

目 录

1. 概述	3
1.1 简介	3
1.2 功能	3
1.3 应用	3
2. 模块芯片使用说明	4
2.1 硬件参数	4
2.2 管脚说明	5
3. 串口通讯协议	6
3.1 通讯格式	6
3.2 通讯指令	7
3.3 芯片返回的数据	9
3.3.1 芯片上电返回的数据	9
3.3.2 曲目播放完毕返回的数据	10
3.3.3 芯片应答返回的数据	10
3.3.4 芯片错误返回的数据	11
3.3.5 设备插入拔出消息	11
3.4 串口指令详解	12
3.4.1 指定歌曲播放指令	12
3.4.2 指定音量播放指令	13
3.4.3 指定播放设备	13
3.4.4 指定文件夹播放	13
3.4.5 指定 MP3 文件夹中的曲目播放	14
3.4.6 插播 ADVERT 文件夹下的广告	15
3.4.7 单个文件夹支持 1000 首曲目	16
3.4.8 全部循环播放指令	17
3.4.9 单曲循环播放指令	17
3.4.10 播放状态查询指令	17
3.4.11 播放停止指令	17
3.4.12 指定文件夹循环播放	18
3.4.13 随机播放设备文件	18
3.4.14 对当前的曲目设置为循环播放	18
3.4.15 开启和关闭 DAC	18
3.4.16 指定文件夹曲目数查询	19
3.4.17 当前设备的总文件夹数目查询	19
3.4.18 多文件夹插播功能	19

3.5 按键接口	20
4、参考电路	21
4.1 串行接口	21
4.2 外接单声道功放	22
4.3 外接耳机电路	22
5、PIO 的特性	23
5.1 中的注意点	23
5.2 串口操作	24
5.2.1 串口操作流程	25
5.2.2 串口编程参考的说明	25
5.2.3 串口编程需要适当延时的注意点	25
6、免责声明	26

1. 概述

1.1 简介

YX6200-16S 是一个提供串口的语音芯片，完美的集成了 MP3、WAV、WMA 的硬解码。同时软件支持 TF 卡驱动，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。

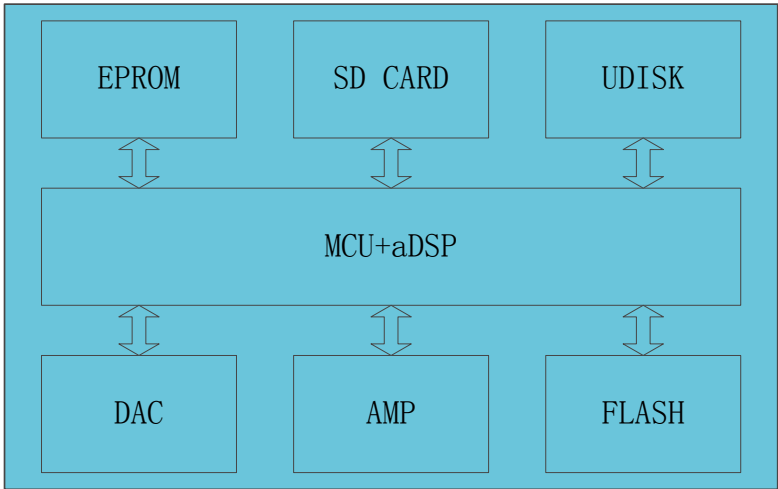
1.2 功能

- 1、支持采样率(KHz):8/11.025/12/16/22.05/24/32/44.1/48
- 2、24 位 DAC 输出，动态范围支持 90dB，信噪比支持 85dB
- 3、完全支持 FAT16、FAT32 文件系统，最大支持 32G 的 TF 卡，支持 32G 的 U 盘、
- 4、多种控制模式可选。IO 控制模式、串口模式、AD 按键控制模式
- 5、广播语插播功能，可以暂停正在播放的背景音乐。广告播放完毕回到背景音继续播放
- 6、音频数据按文件夹排序，最多支持 100 个文件夹，每隔文件夹可以分配 255 首曲目
- 7、30 级音量可调，6 种 EQ 可选

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. 模块芯片使用说明

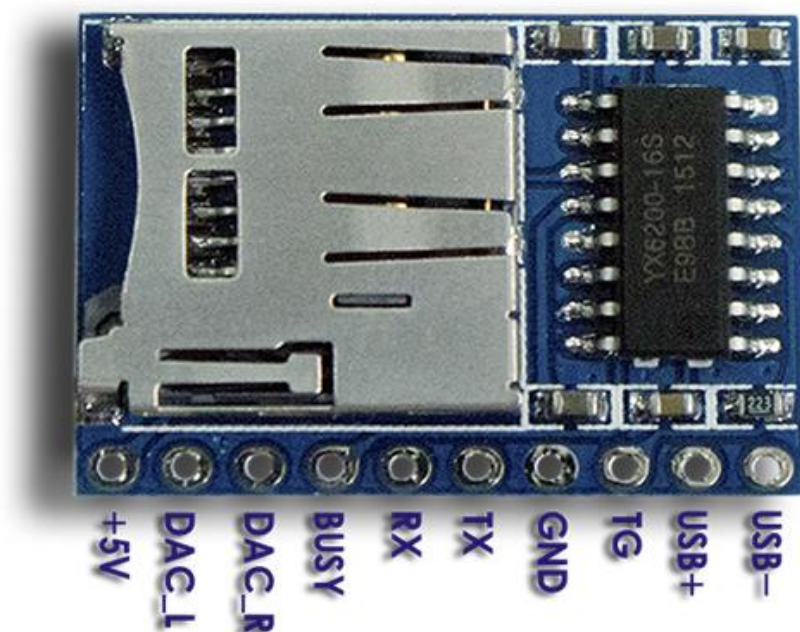


芯片选用的是 SOC 方案，集成了一个 16 位的 MCU，以及一个专门针对音频解码的 aDSP，采用硬解码的方式，更加保证了系统的稳定性和音质。小巧的封装尺寸更加满足嵌入其它产品的需求

