

# 驱动你的 VS1003

——By topsun



**说明:** VS1003 配套文档基于 ICDev 的 VS1003 模块,本教程由 IC 开发网(www.ICDev.com.cn) 2006 年 8 月 21 日发布,仅供参考,更多资料和疑问请到本站论坛下载和发帖,转载请注明出处。

## 一. 准备工作:

### 1. 硬件:

VS1003 模块和 LPC2103 核心模块 (或其他带 SPI 总线接口的微处理器模块皆可);

### 2. 软件:

VLSI 提供的 MP3 播放器的完整实例,对自己编写 VS1003 驱动、播放 mp3 等很有参考价值;

下载地址: <http://www.icdev.com.cn/bbs/board.aspx?boardid=7>

### 3. 文档:

一个是 VS1003 的数据手册,一个是 VS10xx 的应用笔记,有这两个足以搞定 VS1003。

## 二. 关于 VS1003:

在编写 VS1003 的驱动程序前须先对它有一定的了解,这就需要阅读 VS1003 的数据手册了。如果你的 E 文不好,那就硬着头皮看吧。

### 1. VS1003 简介:

VS1003 是由芬兰 VLSI 公司出品的一款单芯片的 MP3/WMA/MIDI 音频解码和 ADPCM 编码芯片,其拥有一个高性能低功耗的 DSP 处理器核 VS\_DSP, 5K 的指令 RAM, 0.5K 的数据 RAM, 串行的控制和数据输入接口, 4 个通用 IO 口, 一个 UART 口; 同时片内带有一个可变采样率的 ADC、一个立体声 DAC 以及音频耳机放大器。

### 2. 支持的音频编解码格式:

- 1) 解码: MP3 (mpeg1 和 mpeg2, 层 3), MP3+V, WMA, WAV, MIDI, SP-MIDI;
- 2) 编码: IMA ADPCM (单声道), 麦克风和线入 (Line input) 两种输入方式。

### 3. 电器特性:

- 1) AVDD (模拟部分): 最大不得超过 3.6V, 推荐值为 2.8V, 最小 2.5V;
  - 2) CVDD (数字部分, 内核): 最大不得超过 2.7V, 推荐值 2.5V, 最小 2.4V;
  - 3) IOVDD (I/O 电压): 最大不得超过 3.6V, 推荐值为 2.8V, 最小值为 CVDD-0.6V
- 注: VS1003 模块 AVDD = IOVDD = 3.3V, CVDD = 2.5V。
- 4) 输入时钟频率: 12M-13MHz, 推荐值为 12.288MHz;
  - 5) VS1003 内部的时钟倍频器: 芯片复位后的值为 1.0 $\times$ , 推荐值为 3.0 $\times$ , 即 3 倍频, 最大值为 4.5 $\times$ , 即可以在 3.0 $\times$  的基础上再增加 1.5 $\times$ , 用于 WMA 的回放。

注: 以上仅列出了 VS1003 的一些基本电器特性, 更具体的内容请参考 VS1003 的数据手册 P9-P11。

#### 4. VS1003 的寄存器（数据手册 P27）:

VS1003 共有 16 个 16 位的寄存器，地址分别为 0x0–0xF；除了模式寄存器（MODE，0x0）和状态寄存器（STATUS，0x1）在复位后的初始值分别为 0x800 和 0x3C 外，其余的寄存器在 VS1003 初始化后的值均为 0。下面将 VS1003 各寄存器逐一介绍：

1) MODE（地址 0x0；RW，可读写）

模式寄存器在 VS1003 中是一个较为重要的寄存器，其每一位都对应着 VS1003 的不同设置。

●bit0: SM\_DIFF

SM\_DIFF = 0 正常音频相位

SM\_DIFF = 1 左声道反转

当 SM\_DIFF 置位时，VS1003 将左声道反相输出，立体声输入将产生环绕效果，对于单声道输入将产生差分（反相）左/右声道信号。

●bit1: SM\_SETTOZERO

置零

●bit2: SM\_RESET

SM\_RESET = 1，VS1003 软复位。软复位之后该位会自动清零。

●bit3: SM\_OUTOFWAV

SM\_OUTOFWAV = 1，停止 WAV 解码。

当你要中途停止 WAV、WMA 或者 MIDI 文件的解码时，置位 SM\_OUTOFWAV，并向 VS1003 持续发送数据（对于 WAV 文件发送 0）直到将 SM\_OUTOFWAV 清零；同时 SCI\_HDAT1 也将被清零。

