

CX800S 语音芯片使用资料

支持 FLASH



CX800S 采用硬解码的方式,保证系统的稳定性和音质。能够自由更换语音内容和控制方式,使用方便,省去了传统语音芯片需要安装上位机更换语音的麻烦。通过简单的串行指令即可完成指定音乐的播放,多种播放路径。由于成本低,使用方便灵活,储存的声音内容更长,该芯片在市场上得到广泛的应用。



目录

1. 产品特点.....	3
1.1 功能.....	3
1.2 应用.....	3
2. 芯片使用说明.....	5
2.1 极限参数.....	5
2.2 电气特性.....	5
2.3 软件参数.....	6
2.4 芯片管脚说明.....	7
3. 控制方式说明.....	7
3.1 通信时序.....	8
3.2 命令说明.....	8
3.21 关键词说明:	8
3.22 控制命令.....	9
3.23 查询命令.....	10
3.3 程序范例 (c 语言)	11
4. 参考电路.....	12
4.1 串行接口.....	12
4.11 供电 3.3V.....	12
4.12 供电 5V.....	13
4.2 外接单声道功放.....	13
4.3 外接耳机电路.....	14
4.4 主控电路 (向客服索取高清文件)	14
5. CX800S-SOP16 封装图.....	15
6. 版本信息.....	15
7. 免责声明.....	15
7.1 开发预备知识.....	15
7.2 EMI 和 EMC.....	16
7.3 ESD 静电放电保护.....	16
附录:	18
1. 音频文件排序.....	18



1. 产品特点

1.1 功能

- 1、支持不同采样率的语音文件；
- 2、支持 USB 线下更换语音内容；
- 3、最大支持 256MBit 的 Flash，最大支持 32G 的 U 盘；
- 4、支持模拟 SPI 串行控制模式；
- 5、支持插播功能，插播过程暂停正在播放的背景音乐。
- 6、支持指定盘符播放，指定曲目播放。
- 7、支持组合播放。
- 8、31 级音量可调；
- 9、支持播放设备切换；
- 10、24 位 DAC 输出，3.6-5.2V 电压；
- 11、内部采用 DSP 硬件解码，动态范围支持 90dB，信噪比支持 85dB；

1.2 应用

汽车（防盗报警器、倒车雷达、GPS 导航仪、电子狗、中控锁）；

智能家居系统；

家庭防盗报警器；

医疗器械人声提示；

家电（电磁炉、电饭煲、微波炉）；

娱乐设备（游戏机、游乐机）；

学习模型（早教机、儿童有声读物）；

智能交通设备（收费站、停车场）；



通信设备（电话交换机、电话机）；

工业控制领域（电梯、工业设备）；

高级玩具。



2. 芯片使用说明

2.1 极限参数

标识	范围值	单位
VDD~GND 电源电压	-0.5~+5.5	V
Vio 输入输出电压	GND-0.3 < Vin < 3.3+0.3	V
Top 工作温度	-40 ~ +70	°C
Tst 存储温度	-40 ~ +85	°C

2.2 电气特性

标识	参量			最小值	典型值	最大值	单位	条件
VDD	工作电压			2.8	3.3	5.5	V	无 U 盘
					4.2			驱动 U 盘
Isb	供电 电流	待机 电流	VDD=3V		200		uA	(定制内容)
			VDD=4.5V		200		uA	
Iop		工作 电流	VDD=3V		15		mA	无负载
			VDD=4.5V		15		mA	
Iih	输入 电流 内部 上拉	弱上拉 (750Ko hms)	3V		-4		Au	Vil=0V
			4.5V		-11			
		高上拉 (33K ohms)	3V		-90			
			4.5V		-225			
Ioh	输出驱动电流		3V		-8		mA	VDD=3V , Voh=2.0V
			4.5V		-12			VDD=4.5V,Voh=3.5V
Iol	输入倒灌电流		3V		18		mA	VDD=3V , Voh=1.0V
			4.5V		25			VDD=4.5V,Voh=1.0V
Pau	音频输出功率		3V		16		mW	Load=32 ohms



Fosc	Oscillation Frequency	-	1	24	48	MHz	VDD=2.0~5.5V
Iv33	LDO 输出电流		120		150	mA	
SNR	信噪比		90	92	95	dB	
Vdac	音频最大幅值				2.8	V	

