



MX6305 语音芯片使用说明书 V1.1

(支持 SPI-Flash、U 盘、TF 卡)



广州美芯电子科技有限公司

出 品



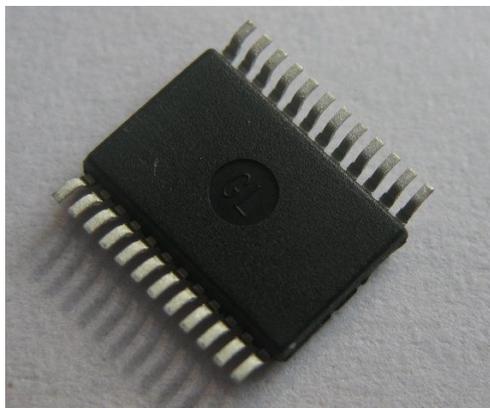
目 录

1. 概述.....	- 3 -
1.1 简介.....	- 4 -
1.2 功能.....	- 4 -
1.3 应用.....	- 5 -
2. 芯片使用说明.....	- 6 -
2.1 硬件参数.....	- 7 -
2.2 更换语音内容.....	- 7 -
2.3 芯片管脚说明.....	- 8 -
3. 控制方式说明.....	- 9 -
3.1 按键接口.....	- 9 -
3.2 通讯格式.....	- 9 -
通信格式:	- 9 -
容错要求:.....	- 10 -
3.3 通讯指令.....	- 10 -
播放控制: (指令:02).....	- 10 -
音量控制:(指令: 03).....	- 12 -
插播控制: (指令: 04)	- 13 -
EQ 设置:(指令: 06).....	- 14 -
组合播放: (指令: 07)	- 14 -
错误信息: (指令: 08)	- 14 -
设备插拨状态信息: (指令: 09)	- 15 -
声道通道切换: (指令:0A).....	- 15 -
系统睡眠指令: (指令:0B).....	- 15 -
4. 参考电路.....	- 15 -
4.1 串行接口.....	- 16 -
协议约定:	- 16 -
4.2 外接单声道功放.....	- 17 -
4.3 外接耳机电路.....	- 17 -
4.4 主控电路 (向客服索取高清文件)	- 18 -
5. MX6305-24SS 封装图.....	- 19 -
6. 注意事项.....	- 20 -
7. 免责声明.....	- 21 -



附录:	- 22 -
MX 系列 MP3 芯片选型参考表.....	- 22 -
检验和程序.....	- 22 -
常用命令程序.....	- 24 -
指定路径播放程序.....	- 26 -
指定路径插播程序.....	- 28 -
组合播放程序.....	- 29 -
短文件名规则.....	- 30 -
文件系统和曲目序号排列说明.....	- 30 -

1. 概述



1.1 简介

MX6305 是一个提供串口的 MP3 芯片，完美的集成了 MP3 的硬解码。同时软件支持 TF 卡驱动，支持电脑直接更新 SPI-flash 的内容，支持 FAT16、FAT32 文件系统。通过简单的串口指令即可完成播放指定的音乐，以及如何播放音乐等功能，无需繁琐的底层操作，使用方便，稳定可靠是此款产品的最大特点。另外该芯片也是深度定制的产品，专为固定语音播放领域开发的低成本解决方案。

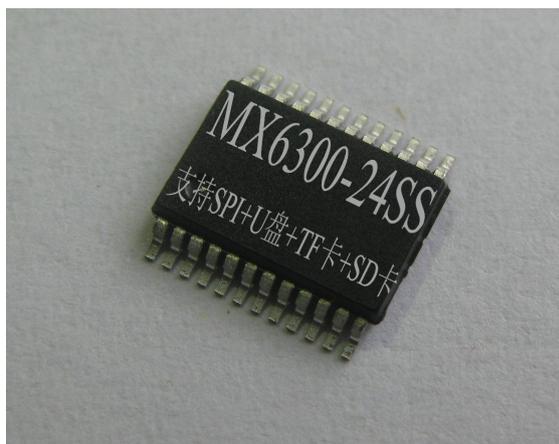
1.2 功能

- ◆ 支持 MP3 WAV 硬件解码
- ◆ 支持 FAT 文件系统
- ◆ 支持采样率(KHz):8/11.025/12/16/22.05/24/32/44.1/48
- ◆ 24 位 DAC 输出内部采用 DSP 硬件解码,非 PWM 输出,动态范围支持 90dB,信比 85dB
- ◆ 多种控制模式、两线串口模式、一线串口控制、ADKEY 等
- ◆ 支持 U 盘（最大支持 32G）、TF 卡（最大支持 32G）、SPIFLASH（最大支持 128M）
- ◆ 支持 USB 声卡，读卡器，HID 控制
- ◆ 支持 SPIFLASH 枚举成 U 盘，直接像操作 U 盘一样更新 SPIFLASH 里的语音
- ◆ 支持上一曲，下一曲，播放、暂停、停止、选曲、等常用功能控制
- ◆ 支持播放曲目序号获取，总曲目及目录总曲目等信息获取
- ◆ 支持各种信息查询，轻松获取语音芯片的当前状态
- ◆ 支持指定曲目插播，即可以暂停当前播放的音乐，播放插播的曲目，播放完后返回原来曲目断点处播放
- ◆ 支持中英文路径插播，除了指定曲目插播，同时也支持指定路径插播，只需要发送插播的路径即可以轻松实现

