

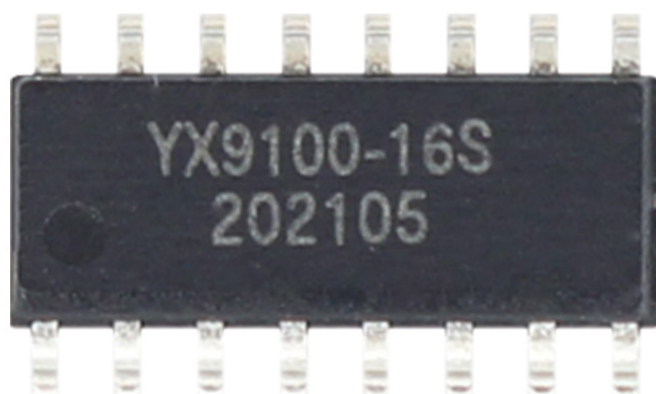


---

# YX9200-16S 语音芯片

---

## 使用手册



广州悦欣电子科技有限公司

Guangzhou Yuexin Electronic Technology Co., Ltd.

## 目录

1、概述.....	3
1.1 简介.....	3
1.2 功能.....	3
1.3 应用.....	3
2、芯片使用说明.....	4
2.1 硬件参数.....	4
2.2 管脚说明.....	4
3. 串口通讯.....	5
3.1 通讯格式.....	5
3.2 通讯指令.....	6
3.2.1、控制指令.....	6
3.2.2、查询指令.....	6
3.3 芯片返回的数据.....	7
3.3.1、芯片上电返回的数据.....	7
3.3.2、曲目播放完毕返回的数据.....	8
3.3.3、芯片应答返回的数据.....	8
3.3.4、芯片错误返回的数据.....	8
3.4 串口控制指令详解.....	9
3.4.1、指定歌曲播放指令.....	9
3.4.2、指定音量播放指令.....	10
3.4.3、单曲循环播放指令.....	10
3.4.4、指定播放设备.....	10
3.4.5、指定文件夹文件名播放.....	11
3.4.6、根目录曲目全部循环播放.....	12
3.4.7、播放停止指令.....	12
3.4.8、指定文件夹循环播放.....	12
3.4.9、随机播放设备文件.....	12
3.4.10、对当前的曲目设置为循环播放.....	13
3.4.11、多文件夹插播.....	13
3.5 串口查询指令详解.....	14
3.5.1、查询当前在线的设备.....	14
3.5.2、播放状态查询指令.....	14
3.5.3、当前设备的音量查询.....	15
4、参考电路.....	15
4.1 串行接口.....	15
4.2 按键接口.....	16
4.2.1、按键接口的接法.....	16
4.2.2、通过 TXT 文件来配置按键功能.....	16
4.2.3、按键配置功能的详细解释.....	17
4.3 外接功放芯片电路.....	18
4.4 外接耳机电路.....	18
4.5 USB 更新语音说明.....	18

4.5.1 USB 更新 SPIFLASH 的语音详细说明.....	19
5、用户使用空白的 FLASH 说明.....	20
6. 芯片使用注意事项.....	23
6.1 GPIO 的特性 .....	23
6.2 应用中的注意点 .....	24
6.3 串口注意事项点 .....	24
6.3.1、芯片上电的工作流程图 .....	25
6.3.2、串口编程参考的说明 .....	25
6.3.3、串口编程需要适当延时的注意点 .....	25
6.3.4、校验的重要说明 .....	26
6.3.6、MCU 的晶振选择.....	26
6.3.7、指定播放的说明 .....	27
6.3.8、串口调试说明 .....	27
6.3.9、校验代码的移植 .....	28
7. PC 端串口调试指令举例 .....	29
7.1 控制指令 .....	29
7.2 查询指令 .....	30
8. 免责声明.....	30
9. 厂家联系方式.....	31
10. 文件版本信息.....	31

# 1、概述

## 1.1 简介

YX9200-16S 是我们公司为满足市场需要和高品质的追求，而研发的一款小巧的新型高品质语音芯片，支持 WAV、MP3 音频解码，使用更灵活方便。支持 SPI-FLASH 和 U 盘作为存储介质。具有 USB 接口，可以通过 PC 机自由更换存储器音频内容。同时软件支持工业级别的串口通信协议，以 FLASH 作为存储介质。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。