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 IS013813-3 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 等音效
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	3.3V-5V
额定电流	15ma[不带 U 盘]
尺寸	DIP10封装
工作温度	-40度 -- 80度
湿度	5% ~ 95%

2.2 管脚说明



引脚序号	引脚名称	功能描述	备注
1	VDD	5V 电源输入	不可以超过5.2V
2	DAC-L	音频输出左声道	驱动耳机、功放
3	DAC-R	音频输出右声道	驱动耳机、功放
4	BUSY	Busy 输出	播放时输出低电平
5	RX	UART 串行数据输入	3.3V 的 TTL 电平
6	TX	UART 串行数据输出	3.3V 的 TTL 电平
7	GND	电源地/音频地	
8	ADKEY1	触发输出口	
9	USB+	USB+ DP	电脑的 USB 口或者 U 盘
10	USB-	USB- DM	电脑的 USB 口或者 U 盘

3. 串口通讯协议

串口作为一种在控制领域常用的通信，我们进行了工业级别的优化，加入的帧的校验、重发、错误处理等措施，大大加强通信的稳定性和可靠性，同时可以在此基础上扩展更加强大的 RS485 进行组网功能，串口的通信波特率可自行设置，默认为 9600

3.1 通讯格式

支持异步串口通讯模式, 通过串口接受上位机发送的命令

通讯标准:9600 bps
数据位 :1
校验位 :none
流控制 :none

格式: \$S VER Len CMD Feedback para1 para2 checksum \$0		
\$S	起始位0x7E	每条命令反馈均以\$开头, 即0x7E
VER	版本	版本信息[目前默认为0xff]
Len	len 后字节个数	校验和不计算在内
CMD	命令字	表示具体的操作, 比如播放/暂停等等
Feedback	命令反馈	是否需要反馈信息, 1反馈, 0不反馈
para1	参数1	查询的数据高字节(比如歌曲序号)
para2	参数2	查询的数据低字节
checksum	校验和[占两个字节]	累加和校验[不计起始位\$]
\$0	结束位	结束位0xEF

举个例子，如果我们发下一曲指令，就需要发送: 7E FF 06 01 00 00 00 FE FA EF
数据长度为 6 ,这 6 个字节分别是[FF 06 01 00 00 00] 。不计算起始、结束、和校验。再然后对得到的结果进去反+1

3.2 通讯指令

1、直接发送的指令，不需要返回参数

CMD 命令(指令)	对应的功能	参数(16位)
0x01	下一曲	
0x02	上一曲	
0x03	指定曲目(NUM)	1-2999
0x04	音量+	
0x05	音量-	
0x06	指定音量	0-30
0x07	指定 EQ0/1/2/3/4/5	Normal/Pop/Rock/Jazz/Classic/Bass
0x08	单曲循环指定曲目播放	0-2999
0x09	指定播放设1/2/3/4/5	U/SD/AUX/SLEEP/FLASH
0x0A	进入睡眠 -- 低功耗	
0x0B	保留	
0x0C	芯片复位	
0x0D	播放	
0x0E	暂停	
0x0F	指定文件夹播放	1-10(需要自己设定)
0x10	扩音设置(无)	[DH=1:开扩音] [DL:设置增益0-31]
0x11	全部循环播放	[1:循环播放][0:停止循环播放]
0x12	指定 MP3文件夹曲目	0--9999
0x13	插播广告	0--9999
0x14	支持15个文件夹	详见下面的详细说明
0x15	停止插播，播放背景	
0x16	停止播放	
0x17	指定文件夹循环播放	详见3.4.13
0x18	随机播放	详见3.4.14
0x19	循环播放	详见3.4.15
0x1A	开启和关闭 DAC	详见3.4.16
0x25	多文件夹插播	详见3.4.19

2、查询系统的参数

CMD 命令详解(查询)	对应的功能	参数(16位)
0x3C	STAY	
0x3D	STAY	
0x3E	STAY	
0x3F	发送初始化参数	0 - 0x0F(低四位每位代表一种设备)
0x40	返回错误, 请求重发	
0x41	应答	
0x42	查询当前状态	
0x43	查询当前音量	
0x44	查询当前 EQ	
0x45	查询当前播放模式	该版本保留此功能
0x46	查询当前软件版本	该版本保留此功能
0x47	查询 UDISK 的总文件数	
0x48	查询 TF 卡的总文件数	
0x49	保留	
0x4A	保留	
0x4B	查询 UDISK 的当前曲目	
0x4C	查询 TF 卡的当前曲目	
0x4D	保留	
0x4A	保留	
0x4B	查询 UDISK 的当前曲目	
0x4C	查询 TF 卡的当前曲目	
0x4D	查询 FLASH 的当前曲目	
0x4E	查询文件夹里的文件总数	
0x4F	查询总文件数	

3.3 芯片返回的数据

芯片在关键地方均会有数据返回。供用户掌控芯片的工作状态

- 芯片上电初始化成功的数据
- 芯片播放完当前曲目的数据
- 芯片成功接收到指令返回的 ACK(应答)
- 芯片接收一帧数据出错[包括数据没收完整、校验出错两种情况]
- 芯片在繁忙时，有数据过来，芯片会返回忙的指令
- U 盘、TF 卡插入拔出，均有数据返回

