目 录

1. 概述

1.1 简介

YX5300-24SS 是一个提供串口的语音芯片，完美的集成了 MP3、WAV 的硬解码。同时软件支持工业级别的串口通信协议，以 SPIFLASH、TF 卡或者 U 盘作为存储介质，用户可以灵活的选用其中的任何一种设备作为语音的存储介质。通过简单的串口指令即可完成播放指定的语音，以及如何播放语音等功能，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。

无需任何烧录器，无需任何软件，USB 直接烧写 FLASH。

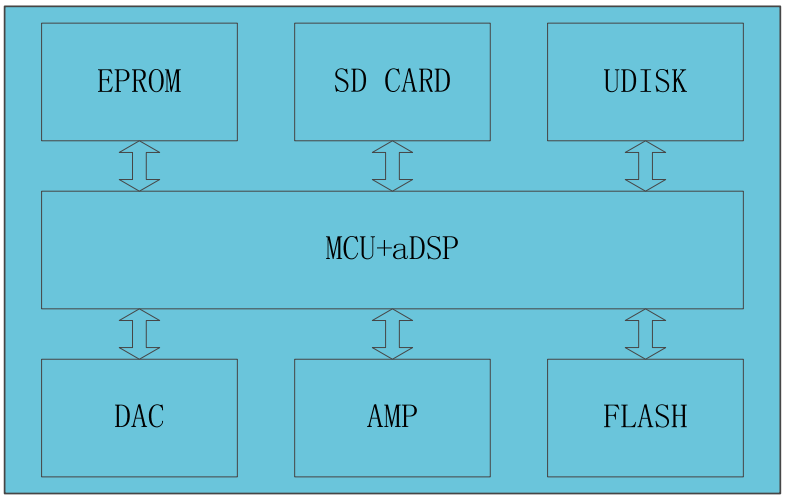
1.2 功能

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、最大支持 16M 字节的 SPIFLASH。例如 W25Q16[2M 字节]、W25Q128[16M 字节]
4、多种控制模式，并口控制模式、串口模式、AD 按键控制模式
5、Miniusb 接口更新语音文件，无需安装任何软件。支持 XP 和 WIN7 系统。
6、支持组合播放功能，可以实现报时、报温度，在一定程度上可以替代一些昂贵的 TTS 方案
7、30 级音量可调，5 级 EQ 可调[此功能暂不开放]
8、自带 3W 的功放，直接外接喇叭即可完成放音
9、支持 8 段语音的触发播放，IO 检测的方式，所以适合碳膜按键等等场合
10、可以同时支持 U 盘、TF 卡以及 SPIFLASH 作为存储介质

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；
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. 方案说明

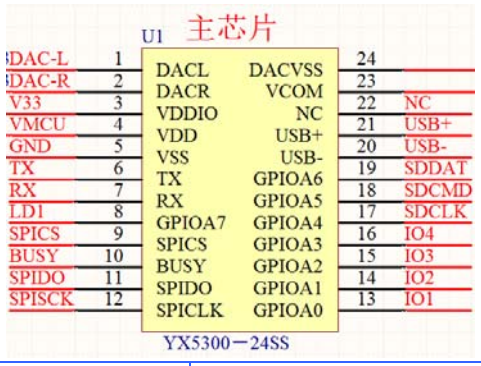


芯片选用的是 SOC 方案，集成了一个 16 位的 MCU，以及一个专门针对音频解码的 aDSP，采用硬解码的方式，更加保证了系统的稳定性和音质。小巧的封装尺寸更加满足嵌入其它产品的需求

2.1 参数说明

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 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[用户不可设]
输入电压	3.3V-5V[7805后级串一个二极管为最佳]
额定电流	10MA[静态]
低功耗电流	<200uA
功率	驱动耳机、功放
尺寸	SSOP24 [单位:mm]
工作温度	-40度 -- 80度
湿度	5% ~ 95%
主芯片型号	YX5300-24SS [SSOP24]

2.2 管脚说明



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 SPI flsah 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	3.3V 的 TTL 电平
7	RX	UART 串行数据输入	3.3V 的 TTL 电平
8	GDPIA7	播放指示灯	建议接三极管驱动
9	SPICS	SPI_CS 片选总线	
10	BUSY	Busy 输出	播放时输出低电平
11	SPIDO	SPI_DO 数据总线	
12	SPICLK	SPI_CLK 数据总线	
13	GPIOA1	P01	触发输出口1
14	GPIOA2	P02	触发输出口2
15	GPIOA3	P03	触发输出口3
16	GPIOA4	P04	触发输出口4
17	GPIOB5	SD_CLK 时钟总线	接 TF 卡或者 SD 卡
18	GPIOB6	SD_CMD 命令总线	接 TF 卡或者 SD 卡
19	GPIOB7	SD_DAT 数据总线	接 TF 卡或者 SD 卡
20	USB-	USB- DM	电脑的 USB 口
21	USB+	USB+ DP	电脑的 USB 口
22	NC	NC	
23	VCOM	退耦	
24	DACVSS	地	

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 $O		
$S	起始位0x7E	每条命令反馈均以$开头,即0x7E
VER	版本	版本信息
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
dat	参数	和前面的 len 相关联，不限制长度
checksum	校验和[占两个字节]	累加和校验[不计起始位$]
$O	结束位	结束位0xEF

举个例子，如果我们指定播放 SPIFLASH，就需要发送:7E FF 06 09 00 00 04 FF dd EF
数据长度为 6 ,这 6 个字节分别是[FF 06 09 00 00 04] 。不计算起始、结束、和校验。然后对得到的结果进行累加，再用 0 减，即“0-checksum=校验数据”。如果这里不明白，可以参考我们的“调试手册”。另外用户也可以直接忽视校验，参考我们的 5.3.4 章节说明。

3.2 通讯指令

我们的通讯分为以下两大块

- 控制指令
- 查询芯片的参数以及状态

3.2.1 控制指令

这里是控制芯片如何工作

CMD 命令	对应的功能	参数(16位)
0x01	下一曲	
0x02	上一曲	
0x03	指定曲目(NUM)	1-255
0x04	音量+	
0x05	音量-	
0x06	指定音量	0-30
0x07	保留	保留
0x08	单曲循环指定曲目播放	详见3.4.3
0x09	指定播放设备	详见3.4.4
0x0A	进入睡眠 -- 低功耗	功耗10MA
0x0B	唤醒睡眠	
0x0C	芯片复位	
0x0D	播放	
0x0E	暂停	
0x0F	指定文件夹文件名播放	详见3.4.5
0x16	停止	
0x17	仅用于 FLASH 存储设备[不支持 TF 卡和 U 盘]	详见3.4.7说明
0x18	保留	保留
0x19	对当前播放的曲目设置为循环播放	详见3.4.8
0x21	开启和关闭芯片的 DAC 输出	详见3.4.9
0x22	组合播放	详见3.4.10