2.3 软件参数

名称	功能
音频格式	支持采样率 8~48KHz，比特率 8~128Kbps 的 MP3 音频文件。(FLASH 支持) 支持采样率 8~48KHz，比特率 8~320Kbps 的 MP3 音频文件。(TF 卡和 U 盘支持) 支持采样率 8/11.025/12/16/22.05/24/32/44.1/48KHz 的 WAV 音频文件。
存储器容量	支持 8Mbit~128Mbit SPI-FLASH 最大支持 32GB TF 卡 最大支持 32GB U 盘
USB 接口	全速 2.0
文件系统	FAT12, FAT16, FAT32(不支持 NTFS)



2.4 芯片管脚说明

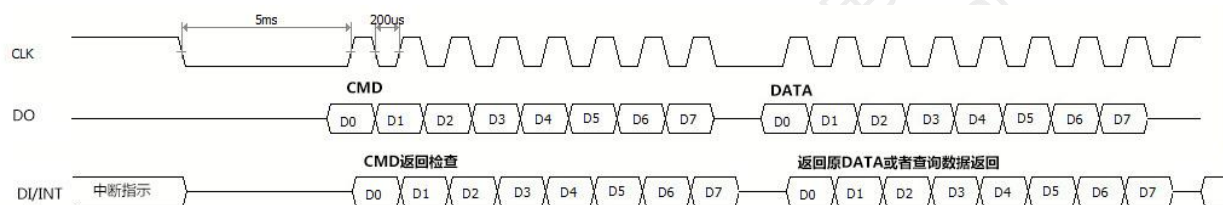
1	SPI-DATA	BUSY	16
2	SPI-CLK	SDI	15
3	SPI-CS	SDO/INT	14
4	CLK	GND	13
5	USB-	VCC	12
6	USB+	3V3OUT	11
7	VCOM	AR	10
8	AGND	AL	9

引脚序号	引脚名称	功能描述	备注
1	SPI_DAT	SPI 数据脚	连接 FLASH
2	SPI_CLK	SPI 时钟脚	
3	SPI_CS	SPI 片选脚	
4	SCLK	通信口时钟	
5	USB-	USB 接口	连接 U 盘或电脑
6	USB+		
7	VCOM	退耦	
8	AGND	音频地	
9	AL	左声道	驱动耳机、功放
10	AR	右声道	
11	3V3OUT	3V3输出	外围电路供电
12	VCC	电源输入	2.8~5.2V
13	GND	地	
14	SDO	通信口从机数据输出	
15	SDI	通信口从机数据输入	
16	BUSY	忙信号输出	播放输出高电平

3. 控制方式说明



3.1 通信时序



通信由片时钟 CLK 和数据输出 DO 脚数据输入 DI 组成，每次通信 16Bit 数据，时钟信号 CLK 拉低 4ms 至 40ms，推荐使用 5ms，时钟宽度介于 200us~1.5ms 之间，推荐使用 200us。

发送数据（DO）低位在先，在时钟的上升沿发送数据。发数据时先发低位，再发高位。前 8Bit 表示命令数据，后 8Bit 表示命令参数或者查询返回值。

接收数据（DI）低位在先，在时钟的下降沿接收数据。发数据时先发低位，再发高位。

DI/INT 为复用引脚，在非通信时间作为 INT 中断指示，当出现中断请求时为低电平，读取 REG0 后，中断自动清零，DI/INT 为高电平。在通信时，作为 DI 引脚，主机端读取数据。

不需要使用读取命令时可不接 DI/INT 引脚，进行单向控制。

注：CLK 拉低时间范围：4ms~40ms，推荐使用 5ms；CLK 周期高电平时间范围：150us~2ms；CLK 周期低电平时间范围：150us~2ms。推荐使用 CLK 高低电平的时间为 200us。通讯 IO 口平时为高，发完数据之后应拉高。

3.2 命令说明

3.2.1 关键词说明：

1) . 文件名称序号

说明：文件名按 3 位数字形式命名，“001-xxxx.mp3”，“002-xxxx.mp3”，“003-xxxx.mp3”
例：文件名为“001-创唯芯欢迎你.mp3”，名称序号为 1。