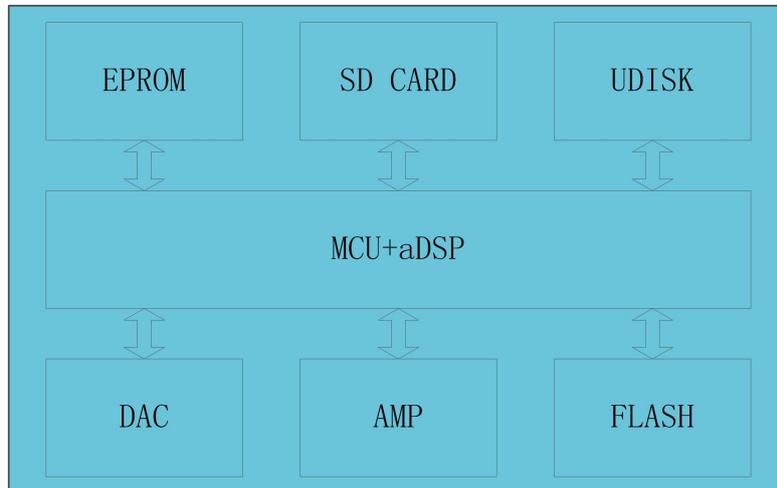
- ◆ 支持跨盘符插播，如插播内容存储在 SPIFLASH
- ◆ 支持指定中英文路径播放，无需知道要播放的内容的序号，直接发送所在的路径即可。
- ◆ 支持智能组合播放，特定的文件夹里的歌曲组合，发送曲目名即可组合。
- ◆ 支持 30 级音量调节
- ◆ 支持 5 种 EQ 调节
- ◆ 支持 7 种播放模式控制，可以适应不同的应用
- ◆ 支持循次数设置，更多应用，更人性化
- ◆ 支持系统深度睡眠，睡眠后电流低至 600 微安，可以通过一线串和指令 IO 唤醒
- ◆ 支持指定时间快进，快退
- ◆ 支持指定时间段复读
- ◆ 支持指定从指定的时候点开始播放
- ◆ 支持外输入音频和 MP3 音频混合，外输入、MP3 输出、外输入和 MP3 混合三种输出切换
- ◆ 专用的 BUSY 信号输出指示
- ◆ 支持各种状态变化信息返回，如设备插拨等
- ◆ 成熟的指令和指令解析，让应用更稳定
- ◆ 专用配套上位机，快速上手，方便调试，指令自动生成
- ◆ 支持广播语插播功能，可以暂停正在播放的背景音乐，支持指定路径下的歌曲播放，支持插播提前结束；
- ◆ 支持组合播放和结束组合播放；
- ◆ 指定盘符播放，指定曲目播放；
- ◆ 指定路径播放(支持中英文)；
- ◆ 支持文件夹切换、支持播放设备切换；

1.3 应用

- 1、 车载导航语音播报；
- 2、 公路运输稽查、收费站语音提示；
- 3、 火车站、汽车站安全检查语音提示；
- 4、 电力、通信、金融营业厅语音提示；
- 5、 车辆进、出通道验证语音提示；
- 6、 公安边防检查通道语音提示；
- 7、 多路语音告警或设备操作引导语音；
- 8、 电动观光车安全行驶语音告示；
- 9、 机电设备故障自动报警；
- 10、 消防语音报警提示；
- 11、 自动广播设备，定时播报。



2. 芯片使用说明



芯片选用的是 SOC 方案，集成了一个 16 位的 MCU，以及一个专门针对音频解码的 aDSP，采用硬解码的方式，更加保证了系统的稳定性和音质。小巧的封装尺寸更加满足嵌入其它产品的需求。

2.1 硬件参数

名称	参数
MP3文件格式	1、支持所有比特率11172-3和 ISO13813-3 layer3 音频解码
	2、采样率支持(KHZ):8/11.025/12/16/22.05/24/32/44.1/48
	3、支持 Normal、Jazz、Classic、Pop、Rock 音效
USB 接口	2.0标准
UART 接口	标准串口, TTL 电平,波特率可设
输入电压	供电在3.2V-5V, 最佳为4.2V
额定电流	20ma[不带 U 盘]
尺寸	标准的 SSOP24封装
工作温度	-40度~70度
湿度	5% ~ 95%

2.2 更换语音内容

此芯片最大的优势在于能够灵活地更换 spi-flash 内的语音内容,省去了用上位机更新音乐的麻烦,直接以 U 盘形式来更新 spi-flash 的内容,方便快捷,一目了然(直接当 U 盘使用)。

通过 usb 接口连接电脑,如果带 tf 卡和 spi-flash 两种设备则会在电脑上显示两个盘符,第一次使用的 spi-flash 会提示格式化,按电脑要求格式化后 spi-flash 里就有文件系统的信息,这样电脑就能认识盘符里的文件,格式化后就可以忘盘符里拷贝音乐等文件,和 U 盘使用的方法一样。

2.3 芯片管脚说明

U1			
1	DACL	DACVSS	24
2	DACR	VCOM	23
3	VDDIO	DACVDD	22
4	LDO5V	USBDP	21
5	VSSIO	USBDM	20
6	P23/P24	P22	19
7	P25	P21	18
8	P26	P20	17
9	P27	P02	16
10	VPP/P46	P05	15
11	P17	P00	14
12	P16	P01	13

引脚序号	引脚名称	功能描述	备注
1	DACL	音频输出左声道	驱动耳机、功放
2	DACR	音频输出右声道	驱动耳机、功放
3	VDDIO	3.3V 电源输出	给 TF 卡、SPI、24C02 供电 最大电流只有 100mA，不建议 给其他非 MP3 电路供电
4	VDD	5V 电源输入	不可以超过 5.2V
5	VSS	电源地	
6	TX	UART 串行数据输出	
7	RX	UART 串行数据输入	
8	AUXL	外部音频输入左	可以独立输出，也可和 MP3 混 合输出
9	AUXR	外部音频输入右	
10	GPIOA0	红外遥控接收	需定制
11	GPIOA1	Busy 输出	输出高电平
12	GPIOA2	SPI_CS 片选总线	
13	GPIOA3	SPI_DO 数据总线	
14	GPIOA4	SPI_CLK 数据总线	
15	GPIOA5	ADKEY2 外接按键	必须接 22K 上拉
16	GPIOA6	ADKEY1 外接按键	必须接 22K 上拉
17	GPIOB4	SD_CLK 时钟总线	串 0 欧电阻接到 24C02 6 脚做记忆
18	GPIOB3	SD_CMD 命令总线	串 0 欧电阻接到 24C02 5 脚做记忆
19	GPIOB2	SD_DAT 数据总线	
20	GPIOB1	USB- DM	接 U 盘和电脑的 USB 口
21	GPIOB0	USB+ DP	接 U 盘和电脑的 USB 口
22	NC	烧写口	
23	VCOM	退耦	
24	DACVSS	地	

3. 控制方式说明

3.1 按键接口

芯片我们采用的是 AD 按键的方式，取代了传统了矩阵键盘的接法，这样做的好处是充分利用了 MCU 越来越强大的 AD 功能。设计简约而不简单，我们芯片默认配置 2 个 AD 口，按键的阻值分配，如果使用在强电磁干扰或者强感性、容性负载的场合，请参考我们的“注意事项”。

(1)、参考按键阻值

电阻	短按	长按	一直按着
22K	上拉		
220K	曲目 1	曲目 6	
100K	曲目 2	曲目 7	
51K	曲目 3	曲目 8	
33K	曲目 4	曲目 9	
24K	曲目 5	曲目 10	
15K	设备切换		
9K1	下一曲		音量加
6K2	上一曲		音量减
3K	单曲停止	全部循环	
0R	播放/暂停	停止	

