLSI 半导体公司生产的 VS1003 就是一款性能优良的 DSP 芯片。VS1003 内部集成有多频率立体声音频 DAC、模拟输出放大器和滤波器, 可以有效地实现数据采样、倍频、高低音加重、滤波除噪等多种解码功能, 同时具

备优越的音频编码功能。采用 VS1003 芯片后, 实现音频流解码系统需要重点解决的就是二进制代码的数据传输问题。采用 SPI (Serial Peripheral Interface, 串行外围设备接口) 协议进行全双工数据传输可以灵活地控制主、从机之间的通信速率, 有效地监视音频流的传输过程, 便于整个音频流解码系统的调试与实现。

1 SPI 协议

SPI 协议是一种同步串行全双工通信协议, 它可以使 MCU 与各种外围设备以串行方式进行通信以交换信息。SPI 协议一般只使用 4 条线: 串行时钟线 SCK、主机输入/从机输出数据线 MISO、主机输出/从机输入数据线 MOSI 和低电平有效的从机选择线 SS^[1]。

SPI 协议最早是 Motorola 公司在其 MC68H CXX 系列处理器上定义的。SPI 接口主要应用在 EEPROM、FLASH、实时时钟、ADC、数字信号处理器和数字信号解码器之间。目前, SPI 协

收稿日期: 2010- 09- 14

作者简介: 徐博(1982-), 男, 山东博山人, 助理工程师, 硕士, 现主要从事煤矿自动化产品可靠性设计方面的研究工作。E-mail: hungryb0424@163.com

议因具有高速、全双工、同步通信以及简单易用的特性而被越来越多的芯片所集成, 如 NXP 公司生产的 LPC2368、VLSI 公司生产的 VS1003 等。

SPI 设备以主从方式通信, 通常有 1 个主设备和 1 个或多个从设备, 至少需要 4 根线, 它们是 SDI(主设备数据输出, 从设备数据输入)、SDO(主设备数据输入, 从设备数据输出)、SCK(时钟信号, 由主设备产生)、CS(从设备使能信号, 由主设备控制), 单向传输时也可以只有 3 根线。其中 CS 是控制芯片是否被选中的, 也就是说只有片选信号为预先规定的使能信号时, 对该芯片的操作才有效。这就使得在同一总线上连接多个 SPI 设备成为可能^[1]。

SPI 设备由 SCK 提供时钟脉冲, SDI、SDO 则基于该脉冲完成数据传输。数据输出通过 SDO 线, 数据在时钟上升沿或下降沿时改变, 在紧接着的下降沿或上升沿被读取, 完成 1 位数据传输, 输入也使用同样原理。这样, 在至少 8 次时钟信号的改变过程中就可以完成 1 B 数据的传输。与普通的串行通信不同, 普通串行通信一次至少连续传送 8 bit 数据, 而 SPI 协议允许数据一位一位地传送, 甚至允许暂停, 这是因为 SCK 时钟线由主设备控制, 当没有时钟跳变时, 从设备不采集或传送数据。也就是说, 主设备通过对 SCK 线的控制可以完成对通信的控制。SPI 协议的缺点是其整个通信机制还只是建立在数据链路层上, 没有指定的流控制, 也没有应答机制确认是否接收到数据。

2 音频流解码系统架构

由于 SPI 协议是主从机通信模式, 所以整个音频流解码系统的架构可以分为主机通信控制模块和从机数据应答模块, 如图 1 所示。主机通信控制模块的主要功能是通过控制通信时钟来控制主从机之间的通信; 从机数据应答模块的主要功能一方面是同步主机的数据传输, 另一方面是及时应答特定的数据请求信号或指令信号, 以起到监视通信的作用。系统以 LPC2368 作为主机通信控制模块, 该芯片是基于 ARM 的可编程微控制器, 适用于为了各种目的而进行串行通信的应用场合, 其特有的 ARM 7TDMI-S 处理器性能优越, 可在高达 72 MHz 的工作频率下运行; 以 VS1003 作为从机数据应答模块^[1]。