●bit4: SM\_PDOWN

SM\_PDOWN = 1，软件省电模式，该模式不及硬件省电模式（由 VS1003 的 xReset 激活）

●bit5: SM\_TESTS

SM\_TESTS = 1，进入 SDI 测试模式。

●bit6: SM\_STREAM

SM\_STREAM = 1，使能 VS1003 的流模式，具体请参考应用笔记 VS10XX。

●bit7: SM\_PLUSV

SM\_PLUSV = 1，MP3 + V 解码使能。

●bit8: SM\_DACT

SM\_DACT = 0，SCLK 上升沿有效；SM\_DACT = 1，SCLK 下降沿有效。

●bit9: SM\_SDIORD

SM\_SDIORD = 0，SDI 总线字节数据 MSB 在前，即须先发送 MSB；

SM\_SDIORD = 1，SDI 总线字节数据 LSB 在前，即须先发送 LSB；

该位的设置不会影响 SCI 总线。

**●bit10: SM\_SDISHARE**

SM\_SDISHARE = 1, SDI与SCI将共用一个片选信号（同时SM\_SDINEW = 1），即将xDCS与xCS这两根信号线合为一条，能省去一个IO口。

**●bit11: SM\_SDINEW**

SM\_SDINEW = 1, VS1002本地模式（新模式）。VS1003在启动后默认进入该模式。

注：这里的模式指的是总线模式。

**●bit12: SM\_ADPCM**

SM\_ADPCM = 1, ADPCM录音使能。

同时置位SM\_ADPCM和SM\_RESET将使能VS1003的IMA ADPCM录音功能。

**●bit13: SM\_ADPCM\_HP**

SM\_ADPCM\_HP = 1, 使能ADPCM高通滤波器。

同时置位SM\_ADPCM\_HP、SM\_ADPCM和SM\_RESET将开启ADPCM录音用高通滤波器，对录音时的背景噪音有一定的抑制作用。

**●bit14: SM\_LINE\_IN**

录音输入选择，SM\_LINE\_IN = 1, 选择线入（line in）；SM\_LINE\_IN = 0, 选择麦克风输入（默认）

**2) SCI\_STATUS (0x1, RW)**

SCI\_STATUS为VS1003的状态寄存器，提供VS1003当前状态信息。

**3) SCI\_BASS (0x2, RW)**

重音/高音设置寄存器。

VS1003的内置的重音增强器VSBE是种高质量的重音增强DSP算法，能够最大限度的避免音频削波。当SB\_AMPLITUDE（bit: 7-4）不为零时，重音增强器将使能。可以根据个人需要来设置SB\_AMPLITUDE。例如，SCI\_BASS = 0x00f6，即对60Hz以下的音频信号进行15dB的增强。当ST\_AMPLITUDE（bit: 15-12）不为零时，高音增强将使能。例如，SCI\_BASS = 0x7a00，即10kHz以上的音频信号进行10.5dB的增强。

**4) SCI\_CLOCKF (0x3, RW)**

在VS1003种对该寄存器的操作有别于VS10x1和VS1002。

●SC\_MULT（bit: 15-13）时钟输入XTALI的倍频设置，设置之后将启动VS1003内置的倍频器。

●SC\_ADD（bit: 12-11）

用于在WMA流解码时给倍频器增加的额外的倍频值。

●SC\_FREQ（bit: 10-0）

当XTALI输入的时钟不是12.288M时才需要设置该位段，其默认值为0，即VS1003默认使用的是12.228M的输入时钟。

**5) SCI\_DECODE\_TIME (0x4, RW)**

解码时间寄存器。当进行正确的解码时，读取该寄存器可以获得当前的解码时长（单位为秒）。可以更改该寄存器的值，但是新值须要对该寄存器进行两次写操作。在每次软件复位或是WAV（PCM、IMA ADPCM、WMA、MIDI）解码开始与结束时SCI\_DECODE\_TIME的值将清零。