3.2 通讯格式

通信采用全双工串口串口通信，
波特率为 9600，数据位：8，停止位 1 位，检验位 N

通信格式：

- 指令码-验证码-数据长度 (n) -数据 1-数据 2-...-数据 n-和检验(SM)
- 指令码：用来区分指令类型
- 验证码：指令码的反码，用来验证指令码
- 数据长度：指令中的数据的字节数
- 数据：指令中的相关数据
- 和检验(SM)：为之前所有字节之和低 8 位，程序请参考附录
- 数据：发送的数据或命令，高 8 位数据在前，低 8 位在后

容错要求:

接受方必须在线实时验证指令码及其反码, 及和校验的正确性; 每条指令数据字节数可能不同, 因此必须实施接收到字节数后确定要接收的后续数据字节数, 若有错误则应丢弃所接受指令, 若一条指令接收时间超过 500ms, 则强制丢弃所接受的指令字节, 重新同步指令接收。

3.3 通讯指令

关键字说明:

SM: 和检验(之前所有字节之和低 8 位)

曲目高: 曲目的索引高 8 位

曲目低: 曲目的索引低 8 位

索引: 芯片会对存储设备的所有曲目进行编号, 方便访问, 具体的编号规则请参考附录--曲目编号规则

播放控制: (指令:02)

播放状态 : 00(停止) 01(播放) 02(暂停)

设备定义: USB:00 SD:01 FLASH:02 NO_DEVICE 0xFF

- 查询播放状态(01)
指令: 02 FD 01 01 01
返回: 02 FD 02 01 播放状态 SM
- 播放(02)
指令: 02 FD 01 02 02
返回: 02 FD 03 0E 曲目高 曲目低 SM
- 暂停(03)
指令: 02 FD 01 03 03
返回: 无
- 停止(04)
指令: 02 FD 01 04 04
返回: 无
- 上一曲(05)
指令: 02 FD 01 05 05
返回: 无
- 下一曲(06)
指令: 02 FD 01 06 06
返回: 无
- 指定曲目(07)

指令：02 FD 03 07 曲目高 曲目低 SM

返回：无

例如：02 FD 03 07 00 08 0F 指定播放当前盘符第 8 首，本说明文档里的曲目顺序都是指文件拷贝顺序，曲目数从 1—65535

- 指定路径播放(08)

指令：02 FD 长度 08 路径 SM

返回：无

例如：SPI-FLASH 中有歌曲 /周华健 /难念的经 MP3

U 盘中有歌曲 /新建文件夹/有没有那么一首歌会让你想起我.MP3

示例程序：

```
MX_SpecifiedPath(“/周华健 /难念的经 MP3”,MX_FLASH);
```

```
MX_SpecifiedPath(“/新建文*/有没有*MP3”,MX_USB);
```

MX_SpecifiedPath()实现方法请参考附录--指定路径播放程序。

中文的编码方式为 GB2312。

输入播放路径为短文件名规则，请参考附录--短文件名规则。

- 查询当前在线设备(09)

指令：02 FD 01 09 09

返回：02 FD 02 09 盘符 SM

说明：在线设备是按位来区分的：USB:BIT(0) SD:BIT(1) SPI-FLASH:BIT(2)

- 查询当前播放设备(0A)

指令：02 FD 01 0A 0A

返回：02 FD 02 0A 盘符 SM

设备盘符定义：USB:00 SD:01 FLASH:02 NO_DEVICE 0xFF

- 切换指定设备(0B)

指令：02 FD 02 盘符 0B

返回：无

如果当前设备在线，可以切换到相对应的盘符播放，建议切换前先查询一下设备是否在线。

- 查询总曲目(0C)

指令：02 FD 01 0C 0C

返回：02 FD 03 0D 总曲目高 总曲目低 SM

说明：查询当前盘符总曲目数

- 查询当前曲目(0E)

指令：02 FD 01 0E 0E

返回：02 FD 03 0E 曲目高 曲目低 SM

- 上一文件夹(0F)

指令: 02 FD 01 0F 0F

返回: 在播放时返回曲目序号

说明: 切换后会播放文件夹里的第一首, 其中返回指令的“0E”是起识别指令的作用。

- 下一文件夹(10)

指令: 02 FD 01 10 10

返回: 在播放时返回曲目序号

说明: 切换后会播放文件夹里的第一首

- 播放结束(11)

指令: 02 FD 01 11 11

说明: 在曲目播放完后返回此命令

- 查询目录首曲目(12)

指令: 02 FD 01 12 12

返回: 02 FD 03 12 曲目高 曲目低 SM

说明: 查询当前文件夹第一首歌曲在全盘里的序号

- 查询目录总曲目(13)

指令: 02 FD 01 13 13

返回: 02 FD 03 13 曲目高 曲目低 SM

说明: 查询当前文件夹总曲目数

音量控制:(指令: 03)

音量 (0~30) 最大为 30, 上电默认 30 (不支持掉电记忆)

- 音量设置(01):

指令: 03 FC 02 01 VOL SM

返回: 无

例如: 03 FC 02 01 14 16 设置音量为 20 级

说明: VOL 为音量等级, 0~30 级

- 音量加(02) :

指令: 03 FC 01 02 02

返回: 无

- 音量减(03) :

指令: 03 FC 01 03 03

返回: 无

插播控制：（指令：04）

说明：此功能支持动态插播，共 50 首，也就是在播放插播的时候也可以插播，这时候插播的曲目会存储起来，等前面的播放晚就会接着播放后插播的，所有的插播完后返回到原来曲目恢复断点播放。

- 指定曲目插播(01)：

指令：04 FB 03 01 曲目高 曲目低 SM

返回：无

例如：04 FB 04 01 00 00 09 0D 插播第 9 首

说明：插播结束后返回插播点继续播放

- 结束曲目插播(01)：

指令：04 FB 01 01 01

返回：无

- 指定路径插播(02)：

USB:00 SD:01 FLASH:02

指令：04 FB 长度 02 盘符 路径 SM

返回：无

说明：插播结束后返回插播点继续播放

例如：SD 卡中的曲目 /广告 /小米手机 MP3

程序示例：

```
MX_InsertSpecifiedPath("/广告 /小米手机 MP3",MX_SD);
```

MX_InsertSpecifiedPath()实现方法请参考附录--指定路径插播程序。

中文的编码方式为 GB2312。

输入播放路径为短文件名规则，请参考附录--短文件名规则。

播放模式：（指令：05）

- 设置循环模式(01):