3.3.1 芯片上电返回的数据

(1)、芯片上电，需要一定的时间初始化，这个时间是需要根据 U 盘、TF 卡等设备的文件多少决定的，一般在 1.5 ~ 3S 这个时间。如果超过这个时间芯片的初始化数据还没有发送出来，说明芯片初始化出错，请复位芯片的电源，另外检测硬件的连接

(2)、芯片初始化数据包括在线设备,譬如发送 7E FF 06 3F 00 00 01 xx xx EF

DL = 0x01 说明上电过程中，只有 U 盘在线。其它的数据请参照下表,各设备之间是或的关系

U 盘 -- 在线	7E FF 06 3F 00 00 01 xx xx EF	各设备之间是或的关系
TF -- 在线	7E FF 06 3F 00 00 02 xx xx EF	
PC -- 在线	7E FF 06 3F 00 00 04 xx xx EF	
U 盘、TF -- 在线	7E FF 06 3F 00 00 03 xx xx EF	

(3)、MCU 必须等待芯片初始化指令发出之后才能发送相应的控制指令，否则发送的指令芯片将不予处理。同时还会影响芯片的正常初始化。

3.3.2 曲目播放完毕返回的数据

U 盘播放完第1曲	7E FF 06 3C 00 00 01 xx xx EF	U 盘播放第1曲完毕
U 盘播放完第2曲	7E FF 06 3C 00 00 02 xx xx EF	U 盘播放第2曲完毕
TF 卡播放完第1曲	7E FF 06 3D 00 00 01 xx xx EF	TF 卡播放第1曲完毕
TF 卡播放完第2曲	7E FF 06 3D 00 00 02 xx xx EF	TF 卡播放第2曲完毕

1、针对很多的触发型的播放需求，我们芯片更正为播放一曲之后自动进入停止状态。如果用户需要此类应用。只需要指定曲目播放即可。这样，曲目播放完毕会自动停止，等待指令

2、另外我们专门开辟一个 IO 作为解码和停止的状态指示。请参见 busy 脚，

(1)、播放状态输出低电平[很多功放有静音脚，可以通过此 IO 直接进行控制]

(2)、播放暂停状态，输出高电平。芯片睡眠状态。也是高电平

3、争对连续播放应用，可以这样实现。假如 U 盘播放第一首歌完毕之后，会返回

7E FF 06 3C 00 00 01 xx xx EF

3C ---- 表示的是 U 盘命令

00 01 ---- 表示播放完毕的曲目。此时再发送播放下一曲的指令，即可顺序循环播放

4、芯片通电之后，初始化正常，芯片会自动进入设备播放状态。并且停止解码，等待用户发送播放的相关指令

5、另外用户在指定设备之后，需要等待 200ms 的时间，再发送指定的曲目，因为一旦指定曲目之后，系统会对指定的设备进行文件系统的初始化，如果立刻发送指定的曲目命令，会导致芯片接收不到。

3.3.3 芯片应答返回的数据

U 盘播放完第3曲	7E FF 06 3C 00 00 03 xx xx EF	U 盘播放第3曲完毕
-----------	-------------------------------	------------

(1)、为了加强数据通信之间的稳定性，我们增加了应答处理，ACKB 字节就是设置是否需要回复应答。这样做的好处是保证每次通信都有握手信号，收到应答就表示 MCU 发送的数据，芯片已经成功收到，马上处理。

(2)、对于一般的应用，客户可以自由选择，不加这个应答处理也是可以的。

3.3.4 芯片错误返回的数据

返回忙	7E FF 06 40 00 00 01 xx xx EF	芯片在文件系统初始化时
当前处于睡眠模式	7E FF 06 40 00 00 02 xx xx EF	睡眠模式只支持指定设备
串口接收错误	7E FF 06 40 00 00 03 xx xx EF	串口一帧数据没接收完毕
校验出错	7E FF 06 40 00 00 04 xx xx EF	和校验出错
指定文件超范围	7E FF 06 40 00 00 05 xx xx EF	文件的指定超过设定的范围
未找到指定的文件	7E FF 06 40 00 00 06 xx xx EF	指定为文件没有被找到
插播指令错误	7E FF 06 40 00 00 07 xx xx EF	当前状态不接受插播

(1)、为了加强数据通信之间的稳定性，我们增加了数据错误处理机制。芯片收到不符合格式的数据，均会有信息反馈出来

(2)、在环境比较恶劣的情况下，强烈建议客户处理此命令。如果应用环境一般，可以不用处理。

(3)、芯片返回忙，基本上是芯片上电初始化的时候才会返回，因为芯片需要初始化文件系统

(4)、芯片上电之后，进入的是设备状态，设备的顺序是 U 盘--TF 卡。如果所有设备都不在线，芯片会进入睡眠状态

(5)、只要参考我们给出的测试 SDK 程序，移植里面的串口操作部分，就不会出现校验出错，在这里强烈建议用户使用我们给出的校验方式。因为谁都不能保证数据的传输不会出错。

(6)、文件指定部分出错，请参考下面的“指定文件名播放详细说明”

3.3.5 设备插入拔出消息

U 盘插入	7E FF 06 3A 00 00 01 xx xx EF	
TF 插入	7E FF 06 3A 00 00 02 xx xx EF	
PC 插入	7E FF 06 3A 00 00 04 xx xx EF	
U 盘拔出	7E FF 06 3B 00 00 01 xx xx EF	
TF 拔出	7E FF 06 3B 00 00 02 xx xx EF	
PC 拔出	7E FF 06 3B 00 00 04 xx xx EF	

(1)、为了加强芯片的灵活性，我们特别增加了，设备插入、拔出的指令反馈。方便用户知道芯片的工作状态。

(2)、设备插入的时候，我们默认进入到设备等待状态，如果用户插入的是带灯的 U 盘，可以看到 U 盘灯闪烁。也可以接收到设备插入的串口消息。

3.4 串口指令详解

以下我们对关键的地方进行详细的说明：

- 指定曲目播放[针对按照存储的物理顺序播放]
- 指定播放的音量
- 指定播放的设备
- 指定文件夹播放[这里有多种方式，参见下面的详细说明]
- 全部循环播放指令