无需烧录器及上位机，USB 直接烧写 FLASH。同时 SPIFLASH、U 盘、TF 卡作为存储介质。

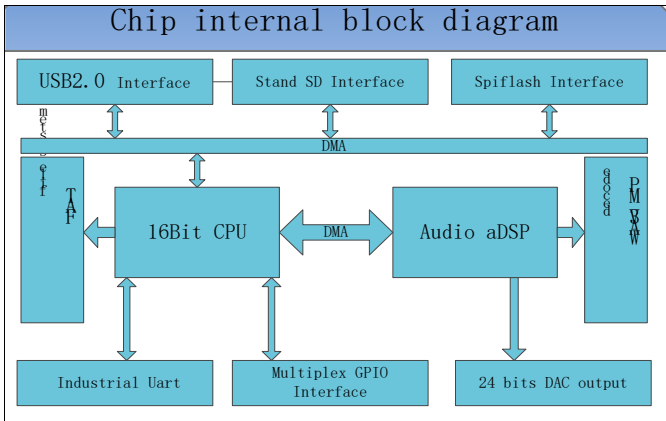
## 1.2 功能

- 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、使用 SPI-FLASH 和 U 盘、TF 卡作为存储器，最大支持 128Mbit flash。
- 4、使用 UART 异步串口通讯，通用标准接口协议，控制更灵活方便。
- 5、30 级音量可调，支持 DAC 输出，可接功放，耳机。
- 6、支持 ADK 触发播放，以适合碳膜按键等等场合。
- 7、AD 按键控制模式下可能过文件配置按键功能。
- 8、USB 接口更新语音文件，无需安装任何软件。

## 1.3 应用

车载导航语音 播报	机电设备故障 自动报警	消防语音报警 提示
公安边防检查 通道语音提示	车辆进、出通 道验证语音提 示	电动观光车安 全行驶语音告 示
公路运输稽查、 收费站语音提 示	电力、通信、 金融营业厅语 音提示	火车站、汽车 站安全检查语 音提示

## 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	支持 192kbps 及以下比特率 11172-3 和 IS013813-3 layer3 音频解码 采样率支持(KHZ):8/11.025/12/16/22.05/24/32/44.1/48
USB 接口	2.0 标准
UART 接口	标准串口，TTL 电平 9600，波特率可设定
输入电压	3.3V-5V
额定电流	10ma[不带 U 盘]
封装尺寸	SOP16 标准
工作温度	-40 度 ~ 80 度
湿度	5% ~ 85%
主芯片型号	YX9200-16S/ YX9100-16S (SOP16)

### 2.2 管脚说明

RX	1	PB7/SD0CLKB/SPI2DOA/AUX1R/ADC9	VCOM	16	VCOM
TX	2	PB6/SD0CMB/SPI2CLKA/AUX1L/ADC8	VSS	15	GND
PGND	3	PGND	DACL	14	DACL
+5V	4	VBAT	ADCR	13	ADCR
3.3V	5	VDDIO	ADC0/PA0/DACR	12	BUSY
SPIDO	SDCLK	6	ADC1/ADC2/AUX0L/MIC/PA3/PA1	11	ADK
SPISCK	SDCMD	7	PC5/SD0CLKA/SPI1DOB/ADC1ADC3/AUX0R/MICBIAS/PA2/PA4	10	USBDP
SPICS	SDDAT	8	PC4/SD0CMDA/SPI1CLKB/ADC11	9	USBDM
			PC3/PC2/SD0DATA/SPI1DIB/ADC10		
			ADC14/USBDM		
			YX9200-SOP16		



引脚序号	引脚名称	功能描述	备注
1	RXD	UART 串行数据输入	3.3V TTL 电平
2	TXD	UART 串行数据输出	3.3V TTL 电平
3	PGND	模拟地	
4	VBAT	电源输入脚	DC3.3V~5V
5	+3.3V	3.3V 电源出	用于 flash 及 TF 卡供电
6	SPI_DATA	SPI_D0 口	Flash 及 TF 卡数据总线
7	SPI_CLK	SPI_CLK 口	Flash 及 TF 卡数据总线
8	SPI_CS	SPI_CS 口	Flash 及 TF 卡片选总线
9	USBDM	USB- 通讯口	连接电脑/U 盘
10	USBDP	USB+ 通讯口	连接电脑/U 盘
11	ADK	触发端口 1	低电平触发
12	BUSY	忙信号输出	空闲输出高/播放时输出低电平
13	DACR	音频输出右声道	驱动耳机、功放
14	DACL	音频输出左声道	驱动耳机、功放
15	GND	电源地	
16	VCOM	退耦	

## 3. 串口通讯

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 :8

校验位 :none

流控制 :none

格式: \$S address Len CMD para checksum \$0

\$S	起始位0x7E	每条命令反馈均以\$开头，即0x7E
VER	地址码	设备地址，默认 FF
Len	字节长度	校验和不计算在内
CMD	操作码	表示具体的操作，比如播放/暂停等等
Feedback	闪灯状态及反馈	用于控制 DI状态和是否需要反馈信息
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\$]
\$0	结束位	结束位0xEF