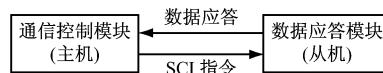


图 1 音频流解码系统架构

在音频流解码系统中, 要以 VS1003 为从机数据应答模块得到顺畅的模拟音频流, 必须依据 VS1003 关于 SPI 协议下其自身特定的 SPI 工作原理对 LPC2368 进行编程控制才能实现。VS1003 特定的 SPI 协议可分为两大类: SCI(Serial Command Interface, 串行命令接口) 传输和 SDI(Serial Data Interface, 串行数据接口) 传输^[3]。

SCI 传输是指 VS1003 为实现基于 SPI 协议的数据传输而率先进行的命令传输, 如图 1 所示。该命令传输也是基于 SPI 协议, 但具体协议内容因命令的不同而不同。LPC2368 必须完全依据 SCI 命令的协议格式去编写代码才有可能实现初步的 SPI 通信^[4]。

由于 SPI 协议本身是全双工的同步通信方式, 因此, 对于 SCI 传输过程, VS1003 定义了读、写 2 套命令操作机制。写命令操作相对于读命令操作更为重要, 主要用于告知从机关于数据通信的具体类别, 而读命令操作只在某些调试中用于监视通信是否正常。

数据应答是衡量 SPI 数据是否被从机正确接收和从机是否准备接收下一批 SPI 数据的重要标识信号。在硬件上该控制位是独立的, 是 VS1003 独有的数据请求返回管脚。毕竟无论是 SCI 传输还是 SDI 传输, VS1003 在整个系统的 SPI 传输中都是完全被动地接收数据的, 而 VS1003 本身还承担着实时处理音频数据的责任, 如果 VS1003 的处理时序与主机的传输时序发生了冲突, 那么 VS1003 被动接收的数据将变得无效, 进而使整个系统数据传输失效; 如果 VS1003 在被动接收数据的前提下能够根据自身的数据处理时序发出数据请求标识, 表明在某一段时间内数据传输可以进行, 在另一段时间内数据传输需要稍微暂停一下, 那么整个 SPI 数据传输就会与音频信号处理时序协调起来, 实现流畅的音频数据传输和处理。因此, 数据应答是一个非常重要的通信标识逻辑, 完全由 VS1003 自动产生。对于主机, 则必须时刻依据数据应答调整 SPI 传输时序。

VS1003 的 SCI 传输协议通常包含一个 8 bit 的指令字节、一个 8 bit 的地址字节和一个 16 bit 的数据字节。具体操作通常由 8 bit 的指令字节来确定。每次 SCI 传输后数据请求返回管脚(DREQ)都会被设置为低, VS1003 是不允许在数据请求返回管脚变为高之前开始新的 SCI 或 SDI 传输的。

SDI 传输是指 VS1003 基于 SPI 协议的数据传

输, 如图 2 所示。SDI 传输的完全是音频流数据, VS1003 在接收 SDI 数据的同时完成片内的音频解码数据处理任务, 以 32 B 为单位进行 SDI 数据的传输和处理, 每传输完 32 B 的二进制音频数据, VS1003 就在数据请求返回管脚上设置一个低电平。该管脚在再次变为高电平之前, 主机不能连续向从机发送 SDI 数据, 否则数据将被自动舍弃无效^[4]。



图 2 主从机 SDI 传输方式

3 SCI 传输与 SDI 传输的实现

图 3 为 LPC2368 与 VS1003 的硬件连接电路, 其中 XCS 管脚传输的就是 SPI 协议中的 CS 信号, DREQ 管脚上传输的就是 VS1003 给 LPC2368 的返回信号^[4]。