3.2.2 查询指令

这里是查询芯片的状态和相关的参数

CMD 命令详解(查询)	对应的功能	参数(16位)
0x3C	保留	
0x3D	保留	
0x3E	保留	
0x3F	发送初始化参数	0x1F(低5位每位代表一个文件夹)
0x40	返回错误，请求重发	
0x41	应答	
0x42	查询当前状态	详见3.4.10
0x43	查询当前音量	
0x44	查询当前 EQ	保留
0x45	保留	该版本保留此功能
0x46	保留	该版本保留此功能
0x47	查询 UDISK 文件总数	设备的总文件数
0x48	查询 TF 文件总数	设备的总文件数
0x49	查询 FLASH 的总文件数	5个文件夹的总文件数
0x4B	查询 UDISK 的当前曲目	物理顺序
0x4C	查询 TF 的当前曲目	物理顺序
0x4D	查询 FLASH 的当前曲目	返回文件夹号和曲目指针
0x4E	查询指定文件夹的曲目总数	详见3.5.2
0x4F	查询当前设备的总文件夹数	详见3.5.3
0x61	查询当前文件夹指针	仅支持 FLASH

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 TF 卡、U 盘、SPIFLASH 设备的文件多少决定的，一般情况在小于 500ms 这个时间。如果超过这个时间模块的初始化数据还没有发送出来，说明模块初始化出错，请检查硬件的连接

(2)、模块初始化返回的数据为当前的有效文件夹,譬如返回 7E FF 06 3F 00 00 03 xx xx EF

==>其中 0x03 代表的是 U 盘和 TF 这两个设备在线

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	
FLASH -- 在线	7E FF 06 3F 00 00 08 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
FLASH 播放完第1曲	7E FF 06 3E 00 01 01 xx xx EF	FOLDER1的第1曲播放完
FLASH 播放完第2曲	7E FF 06 3E 00 02 02 xx xx EF	FOLDER2的第2曲播放完

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 作为解码和暂停的状态指示。请参见第 5 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。也是低电平

3、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

4、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

芯片返回 ACK	7E FF 06 41 00 00 00 xx xx EF	说明成功接收数据

(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
数据不符合规则	7E FF 06 40 00 00 08 xx xx EF	如最小为1的地方，发送为0

(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，设备是 SPIFLASH。如果 SPIFLASH 不在线的话，会自动进入睡眠状态。

(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，请参考下面的详解

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 串口控制指令详解

以下我们对关键的地方进行详细的说明--针对控制指令:

- 指定曲目播放
- 指定播放的音量
- 指定播放的设备
- 全部循环播放指令
- 组合播放功能[亮点]
- 带音量参数的指定曲目播放

3.4.1 指定歌曲播放指令[可以直接参考 3.4.7]

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 0~255.其实是可以支持更多的，因为涉及到文件管理的原因，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 FF 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 --- 代表产品编号

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)、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 单曲循环播放指令

循环播放指定曲目	7E FF 06 08 00 00 01 xx xx EF	循环播放第一曲
	7E FF 06 08 00 00 02 xx xx EF	循环播放第二曲
	7E FF 06 08 00 01 01 xx xx EF	循环播放 FOLDER1的第1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。在操作 TF 卡或者 U 盘时，按照的是文件存储的物理顺序指定，这点请用户注意。但是在操作 FLASH 时，是按照文件夹分区指定的，请参考上面的测试指令。

(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停止来关闭循环播放状态

3.4.4 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。

(4)、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-TF 卡	7E FF 06 09 00 00 02 xx xx EF	
指定播放设备-PC	7E FF 06 09 00 00 04 xx xx EF	
指定播放设备-FLASH	7E FF 06 09 00 00 05 xx xx EF	
指定播放设备-SLEEP	7E FF 06 09 00 00 06 xx xx EF	

3.4.5 指定文件夹文件名播放

文件夹01的001xxx.mp3	7E FF 06 0F 00 01 01 xx xx EF	TF 卡或者 U 盘
文件夹11的100xxx.mp3	7E FF 06 0F 00 0B 64 xx xx EF	TF 卡或者 U 盘
文件夹99的255xxx.mp3	7E FF 06 0F 00 63 FF xx xx EF	TF 卡或者 U 盘
FOLDER1的第1曲	7E FF 06 0F 00 01 01 xx xx EF	[FLASH]
FOLDER2的第1曲	7E FF 06 0F 00 02 01 xx xx EF	[FLASH]

(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，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

(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]

(5)、SPIFLASH 最多支持 5 个 FOLDER，请用户操作时，不要超过此范围。

3.4.6 指定文件夹开始循环播放

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
指定 FOLDER 循环播放	7E FF 06 17 00 03 01 xx xx EF	FOLDER3的第1曲循环播放

(1)、对于 TF 卡和 U 盘，文件夹的命名方式必须是”01” --- “99”。不可以超过 99

(2)、对于 SPIFLASH，用户可以对 5 个文件夹的任意一个循环播放，请参考上面的参考指令。

03 指定的文件夹为 FOLDER3

01 指定文件夹的第一曲开始，如果这里为 02。那么从第 2 曲开始循环播放此文件夹

(3)、用户可以发送停止指令来结束循环播放。

3.4.7 对当前的曲目设置为循环播放

循环播放开启关闭	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。

3.4.8 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，这样用户就可以共用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。

(2)、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，只是没有声音了而已。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

3.4.10 带音量参数的指令播放

带音量播放	7E FF 06 22 00 1E 01 xx xx EF	30级音量播放第1曲
	7E FF 06 22 00 0F 02 xx xx EF	15级音量播放第2曲

(1)、针对一些用户希望，对不同的语音设置不同的音量进行播放，如果按照以前的老方法，就是先设置完音量，再指定曲目播放，这样操作繁琐，不方便。特此我们增加此条指令 0x22

(2)、具体的操作可以参考上面给出的两条测试指令。

(3)、对于 U 盘或者 TF 卡，我们按照是物理顺序指定播放的。对于 FLASH，则是默认是 FOLDER1 文件夹下面。

3.5 串口查询指令详解

以下我们对关键的地方进行详细的说明--针对查询指令:

- 播放状态查询指令
- 指定文件夹曲目总数查询
- 当前设备的总文件夹数查询

3.5.1 查询当前在线的设备

查询在线设备	7E FF 06 3F 00 00 00 FE BC EF	U 盘正在播放

(1)、芯片在工作过程中，会不断的检测设备的在线情况，用户也可以通过 0x3F 这条指令进行查询

(2)、举例说明，如果芯片返回的数据为 7E FF 06 3F 00 00 0A xx xx EF

DL=0x0A = 0000 1010 代表了 TF 卡和 FLASH 在线

如果 DL=0x1F= 0000 1111 代表了 U 盘、TF 卡、PC、FLASH 均在线

(3)、0x0F--低四位均代表一种设备。

3.5.2 播放状态查询指令

正在播放	7E FF 06 42 00 01 01 xx xx EF	U 盘正在播放
暂停播放	7E FF 06 42 00 02 02 xx xx EF	TF 卡播放过程中被暂停
停止播放	7E FF 06 42 00 04 00 xx xx EF	在 USB 模式
	7E FF 06 42 00 08 01 xx xx EF	FLASH 正在播放
	7E FF 06 42 00 10 00 xx xx EF	芯片处于睡眠

(1)、模块在解码过程中会有 3 种状态对用户开放。用户可以通过指令查询获取模块的当前状态

(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，模块就处于播放停止的状态

(3)、如果返回的数据为 7E FF 06 42 00 02 02 xx xx EF 代表的意思详解如下：

DH = 0x02 --- 代表的是当前是 TF 卡设备，

DL = 0x02 --- 代表的是当前“TF 卡播放过程中被暂停”

(4)、如果返回的数据为 7E FF 06 42 00 02 02 xx xx EF 代表的意思详解如下：

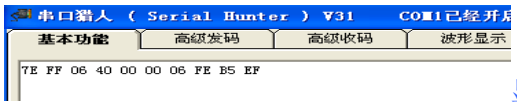
DH 的含义		DL 的含义	
0x01	当前设备是 U 盘	0x00	当前处于播放停止状态
0x02	当前设备是 TF 卡	0x01	当前处于正在播放状态
0x04	当前设备是 USB 盘	0x02	当前处于正在暂停状态
0x08	当前设备是 FLASH 盘		
0x10	当前是 SLEEP 模式		

3.5.2 指定文件夹曲目总数查询

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。

(2)、如果查询的文件夹为空[表示无有效文件]，那么串口会直接返回以下信息



显示查询文件夹出错

(3)、如果是 FLASH 的查询，直接查询我们给出的 FOLDER1--FOLDER5 即可。

3.5.3 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
	7E FF 06 4F 00 00 03 FE AC EF	FOLDER1和 FOLDER2有效

(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。

(2)、假如设备中有 5 个有效文件夹[文件夹里面有 MP3/WAV 文件]，一个空文件夹，那么查询文件夹的总数时，会返回有 6 个文件夹。所以建议用户不要建立空的文件夹。

(3)、在 FLASH 模式下，查询文件夹的数目返回的数据和 TF 卡以及 U 盘不一样，举例说明：

如果返回的数据：7E FF 06 4F 00 00 03 FE AC EF

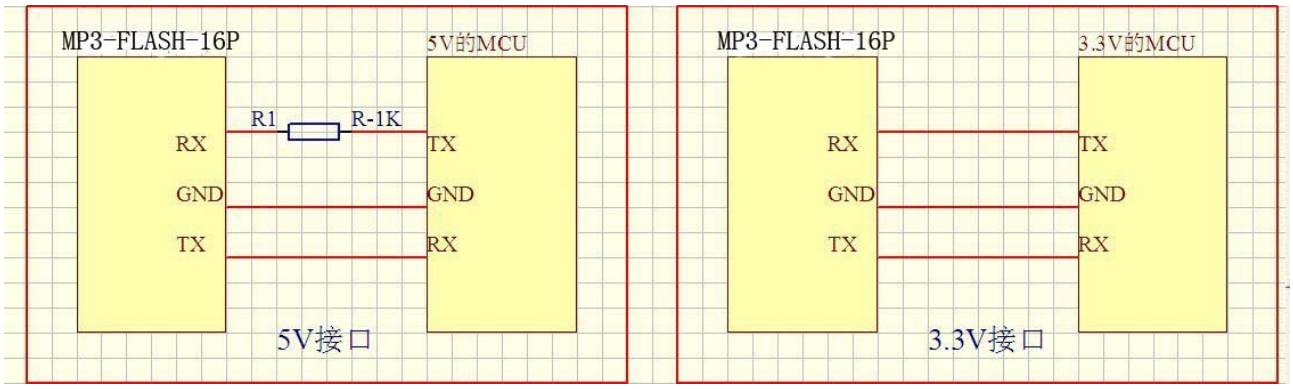
其中 DL = 0x03 ，代表的是 FOLDER1 和 FOLDER2 两个文件夹有语音文件，而 FOLDER3-FOLDER5 则没有有效文件。

4. 参考电路

争对芯片的应用，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

- 串行通信接口，波特率默认 9600，可以根据客户的要求修改
- 外部的 IO 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、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。

2、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，

3、如果应用于强电磁干扰的场合，请参考“注意事项”的说明。

4、芯片在 5V 和 3.3V 的系统中均正常的测试过，一切正常。均采用的是直连的方式，并没有串 1K 的电阻。一般的芯片都是能够兼容 3.3V 和 5V 的电平。

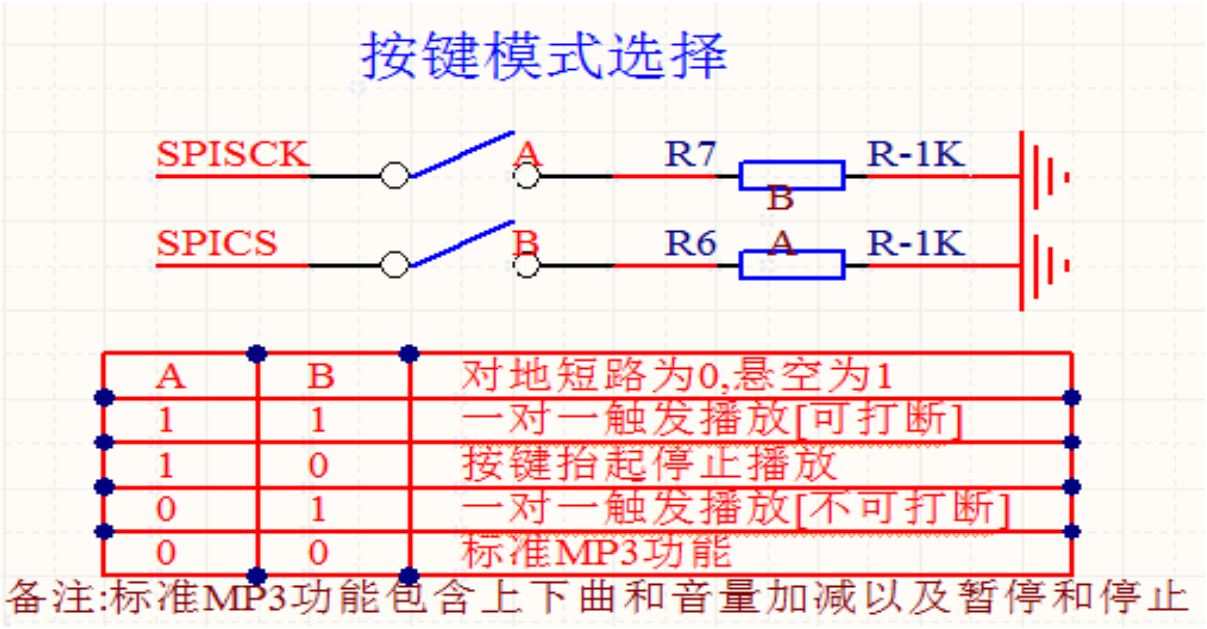
5、但是用户在实际的产品开发过程中，一定要严格的测试，留意电平的转换。强烈建议用户在能修改的条件下，使用 3.3V 的 MCU，响应环保、低功耗的号召。