## 3.2 通讯指令

我们的通讯指令分为以下两大块

- 控制指令
- 查询芯片的参数以及状态

### 3.2.1、控制指令

1、直接发送的指令，这里是控制芯片如何工作

CMD 命令(指令)	对应的功能	参数
0x01	下一曲	
0x02	上一曲	
0x03	指定曲目(NUM)	详见 3.4.1
0x04	音量+	
0x05	音量-	
0x06	指定音量	详见 3.4.2
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	保留	保留
0x0B	设定波特率	该版本暂无
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
0x11	根目录曲目全部循环播放	详见 3.4.6
0x13	保留	保留
0x15	保留	保留
0x16	停止	详见 3.4.7
0x17	指定文件夹循环播放	详见 3.4.8
0x18	指定整个设备随机播放	详见 3.4.9
0x19	对当前播放的曲目设置为循环播放	详见 3.4.10
0x1A	保留	保留
0x21	保留	保留
0x25	多文件夹插播	详见 3.4.11
0XC0	设置芯片地址	该版本暂无

### 3.2.2、查询指令

CMD 命令(查询)	对应的功能	参数
0x3C	保留	
0x3D	保留	

0x3E	保留	
0x3F	查询在线的设备	详见 3.5.1
0x40	返回错误，请求重发	
0x41	应答	
0x42	查询当前状态	详见 3.5.2
0x43	查询当前音量	
0x44	保留	保留
0x45	保留	保留
0x46	保留	保留
0x47	查询 UDISK 文件总数	设备的总文件数
0x48	保留	保留
0x49	查询 FLASH 的总文件数	FLASH 内部总文件数
0x4B	查询 UDISK 的当前曲目	物理顺序
0x4C	保留	保留
0x4D	查询 FLASH 的当前曲目	返回文件夹号和曲目标针

### 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

#### 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 SPIFLASH 设备的文件多少决定的，一般在小于 500ms 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，请检查硬件的连接

(2)、芯片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 08 xx xx EF  
 ==>其中 0x08 代表的是 SPIFLASH 设备在线  
 xx xx 在整个说明书中为校验位  
 前两个 xx ---校验的高字节  
 后两个 xx --- 校验的低字节

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
FLASH、PC -- 在线	7E FF 06 3F 00 00 0C 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

FLASH 播放完第 1 曲	7E FF 06 3E 00 00 01 xx xx EF	SPI FLASH 第 1 曲播放完
FLASH 播放完第 2 曲	7E FF 06 3E 00 00 02 xx xx EF	SPI FLASH 的第 2 曲播放完
FLASH 播放完第 10 曲	7E FF 06 3E 00 00 0A xx xx EF	SPI FLASH 的第 10 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 BUSY 脚。
- 3、播放状态输出低电平，可以通过此 AD 按键直接进行控制。
- 4、播放暂停状态，输出高电平；芯片睡眠状态，也是低电平。
- 5、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。
- 6、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。

## 3.4 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

### 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 0~3000，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。此指令是按照存储的物理顺序指定的，FLASH 默认为根目录。

(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 FE F7 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

FE --- 校验的高字节

F7 --- 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)、芯片内部设置的音量细分级数为 0—30，请用户注意。

(3)、例如指定音量为 15 级，串口发送的指令：7E FF 06 06 00 00 0F FE E6 EF

7E --- 起始命令

FF --- 版本信息

06 --- 数据长度(不包含校验)

06 --- 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 --- 音量级的高字节[DH]

0F --- 音量级的低字节[DL], 这里代表的是音量为 15 级

FE --- 校验的高字节

E6 --- 校验的低字节

EF --- 结束命令

(4)、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	根目录循环播放第一曲
	7E FF 06 08 00 00 02 xx xx EF	根目录循环播放第二曲
	7E FF 06 08 00 00 03 xx xx EF	根目录循环播放第三曲

(1)、针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 3 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)、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。

(3)、看下表，选择合适的指令发送。

指定播放设备-U 盘	7E FF 06 09 00 00 01 xx xx EF	
指定播放设备-TF 卡	7E FF 06 09 00 00 01 xx xx EF	
指定播放设备-PC	7E FF 06 09 00 00 03 xx xx EF	指[读卡]模式

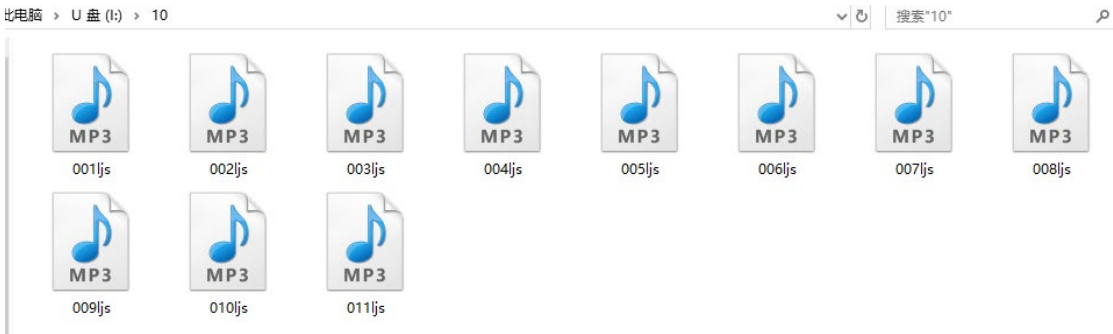
指定播放设备-FLASH	7E FF 06 09 00 00 04 xx xx EF	
--------------	-------------------------------	--