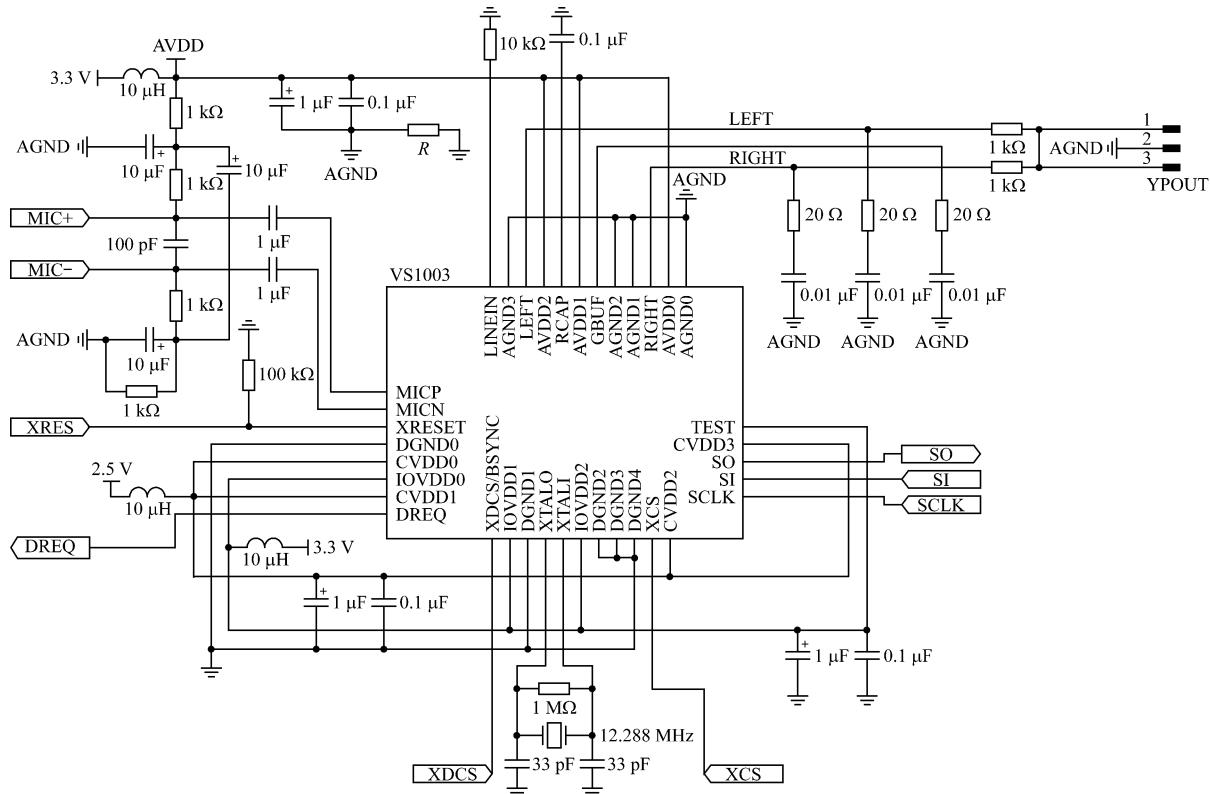


图 3 LPC2368 与 VS1003 的硬件连接电路

SCI 读数据过程如图 4 所示。XCS 线就是该系统的 CS 信号传输线。它本身只是一个片选和使能的功能。VS1003 的内部地址里都有固定的参数作为内存数据，当对这些内部地址进行读操作时就会把里边的参数读出来。因此，首先 LPC2368 的 XCS 线上信号需要拉低以选择芯片，这是 SPI 协议的一个关键。然后，读指令代码 0x03 将会通过 LPC2368 的 SI 管脚被传输至 VS1003。读指令代码之后紧跟一个 8 bit 的地址字节。在该地址被读入之后，任何在 SI 管脚上传输的数据都将被忽略。刚才所传输地址空间中包含的 16 bit 固定参数将会紧跟着在 SO 管脚上返回给 LPC2368，用于监视通信。数据传输完成后，XCS 线上的信号需要被拉高。最后，VS1003 的 DREQ 管脚上的信号会因为