3.4.1 指定歌曲播放指令

我们给出的指令是支持指定曲目播放的，歌曲的选择范围为 0~2999.其实是可以支持更多的，因为涉及到文件系统的原因，支持过多的歌曲，会导致系统操作缓慢，一般的应用也不需要支持这么多的文件。如果客户有非常规的应用，请事前和我们沟通。

(1)、例如选择第一首歌播放，串口的发送部分 7E 10 06 03 00 00 01 FF E6 EF

7E --- 起始命令

FF --- 版本信息

06 --- 数据长度(不包含校验)

03 --- 代表命令字节

00 --- 是否需要应答[0x01:需要应答，0x00:不需要返回应答]

00 --- 曲目的高字节[DH]

01 --- 曲目的低字节[DL],这里代表的是第一首歌播放

FF --- 校验的高字节

E6 --- 校验的低字节

EF --- 结束命令

(2)、对于选曲，如果选择第 100 首，首先将 100 转化为 16 进制,默认为双字节,就为 0x0064。

DH = 0x00 ; DL = 0x64

(3)、如果选择第 1000 首进行播放，首先将 1000 转化为 16 进制,默认为双字节,就为 0x03E8

DH = 0x03 ; DL = 0xE8

(4)、其它的操作依次类推即可，因为在嵌入式领域采用 16 进制是最为方便的一种操作。

3.4.2 指定音量播放指令

- (1)、我们系统上电默认的音量为 30 级，如果要设置音量的话,直接发送相应的指令即可
- (2)、例如指定音量为 15 级,串口发送的指令:7E FF 06 06 00 00 0F FF D5 EF
- (3)、DH = 0x00 ; DL = 0x0F ， 15 转化为 16 进制为 0x000F。可以参照播放曲目部分的说明

3.4.3 指定播放设备

- (1)、我们的芯片默认是支持 4 种类型的播放设备,只有设备在线才能指定设备去播放
设备是否在线，我们软件会自动检测，无需用户关系。
- (2)、看下表，选择合适的指令发送
- (3)、指定设备之后。芯片会自动进入停止解码状态，等待用户指定曲目播放。从接收到指定设备到芯片内部完成初始化文件系统。大概需要 200ms。请等待 200ms 之后再发送指定曲目的指令。

指定播放设备-U 盘	7E FF 06 09 00 00 01 xx xx EF	xx xx: 代表校验
指定播放设备-SD 盘	7E FF 06 09 00 00 02 xx xx EF	
指定播放设备-AUX	7E FF 06 09 00 00 03 xx xx EF	
指定播放设备-PC	7E FF 06 09 00 00 05 xx xx EF	指[读卡]模式
指定播放设备-SLEEP	7E FF 06 09 00 00 06 xx xx EF	

3.4.4 指定文件夹播放

指定文件夹01里面的001xxx.mp3	7E FF 06 0F 00 01 01 xx xx EF
指定文件夹11里面的100xxx.mp3	7E FF 06 0F 00 0B 64 xx xx EF
指定文件夹99里面的255xxx.mp3	7E FF 06 0F 00 63 FF xx xx EF

- (1)、指定文件夹播放是我们制定的扩展功能，默认文件夹的命名方式为"01","11"这样的方式因为我们的芯片不支持汉字名称的文件夹名称识别，为了系统的稳定性和歌曲切换的速度，每个文件夹下默认最大支持 255 首歌,最多支持 99 个文件夹的分类，如果客户有特殊要求，需要按照英文名称来分类，我们也是可以实现的，但是名称只能是“GUSHI”、“ERGE”等英文名称组成。但是 mp3 文件是需要增加前缀的，可以在“不得不爱.mp3”基础上改成“002 不得不爱.mp3”。

- (2)、例如指定"01"文件夹的 100xxx.MP3 文件,串口发送的指令为:7E FF 06 0F 00 01 64 xx xx EF

DH:代表的是文件夹的名字,默认支持 99 个文件,即 01 -- 99 的命名

DL:代表的是曲目,默认最多 255 首歌，即 0x01 ~ 0xFF

- (3)、为了芯片的标准性，必须同时指定文件夹和文件名，来锁定一个文件。单独指定文件夹或者单独指定文件名也是可以的，但是这样文件的管理会变差。指定文件夹和指定曲目是支持 MP3、WAV

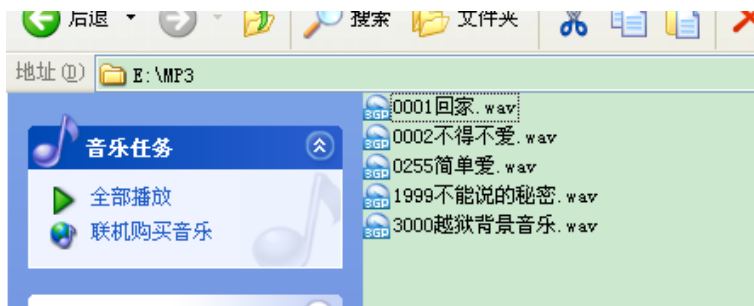
- (4)、下面截两个图说明文件夹和文件名的指定[分左右两个图]



3.4.5 指定 MP3 文件夹中的曲目播放

指定 MP3文件夹	7E FF 06 12 00 00 01 FE E8 EF	"MP3"文件夹，曲目为"0001"
	7E FF 06 12 00 00 02 FE E7 EF	"MP3"文件夹，曲目为"0002"
	7E FF 06 12 00 00 FF FD EA EF	"MP3"文件夹，曲目为"0255"
	7E FF 06 12 00 07 CF FE 13 EF	"MP3"文件夹，曲目为"1999"
	7E FF 06 12 00 0B B8 FE 26 EF	"MP3"文件夹，曲目为"3000"