2) . 文件索引

说明：按下载顺序编号，第一个下载的文件索引为 1。

下载方法：建议文件名按 3 位数字形式命名，“001-xxxx.mp3”，“002-xxxx.mp3”，“003-xxxx.mp3”，全选文件，对着第一个文件右击，复制，粘贴到存储盘符中，那么下载顺序就跟文件名顺序一一对应，详细参照<附录 1. 音频文件排序>。

3) . 插播

说明：打断正在播放的曲目，播放放新曲目，播放完成后返回原来播放

例：当前播放背景音乐 30 秒处，插入播放提示音，提示音播放完后返回背景 30 秒处。

4) . 组合播放

说明：发出组合播放命令不会打断当前播放，按发送的顺序逐个播放。



例：存储器中有音频“零”...“十”，应用中需要报出数字“三十二”，可使用组合播放命令发“三”，“十”，“二”。

5). 无缝衔接

循环播放曲目，一般方式是播放结束后重新打开文件重新播放同一文件，出现大约 10~50ms 停顿，播放环境音时出现卡顿。CX800S 通过虚拟文件技术实现完全连贯循环的背景音播放。

3.22 控制命令

CMD	FUN	para
0x11	MP3 通用功能	1: 播放 2: 暂停 3: 停止 4: 下一曲 5: 上一曲 6: 音量加 7: 音量减 8: 停止插播
0x12	指定名称序号播放	1~255
0x13	指定名称组合播放	
0x14	指定名称序号循环播放 (无缝衔接)	
0x15	指定名称序号插播	
0x16	设置音量	0~30
0x17	设备选择	0: U 盘 1: Res 2: FLASH
0x18	设置循环模式	0: 全部循环 1: 单曲循环 2: 文件夹循环 3: 随机播放 4: 单曲停止 5: 顺序播放
0x19	指定索引播放	1~255
0x1A	指定索引组合播放	
0x1B	指定索引循环播放 (无缝衔接)	
0x1C	指定索引插播	
0x1D	播放停止中断允许	0: 禁止 1: 允许



3.23 查询命令

CMD	FUN	para
0x80	读状态寄存器	BIT(0) 忙 BIT(1) 设备错误 BIT(2) 播放文件错误 BIT(3) 播放结束 BIT(4) 连接电脑 BIT(5) Res BIT(6) U 盘在线 BIT(7) Flash 在线
0x81	读播放状态	0. 播放 1. 暂停 2. 停止 3. 非激活
0x82	读当前索引	1~255
0x83	res	
0x84	读当前音量	0-30
0x85	res	
0x86	读当前设备	0: U 盘 1: Res 0. 2: FLASH
0x87	读当前循环模式	0: 全部循环 1: 单曲循环 2: 文件夹循环 3: 随机播放 4: 单曲停止 5: 顺序播放
0x88	读文件总数	0~255

注：状态寄存器说明（0x80 命令）

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Flash 在线	U 盘在线	Res	连接电脑	播放结束	文件错误	设备错误	忙

说明：当状态寄存器产生变化时将会产生中断，复用引脚 DI/INT 变成低电平，通知主机读取状态寄存器，主机读取状态寄存器后，错误标志位（低 4 位）自动清零，DI/INT 变高电平，在线设备变化时（设备插入或者拔出），也产生中断。

BIT(0)：忙，在不能响应命令时产生此消息，例如插入 pc 时，插播时，发普通播放命令，

BIT(1)：设备错误，没有插入设备或者设备无播放文件



BIT(2): 播放文件错误
BIT(3): 播放结束, 受 CMD_END_MSG_EN (0x1d) 控制
BIT(4): 连接电脑标志位, 0, 未连接 1, 连接
BIT(5): SD 卡在线标志位, 0, 未连接 1, 连接
BIT(6): Res
BIT(7): Flash 在线标志位, 0, 未连接 1, 连接

3.3 程序范例 (c 语言)

```
/******  
//      Port  
//说明: 移植对 IO 操作  
*/  
#ifndef u8  
#define u8    unsigned char  
#endif  
  
#define  CLK_L()      //clk =0  
#define  CLK_H()      //clk =1  
  
#define  D0_L()      //D0 =0  
#define  D0_H()      //D0 =1  
  
#define  READ_DI()    //读 DI  
//延时  
#define  delayms(m)    //延时毫秒: m*1ms  
#define  delayus(u)    //延时微秒: u*1us  
u8 CXError=0;  
u8 CX800S_Send(u8 cmd, u8 data)  
{  
    u8 Rx = 0;  
    u8 temp;  
    u8 i;  
    CLK_L();  
    delayms(5);  
    temp = cmd;  
    for (i = 0; i < 8; i++) {  
        Rx >>= 1;  
        if (temp & 0x01) {  
            D0_H();  
        }  
        else {  
            D0_L();  
        }  
        delayus(200);  
    }  
}
```