### 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
文件夹 11 的 100xxx.mp3	7E FF 06 0F 00 0B 64 xx xx EF
文件夹 99 的 255xxx.mp3	7E FF 06 0F 00 63 FF xx xx EF

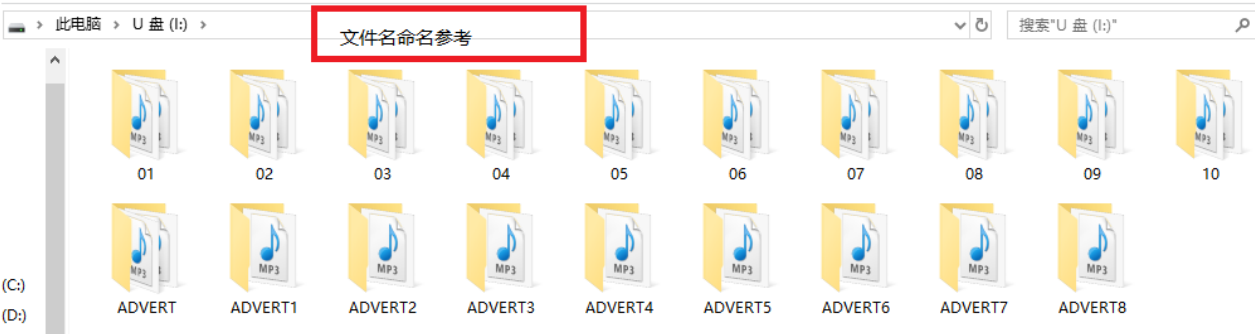
- (1)、指定文件夹播放是我们制定的扩展功能，默认文件夹的命名方式为“01”，“05”，“11”这样的方式，为了系统的稳定性和歌曲切换的速度，每个文件夹下默认最大支持 255 首歌，最多支持 99 个文件夹。
- (2)、例如：指定“01”文件夹的 001xxx.MP3 文件，串口发送的指令为:7E FF 06 0F 00 01 01 xx xx EF；001xxx.MP3 文件：前面三位按 001~255 来命名，否则读取无反应；后面三位 xxx 任意取，也可以只取前面三位。

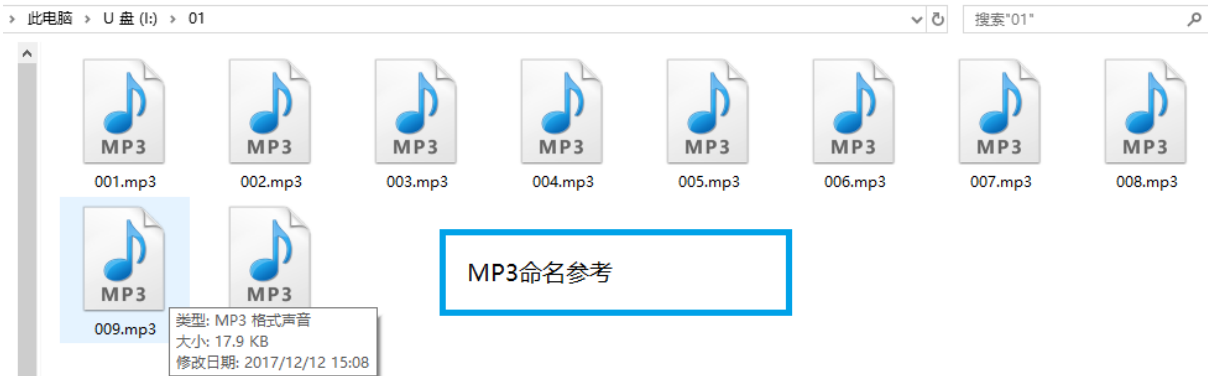
例如：这样命名“010 你是我的小苹果.MP3”，“101 忘了没有.MP3”等等。



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[上下图]





### 3.4.6、根目录曲目全部循环播放

循环播放	7E FF 06 11 01 00 01 xx xx EF	从第一首曲目到最后一首曲目重复循环
	7E FF 06 11 01 00 00 xx xx EF	发送命令后，待当前曲目播放完就停止

- (1)、此指令是按顺序并且循环播放设备里面存储的所有语音文件，是按照物理顺序循环播放，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。
- (2)、在循环播放后，下一曲也是按顺序的，上一曲也是按顺序的，就算是暂停再播放也是按顺序的，只能切换状态来改变。如果不改变状态，从第一首曲目到最后一首曲目重复循环。