4.2 按键接口

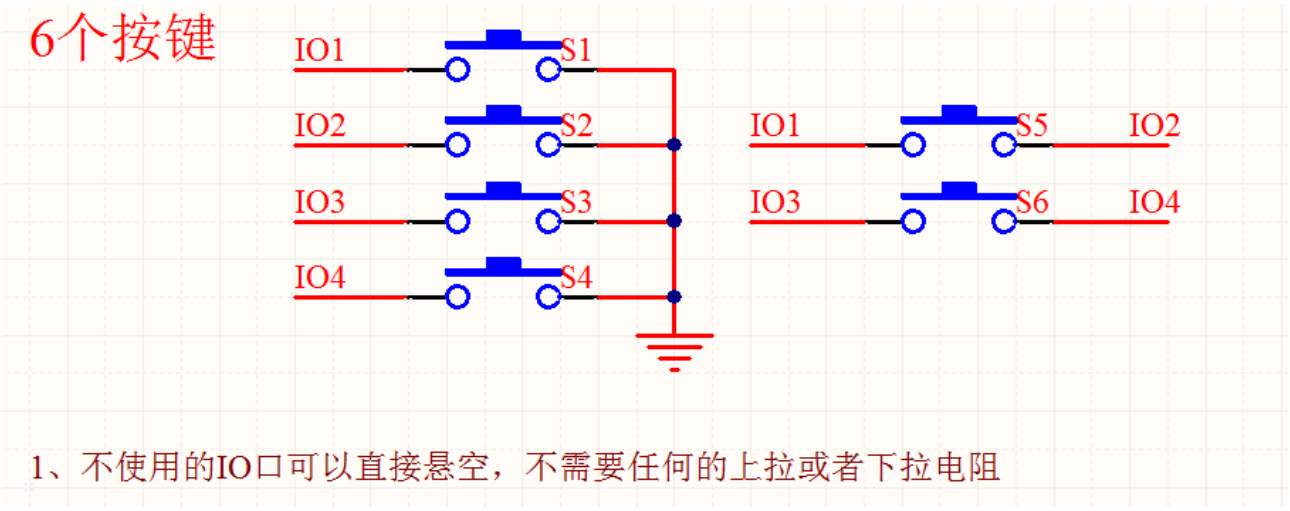
芯片我们采用的是 IO 按键的方式，取代了 AD 键盘的接法，这样做的好处是充分利用了 MCU 越来越来越多的 GPIO。设计繁琐但不简单，我们芯片默认配置 6 个按键的功能分配，可以在任何恶劣的场合随意的控制，甚至也可以作为与 MCU 的通信接口。我们的按键分配 4 种不同类型的功能，根据两个电阻的对地选择，请联系技术支持。

- 一对一触发播放，可打断
- 一对一触发播放，电平保持可循环
- 一对一触发播放，不可打断
- 标准的播放功能，如上下曲、播放暂停等等

丝印 1 对应 A 丝印 2 对应 B[详见 PCB]



上表为选择 10 个按键的功能逻辑表。两引脚悬时为[一对一触发播放可打断]



芯片留出来的按键为 S1—S4.其它的 S5—S6 需要用户自行按照上图的原理进行接线

(1)、一对一触发功能[可打断]

一对一触发	短按	长按	按着不松	按键抬起
S1	第1段[FOLDER1]			
S2	第2段[FOLDER1]			
S3	第3段[FOLDER1]			
S4	第4段[FOLDER1]			
S5	第5段[FOLDER1]			
S6	第6段[FOLDER1]			
备注:此为一对一触发播放，可打断[对地短路触发80ms 有效]				

(2)、按键抬起停止播放功能

按键抬起停止	短按	长按	按着不松	按键抬起
S1		第1段循环[FOLDER1]		停止
S2		第2段循环[FOLDER1]		停止
S3		第3段循环[FOLDER1]		停止
S4		第4段循环[FOLDER1]		停止
S5		第5段循环[FOLDER1]		停止
S6		第6段循环[FOLDER1]		停止
备注:此为抬起停止，长按循环播放[对地短路触发800ms 有效]				

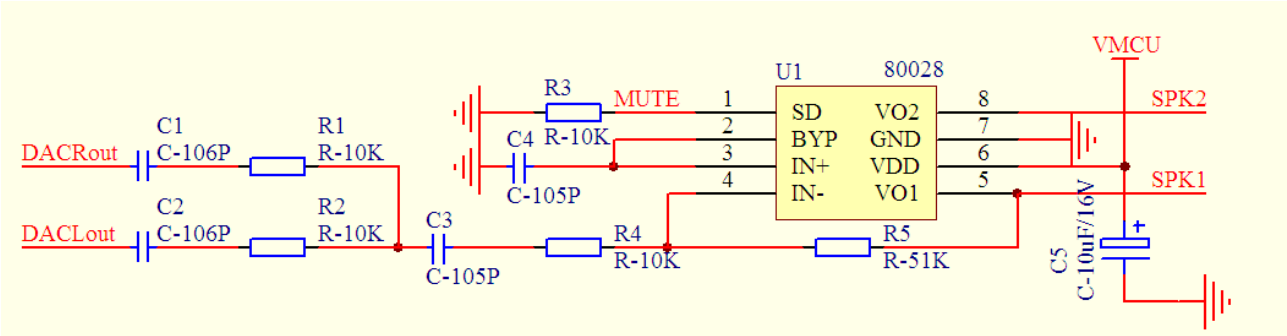
(3)一对一触发不可打断

触发不打断	短按	长按	按着不松	按键抬起
S1	第1段不可打断[FOLDER1]			
S2	第2段不可打断[FOLDER1]			
S3	第3段不可打断[FOLDER1]			
S4	第4段不可打断[FOLDER1]			
S5	第5段不可打断[FOLDER1]			
S6	第6段不可打断[FOLDER1]			
备注:此为一对一触发播放，不可打断[对地短路触发80ms 有效]				