- (1)、在指定文件夹和文件名的基础上，我们扩展单个文件夹的功能，文件夹的命名必须为“MP3”
- (2)、最多支持 65536 首曲目，但是鉴于文件系统的操作速度，可能会随着文件的增大，曲目切换的速度会相应的减慢。
- (3)、指定的文件命名如下：



3.4.6 插播 ADVERT 文件夹下的广告

插播广告	7E FF 06 13 00 00 01 FE E7 EF	"ADVERT"文件夹, 曲目为"0001"
	7E FF 06 13 00 00 02 FE E6 EF	"ADVERT"文件夹, 曲目为"0002"
	7E FF 06 13 00 00 FF FD E9 EF	"ADVERT"文件夹, 曲目为"0255"
	7E FF 06 13 00 07 CF FE 12 EF	"ADVERT"文件夹, 曲目为"1999"
	7E FF 06 13 00 0B B8 FE 25 EF	"ADVERT"文件夹, 曲目为"3000"

(1)、我们支持选曲播放过程中插播其它的曲目进行播放, 这样就可以满足需要在背景音乐播放的过程中插播一下广告的需求。

(2)、发送 0x13 指令之后, 系统会存储当前正在播放的曲目的 IDV3 信息, 然后播放指定的插播曲目。等到插播的曲目播放完毕之后, 系统会回到保存的那个播放断点处继续播放的。直至播放完毕。

(3)、设置的格式是, 在设备中建立一个“ADVERT”名字的文件夹, 在里面存储需要插播的曲目即可, 曲目设置为“0xxx+曲目名.MP3/WAV”

(4)、另外系统如果当前处于暂停或者停止状态, 发送插播指令, 均得不到响应, 并且会有错误信息返回。如果在插播的过程中, 可以继续插播其它曲目, 但是播放完毕之后, 还是回到第一次存储的 IDV3 信息处。

(5)、指定插播广告设置如下:



3.4.7 单个文件夹支持 1000 首曲目

支持1000首	7E FF 06 14 00 10 FF FD D8 EF	指定为“01”的文件夹，曲目为“0255”
	7E FF 06 14 00 17 CF FE 01 EF	指定为“01”的文件夹，曲目为“1999”
	7E FF 06 14 00 C0 01 FE 26 EF	指定为“12”的文件夹，曲目为“0001”
	7E FF 06 14 00 C0 FF FD 28 EF	指定为“12”的文件夹，曲目为“0255”
	7E FF 06 14 00 C7 CF FD 51 EF	指定为“12”的文件夹，曲目为“1999”

针对很多客户要求 10 个文件夹，每个文件夹能管理 1000 首曲目的要求，我们特别增加此条指令工
用户调用，详细的说明如下：

(1)、串口的命令字节为 0x14

(2)、参数为两个字节，假如”指定为“12”的文件夹，曲目为“1999”

串口的数据:7E FF 06 14 00 C7 CF FD 51 EF

其中 0xC7 和 0xCF 为参数，组合起来就是 0xC7CF.总共为 16 位

其中高 4 位代表文件夹的命名，这里的 C 表示的是 12

其中低 12 位表示文件名的命名，这里 7CF 代表是 1999，也就是文件前缀为“1999”的曲目

(3)、文件夹的命名如下：



3.4.8 全部循环播放指令

0x11	循环播放	[1:循环播放][0:停止循环播放]
------	------	--------------------

(1)、争对一些需要循环播放根目录下面曲目的要求，我们加多这一条控制指令 0x11。

循环播放开始	7E FF 06 11 00 00 01 xx xx EF	循环播放所有曲目
循环播放停止	7E FF 06 11 00 00 00 xx xx EF	停止循环播放曲目

(2)、在循环播放的过程中，可以正常的操作播放/暂停，上一曲、下一曲、音量调节，包括 EQ 等等

(3)、循环播放开始之后，芯片会不停的播放设备里面的曲目，按照存储的物理顺序。播完一遍之后会继续再播放一边，直到接收到播放完成，或者暂停等等指令

3.4.9 单曲循环播放指令

0x08	单曲循环指定曲目播放	0-2999
------	------------	--------

循环播放开始	7E FF 06 08 00 00 01 xx xx EF	循环播放第一曲
循环播放停止	7E FF 06 08 00 00 02 xx xx EF	循环播放第二曲

(1)、争对一些需要单曲循环播放的要求，我们改进这一条控制指令 0x08。

(2)、在循环播放的过程中，可以正常的操作播放/暂停，上一曲、下一曲、音量调节，包括 EQ 等等并且状态仍然是循环播放.可以通过指定单曲触发播放或者进入睡眠来关闭循环播放状。

3.4.10 播放状态查询指令

正在播放	7E FF 06 42 00 00 01 xx xx EF	正在播放
暂停播放	7E FF 06 42 00 00 02 xx xx EF	播放过程中被暂停
停止播放	7E FF 06 42 00 00 00 xx xx EF	播放完毕
当前在睡眠状态	7E FF 06 42 00 00 08 xx xx EF	无设备在线或者被指定睡眠

(1)、芯片在解码过程中会有四种状态对用户开放。用户可以通过指令查询获取芯片的当前状态

(2)、播放暂停是指，正在播放一首曲目，人为的发送指令暂停播放，
播放停止是指，一首曲目播放完毕，芯片就处于播放停止的状态

3.4.11 播放停止指令

停止播放广告	7E FF 06 15 00 00 00 FE E6	停止当广告，回到背景音乐继续
--------	----------------------------	----------------