### 3.4.7、播放停止指令

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

在芯片的播放过程中，可以发送此指令停止所有播放任务。

### 3.4.8、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 0B FE D9 EF	指定 11 文件夹循环播放
	7E FF 06 17 00 00 63 FE 81 EF	指定 99 文件夹循环播放

- (1)、文件夹的命名方式必须是“01”到“99”。不可以超过99。
- (2)、一旦指定文件夹循环之后，可以使用播放/暂停/上一曲/下一曲，这些操作命令都不会打断当前的文件夹循环播放状态。也就是说发送下一曲指令之后，还是会循环当前的文件夹。
- (3)、用户可以发送停止指令来结束循环播放，返回至触发播放状态。

### 3.4.9、随机播放设备文件

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



- (1)、此指令是随机播放设备里面存储的所有语音文件，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。
- (2)、在随机播放后，下一曲也是随机的，上一曲也是随机的，就算是暂停再播放也是随机的，只能切换状态来改变。随机播放还是指定文件夹循环播放，如果不给单曲循环播放指令，无论是上一曲或者下一曲还是随机播放或者指定文件夹循环播状态，就退不出这个状态，除非断电再上电。

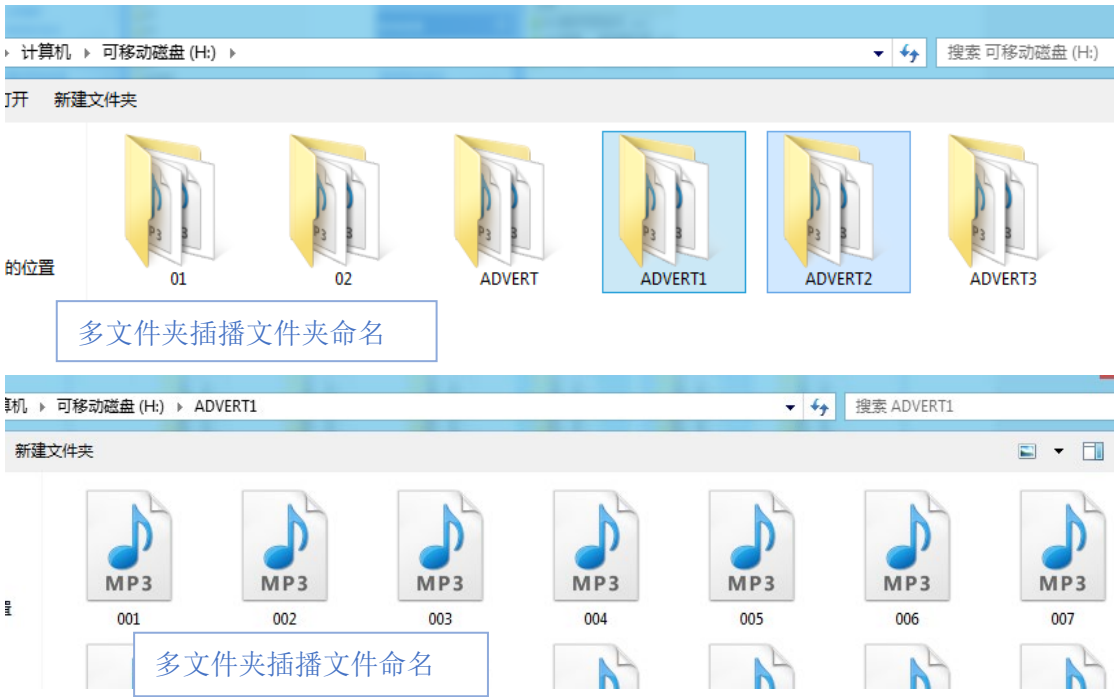
3.4.10、对当前的曲目设置为循环播放

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。

3.4.11、多文件夹插播

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。

### 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：

- 查询当前在线的设备
- 播放状态查询指令
- 当前设备的音量查询

#### 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
--------	-------------------------------

- (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  
DL=0x0C = 0000 1100 代表了 PC 和 FLASH 在线  
DL=0x08 = 0000 1000 只代表了 FLASH 在线  
如果 DL=0x0F= 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

停止播放	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
正在播放	7E FF 06 42 00 04 01 xx xx EF	FLASH 正在播放
	7E FF 06 42 00 08 01 xx xx EF	在 PC 连接下载模式
	7E FF 06 42 00 10 08 xx xx EF	芯片处于睡眠

- (1)、芯片在解码过程中会有 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)、如果返回的数据为 7E FF 06 42 00 04 00 xx xx EF 代表的的意思详解如下：

DH = 0x04 --- 代表的是当前设备是 FLASH 盘  
DL = 0x00 --- 代表的是当前处于播放停止状态

(4)、如果返回的数据为 7E FF 06 42 00 01 02 xx xx EF 代表的意思详解如下：

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

### 3.5.3、当前设备的音量查询

查询当前音量	7E FF 06 43 00 00 00 xx xx EF	查询当前设备的音量
--------	-------------------------------	-----------