(4)、标准播放功能

播放功能按键	短按	长按	按着不松	按键抬起
S1	下一曲		音量+	
S2	上一曲		音量-	
S3	播放暂停			
S4	停止			
S5	音量+			
S6	音量-			
备注:此为标准的 MP3功能				

4.3 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。

4.4 用户调节功放的音量

我们的芯片默认是上电音量最大，如果用户想自己减小音量，可以从如下 2 个地方修改

(1)、修改芯片的 DAC 输出的限流电阻，我们默认是贴 1K 的电阻，用户想减小音量，可以适当的加大此电阻。DAC 的电阻位置参照下图的“1”的红框框标记

(2)、修改功放的放大倍数，我们默认贴的是“51K”的电阻，用户如果想减小音量，可以适当的减小此电阻，可以修改为 47K 或者 33K。参见下图红框框“2”的标记

4.4 USB 更新语音说明

我们的模块可以使用手机充电线直接更新语音，方便、灵活。我们的优势如下

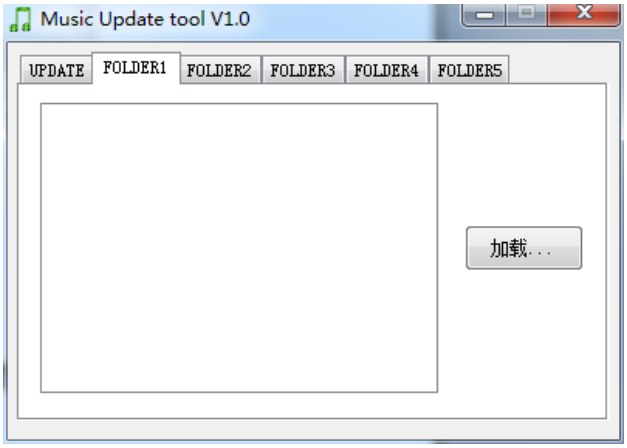
- 可以按照客户的要求，更正下载语音的窗口信息
- 无需安装任何软件，直接更新，也不需要专用下载器
- 对音质无任何压缩和损坏，保证更高的音质体验

1、插上我们的手机充电线，称之为 MicroUSB 线。电脑会弹出如下界面，然后把电脑的 360 软件，或者杀毒软件关了，或者插 USB 后弹出以下窗口选允许程序运行：

2、电脑会弹出如下界面

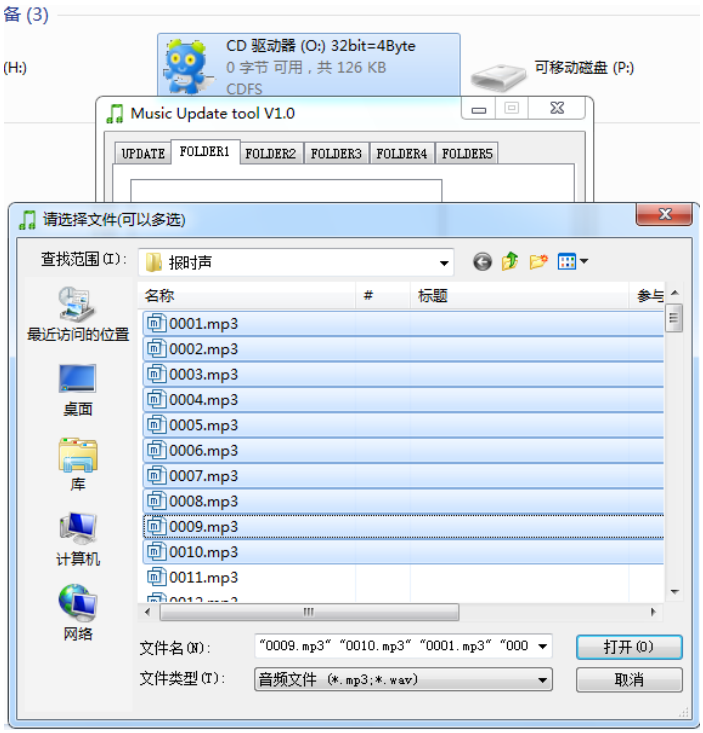
1、双击鼠标左键，电脑会弹出如下界面（1）

（1）



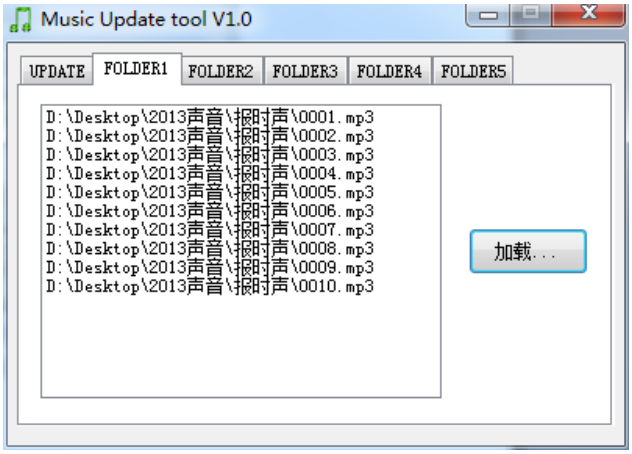
（2）

2、选中“FOLDER1”文件夹，如上图（2）并且点击加载，就会弹出一个加载语音的窗口，如下图



点鼠标左键拖动选中，或者按着键盘 Ctrl 键，一个一个的选中声音文件

3、此时选中需要加载的语音，点击“打开”就添加在软件中了。



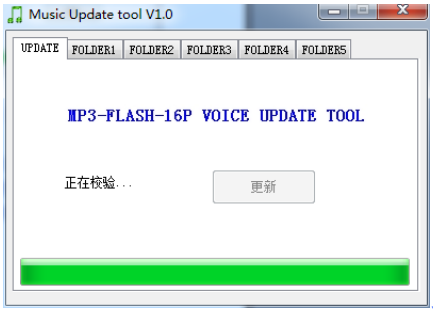
4、最后，回到“UPDATE”界面，点击“更新”按钮，会出现如下界面

5、

从左至右依次 3 个窗口，最后一个窗口就代表更新完毕，直接关闭此窗口，拔掉 USB 线就可以了

5、目前我们测试过的 SPIFLASH 包括华邦、GD、旺宏等市场上量最大的 FLASH。所以 FLASH 的兼容性不存在问题。我们测试过的 FLASH。最大为 16M 字节[W25Q128],均是没问题。

6、由于我们更新的窗口的更新进度条最大只支持 8M 字节，所以使用 16M 字节的 FLASH 时，会出现如下界面，请用户朋友不用担心，这是正常的



虽然进度条没变，但是还是在更新

7、更新的说明：

(1)、我们的更新窗口预留了 5 个区域进行更新，分别命名为 FOLDER1--FOLDER5.此版本支持 5 个区域的指定播放操作

(2)、FOLDER1--FOLDER5 这 5 个区域并没有分配空间，是可伸缩的。譬如，我只是用 FOLDER1，不使用其它的 4 个文件区域，那这 4 个文件区域会占空间吗？会的，占用不到 10 个字节,可忽略。