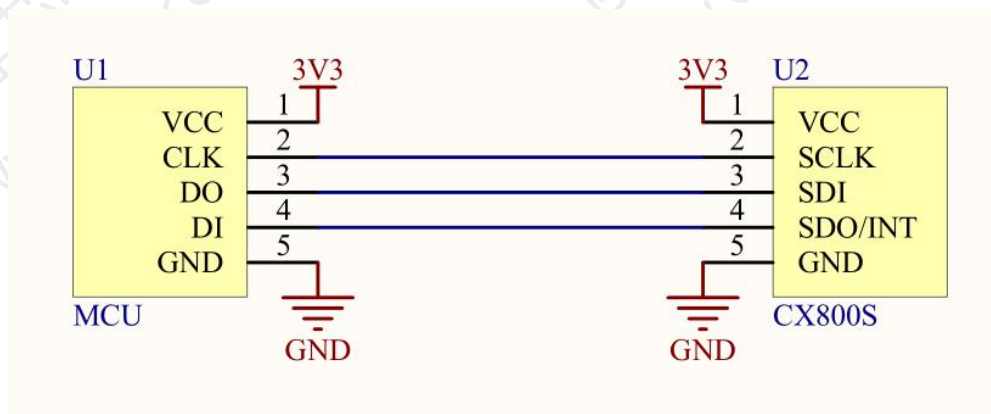
```
    CLK_H();
    delayus(200);
    CLK_L();
    if (READ_DI())
        Rx |= 0x80;
    temp >>= 1;
}
if (Rx != cmd) {
    CXError++;
    return 0xff; //error
}
Rx = 0;
delayus(200);
temp = data;
for (i = 0; i < 8; i++) {
    Rx >>= 1;
    if (temp & 0x01) {
        DO_H();
    }
    else {
        DO_L();
    }
    delayus(200);
    CLK_H();
    delayus(200);
    CLK_L();
    if (READ_DI())
        Rx |= 0x80;
    temp >>= 1;
}
if (((cmd & 0x80) == 0) && (Rx != data)) {
    CXError++;
    return 0xff; //error
}
delayus(200);
CLK_H();
delayus(200);
return Rx;
}
```

4. 参考电路



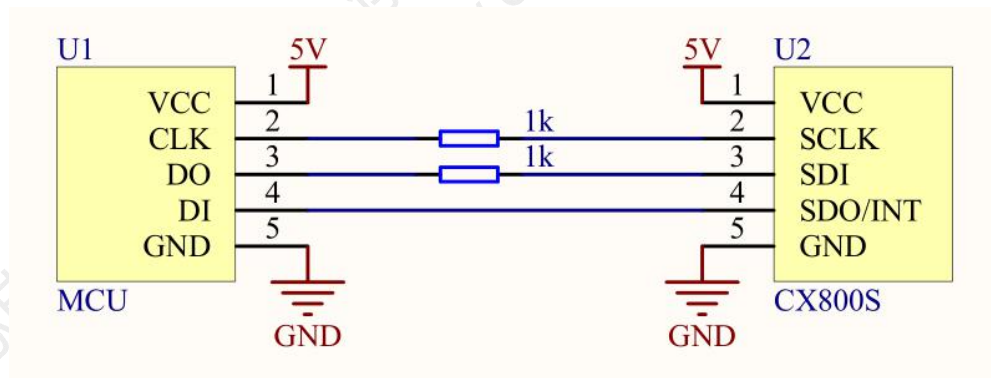
4.1 串行接口

4.1.1 供电 3.3V



CX800S 内置 LDO，内部使用 LDO 输出电压 3V3，IO 电压为 3.3V TTL 电平，当 MCU 和 CX800S 同一供电 3V3 时，IO 电平一致可以直接相连。