VS1003 的处理而有短暂的拉低, 很快会自动变为高电平以迎接下一个 SCI 指令。

SCI 写数据过程如图 5 所示。它与 SCI 读数据类似，也是先将 XCS 线拉低以选择芯片。然后，写指令代码 0x02 经由 SI 管脚被传输至 VS1003。写指令代码之后也是一个 8 bit 的地址字节，只不过该地址是用来存放将要传输来的 SPI 数据的。整个过程中 SO 管脚是闲置的。

在整个数据传输过程中,SCI 传输与 SDI 传输之间是紧密跟进的,因此,在编写代码时必须非常注意真正的时序,尤其是注意 DREQ 信号的高低状态和 XCS 线的高低选择。对于图 6 所示的 2 个相邻的 SCI 命令传输过程,XCS 信号必须在 2 个 SCI 传输中被拉高,当检测到 DREQ 线上信号变高后再拉

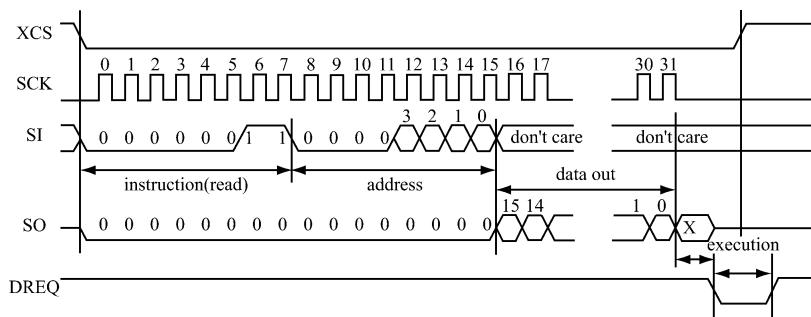


图 4 SCI 读数据过程

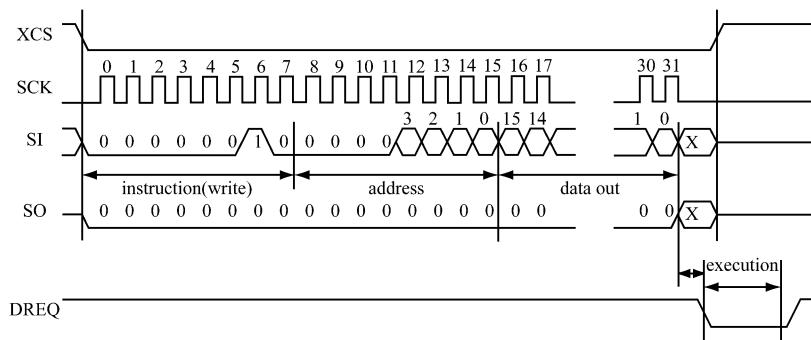


图 5 SCI 写数据过程

低以进行 SCI 传输。而对于图 7 所示的 2 个相邻的 SDI 字节传输过程,由于传输的只是 SDI 数据,完全没有必要采取 DREQ 的中断,因此, XCS 信号可持续走高以放心通信,直到 32 B 的 SDI 数据传完,再看 DREQ 管脚上的信号以判断是否继续通信。2 个 SDI 传输之间的 SCI 传输过程(见图 8)表明,只要有 SCI 传输,在传输完成之后就一定存在一个 DREQ 低电平返回,DREQ 与 SCI 传输是匹配的。