#### 6) SCI\_AUDATA (0x5, RW)

当进行正确的解码时，该寄存器的值为当前的采样率（bit: 15-1）和所使用的声道（bit: 0）。采样率须为2的倍数；bit0 = 0，单声道数据，bit0 = 1，立体声数据。写该寄存器将直接改变采样率。

#### 7) SCI\_WRAM (0x6, RW)

该寄存器用来加载用户应用程序和数据到VS1003的指令和数据RAM中。起始地址在SCI\_WRAMADDR中进行设置，且必须先于读写SCI\_WRAM。对于16位的数据可以在进行一次SCI\_WRAM的读写中完成；而对于32位的指令字来说则需要两次连续读写。字节顺序是大端模式，即高字节在前，低字节在后。在每一次完成全字读写后，内部指针将自动增加。

#### 8) SCI\_WRAMADDR (0x7, RW)

用于设置RAM读写的首地址。地址范围见数据手册P32。

#### 9) SPI\_HDAT0和SPI\_HDAT1 (0x8, 0x9, R)

这两个寄存器用来存放所解码的音频文件的相关信息，为只读寄存器。

- 当为WAV文件时，SPI\_HDAT0 = 0x7761，SPI\_HDAT1 = 0x7665；
- 当为WMA文件时，SPI\_HDAT0的值为解码速率（字节/秒），要转换为位率的话则将SPI\_HDAT0的值乘8即可，SPI\_HDAT1 = 0x574D；
- 当为MIDI文件时，SPI\_HDAT0的值请参考数据手册P33，SPI\_HDAT1 = 0x4D54；
- 当为MP3文件时，SPI\_HDAT0和SPI\_HDAT1包含较为复杂的信息（来自于解压之后的MP3文件头），包括当前正在解码的MP3文件的采样率、位率等，具体请参考数据手册P33-P34。复位后SPI\_HDAT0和SPI\_HDAT1将清零。

#### 10) SCI\_AIADDR (0xA, RW)

用户应用程序的起始地址，初始化先于SCI\_WRAMADDR和SCI\_WRAM。如果没有使用任何用户应用程序，则该寄存器不应进行初始化，或是将其初始化为零，具体请参考应用笔记VS10XX。

#### 11) SCI\_VOL (0xB, RW)

音量控制寄存器。高八位用于设置左声道，低八位用于设置右声道。设置值为最大音量的衰减倍数，步进值为0.5dB，范围为0到255。最大音量的设置值为0x0000，而静音为0xFFFF。例如，左声道：-2.0dB，右声道：-3.5dB，则SCI\_VOL = (4 × 256) + 7 = 0x0407。硬件复位将使SCI\_VOL清零（最大音量），而软件复位将不改变音量设置值。

注：设置静音（SCI\_VOL = 0xFFFF）将关闭模拟部分的供电。

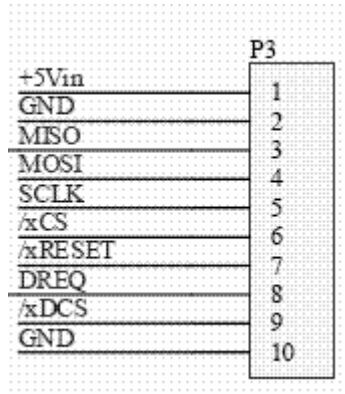
#### 12) SCI\_AICTRL[x] (0xC-0xF, RW)

用于访问用户应用程序。

### 三. 初始化VS1003

#### 1. 首先确保VS1003与微处理器的正确与可靠连接。

这一步很关键，必须确保连线的正确而且是确实连接上的（在插线之前最好用万用表测一下导线的连通性），否则以后的工作将是浪费时间。下图为VS1003模块接头定义：



对于LPC2103模块来说，总共需要其7个IO口。MISO、MOSI与SCLK分别接LPC2103的P0.5、P0.6和P0.4（LPC2103的SPI0），其余的四根线接到LPC2103其他的四个通用IO口即可。对于其他微处理器来说接法类似。