4.1.2 供电 5V

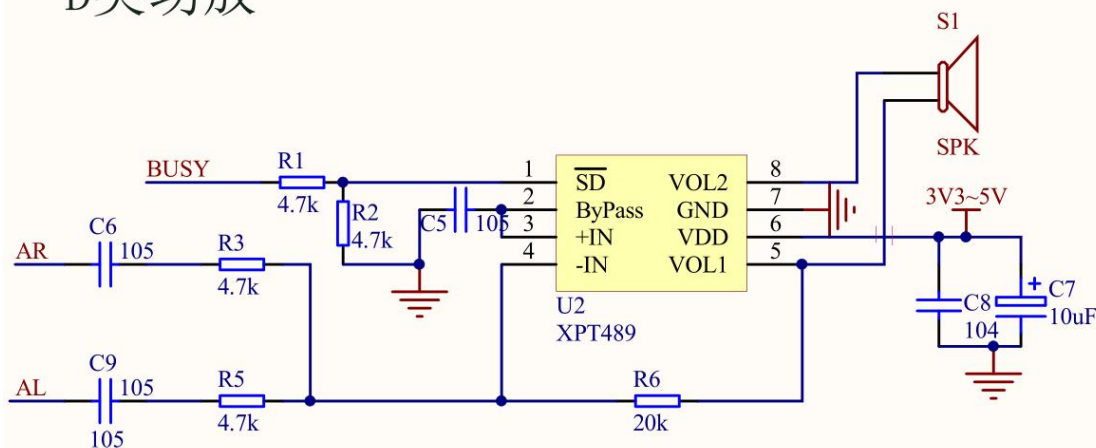


CX800S 内置 LDO，内部使用 LDO 输出电压 3V3，IO 电压为 3.3V TTL 电平，当 MCU IO 电压为 5V 时和 CX800S IO 电平不一致不可以直接相连，需做隔离，使用 1K 电阻串联或者使用二极管隔离。