指令：05 FA 02 01 循环模式 SM

返回：无

举例：设置为单曲循环：05 FA 02 01 01 03

说明：播放模式支持掉电记忆

全部循环(00)：按顺序播放全盘曲目，播放完后循环播放

单曲循环(01)：一直循环播放当前曲目

文件夹内循环(02)：按顺序播放当前文件夹内曲目，播放完后循环播放

- 随机播放(03): 随机播放盘符内曲目
- 单曲停止(04): 播放完当前曲目一次停止
- 顺序播放(05): 按顺序播放全盘曲目, 播放完后停止

EQ 设置:(指令: 06)

EQ 定义: NORMAL:00 POP:01 ROCK:02 JAZZ:03 CLASSIC:04

- EQ 设置(01)
指令: 06 F9 02 01 EQ SM
返回: 无
例如: 06 F9 02 01 02 04 设置 EQ 为 ROCK
说明: EQ 默认设置为 NORMAL (掉电不记忆)

组合播放: (指令: 07)

- 组合播放(01)
指令:
07 F8 长度 00 曲目 1 高字节, 曲目 1 低字节..... 曲目 n 高字节, 曲目 n 低字节 SM
返回: 无
例如: 07 F8 07 01 00 04 00 01 00 05 11, 第 4、1、5 三首歌曲组合播放
说明: 每首曲目都由高低两个字节来组成, 最大组合 15 首。每个曲目播放结束都会返回“播放结束”指令, 组合播放中的曲目不存在时会自动跳过。
- 结束组合播放(02)
指令: 07 F8 02 02 03
返回: 无
说明: 结束组合播放, 返回组合前的播放状态

错误信息: (指令: 08)

- 08 F7 01 00 00 串口接收数据错误
- 08 F7 01 02 02 批定的盘符找不到
- 08 F7 01 03 03 没有可以播放盘符
- 08 F7 01 04 04 文件播放错误, 如文件找不到等

设备插拔状态信息：（指令：09）

- 09 F6 01 00 00 U 盘插入
- 09 F6 01 01 01 U 盘拔出
- 09 F6 01 02 02 SD 卡插入
- 09 F6 01 03 03 SD 卡拔出

声道通道切换：（指令：0A）

我们芯片声音为 dac 输出，此指令为选择 dac 输出哪几个通道的声音

CH_MP3 0 说明：播放 MP3 通道，dac 输出的声音为音乐播放的声音。

CH_AUX 1 说明：AUX 通道，dac 输出的声音为 P26 和 P27 输入的声音

MP3_AUX 2 说明：AUX 通道 MP3 同时打开，DAC 输出的声音是音乐播放和输入声音的混音

- 设置通道(01)
指令：0A F5 02 01 通道 SM
返回：无

系统睡眠指令：（指令：0B）

系统进入低功耗睡眠，功耗约几百微安，用一线串口话 AD 触发以唤醒芯片，唤醒后会接着原来的工作。

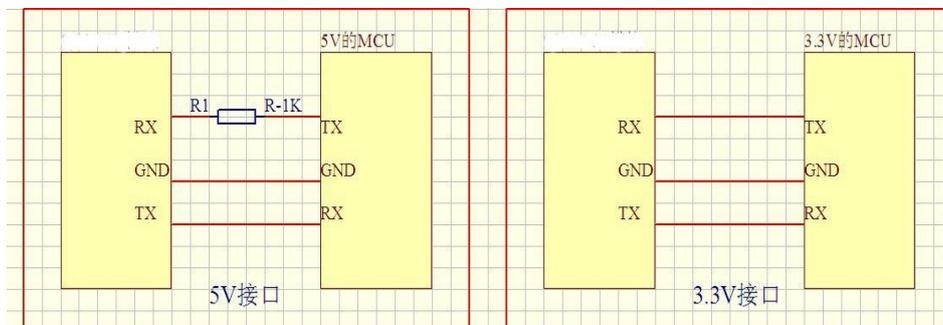
- 睡眠(01)
指令：0B F4 01 01 01
返回：无

4. 参考电路

针对芯片的应用，我们提供了详细的设计参考，让您更快的上手体验到该芯片的强大功能：

- 串行通信接口，波特率默认 9600；
- 外部 AD 按键的接口电路，按键的功能可以按照客户需求订制；
- 外部单声道功放参考电路。

4.1 串行接口

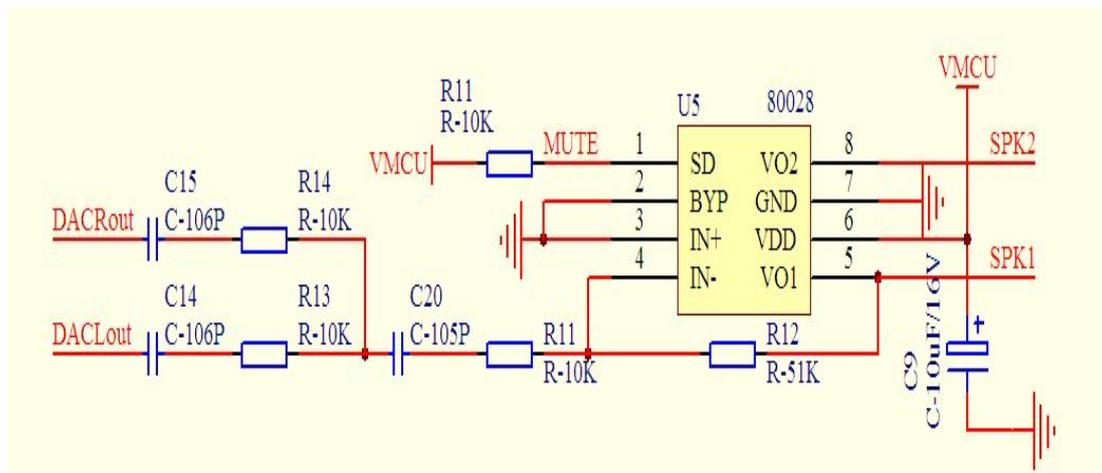


- 1、我方作为从机，上电默认等待状态，所以播放操作全由主机控制。
- 2、从机在状态发生改变是会主动发起数据传输，发送相关状态，比如拔卡，插入 U 盘等。
- 3、串口通信电平为：TTL（输出高电平 $>2.4V$ ，输出低电平 $<0.4V$ ），由于本芯片 io 供电为 3.3v，如果主机系统是 5v，请在中间串 1K 电阻。如果主机串口电平为 RS232（逻辑 1：-3~-15V 逻辑 0：+3~-15V）或 RS485（逻辑 1：电压差为+（2~6）V 逻辑 0：电压差为-（2~6）V），请加电平转换芯片。
- 4、如未特别说明，协议中所有数据都是表示十六进制数据。