	EF	播
停止播放	7E FF 06 16 00 00 00 FE E5 EF	停止软件解码

(1)、在模块的播放过程中，我们有两种停止方式，一种是停止当前的插播广告，回到当前断点处继续播放背景音乐。另一种是停止所有的播放，包括背景音乐

(2)、假如当前在播放插播广告，这时发送停止指令 0x16，芯片会停止所有播放任务

3.4.12 指定文件夹循环播放

指定文件夹循环播放	7E FF 06 17 00 00 02 FE E2 EF	指定02文件夹循环播放
	7E FF 06 17 00 00 01 FE E3 EF	指定01文件夹循环播放

(1)、文件夹的命名方式必须是”01” --- “99”。不可以超过 99

(2)、指定文件夹之后，就会在指定的文件夹内部循环播放，不会停止，除非发送停止指令等等

3.4.13 随机播放设备文件

随机播放	7E FF 06 18 00 00 02 FE E3 EF	整个设备的随机播放
------	-------------------------------	-----------

(1)、此指令时随机播放设备里面存储的语音文件，是按照物理顺序随机播放，不分设备里面是否带有文件夹。并且播放的第一个语音文件必须是设备里面的第一个语音文件

3.4.14 对当前的曲目设置为循环播放

指定文件循环播放	7E FF 06 19 00 00 00 FE E2 EF	单曲循环播放开启
	7E FF 06 19 00 00 01 FE E1 EF	单曲循环播放关闭

(1)、在播放的过程中发送此指令，会循环播放当前的曲目。如果当前是处理暂停或者停止状态，则芯片不会响应此指令

(2)、如果要关闭单曲循环播放，发送关闭的指令即可，这样会把当前的曲目播放完毕之后，就停止。

3.4.15 开启和关闭 DAC

设置 DAC	7E FF 06 1A 00 00 00 FE E1 EF	开 DAC
	7E FF 06 1A 00 00 01 FE E0 EF	关 DAC[高阻]

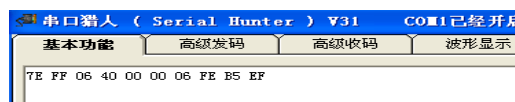
(1)、在一些用户需要叠加自己音源的场合，可以先暂停当前播放的语音，再将我们芯片的 DAC 输出设置为高阻，这样用户就可以一个功放来播放自己的音源了，但是 DAC 的开启和关闭，会有一声 po 音，请用户朋友们注意。

、芯片任何时候都可以关闭 DAC。如果当前正在播放语音，关闭了 DAC，芯片还会继续播放，只是没有声音而已了。芯片上电之后是默认开启 DAC 的，只有被设置为关闭之后，才会被关闭。如果再需要打开，就需要通过指令打开 DAC 了

3.4.16 指定文件夹曲目数查询

查询文件夹曲目总数	7E FF 06 4E 00 00 01 FE AC EF	查询01文件夹的总曲目数
	7E FF 06 4E 00 00 0B FE A2 EF	查询11文件夹的总曲目数

(1)、如果用户按照我们设定的规则命名文件，“01”、“02”等等，这样就可以对这些文件夹里面的曲目总数进行查询。查询的有效文件包括 MP3、WAV。其它格式的文件忽视。



显示查询文件夹出错

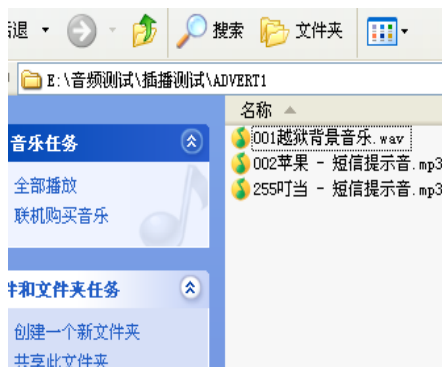
3.4.17 当前设备的总文件夹数目查询

查询文件夹总数	7E FF 06 4F 00 00 00 FE AC EF	查询当前设备的文件夹总数
---------	-------------------------------	--------------

(1)、用户可以对当前的设备进行文件夹总数的查询。我们只支持“根目录”下的文件夹的数目查询。不支持文件夹里面包含文件夹。另外请用户不要建立空的文件夹。

3.4.18 多文件夹插播功能

插播广告-多文件夹	7E FF 06 25 00 01 01 FE D4 EF	“ADVERT1”的文件夹，曲目为“001”
	7E FF 06 25 00 01 02 FE D3 EF	“ADVERT1”的文件夹，曲目为“002”
	7E FF 06 25 00 02 01 FE D3 EF	“ADVERT2”的文件夹，曲目为“001”



(1)、在之前我们的插播基础上，我们增加了多文件夹的插播功能，命名的方式如上图

(2)、文件夹最多支持 9 个，也就是从 ADVERT1 --- ADVERT9，文件夹的命令请一定按照我们给出的规则，否则会导致出错。请注意上图的文件夹命名格式