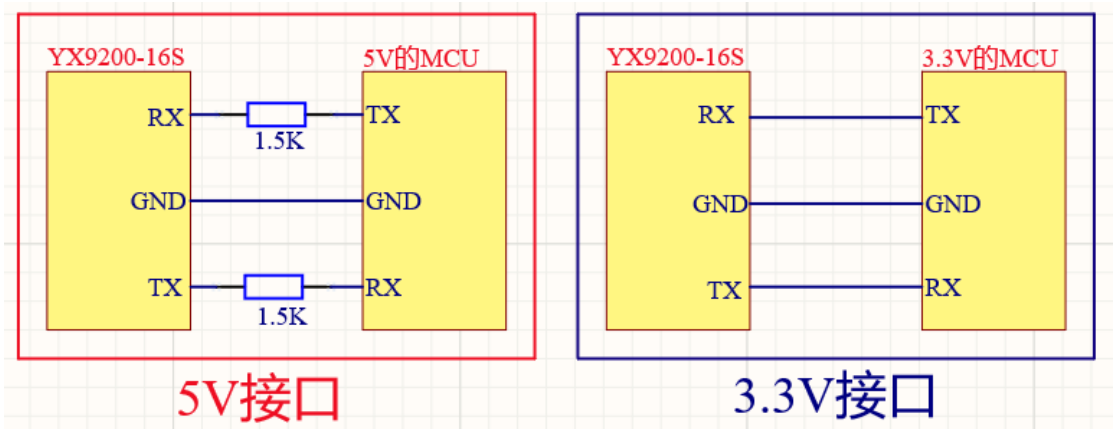
用户可以对当前的设备进行音量的查询。支持设备的音量查询。  
如果返回的数据为 7E FF 06 43 00 00 05 xx xx EF 代表的意思详解如下：05 表示当前音量为 05 级音量，一共只有 30 级音量级，所以最多返回 1E，0x1E=30 级音量。

## 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，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 1.5K 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。



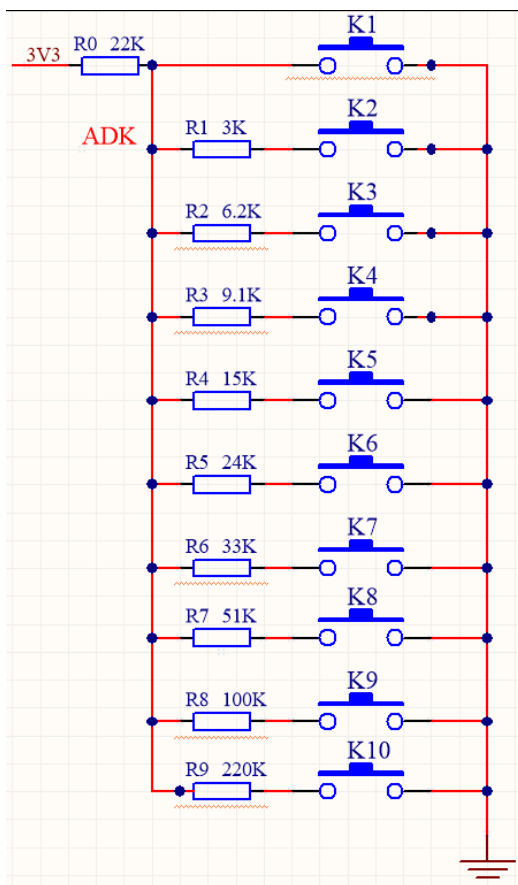
- (4)、芯片在 5V 和 3.3V 的系统中均正常的测试过，一切正常。均采用的是直连的方式，并没有串 1.5K 的电阻，一般的芯片都是能够兼容 3.3V 和 5V 的电平。
- (5)、但是用户在实际的产品开发过程中，一定要严格的测试，留意电平的转换。强烈建议用户单片机 5V 的情况下，TX 和 RX 都能各串 1 个 1.5K 左右的电阻。

## 4.2 按键接口

芯片我们采用的是 AD 按键的方式，这样做的好处是可以接多个按键，用文件后缀.txt 的文件来做为按键功能设定和选择；按原理图定义管脚，11 脚为 ADK；可以根据自己的需要来选择多少个按键，最多有 10 个按键。