(3)、每一个 FOLDER 区域最多支持 255 段语音[在空间允许的情况下]。单个语音的大小是没有要求的。目前 FLASH 最大支持 128Mbit。也就是 16M 字节的语音存储，足以满足大部分的应用

(4)、如果用户对音质要求不高，可以自行选用语音压缩软件，对 MP3 或者 WAV 文件进行压缩，以减少文件所占的存储空间。为了体现我们产品的高音质，不建议用户这么使用

(5)、USB 更新语音的时间，是根据语音文件大小决定的，语音越大，更新的时间越慢。

(6)、一旦更新过语音之后，之前保存的语音将会全部删除。

7、备注：

(1)、我们的模块无需安装任何软件，插入电脑弹出的软件实际是烧录至 SPIFLASH。占 200kb 的空间。很小，所以用户可以直接忽略。

(2)、用户如果需要自行换上自己的 SPIFLASH。请向我司技术支持索取更新的固件。

(3)、如果电脑第一次使用我们的模块，刚插入电脑时，会需要一定的时间，因为此时电脑会自动的对我的模块进行枚举，确定身份等等操作。可能需要 1 分钟左右，直至弹出更新窗口。

4.5 用户使用空白的 FLASH 说明

用户在调试的过程中，会按照自己的需求更换 flash 的大小来满足自己的需求，这样就需要以下三个步骤来完成 FLASH 的替换。

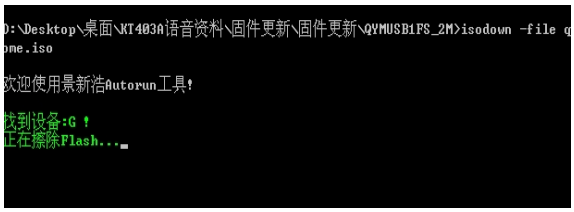
- 问我们获取 FLASH 的更新固件
- 通过 USB 接口对空白的 FLASH 进行固件的烧录
- 拔掉 USB 线缆，再通电就可正常使用

1、我们的固件分为 4 个部分，下面详细解释一下

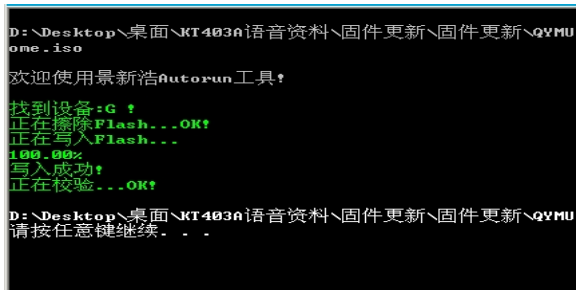
用户根据自己的 FLASH 大小，来选择更新的固件。固件的大小都是一样的。

右图实际上是通过”双击本文件烧写模块固件.bat”这个批处理调用“ISDDown.exe”软件将“32bit=4Byte.iso.iso”镜像文件写入 FLASH 中。这个是开源工具，有兴趣的可以自行查阅资料，来了解整个的流程。

2、通过 USB 插入电脑之后，点击“mydown.bat”，会弹出如下窗口



3、更新完成之后，会出现如下界面：



4、更新完成之后，就可以看到电脑已经虚拟出来了一个”CDROM”，到此就说明成功了。

备注：如果在更新过程中，出现任何异常，都是不正常的，可以更换 FLASH 来确定问题。

5. 注意事项

芯片的使用,关键的地方做如下说明:

- 芯片的 GPIO 的特性
- 应用的中注意事项
- 串口编程部分的注意

5.1 GPIO 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*VDD	V	VDD=3.3V
V_{IH}	High-Level Input Voltage	0.7VDD	-	VDD+0.3	V	VDD=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	VDD=3.3V
V_{OH}	High-Level Output Voltage	2.7	-	-	V	VDD=3.3V

5.2 应用中的注意点

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

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

4、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 10MA 左右，播放中，依据音量的大小，峰值电流可以达到 1A。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、用户如果直接使用我们芯片自带的功放，请选择合适的喇叭即可。推荐使用 4 欧姆/3W。这个是使用效果最好的配置。选用其它的喇叭，请注意负载大小，以及功率这两个参数

7、该芯片支持 MP3、WAV 二种主流的音频格式。

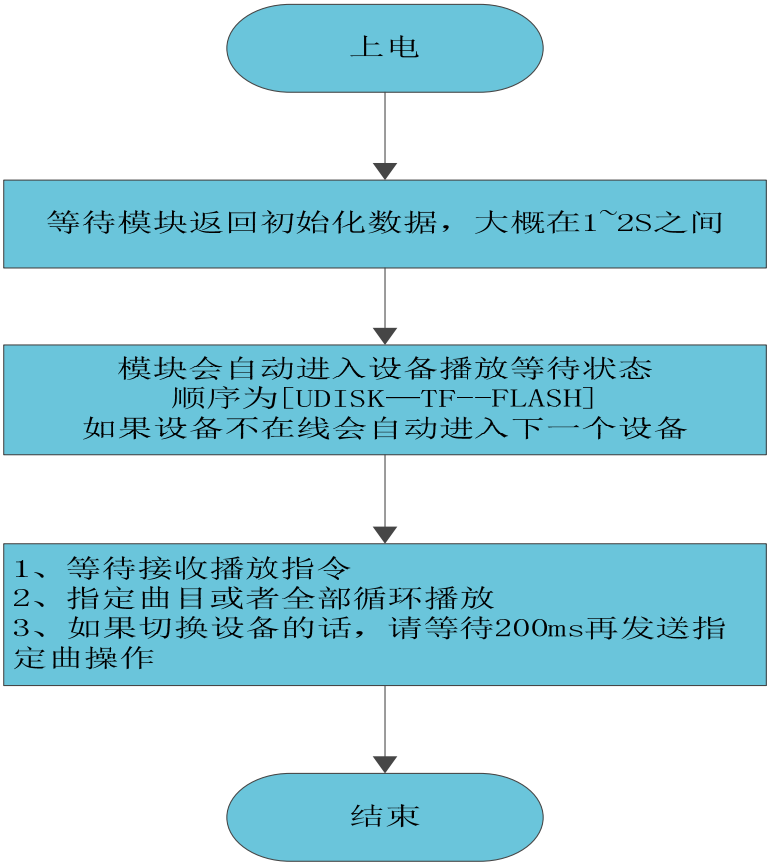
8、我们的芯片支持 8/11.025/12/16/22.05/24/32/44.1/48KHZ 采样率的音频文件，这些也是网络上绝大多数的音频文件的参数。如果用户的音频文件的采样率不在此范围内，是不支持播放的，但是可以通过专用的软件转换一下即可。我们的优势就是无损播放和高音质，所以不太建议用户对音频进行压缩。