协议约定：

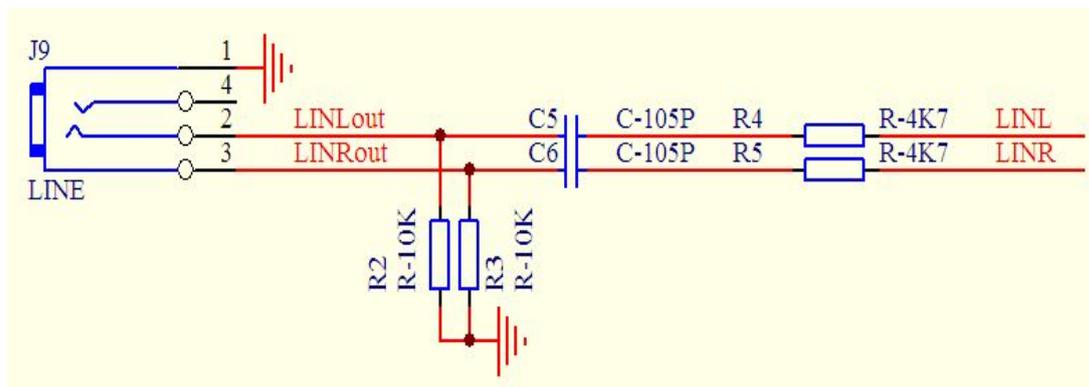
- 新曲目播放时会发送当前曲目序号，这里的序号为全盘统一编号的序号，本协议如无特殊说明都是批全盘统一编号的序号；
- 盘符拔出时只会发送盘符拔出指令，不做换盘符处理，等待操作指令；
- 协议中所有数据都表示 HEX；
- SM 表示之前所有字节之和低 8 位；
- 协议中指的的文件都是指可以播放的文件数，不包括非播放文件；
- 模式切换过去都处理停止播放状态，要发送播放命令才会播放，上电默认为停止状态，循环模式默认为全部循环；
- 掉电记忆：曲目号、音量、播放模式，需要有 24C02 或 SPI-FLASH。

4.2 外接单声道功放



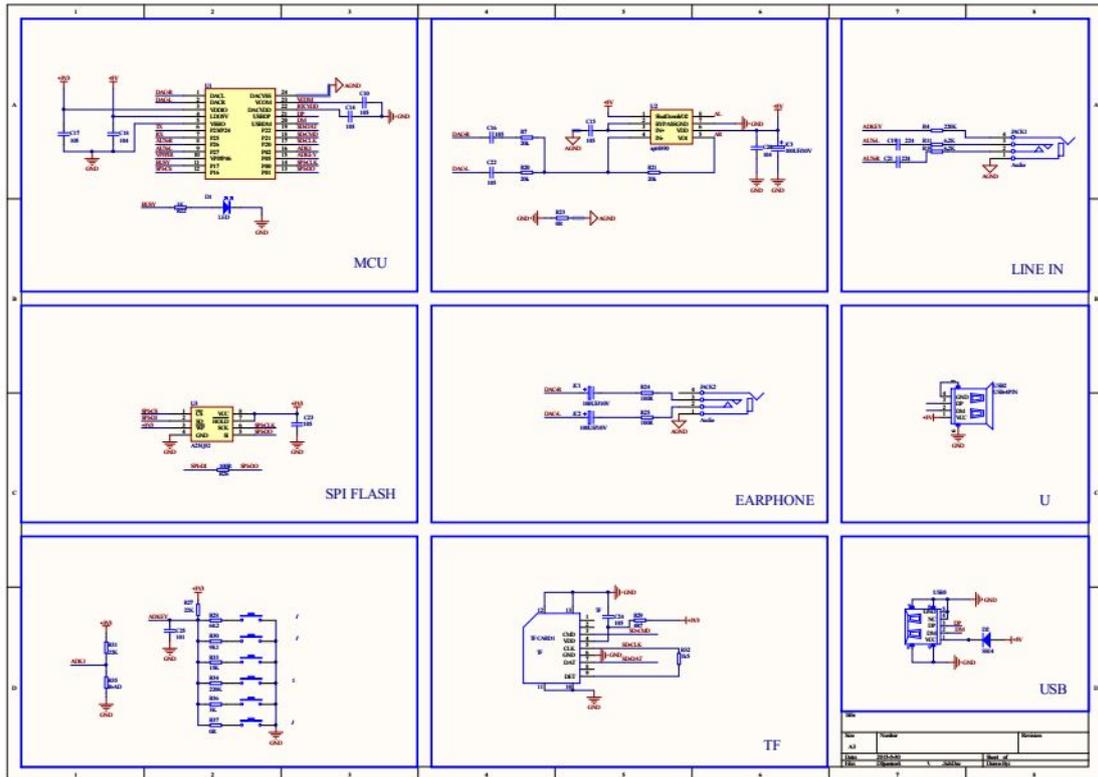
这里功放我们采用的是4890，具体参数请参考IC的datasheet。应用于一般场合足以，如果追求更高的音质，请客户自行寻找合适的功放。

4.3 外接耳机电路



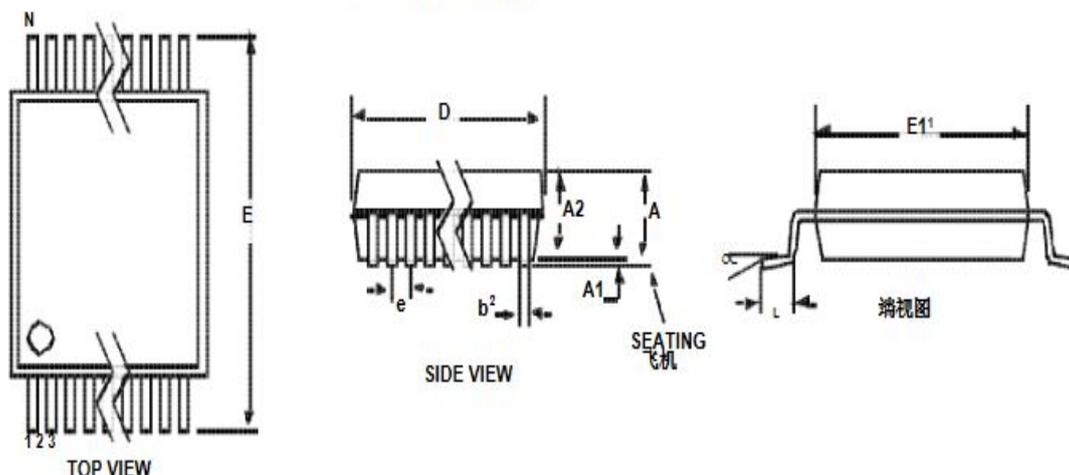
这里R4 和R5 为限幅电阻，防止外部音源幅度过大(V_{p-p} 最大值为3.0V)，影响系统的稳定性，C1和C2 为隔直电容，防止外部音源的直流电平影响到芯片内部的偏置；R2 和R3 预留电阻给大功放设计用。