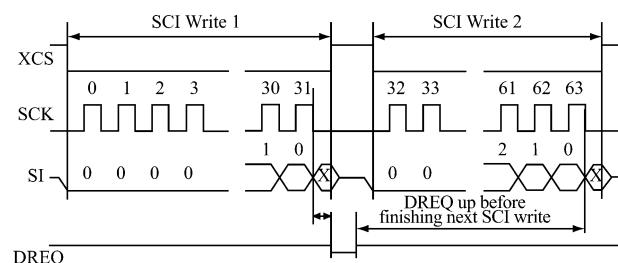


图 6 2 个相邻的 SCI 命令(4 B) 传输过程

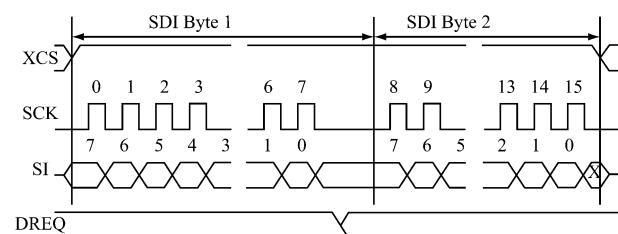


图 7 2 个相邻的 SDI 字节传输过程

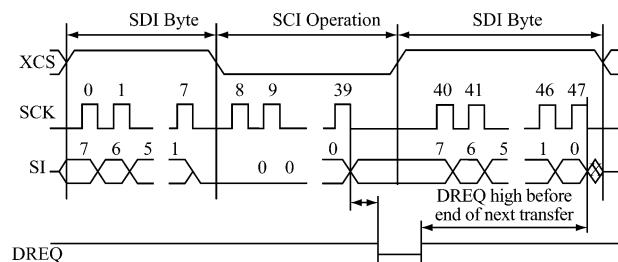


图 8 2 个 SDI 传输之间的 SCI 传输过程

4 系统软件的设计与实现

基于 SPI 协议的音频流解码系统软件的实现完全依赖于主机的硬件特性。LPC2368 具备 2 种基于 SPI 协议的硬件接口: SPI 接口和 SSP 接口。这两种接口都有各自的一套 SPI 控制寄存器。

SPI 接口的 SPI 控制寄存器包含许多可编程位来实现 SPI 协议传输。SSP 接口的 SPI 控制寄存器也是如此。该寄存器必须在数据传输之前设定。SPI 控制器是 SPI 控制寄存器中的一种,它具有一个使能位,该位控制激活 SPI 的 I/O 口和使能 SPI 内部状态机^[7]。SPI 状态寄存器包含只读位,用于监视 SPI 接口的状态,包括一般性功能和异常状况^[6]。该寄存器的主要用途是检测数据传输是否完成,通过 SPIF 位来实现,其它位用于指示异常状况。SPI 数据寄存器用于提供发送和接收的数据字节。串行数据实际的发送和接收通过 SPI 模块逻辑

中的内部移位寄存器来实现。在发送时向 SPI 数据寄存器写入数据。SPI 数据寄存器和内部移位寄存器之间没有缓冲区, 写数据寄存器会使数据直接进入内部移位寄存器。因此, 数据只能在没有执行数据发送时写入该寄存器。读数据带有缓冲区, 当传输结束时, 接收到的数据转移到一个单字节的数据缓冲区, 在下次传输时将其读出。读 SPI 数据寄存器将返回读数据缓冲区的值。SPI 时钟计数器寄存器用于控制时钟速率, 该寄存器必须在数据传输之前设定。

SSP 是一个同步串行端口控制器, 可在 SPI、4 线SSI 或 Microwire 总线上操作。它与总线上多个主机和从机相互作用。在给定的数据传输过程中, 总线上只能有 1 个主机和 1 个从机进行通信。数据传输原则上是全双工的, 4~16 bit 帧的数据由主机发送到从机或由从机发送到主机。但实际上, 大多数情况下只有一个方向上的数据流包含有意义的数据。