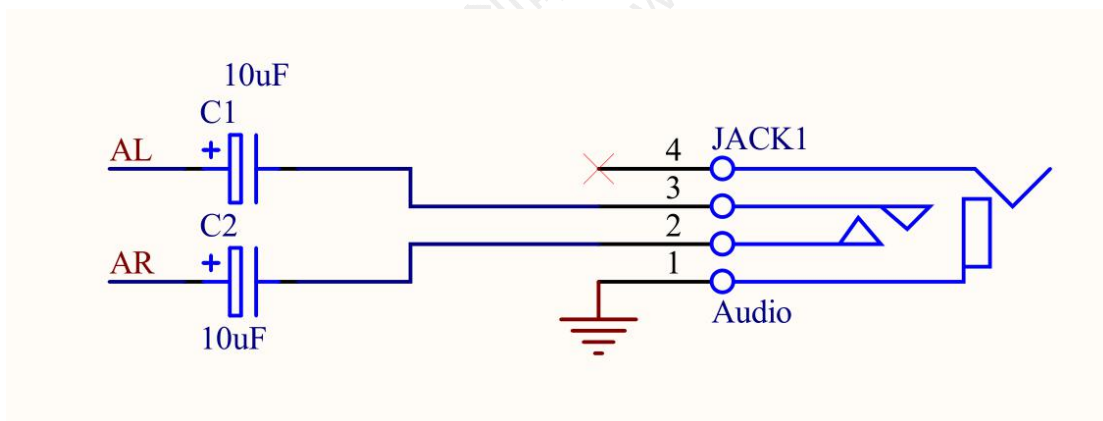
4.2 外接单声道功放

D类功放



采用xpt4890，具体参数请参考IC的datasheet。应用于一般场合足以，如果追求更高的音质，请客户自行寻找合适的功放。

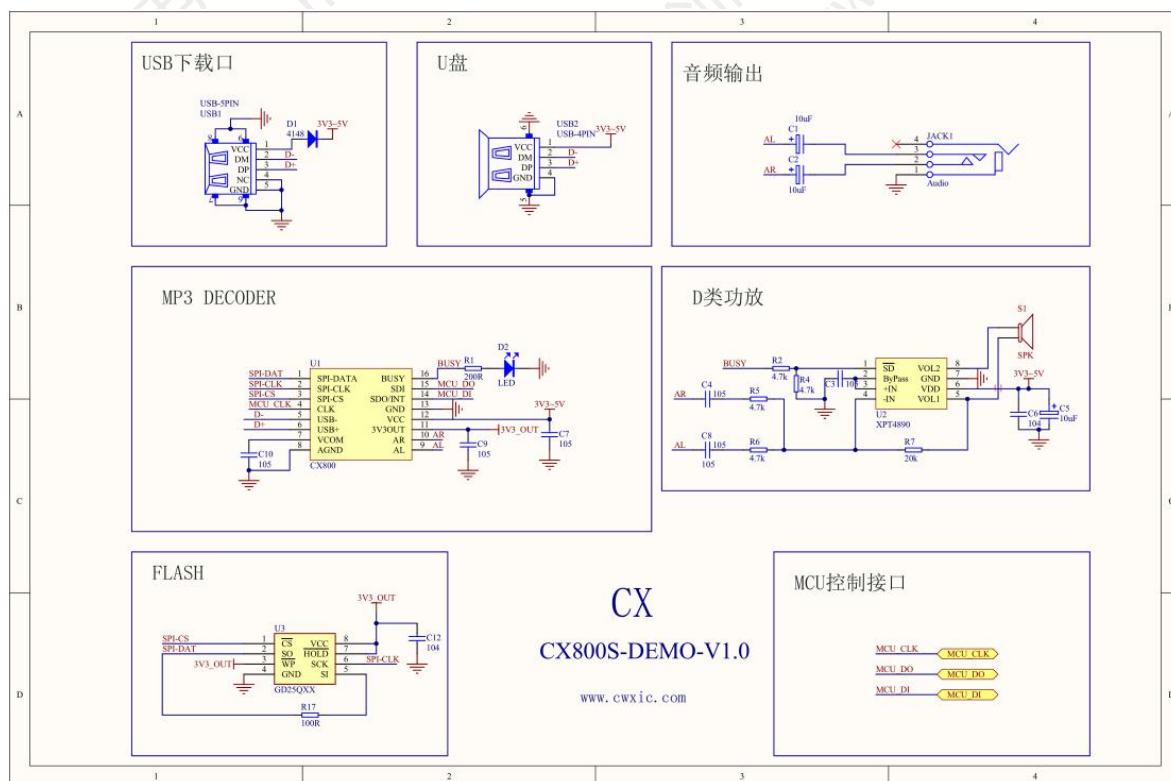
4.3 外接耳机电路



这里R4 和R5 为限幅电阻，防止外部音源幅度过大(V_{p-p} 最大值为3.0V)，影响系统的稳定性，C1和C2 为隔直电容，防止外部音源的直流电平影响到芯片内部的偏置；R2 和R3 预留电阻给大功放设计用。