5.3 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

5.3.1 串口操作流程



1、我司提供的所有芯片的串口部分的操作，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。

5.3.2 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

5.3.3 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 时间进行初花的相关操作，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会这些数据

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

4、芯片内部对串口的处理是 10MS 处理一次，所以连续的指令发送时，必须要间隔 20MS 的延时。否则前面的指令将会被覆盖而得不到执行

5、如果指定文件夹文件名播放[0x0F 指令]，延时必须大于 40ms，因为芯片锁定文件是需要时间的。只要涉及到文件夹文件名查找的相关指令，40MS 的延时是必不可少的。如果芯片当前正在查找文件，串口的数据过来太频繁，会导致芯片的复位

5.3.4 校验的重要说明

1、争对很多用户不太习惯校验的通信方式，我们特别推出了带校验和不带校验的兼容方式。举例说明。如果我们发送组合播放指令如下:

下一曲播放[不带校验]	7E FF 06 01 00 00 00 EF	
下一曲播放[带校验]	7E FF 06 01 00 00 00 FE FA EF	

比较两条指令的区别，就是省略掉的校验的 2 个字节。这两帧数据均可以被芯片所接收。

2、因为很多用户在使用的过程中，很多都是使用不带晶振的 MCU。这样的话，我们必须建议您加上校验这种方式，来保证通信的稳定性。

3、假如用户使用 STM32 或者 STC 等等 MCU，并且是外挂晶振的，就可以适当的省掉校验。因为不带晶振的 MCU，时钟是相对不那么准的，所以串口会存在误差，一旦误差过大，会导致通信出错。请用户朋友自行斟酌。

5.3.5 MCU 的晶振选择

1、原则上我们建议用户，使用 11.0592MHZ 或者相倍数的晶振。这样可以让串口产生 9600 的波特率会更准确。我们的芯片串口误差是允许在 3%以内的
2、如果用户在 12M 的晶振时。首先要做如下判断，
(1)、看是什么 MCU，51 或者 PIC、STM32 等等，基本都自带波特率发生器，所以产生 9600 的波特率基本没压力。
(2)、看 MCU 是否为硬件串口，如果是 IO 模拟的串口的话，强烈建议用户使用 11.0592 的晶振
(3)、标准的 51，如：STC89C52 或者 AT89C52 等等都是采用定时器产生波特率的，经过简单的计算就可以算出，12M 晶体做 9600 波特率的误差是是 0.16%，正常运行是没有任何问题，但还是需要用户进行全面测试

6. 免责声明

■ 开发预备知识

本公司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

本公司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。本公司芯片芯片的 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.

本公司芯片的 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。我们将努力完善本公司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

本公司能保留任何时候在不事先声明的情况下对本公司芯片产品相关文档的修改权力

■ ESD 静电放点保护

本公司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，安装本公司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

7. 参考例程

```
/*******************************************************************************
 - 实现功能：实现芯片上电分别指定播放第一曲和第二曲，基本的程序供用户测试
 - 日期      ：2013-05-06
 - 运行环境：STC     晶振：11.0592M        波特率:9600
 - 备注      ：在普中科技的 51 开发板上调试 OK --- STC89C516RD+
          该测试程序必须是芯片或者芯片方案中有设备在线，譬如 U 盘、TF 卡、FLASH
*******************************************************************************/
#include "REG52.h"

#define COMM_BAUD_RATE    9600      //串口波特率
#define OSC_FREQ          11059200      //运行晶振：11.05926MHZ
static INT8U Send_buf[10] = {0} ;

void Delay_Ms(INT32U z)
{
     INT32U x=0 , y=0;
     for(x=110 ; x>0 ;x--)
     for(y=z; y>0;y-- );
}

/*******************************************************************************
 - 功能描述：  串口 1 初始化
 - 注：        设置为 9600 波特率
*******************************************************************************/
void Serial_init(void)
{
     TMOD = 0x20;                    // 设置 T1 为波特率发生器
     SCON = 0x50;                    // 0101,0000 8 位数据位，无奇偶校验
     PCON = 0x00;                    //PCON=0;
     TH1=256-(OSC_FREQ/COMM_BAUD_RATE/32/12);//设置为 9600 波特率
     TL1=256-(OSC_FREQ/COMM_BAUD_RATE/32/12);
     TR1       = 1;                  //定时器 1 打开
     REN       = 1;                  //串口 1 接收使能
     ES        = 1;                  //串口 1 中断使能
}
void Uart_PutByte(INT8U ch)
{
     SBUF    = ch;
     while(!TI){;}
     TI = 0;
}

/*******************************************************************************
 - 功能描述：  串口向外发送命令[包括控制和查询]
 - 参数说明：  CMD:表示控制指令，请查阅指令表，还包括查询的相关指令
               feedback:是否需要应答[0:不需要应答，1:需要应答]
               data:传送的参数
*******************************************************************************/
void SendCmd(INT8U len)
```

```
{
    INT8U i = 0 ;
    Uart_PutByte(0x7E); //起始
    for(i=0; i<len; i++)//数据
    {
        Uart_PutByte(Send_buf[i]) ;
    }
    Uart_PutByte(0xEF) ;//结束
}

/********************************************************************************
 - 功能描述：求和校验 --- 用户也可以省略此校验，参见 5.3.4 的说明
 - 和校验的思路如下：
    发送的指令，去掉起始和结束。将中间的 6 个字节进行累加，最后取反码。接收端就将接收到
的一帧数据，去掉起始和结束。将中间的数据累加，再加上接收到的校验字节。刚好为 0.这样就代
表接收到的数据完全正确。
********************************************************************************/
void DoSum( INT8U *Str, INT8U len)
{
    INT16U xorsum = 0;
    INT8U i;
    for(i=0; i<len; i++)
    {
        xorsum    = xorsum + Str[i];
    }
    xorsum        = 0 -xorsum;
    *(Str+i)     = (INT8U)(xorsum >>8);
    *(Str+i+1) = (INT8U)(xorsum & 0x00ff);
}

void Uart_SendCMD(INT8U CMD ,INT8U feedback , INT16U dat)
{
    Send_buf[0] = 0xff;      //保留字节
    Send_buf[1] = 0x06;      //长度
    Send_buf[2] = CMD;       //控制指令
    Send_buf[3] = feedback;//是否需要反馈
    Send_buf[4] = (INT8U)(dat >> 8);//datah
    Send_buf[5] = (INT8U)(dat);       //datal
    DoSum(&Send_buf[0],6);            //校验
    SendCmd(8);           //发送此帧数据
}

void main()
{
    Serial_init() ;//串口寄存器的初始化设置
    Uart_SendCMD(0x03 , 0 , 0x01) ;//播放第一首
    Delay_Ms(1000) ;//延时大概 6S
    Uart_SendCMD(0x03 , 0 , 0x02) ;//播放第二首
    Delay_Ms(1000) ;//延时大概 6S
    Uart_SendCMD(0x03 , 0 , 0x04) ;//播放第四首
    while(1);
}
```