4.4 主控电路（向客服索取高清文件）



5. MX6305-24SS 封装图

24L SSOP封装图



DIM	INCHES			MILLIMETERS			注:
	MIN	NOM	MAX	MIN	NOM	MAX	
A	--	--	0.084	--	--	2.13	
A1	0.002	0.006	0.010	0.05	0.13	0.25	
A2	0.064	0.068	0.074	1.62	1.73	1.88	
b	0.009	--	0.015	0.22	--	0.38	2,3
D	0.311	0.323	0.335	7.90	8.20	8.50	1
E	0.291	0.307	0.323	7.40	7.80	8.20	
E1	0.197	0.209	0.220	5.00	5.30	5.60	1
e	0.022	0.026	0.030	0.55	0.65	0.75	
L	0.025	0.03	0.041	0.63	0.75	1.03	
α	0°	4°	8°	0°	4°	8°	

JEDEC #: MO-150

控制尺寸为毫米。

注: 3. "D"和"E1"是参考数据,不包括塑模毛边或突起,但不包括模具不匹配,并测量在分模线上,模具毛边或突起不得超过0.20毫米,每边。

4. 尺寸"b"不包括丹巴尔症/入侵,应允许丹巴尔症在"B"尺寸超过0.13 mm总在最大的物质条件,丹巴尔入侵不得减少尺寸"b"至少大于0.07毫米的物质条件。

5. 这些尺寸适用于0.10和0.25毫米的导线头间的导线的扁平部分。

6. 注意事项

IO 输入特性						
符号	参数	最小	典型	最大	单位	测试条件
V_{IL}	Low-Level Input Voltage	-0.3	-	$0.3 \cdot V_{DD}$	V	$V_{DD}=3.3$ V
V_{IH}	High-Level Input Voltage	$0.7V_{DD}$	-	$V_{DD}+0.3$	V	$V_{DD}=3.3$ V
IO 输出特性						
符号	参数	最小	典型	最大	单位	测试条件
V_{OL}	Low-Level Output Voltage	-	-	0.33	V	$V_{DD}=3.3$ V
V_{OH}	High-Level Output Voltage	2.7	-	-	V	$V_{DD}=3.3$ V

- 1、芯片对外的接口均是 3.3V 的 TTL 电平，所以在硬件电路的设计中，请注意电平的转换问题。另外在强干扰的环境中，请注意电磁兼容的一些保护措施，GPIO 采用光耦隔离，增加 TVS 等等。
- 2、ADKEY 的按键取值均按照一般的使用环境，如果在强感性或者容性负载的环境下，请注意芯片的供电，建议采用单独的隔离供电，另外再配上磁珠和电感对电源的滤波，一定要尽可能的保证输入电源的稳定和干净。如果实在无法保证，请联系我们，减少按键的数量，重新定义更宽的电压分配。
- 3、串口通信，在一般的使用环境下，注意好电平转换即可。如果强干扰环境，或者长距离的 RS485 应用，那么请注意信号的隔离，严格按照工业的标准设计通信电路。可以联系我们，我们提供设计参考。

7. 免责声明

■ 开发预备知识

MX 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。

■ EMI 和 EMC

MX 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。MX 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。

MX 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 MX 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。

■ 修改文档的权力

广州美芯电子科技有限公司有保留任何时候在不事先声明的情况下对 MX 系列产品相关文档的修改权力。

■ ESD 静电放点保护

MX 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，例如佩戴可靠接地的静电环，触摸接入大地的自来水管等。

附录 1：MX 系列 MP3 芯片选型参考表

MX 系列 MP3 芯片选型参考表 10-20 更新					
名称 (封装)	形式	功能	支持存储设备	串口通信方式	主要功能
MX6005	SOP16	播放芯片	U 盘（最大 32G） SPI（最大 128M）	一线 串口	(1) 支持 SPI-flash 模拟成 U 盘 (2) 支持 AD 按键控制 (3) 支持音量和当前曲目掉电记忆
MX6100	SOP16	播放芯片	U 盘（最大 32G） TF 卡（最大 32G） SD 卡（最大 32G）	UART 9600bps	(1) 支持组合播放、选曲播放、路径播放、 曲目插播、路径插播 (2) 支持 AD 按键控制
MX6102	SOP16				(1) 支持 AD 按键 8 路一对一播放（8 个按键 分别对应 8 首歌曲） (2) 播放模式默认单曲停止
MX6104	SOP16				(1) 支持当前歌曲短文件名查询、上一曲不 播放、下一曲不播放、选曲不播放 (2) 支持 AD 按键控制
MX6105	SOP16				(1) 在 6100 的基础上，增加上电全部循环播 放功能
MX6106	SOP16				(1) 缩短了延时时间，最小 10MS 即可复位。 (原来需要 150MS-800MS) (2) 单曲停止
MX6107	SOP16				(1) 声卡功能，其它如 6200 一样。
MX6108	SOP16				(2) 客户定制，不对外出货
MX6200	SOP16				播放芯片
MX6201	SOP16	波特率为 4800，其它功能和 6200 一样			
MX6202	SOP16	(1) 定制功能：上电播放，全部循环。 (2) 其它功能和 6200 一样 (3) 不支持 U 盘			