应用 LPC2368 自有的 SPI 协议寄存器来实现 SPI 数据传输简单、方便, 但调试起来往往困难较多。原因主要是数据传输的时序完全由 LPC2368 自身控制, 用户代码仅仅起到控制通信和监视中断的功能, 不能完全实现对传输数据的监视。因此, 为了更稳定地实现 SPI 的数据传输, 在软件上考虑使用纯代码模拟 SPI 通信的方式实现。选定 SO、SI、SCLK、CS 等管脚后, 首先针对 SCLK 管脚人为地模拟产生固定频率的时钟脉冲信号, 然后选定在脉冲的上升沿或下降沿触发 SI 管脚上相应的电平, 以字节为单位按照脉冲依次输出; 对于 SO 管脚则依次接收。整个过程需要注意的是时钟脉冲信号的输出频率和 VS1003 的 DREQ 信号返回, 频率过高和过低或者 DREQ 信号捕捉不到都会造成数据传输失败。模拟 SPI 协议核心代码如下:

```
unsigned int Self_SendData(unsigned char data)
{
    unsigned int i, temp, rev_data= 0;
    temp = data;
    for(i = 0; i <= 7; i++)
    {
        IO0CLR |= SCLK; // 开始给时钟位写 0
        if(temp & 0x80)
            {IO0SET |= SI;} // 最左边位为 1 时, SI 线上给一个高电平
        else
            {IO0CLR |= SI;} // 最左边位为 0 时, SI 线上给一个低电平
        rev_data <<= 1;
        if(IO0PIN & SO)
```

```
{rev_data |= 0x01;}
else
{rev_data&= 0xFE;}
delay(1);
// for(j= 0;j == 10;j++)
IO0SET |= SCLK; // 再给时钟位写 1
delay(1);
// for(j= 10;j == 0;j--)
temp <<= 1; // 下一位
}
return(rev_data); // 读来的数据
}
```

该段代码就是整个系统通信的最核心代码, 依此可写出针对 SCI 写数据传输、SCI 读数据传输的代码段, 本文不再赘述。如果采用 LPC2368 的经典 SPI 和 SSP 寄存器, 代码理所应当要简单, 但是因为失去了对整个通信的底层监视, 常常通信不成功, 也难以查错。该段代码的优点在于完全针对通信过程的最底层进行透明的编写, 在调试过程中可以结合编译器进行一位一位的监视, 以方便查错, 控制通信。

5 系统的初始化调试与正弦波测试

基于 SPI 协议的通信实现之后, 整体系统架构中的核心芯片 VS1003 应该首先初始化。VS1003 的所有 DVDD、AVDD 管脚以及 xReset、TEST 应该接 +3.0 V 电压, 然后测量 RCAP 管脚, 其输出应在 1.3 V 左右, 否则芯片模拟部分无法正常工作。

对于初始化调试, 向 VS1003 的音量控制寄存器 SCI_VOL 循环写入最高值和最低值, 正常情况下能从耳机听到“滴滴”的声音, 具体步骤如下:

- (1) 拉低 xCS;
- (2) 设置音量最高: SCI_VOL = 0x0000;
- (3) 拉高 xCS;
- (4) 等待 500 ms;
- (5) 拉低 xCS;
- (6) 设置音量最低: SCI_VOL = 0xFFFF;
- (7) 等待 500 ms;
- (8) 拉高 xCS;
- (9) 循环, 否则以上步骤无法识别。

还可以按以下方法进行初始化调试:

- (1) 拉低 xCS;
- (2) 写音量控制寄存器: SCI_VOL = 0xA2F5;
- (3) 适当延时, 等待 DREQ 为高;
- (4) 读音量控制寄存器, 看读回的值是否与写入的一致, 如果不为 0xA2F5, 则说明 SCI 读写操作

有问题。

正弦波测试步骤如下:

- (1) 进入 VS1003 的测试模式: SPI_MODE = 0x0820;
- (2) 等待 DREQ 为高;
- (3) xDCS 拉低(xCS 置为 1), 选择 VS1003 的数据接口;
- (4) 向 VS1003 发送正弦测试命令: 0x53 0xEF 0x6E 0x30 0x00 0x00 0x00 0x00;

- (5) 延时 500 ms;
- (6) 退出正弦波测试, 发送命令: 0x45 0x78 0x69 0x74 0x00 0x00 0x00 0x00;
- (7) 延时 500 ms;
- (8) 循环。

正弦波测试通过后, 可以在 VS1003 的模拟信号输出端接耳机听到固定频率的尖锐的正弦测试音, 按照代码规律手动编写二进制文件, 还可以听到变化频率的正弦音。使用 WinHEX 等软件将 MP3 音乐的二进制代码拷出作为源文件的一部分传输给 VS1003, 可以初步实现音频流解码系统对模拟信号的播放功能。