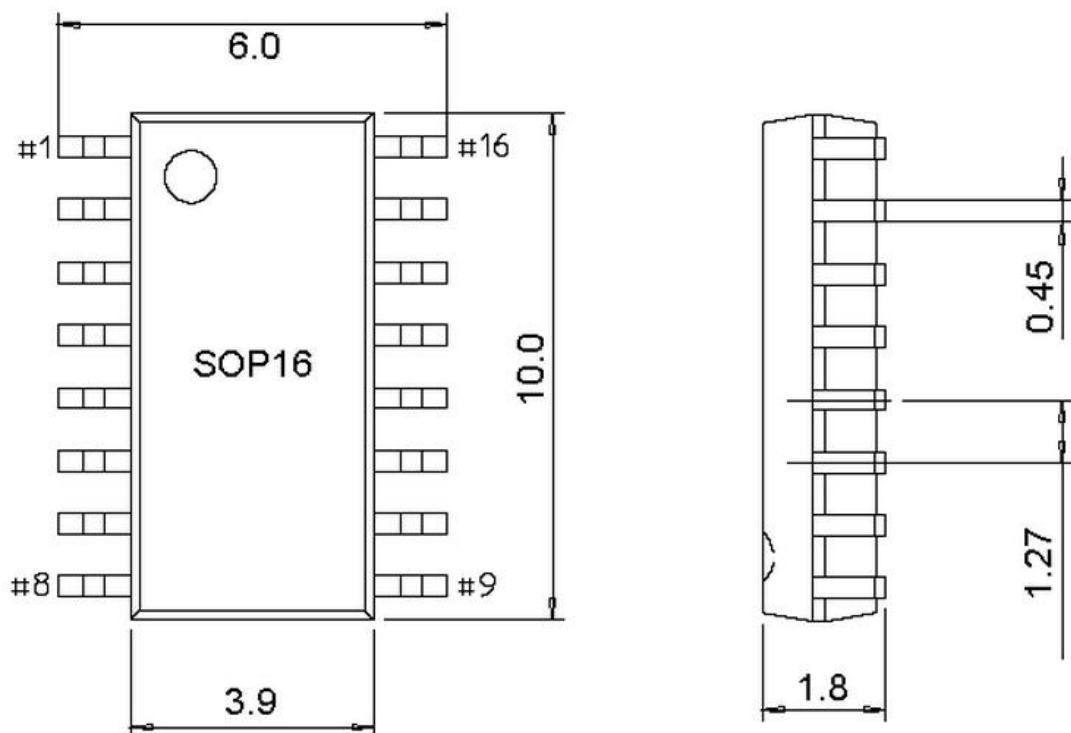


4.4 主控电路（向客服索取高清文件）





SOP16



6. 版本信息

版本	日期	更新内容
V1.0	2017-05-11	原始版本
V1.1	2018-09-10	修改通信协议信息
V1.2	2018-09-18	通信增加校验数据返回

7. 免责声明

7.1 开发预备知识

CX 系列产品将提供尽可能全面的开发模版、驱动程序及其应用说明文档以方便用户使用但也需要用户熟悉自己设计产品所采用的硬件平台及相关 C 语言的知识。



7.2 EMI 和 EMC

CX 系列芯片机械结构决定了其 EMI 性能必然与一体化电路设计有所差异。CX 系列芯片的 EMI 能满足绝大部分应用场合，用户如有特殊要求，必须事先与我们协商。

CX 系列芯片的 EMC 性能与用户底板的设计密切相关，尤其是电源电路、I/O 隔离、复位电路，用户在设计底板时必须充分考虑以上因素。我们将努力完善 CX 系列芯片的电磁兼容特性，但不对用户最终应用产品 EMC 性能提供任何保证。

7.3 ESD 静电放电点保护

CX 系列产品部分元器件内置 ESD 保护电路，但在使用环境恶劣的场合，依然建议用户在设计底板时提供 ESD 保护措施，特别是电源与 IO 设计，以保证产品的稳定运行，安装 QY 系列产品为确保安全请先将积累在身体上的静电释放，例如佩戴可靠接地的静电环，触摸接入大地的自来水管等。



佛山创唯芯电子科技有限公司是一家专注于语音科技研究的高新技术企业。从事语音产品开发、设计、生产与销售。同时为有特别需求的客户制订语音产品开发方案，并且落实执行该方案，完成产品的研发、测试，声音处理，直至产品的实际应用指导等一系列服务。目前主要从事各种语音芯片 IC、语音模块、语音播放板、语音提示器、MP3 方案开发、人体感应提示器等。应用于家用电器、汽车电子、医疗设备、暗防门禁、智能家居、工业产品、玩具等。公司拥有一支专注于开发的技术团队、优质服务、诚实有信的商业原则，赢取广大客户及社会的支持与信任！



附录:

1. 音频文件排序

CX800S 的音频文件索引排序, 是以音频文件先后存放到 存储器的顺序排序, 并非按照文件名排列顺序。因此 CX800S 按

照索引播放文件时、排列的顺序和文件名无关, 只和拷贝到存储器中顺序有关。

我们可以在电脑上面建一个文件夹、将文件全部拷贝到文件夹中, 然后在文件夹内把文件按照自己想要的顺序排列好、

接下来、按照以下两种方法将文件拷贝到 TF 卡或者 U 盘中。(如果是想要下载到 SPI 中的、可以先拷贝到 TF 卡或者 U 盘中,

然后发指令或者通过拷贝按键拷贝到 SPI 中)

有两种常用的复制方法:

一、是用快捷键“Ctrl+C”和“Ctrl+V”, 但注意鼠标不能点击到任何选中待发送的文件, 否则会因鼠标所点击的文件

开始发送的。这样就会打乱了文件的顺序了。

二、是排列好文件的顺序, 选中所要发送的文件, 然后右键点击第一个文件(例如 001-欢迎光临.mp3), 在右键菜单中选

择发送到存储盘的根目录。(注意右键点击的是要发送的第一个文件, 系统会从此文件开始发送的)。