MX6203	SOP16				(1) 支持 AD 按键 8 路一对一播放 (8 个按键分别对应 8 首歌曲) (2) 播放模式、当前播放曲目支持掉电记忆
MX6204	SOP16				(1) 把音量调为 18 (最大 30 级) (2) 单曲停止播放
MX6205	SOP16				(1) 缩短了延时时间, 最小 10MS 即可复位。 (原来需要 150MS-800MS)
MX6206	SOP16				(1) 定制功能: 上电播放, 单曲停止。 (2) 其它功能和 6200 一样 (3) 支持 U 盘
MX6207	SOP16				(1) 把音量调为 18 (最大 30 级) (2) 单曲停止播放 (3) 缩短了延时时间, 最小 10MS 即可复位。 (原来需要 150MS-800MS) (4) 主推此功能, 打算代替原来的 6200
MX6300	SSOP24	播放芯片	U 盘 (最大 32G) TF 卡 (最大 32G) SD 卡 (最大 32G) SPI- (最大 128M)	UART 9600bps	(1) 支持 AD 按键 5 路一对一播放 (5 个按键分别对应 5 首曲目) (2) 支持组合播放、选曲播放、路径播放、曲目插播、路径插播 (3) 播放模式、当前播放曲目支持掉电记忆
MX6294	SSOP24	播放芯片	U 盘 (最大 32G) TF 卡 (最大 32G) SD 卡 (最大 32G) SPI- (最大 128M)	UART 9600bps	支持中文长文件名功能, 全显汉字
MX6305	SSOP24	播放芯片	U 盘 (最大 32G) TF 卡 (最大 32G) SD 卡 (最大 32G) SPI- (最大 128M)	UART 9600bps	(1) 支持 AD 按键 8 路播放 (2) 支持组合播放、选曲播放、路径播放、曲目插播、路径插播 (3) 支持 SPI-flash 模拟成 U 盘 (4) 缩短接收命令之间延时至 10ms (5) 指定盘符路径插播, 例如 flash 放提醒语音, sd 卡放音乐, 只需指定 flash 插播提醒语音, 无需来回切换存储设备。 (6) 单曲停止等

附录 1：检验和程序

```
/**
 * @功能：将输入数据相加，返回低 8 位
 * @输入参数: *str: 需产生和校验的数据指针
 *           len: 数据的长度
 * @返回值: 检验和低 8 位
 */
u8 SumCheck ( u8 *Str, u8 len )
{
    int xorsum = 0;
    int i;
    for ( i = 0; i < len; i++ )
    {
        xorsum = xorsum + ( *Str++ );
    }
    return ( (u8) ( xorsum & 0x00ff ) );
}
```

常用命令程序

```
#define MX_PLAY           0x020102    // 播放
#define MX_BREAK         0x020103    // 暂停
#define MX_STOP          0x020104    // 停止
#define MX_PREV          0x020105    // 上一曲
#define MX_NEXT          0x020106    // 下一曲
#define SPECIFIED_SONG   0x020307    // 指定歌曲索引播放
#define MX_SEL_DEV       0x02020B    // 选择存储设备

typedef enum {
    MX_USB=0,           //U 盘
    MX_SD,              //sd 卡
    MX_FLASH            //flash
}MX_DevType;

#define MX_PREV_FOLDER   0x02010F    // 前一文件夹
#define MX_NEXT_FOLDER   0x020110    // 下一文件夹
```

```
#define MX_SET_VOL          0x030201    // 设置音量
#define MX_ADD_VOL         0x030102    // 音量加
#define MX_DEC_VOL         0x030103    // 音量减

#define MX_SLEEP           0x0B0101    // 睡眠

#define MX_SET_CYCLICAL    0x050201    // 设置循环模式
enum{
MX_CYCLICAL_ALL      = 0,    //全部循环
MX_CYCLICAL_SINGLE,    //单曲循环
MX_CYCLICAL_SINGLE_STOP ,    //单曲停止
MX_CYCLICAL_RANDOM,    //全盘随机播放
MX_CYCLICAL_FOLDER ,    //文件夹循环
MX_CYCLICAL_FOLDER_RANDOM ,    //文件夹随机
MX_CYCLICAL_FOLDER_ODER,    //文件夹顺序播放
MX_CYCLICAL_ODER      //全盘顺序播放
};

#define MX_SET_EQ          0x060201    // 设置音效
enum{
MX_EQ_NORMAL  =0,        //标准音效
MX_EQ_POP ,            //流行
MX_EQ_ROCK ,          //摇滚
MX_EQ_JAZZ ,          //爵士
MX_EQ_CLASSIC        //古典
};

/**
 * @功能: MX6305 常用命令发送
 * @输入参数: cmd: 命令宏
 *           dat: 命令参数
 * @调用函数: void UART_SendMByte(u8 *pdata, u8 len);
 *           此函数由单片机提供, 把多个字节通过串口发送出去
 *           u8 SumCheck ( u8 *Str, u8 len )
 *           和校验
 */

void MX_6305SendCmd(u32 cmd,u32 dat)
{
    u8 buff[20];
    u8 category=(u8)(cmd>>16); //宏定义命令的第三个字节为命令码
```

```
u8 datalen= (u8)( cmd>>8); //宏定义命令的第二个字节为数据区长度
u8 cmdlist= (u8) cmd; //宏定义命令的第一个字节为命令操作码,属于数据区

buff[0] = category; //命令码
buff[1] = ~category; //命令码反码
buff[2] = datalen; //数据区长度
buff[3] = cmdlist; //子命令
if(datalen>1) //数据区长度>1 说明这个命令带参数
{ //装载参数
    u8 i;
    for(i=datalen-1;i>0;i--)
    { buff[3+i] = (u8)dat;
      dat>>=8;
    }
}
buff[datalen+3]= SumCheck (buff,datalen+3);//校验和
UART_SendMByte(buff,datalen+4); //把缓存区的数据通过串口发送出去, 此函数由底层提供
}
```

使用范例:

```
MX_6305SendCmd(MX_PLAY, NULL); //播放命令
MX_6305SendCmd( MX_SEL_DEV, MX_SD ); //选择 S D 卡
MX_6305SendCmd( SPECIFIED_SONG , 123); //指定播放第 123 首歌
MX_6305SendCmd(MX_SET_VOL, 20); //设置音量为 20
MX_6305SendCmd(MX_SET_CYCLICAL, MX_CYCLICAL_SINGLE_STOP); //设置循环模式为
//单曲停止
MX_6305SendCmd(MX_SET_EQ, MX_EQ_ROCK); //设置音效为摇滚
MX_6305SendCmd(MX_SLEEP, NULL); //低功耗睡眠命令
```