(3)、单个插播文件夹下的文件最大不能超过 255，也就是“255xxx.MP3/WAV”，请注意上图的文件名的命名格式。

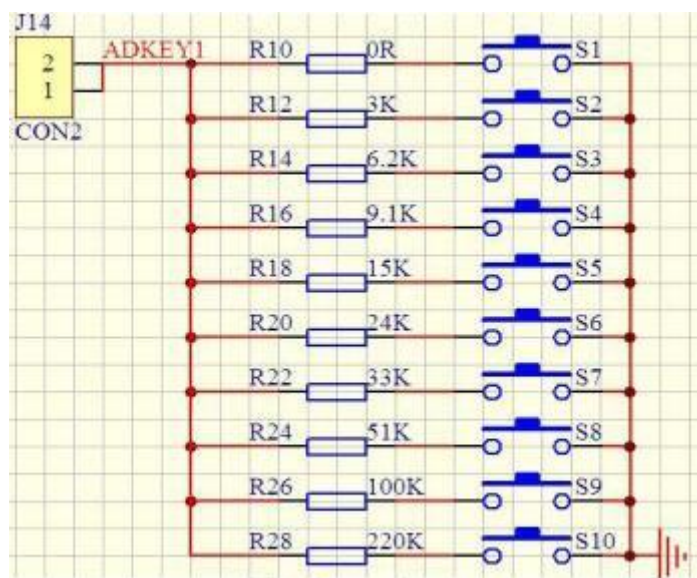
(4)、完善了插播的相关应用，如当前播放的文件处于单曲循环或者当前文件夹循环，即使有插播进来，也不会改变当前的状态，还会是单曲循环或者文件夹循环播放，除非用户使用停止指令或者其他的。

(5)、插播的原理请参见 3.4.6.

3.5 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 1 个 AD 口，10 个按键的阻值分配，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)、参考原理图



(2)、20 个按键的功能分配表

按键	阻值【22K 上拉】	短按	长按	备注
S1	0R—0V	1[单曲]	无	
S2	3K—0.7V	2[单曲]	无	
S3	6K2—0.7V	3[单曲]	无	
S4	9K1—1V	4[单曲]	无	
S5	15K—1.3V	5[单曲]	无	
S6	24K—1.7V	6[单曲]	无	
S7	33K—2V	7[单曲]	无	
S8	51K—2.3V	8[单曲]	无	

S9	100K—2.7V	9[单曲]	无	
S10	220K—3V	停止	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，让您可以更快的上手体验到该芯片的强大功能

3、串行通信接口，波特率默认 9600，可以根据客户的要求修改

4、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

5、外部单声道功放参考电路

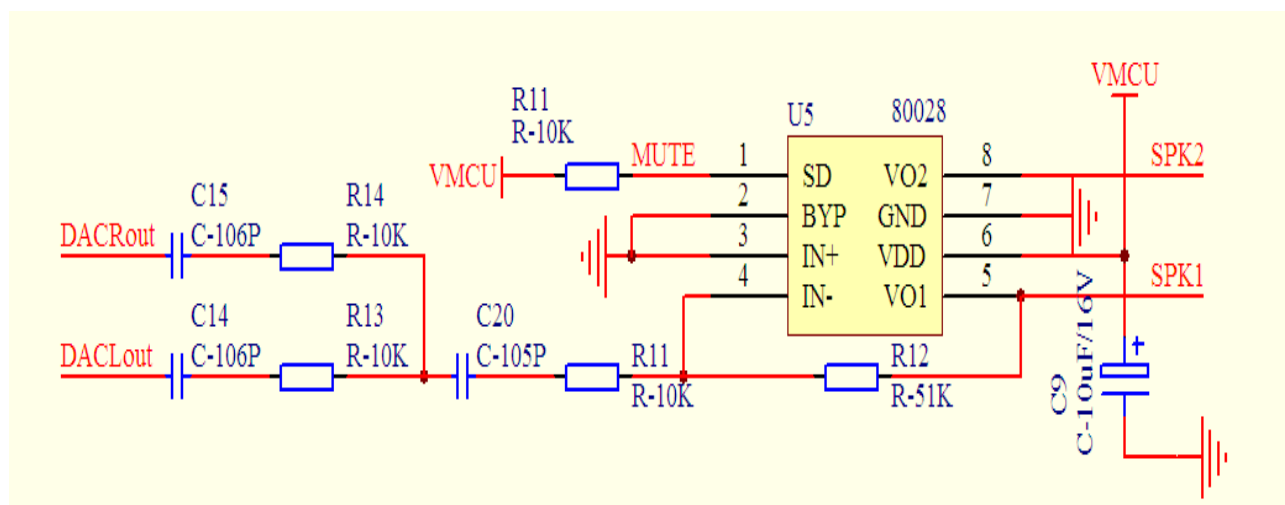
4.1 串行接口



芯片的串口为 3.3V 的 TTL 电平，所以默认的接口的电平为 3.3V。如果系统是 5V。那么建议在串口

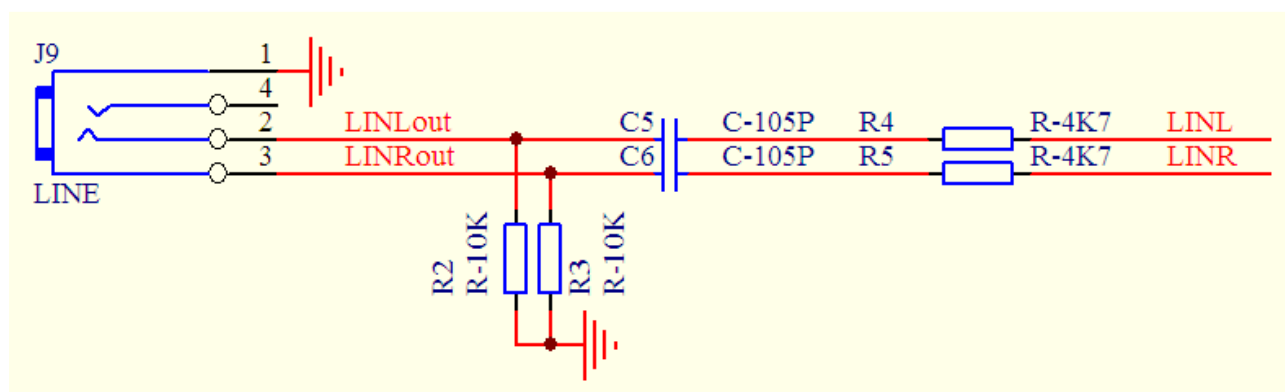
的对接接口串联一个 1K 的电阻。这样足以满足一般的要求，如果应用于强电磁干扰的场合，请参考“注意事项”的说明。芯片在 5V 和 3.3V 的系统中均正常的测试过，一切正常。均在采用的是直连的方式，并没有串 1K 的电阻。