8. PC 端串口调试指令举例

用户可以通过电脑端的串口软件，对芯片进行测试。我们的芯片串口为 TTL 电平，请注意电平转换

- 控制指令
- 查询参数指令

9.1 控制指令

串口调试助手进行测试	发送的命令[带校验]	发送的命令[不带校验]	备注
[下一首]	7E FF 06 01 00 00 00 FE FA EF	7E FF 06 01 00 00 00 EF	
[上一首]	7E FF 06 02 00 00 00 FE F9 EF	7E FF 06 02 00 00 00 EF	
[指定曲目]	7E FF 06 03 00 00 01 FE F7 EF	7E FF 06 03 00 00 01 EF	指定第一首播放
	7E FF 06 03 00 00 02 FE F6 EF	7E FF 06 03 00 00 02 EF	指定第二首
	7E FF 06 03 00 00 0A FE EE EF	7E FF 06 03 00 00 0A EF	指定第10首
音量加	7E FF 06 04 00 00 00 FE F7 EF	7E FF 06 04 00 00 00 EF	
音量减	7E FF 06 05 00 00 00 FE F6 EF	7E FF 06 05 00 00 00 EF	
[指定音量]	7E FF 06 06 00 00 1E FE D7 EF	7E FF 06 06 00 00 1E EF	指定音量为30级
[指定 EQ]	7E FF 06 07 00 00 01 FE F3 EF	7E FF 06 07 00 00 01 EF	保留
[循环播放曲目]	7E FF 06 08 00 00 01 FE F2 EF	7E FF 06 08 00 00 01 EF	循环播放第一首
	7E FF 06 08 00 00 02 FE F1 EF	7E FF 06 08 00 00 02 EF	循环播放第二首
	7E FF 06 08 00 00 0A FE E9 EF	7E FF 06 08 00 00 0A EF	循环播放第十首
	7E FF 06 08 00 01 01 FE F1 EF	7E FF 06 08 00 01 01 EF	循环播放 FLASH 的 FOLDER1的第一曲
	7E FF 06 08 00 02 01 FE F0 EF	7E FF 06 08 00 02 01 EF	循环播放 FLASH 的 FOLDER2的第一曲
[指定播放设备]	7E FF 06 09 00 00 01 FE F1 EF	7E FF 06 09 00 00 01 EF	指定播放 UDISK
	7E FF 06 09 00 00 02 FE F0 EF	7E FF 06 09 00 00 02 EF	指定播放 TF
	7E FF 06 09 00 00 03 FE EF EF	7E FF 06 09 00 00 03 EF	指定播放 FLASH
[进入睡眠模式]	7E FF 06 0A 00 00 00 FE F1 EF	7E FF 06 0A 00 00 00 EF	
[唤醒睡眠]	7E FF 06 0B 00 00 00 FE F0 EF	7E FF 06 0B 00 00 00 EF	
[芯片复位]	7E FF 06 0C 00 00 00 FE EF EF	7E FF 06 0C 00 00 00 EF	
[播放]	7E FF 06 0D 00 00 00 FE EE EF	7E FF 06 0D 00 00 00 EF	
[暂停]	7E FF 06 0E 00 00 00 FE ED EF	7E FF 06 0E 00 00 00 EF	
[指定文件夹文件名]	7E FF 06 0F 00 01 01 FE EA EF	7E FF 06 0F 00 01 01 EF	"01"的文件夹，曲目为"001"
	7E FF 06 0F 00 01 02 FE E9 EF	7E FF 06 0F 00 01 02 EF	"01"的文件夹，曲目为"002"

停止播放	7E FF 06 16 00 00 00 FE E5 EF	7E FF 06 16 00 00 00 EF	停止软件解码
指定文件夹循环播放	7E FF 06 17 00 02 00 FE E2 EF	7E FF 06 17 00 02 00 EF	指定02文件夹循环播放
	7E FF 06 17 00 01 00 FE E3 EF	7E FF 06 17 00 01 00 EF	指定01文件夹循环播放
随机播放	7E FF 06 18 00 00 00 FE E3 EF	7E FF 06 18 00 00 00 EF	随机播放
单曲循环播放	7E FF 06 19 00 00 00 FE E2 EF	7E FF 06 19 00 00 00 EF	单曲循环播放开启
	7E FF 06 19 00 00 01 FE E1 EF	7E FF 06 19 00 00 01 EF	单曲循环播放关闭
带音量播放	7E FF 06 22 00 1E 01 FE BA EF	7E FF 06 22 00 1E 01 EF	30级音量播放第1曲
	7E FF 06 22 00 0F 01 FE C9 EF	7E FF 06 22 00 0F 01 EF	15级音量播放第1曲
	7E FF 06 22 00 0F 02 FE C8 EF	7E FF 06 22 00 0F 02 EF	15级音量播放第2曲

9.2 查询参数指令

串口调试助手进行测试	发送的命令[带校验]	发送的命令[不带校验]	备注
查询当前状态	7E FF 06 42 00 00 00 FE B9 EF	7E FF 06 42 00 00 00 EF	
[查询音量]	7E FF 06 43 00 00 00 FE B8 EF	7E FF 06 43 00 00 00 EF	
[查询当前 EQ]	7E FF 06 44 00 00 00 FE B7 EF	7E FF 06 44 00 00 00 EF	
U 盘总文件数	7E FF 06 47 00 00 00 FE B4 EF	7E FF 06 47 00 00 00 EF	当前设备的总文件数
TF 总文件数	7E FF 06 48 00 00 00 FE B3 EF	7E FF 06 48 00 00 00 EF	
FLASH 总文件数	7E FF 06 49 00 00 00 FE B2 EF	7E FF 06 49 00 00 00 EF	
U 盘当前曲目	7E FF 06 4B 00 00 00 FE B0 EF	7E FF 06 4B 00 00 00 EF	当前播放的曲目
TF 当前曲目	7E FF 06 4C 00 00 00 FE AF EF	7E FF 06 4C 00 00 00 EF	
FLASH 当前文件夹曲目指针	7E FF 06 4D 00 00 00 FE AE EF	7E FF 06 4D 00 00 00 EF	
指定文件夹曲目总数查询	7E FF 06 4E 00 00 01 FE AC EF	7E FF 06 4E 00 01 00 EF	
查询 TF/U 盘总文件夹数	7E FF 06 4F 00 00 00 FE AC EF	7E FF 06 4F 00 00 00 EF	只支持 TF 卡和 U 盘