注意:芯片上电 500ms 后方可接收命令,命令之间需要一定的延时, 最短 10ms。

指定路径播放程序

```
#include <string.h>
typedef enum {
    MX_USB=0,
    MX_SD,
    MX_FLASH
}MX_DevType;
```

```
/**
 * @功能: 当前盘符指定路径播放
 * @输入参数: *path: 路径字符串
 *           Dev: 指定存储设备 ( MX_USB, MX_SD, MX_FLASH)
 * @调用函数: void UART_SendMByte(u8 *pdata, u8 len);
 *           此函数由单片机提供, 把多个字节通过串口发送出去
 *           u8 SumCheck ( u8 *Str, u8 len )
 *           和校验
 */

void MX_SpecifiedPath(const char * path, MX_DevType dev )
{
    u8 i;
    char Temp[30]; //注意缓存区不要溢出
    u8 pathlen=strlen(path);//路径的长度
    u8 datalen=pathlen+6;//全部数据的长度, 除路径还有 6 个字节

    Temp[0]=0x02; //命令码
    Temp[1]=~0x02;//命令码反码
    Temp[2]=2+pathlen;//数据区长度
    Temp[3]=0x08;
    Temp[4]=dev ;//指定存储设备
    for(i=0;i<pathlen;i++)//把路径复制到缓存区
    {
        Temp[5+i]=*(path+i);
    }
    Temp[datalen-1] = SumCheck((unsigned char *)Temp,datalen-1);//最后一个字节为校验和
    UART_SendMByte((u8 *)Temp,datalen);//把缓存区的数据通过串口发送出去, 此函数由底层
    提供
}
```

使用范例:

```
//播放 S D 卡中的 ——新建文件夹——m y s o n g ——好久不见.MP3
MX_SpecifiedPath("新建文*/mysong /好久不见 MP3", MX_SD );
```

路径使用短文件名规则, 请参考——短文件名规则。

指定路径插播程序

```
#include <string.h>

typedef enum {
    MX_USB=0,
    MX_SD,
    MX_FLASH
}MX_DeVType;

/**
 * @功能: 指定路径插播, 播放完后返回断点继续播放
 * @输入参数: *path: 路径字符串
               Dev: 指定存储设备 ( MX_USB, MX_SD, MX_FLASH)
 * @调用函数: void UART_SendMByte(u8 *pdata, u8 len);
 *             此函数由单片机提供, 把多个字节通过串口发送出去
 *             u8 SumCheck ( u8 *Str, u8 len )
 *             和校验
 */
void MX_InsertSpecifiedPath(const char * path, MX_DeVType dev)
{
    u8 i;
    char Temp[30]; //注意缓存区不要溢出
    u8 pathlen=strlen(path); //路径的长度
    u8 datalen=pathlen+6; //全部数据的长度, 除路径还有 6 个字节

    Temp[0]=0x04; //命令码
    Temp[1]=~0x04; //命令码反码
    Temp[2]=2+pathlen; //数据区长度
    Temp[3]=0x02;
    Temp[4]=dev;
    for(i=0;i<pathlen;i++) //把路径复制到缓存区
    {
        Temp[5+i]=*(path+i);
    }
    Temp[datalen-1] = SumCheck((unsigned char *)Temp,datalen-1); //最后一个字节为校验和
    UART_SendMByte((u8 *)Temp,datalen); //把缓存区的数据通过串口发送出去, 此函数由底层
    提供
}

```

使用范例:

```
//插播 u 盘卡中的 ——m u s i c ——美芯欢迎词.MP3  
MX_InsertSpecifiedPath("music /美芯欢*MP3", MX_USB );
```

路径使用短文件名规则，请参考——短文件名规则。

组合播放程序

```
/**  
 * @功能: 组合播放  
 * @输入参数: *stab: 组合播放曲目列表数组  
 *           len: 数组长度  
 * @调用函数: void UART_SendMByte(u8 *pdata, u8 len);  
 *           此函数由单片机提供，把多个字节通过串口发送出去  
 *           u8 SumCheck ( u8 *Str, u8 len )  
 *           和校验  
 */  
  
void MX_CombinPlay(const u16 *stab,u8 len)  
{  
    u8 i;  
    u8 buff[30]; //注意缓存区不要溢出  
    buff[0] = 0x07; //命令码  
    buff[1] = ~0x07; //命令码反码  
    buff[2] = len+1; //数据区长度  
    buff[3] = 0x01; //子命令属于数据区  
    for(i=0;i<len;i+=2) //装载组合信息  
    {  
        buff[4+i] =(*stab)>>8;//高位  
        buff[5+i] = (u8)*stab++;//低位  
    }  
    buff[4+len] = SumCheck((unsigned char *)buff,4+len);//校验和  
  
    UART_SendMByte((u8 *)buff,5+len);//通过串口把缓冲区发出去  
}
```

使用范例:

```
//组合播放第 1, 35, 64, 12, 20 曲目  
const u16 myCombinSong[]={1,35,64,12,20};
```

```
MX_CombinPlay(myCombinSong,sizeof(myCombinSong));
```

短文件名规则

短文件名不超过 8 个字节，1 个 ASCII 字符占 1 个字节，1 个中文字符占 2 个字节。也就是说文件名不超过 8 个 ASCII 字符，4 个中文字符。

文件名超过 8 个字节处理：

使用通配符 *（星号）用来模糊搜索文件，*表示代替一个或多个真正的字符，例如：在存储设备里歌曲《有没有那么一首歌会让你想起我.MP3》，指定文件名使用通配符为《有没有*MP3》

文件名少于 8 个字节处理：

不足部分使用‘ ’空格符号填充。例如：/周华健 /难念的经 MP3，有一个文件夹名为“周华健”3 个中文占 6 个字节，不足 8 个字节，填充两个空格。注意要使用英文标点的空格符。

文件夹名与文件名规则一样。

文件系统和曲目序号排列说明

本文件系统只支持 FAT32 和 FAT16，支持 sd 卡 U 盘和 sip-flash 三种存储设备播放，文件路径深度最大支持 9 级。

曲目编号规则：

总体按目录编号，设备根目录为第 0 级目录，根目录下的文件夹为第 1 级目录，文件夹的子文件夹按对应的目录增加，在曲目编号是先按目录编号，所以会从根目录开始编号，不管根目录的歌曲是什么时候拷贝进去，文件夹及子文件夹也是按同一目录编号，当前目录下所有子目录全部编号完成后才继续另外的目录编号，目录刚按建立的先后顺序排列，录音文件里的文件会放在最后来编号，不管里面的文件是什么时候建立的。



各级目录之间编号关系:

所有编号是按目录编号的，统计目录编号按目录建立的先后顺序编号，比如有 A 文件夹和 b 文件夹，A 文件夹先建立，则 A 文件夹及其子文件夹的编号都会限于 b 文件夹内的编号，不管其文件夹是什么时候建立的。子目录的编号会接着父目录继续编号，比如第二级目录会接着第一级目录编号，比如 a 文件夹里有 c 和 d 两文件夹，c 先 d 建立，系统在 a 目录下的歌曲编号完成后去处理 c、文件夹里的歌曲编号，c 文件夹编号完成后才回去编号 d 文件夹里的编号，如果在后期 c 文件夹里的个歌曲增加了，增加的文件还是会先于 d 文件夹里的歌曲编号。