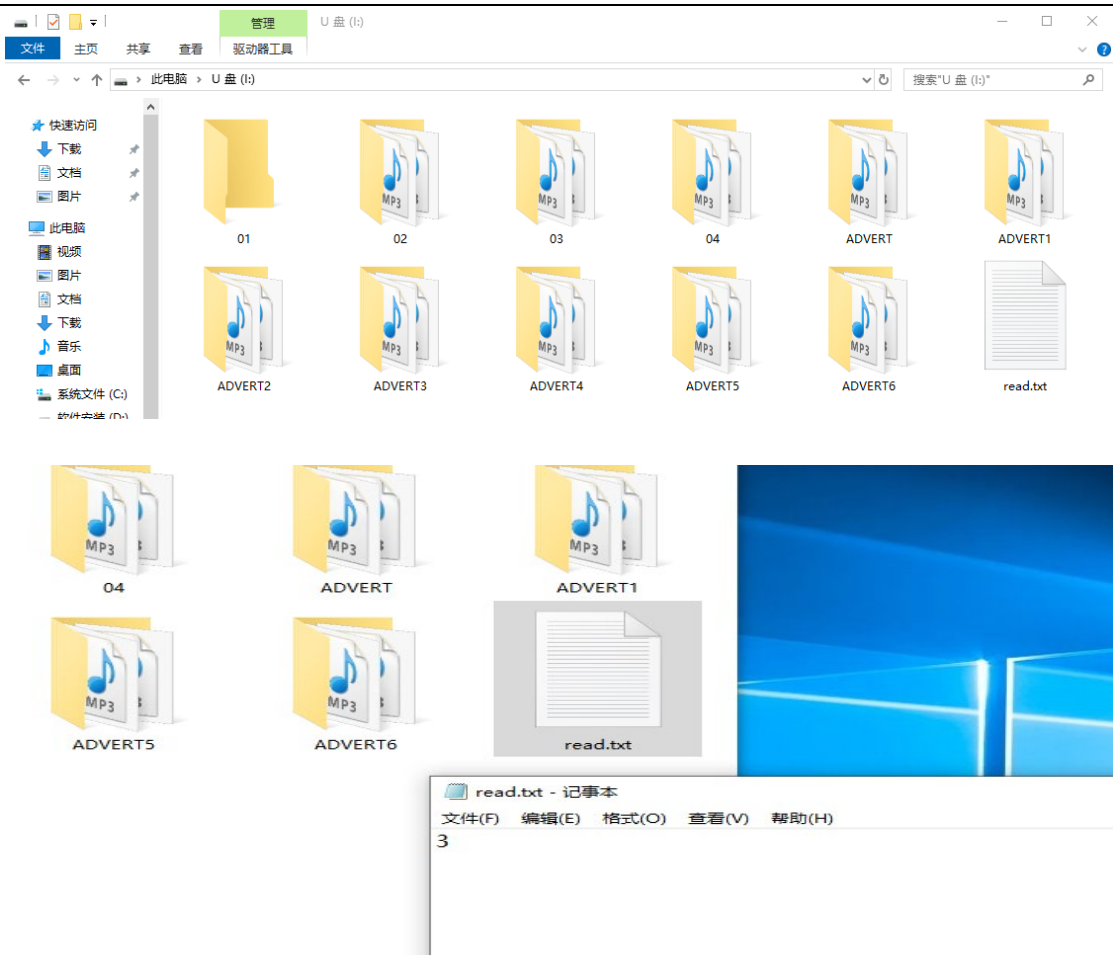
- 按键接口的接法
- 通过文件来配置按键的功能

### 4.2.1、按键接口的接法



### 4.2.2、通过 TXT 文件来配置按键功能

由于我们的方案支持 FAT 文件系统，所以设备中的文件的内容，可以很轻易的读出来，所以我们就扩展了此功能，通过设备内的 xxx.txt 文件来读取按键的配置，这个 xxx.txt 文件可以通过 USB 接口连接电脑来创建，后期我们还会扩展更多的功能，请拭目以待。



- (1)、配置文件必须为 xxx.txt，这样才能被系统所识别，其它的命名系统则不识别，原理是这样的，芯片上电之后初始化文件系统，会首先搜索 xxx.txt 这个文件，搜索到之后，对里面的数据进行读取和处理，请严格的按照我们给出的配置参数的方法。
- (2)、配置文件在同一个设备中，最多只能有一个，可以没有，但是一定不能出现两个或者以上，否则会导致识别出错。
- (3)、如果要配置 FLASH 模式下的按键功能，必须要要在 FLASH 里面建立一个 xxx.txt 文件。如果要配置 TF 卡模式下的按键功能，也要在 TF 卡里面建立一个 read.txt 文件。U 盘也是一样，也就是说任何一个设备需要配置按键功能，都需要建立一个 read.txt 在相应的设备中。

### 4.2.3、按键配置功能的详细解释

**“0” 配置【脉冲可打断】：**10 个触发口各对应 10 个声音，检测下降沿有效（即按下就响），按下响完曲就停，如果播放过程中，反复按是要打断从头开始播放，别的按键能打断。如果按键有短路地的，插 USB 要能出来移动硬盘，如果某个按键一直按着，别的按键也要有效，能播放最新的按键指令。如果内存里没有配置文件，就默认此模式。短路 K1 到地，上电能响 K1 的声音。