4.2 外接单声道功放



这里功放我们采用的是8002，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 预留电阻给大功率设计用

5、PIO 的特性

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	-	$0.3 \times V_{DD}$	V	$V_{DD}=3.3V$
V_{IH}	High-Level Input Voltage	$0.7V_{DD}$	-	$V_{DD}+0.3$	V	$V_{DD}=3.3V$
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	$V_{DD}=3.3V$
V_{OH}	High-Level Output Voltage	2.7	-	-	V	$V_{DD}=3.3V$

5.1 中的注意点

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

1、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。

6、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、我们支持音频文件的采样率最低为 8KHZ。也就是说低于 8KHZ 的音频文件是不支持的，不能正常解码播放。用户可以使用音频处理软件，提高音频文件的采样率来解决这个问题。

5、芯片在睡眠状态的电流在 12ma 左右，播放 TF 卡，在 15ma 左右。功耗会比较大。如果使用在低功耗场合，请用户控制芯片或者芯片的供电。这样可以减小芯片的功耗

6、该芯片支持 MP3、WAV、WMA 三种主流的音频格式。但是默认发货烧录的软件支持的是 MP3、WAV 这两种格式。如果特别需要支持 WMA 格式的需求，请提前说明

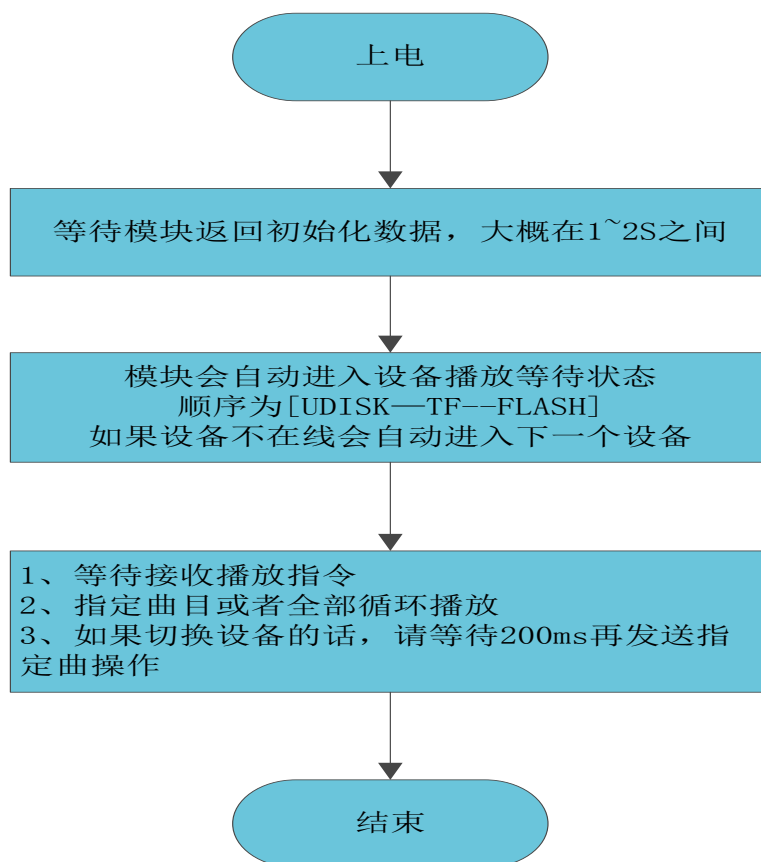
2、我们的芯片支持 8/11.025/12/16/22.05/24/32/44.1/48KHZ 采样率的音频文件，这些也是网络上绝大多数的音频文件的参数。如果用户的音频文件的采样率不在此范围内，是不支持播放的，但是可以通过专用的软件转换一下即可。

5.2 串口操作

串口部分的操作，参见下面的流程，我们提供了完整的参照例程，供用户参考：

- 串口的操作流程
- 串口编程参考的说明
- 串口操作需要延时的注意事项

5.2.1 串口操作流程



1、我司提供的所有芯片和模块的串口部分的操作，均是一样的协议，所以不用担心不同芯片和模块的不兼容

2、如果对串口的操作，有任何不明白的，请一定联系我们，索取串口编程参考例程。

3、我们产品的更新，也一定会按照当前的协议版本，做到向下兼容。

5.2.2 串口编程参考的说明

目前我们提供的串口编程参考代码，有两部分，第一部分是我们测试版的测试代码，相关的串口操作比较全面，另一个是基本版，只是指定曲目的范例。请用户耐心消化

5.2.3 串口编程需要适当延时的注意点

1、芯片上电之后，需要大概 1S-1.5S 时间进行初始化的相关操作，初始化完毕之后，会有初始化的相关数据发送出来。用户也可以直接不理睬这些数据

2、当指定设备播放之后，需要延时 200ms 的时间，再发送指定曲目等等相关指令。

3、因为芯片自带文件系统，正常情况下，在曲目不大于 1000 首的话，响应速度是低于 50ms 的
曲目超过 3000 首之后，文件系统的切换速度会变慢一点，响应速度在 100ms --- 1S 之间不等

6、免责声明

■ 开发预备知识

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

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。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。

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

本公司有保留任何时候在不事先声明的情况下对相关文档的修改权力

■ ESD 静电放点保护

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。