#### 2. 微处理器模块的正确初始化（以LPC2103为例）

##### 1) 正确配置IO口

注意将与DREQ连接的IO口配置为输入，其余配置为输出（xCS、xDCS和xReset）；将P0.4、P0.5和P0.6配置为SPI总线接口。

##### 2) 初始化SPI

所有对VS1003的操作将通过SPI总线来完成。在默认情况下，数据位将在SCLK的上升沿有效（被读入VS1003），因此需要在SCLK的下降沿更新数据；并且字节发送以MSB在先。VS1003的SPI总线的输入时钟最大值为 $CLKI / 6$  MHz，其中 $CLKI$ （内部时钟）=  $XTALI \times$  倍频值。（注： $CLKI / 6$ 为SCI读的时钟最大值，SCI和SDI写的时钟最大值为 $CLKI / 4$ 。数据手册P11）SPI的初始化程序代码如下：

```

/*****/
/* 函数名称 :   MSPI_Init                               */
/* 函数功能 :   初始化SPI接口，设置为主机。           */
/* 参数      :   无                                     */
/* 返回值   :   无                                     */
/*-----*/
void MSPI_Init(void)
{
    PINSEL0 = (PINSEL0 & 0xFFFF00FF) | 0x00005500; //选择 SPI
    S0SPCCR = 0x08;                               // SPI 时钟设置, Fpclk / 8
    S0SPCR  = (0 << 3) |                           // CPHA = 0,
              (0 << 4) |                           // CPOL = 0,
              (1 << 5) |                           // MSTR = 1,
              (0 << 6) |                           // LSBF = 0,
              (0 << 7);                             // SPIE = 0,

```

```
}
```

当前处理器的时钟频率设置为 $11.0592 \times 5 = 55.296\text{M}$ （晶振11.0592，5倍频）； $F_{\text{pclk}} = F_{\text{ccclk}} / 4 = 55.296 / 4 = 13.824\text{M}$ （外设时钟为处理器时钟的四分频），则 $SCLK = 13.824 / 8 = 1.728\text{MHz}$ 。

### 3. VS1003模块的初始化

上述两步完成后就可以通过SPI总线对VS1003进行初始化了。初始化的一般流程如下：

- 硬复位， $x\text{Reset} = 0$ ；
- 延时， $x\text{DCS}$ 、 $x\text{CS}$ 、 $x\text{Reset}$ 置1；
- 等待DREQ为高；
- 软件复位： $\text{SPI\_MODE} = 0x0804$ ；
- 等待DREQ为高（软件复位结束）；
- 设置VS1003的时钟： $\text{SCI\_CLOCKF} = 0x9800$ ，3倍频；
- 设置VS1003的采样率： $\text{SPI\_AUDATA} = 0xBB81$ ，采样率48k，立体声；
- 设置重音： $\text{SPI\_BASS} = 0x0055$ ；
- 设置音量： $\text{SCI\_VOL} = 0x2020$ ；
- 向vs1003发送4个字节无效数据，用以启动SPI发送。

## 四. 全面测试你的VS1003

初始化完成了，可以开始播放MP3了么？答案是否定的（当然你运气特别好除外），开发过程不可能这么一帆风顺。如何知道你的VS1003是否正常工作、是否被你正确初始化？下面我们就来全面测试一下。

### 1. 察看芯片供电是否正常；

2. 模拟部分是否正常：VS1003的所有DVDD、AVDD管脚以及 $x\text{Reset}$ 、TEST（Pin 32）接+3.0V，然后测量RCAP在应在1.3V左右，否则芯片模拟部分未正常工作。