6 结语

SPI 协议下的音频流解码系统由于采用了先进

的 VS1003 芯片, 因此, 整个系统的解码模块本身并不是主要研究方向, 系统实现的关键是怎样将二进制的 MP3/WMA/MIDI 等编码依据 VS1003 特定的 SPI 传输原理进行通信。结合 LPC2368 的可编程特点, 人为地模拟 SPI 底层通信协议, 能够实现对 SCI 传输和 SDI 传输的监视, 结合正弦波测试的结果分析, 较好地实现了 SPI 协议下音频流解码功能。

参考文献:

- [1] 胡邦楠. SPI 总线一主多从方式的应用 [J]. 宜春学院学报: 自然科学版, 2006(6): 64-66.
- [2] 周立功. LPC23XX 系列 Flash 单片机应用技术 [M]. 北京: 北京航空航天大学出版社, 2004.
- [3] 李桐宇, 杨家伟. 用 SPI 总线实现 DSP 和 MCU 之间的高速通信 [J]. 电子元器件应用, 2006(11): 28-30.
- [4] 孙冰洁, 周洪利. SPI 模式下单片机对 MMC 卡的读写控制 [J]. 电脑知识与技术, 2006(5): 147, 155.
- [5] 耿德根. AVR 高速嵌入式单片机原理与应用 [M]. 北京: 北京航空航天大学出版社, 2001.
- [6] 刘永平, 王威, 江豪, 等. 基于 SPI 总线的 PIC 与 ISD4003 语音接口电路 [J]. 微计算机信息, 2006(9-2): 95-96.
- [7] 刘永亮, 仇三山. 用 SPI 实现 dsPIC 与 ISD 语音芯片的通信 [J]. 单片机与嵌入式系统应用, 2005(11): 42-44, 97.

各项参数要求尤为严格。该大功率高压晶闸管是由株洲南车时代电气股份有限公司电力电子事业部提供的, 经多次试验后通过了 6 000 A 融冰试验的考验。

(封疆)

中国南车为中国首套 500 kV 固定直流兼静止无功补偿装置 6 000 A 融冰试验提供核心器件

2010 年 12 月 14 日, 由中国南车提供核心器件的中国首套 500 kV 固定直流兼静止无功补偿装置在湖南益阳 500 kV 复兴变电站成功完成 6 000 A 融冰试验, 这是我国国内融冰装置进行的最大电流的融冰试验。

直流融冰兼静止无功补偿装置兼具融冰和无功补偿功能, 在线路无融冰需求时, 通过对装置进行切换、重构, 使装置作为静止无功补偿器运行, 为电网提供动态无功补偿, 支撑系统电压, 改变系统无功潮流, 改善系统电能质量。该装置额定容量为 114 MW, 额定电流为 6 000 A, 作为国内首套、世界第二套直流融冰兼静止无功补偿装置, 其主要技术指标均居国内之最, 对该装置中核心部件换流阀的关键基础器件——12.7 cm 的 3 400 A、6 500 V 大功率高压晶闸管的

天地(常州)自动化股份有限公司有五项科研项目通过天地科技股份有限公司的验收

2010 年, 天地(常州)自动化股份有限公司有 5 项科研项目通过天地科技股份有限公司的验收, 它们分别是“矿井高精度位置监测系统软件技术研究”((TZ-JJ-09-CZ-1) Z051)、“在用煤矿井下监控分站性能检验检测方法的研究”((TZ-JJ-09-CZ-3) Z053)、“基于 WiFi 的漫游切换无缝接入技术研究”((TZ-JJ-09-CZ-4) Z054)、“基于 FPGA 的监控分站的研究”((TZ-JJ-09-CZ-5) Z055) 和“矿用设备的三防比较研究”((TZ-JJ-09-CZ-6) Z056)。

(李娟)