3. 检查SCI命令是否正确写入。给音量控制寄存器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：

- 拉低 $x\text{CS}$ ；
- 设置音量最高： $\text{SCI\_VOL} = 0x0000$ ；
- 拉高 $x\text{CS}$ ；
- 等待500ms，这个时间控制在0.5到1s之间，太大太小可能都无法听到；
- 拉低 $x\text{CS}$ ；
- 设置音量最低： $\text{SCI\_VOL} = 0xFFFF$ ；
- 等待500ms，道理同上；
- 拉高 $x\text{CS}$ ；
- 循环，否则以上步骤无法识别。

### 4. 可以对SCI的读写做进一步的测试：

- 拉低 $x\text{CS}$ ；
- 写音量控制寄存器： $\text{SCI\_VOL} = 0xA2F5$ ；
- 适当延时，等待DREQ为高；
- 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的话说明SCI读

写有问题。

## 5. 正弦测试 (SDI测试)

VS1003的SPI总线用来传送MP3数据和控制命令。当要传送MP3数据时xDCS须置为低电平，而xCS置1。此时SPI总线称作SDI (串行数据接口)。VS1003拥有以下几种测试模式：存储器测试、SCI总线测试和正弦测试。所有的这些测试都有相同的步骤：硬件复位，置位模式寄存器SPI\_MODE的bit5: SM\_TESTS，发送测试命令到SDI总线上。测试命令总共包含8个字节的数据，前4个字节为命令代码，后四个字节为0。正弦测试属于芯片内部的测试功能，如果写SDI总线无误的话可以从耳机里听到单一频率的正弦音（可以通过命令更改频率），强烈建议大家对此项步骤测试时不要将耳塞直接塞入耳中，因为系统不同可能导致声音大小会不同，有可能极其刺耳。正弦测试步骤如下：

- 进入VS1003的测试模式：SPI\_MODE = 0x0820；
- 等待DREQ为高；
- xDCS拉低（xCS置1），选择vs1003的数据接口；
- 向VS1003发送正弦测试命令：0x53 0xEF 0x6E 0x30 0x00 0x00 0x00 0x00；
- 延时500ms；
- 退出正弦测试，发送命令：0x45 0x78 0x69 0x74 0x00 0x00 0x00 0x00；
- 延时500ms；
- 循环。

如果这一步通过了，说明你的VS1003已经做好了为你播放MP3的准备。下面的你的工作就很轻松了，把MP3文件数据有条不紊的发给VS1003，让它为你完成MP3的解码和播放任务。至此，VS1003的驱动任务已经完成。

## 五. 播放MP3

MP3的存储介质可以是U盘、SD卡、MMC卡或是移动硬盘等。不过众多的存储介质对于VS1003来说都是一样的，只要你能将MP3文件按照正确的方法发给VS1003，它就可以为你放出音乐。如何从存储介质中获得MP3文件，然后如何将其发给VS1003都要靠你自己去实现了。现在以U盘为例。较为关键的步骤是：从U盘读出MP3文件到LPC2013的缓冲区，缓冲区的大小设置为512Byte，一般一次读一个扇区，然后将数据发往VS1003。由于VS1003有32Byte的数据缓冲区，一次可以发32个字节的数据，然后检测DREQ，当DREQ为高时送下一个32Byte的数据，直到发完为止。DREQ为高表明VS1003可以接受新的数据，如果不考虑DREQ的话直接连续的给VS1003发的mp3数据的话将会出现音乐断断续续的情况。具体流程如下：

- 1) 打开一个指定的MP3文件：CH375FileOpen();
- 2) 读一个扇区数据：CH375FileRead();
- 3) 发32个字节的数据到VS1003;
- 4) 检测DREQ，当DREQ为高时发下一个32个字节的数据;
- 5) 是否发完512个字节，否，则回到3);
- 6) 是否发完打开的mp3文件，否，则回到2);
- 7) 关闭打开的文件：CH375FileClose( )。

对于SD卡来说过程类似。

播放流程也可以参考VS10xx的应用笔记P16: How to play a song，在这里就不重复了。

## 六. 结语

希望你在音乐出来的那一刻能像当时的我们那样激动和陶醉，这样本文就达到目的了；另外由于E文水平有限，有写得不对的地方请大家多多指正。