**“1” 配置【电平保持可循环】：**10 个触发口各对应 10 个声音，检测下降沿有效（即按下就响），一直按一直循环，松开马上停，如果播放过程中，别的按键有效，则切换别的按键循环播放。短路任意触发口插 USB 要能出来移动硬盘。短路 K1 到地，上电能循环响 K1 的声音，直到掉电。

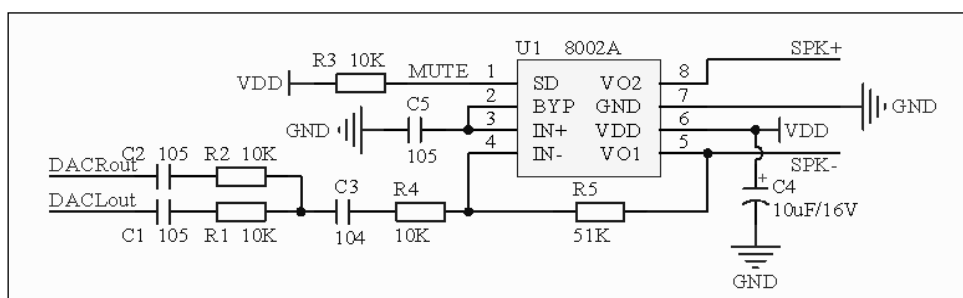
**“2” 配置【脉冲不打断】：**10 个触发口各对应 10 个声音，检测下降沿有效（即按下就响），按一下

播放，再按无效，要等播放完才有效，如果播放过程中，别的按键无效。播放过程中插 USB 要能出来移动硬盘。

**“3” 配置：**K1 短路上电就全部循环，大循环/暂停。K2 短路上电响一遍内存声音后停，循环一遍/暂停。K3 到 K10 都是空白。

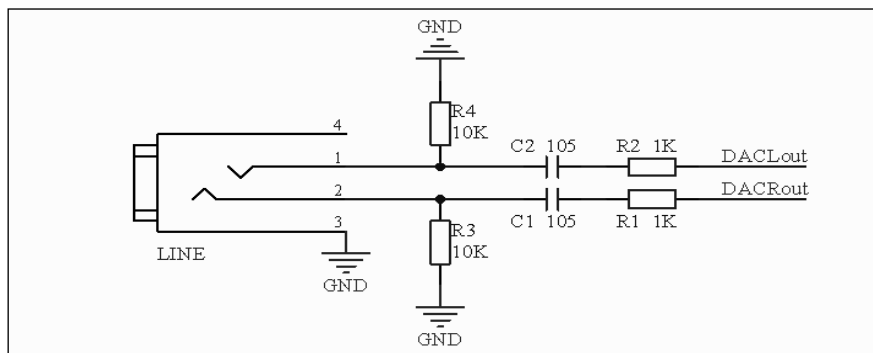
**“4” 配置【脉冲播放单曲循环】：**10 个触发口各对应 10 个声音，检测下降沿有效（即按下就响），脉冲播放一次，检测到快要播放完的时候，还有触发，继续播放，如果播放过程中，自身的触发不能打断播放，别的按键信号能打断播放新的。短路任意触发口插 USB 要能出来移动硬盘。

### 4.3 外接功放芯片电路



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 外接耳机电路



这里 R1 和 R2 为限幅电阻，防止外部音源幅度过大影响系统的稳定性，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；R3 和 R4 预留电阻给大功放设计用。

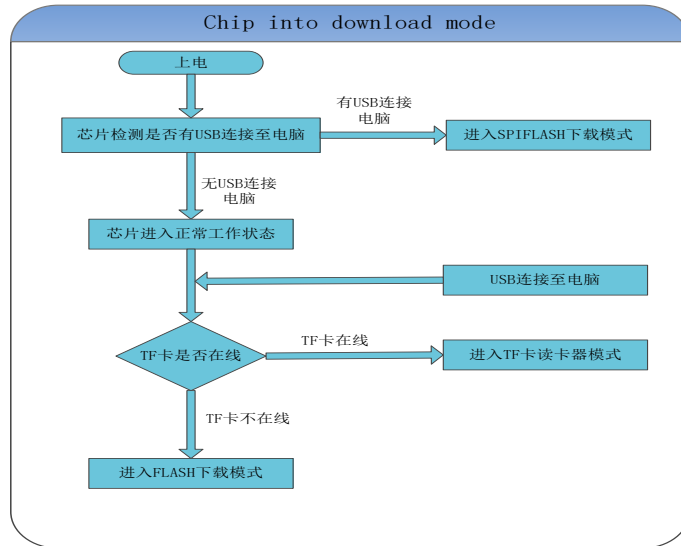
### 4.5 USB 更新语音说明

我们的芯片可以使用手机充电线直接更新语音，方便、灵活。这里分两种设备

- ◆ USB 更新 SPIFLASH 的语音模式
- ◆ USB 更新 TF 卡内的语音模式

其实 SPIFLASH 和 TF 卡，在插入 USB 连接电脑，原则上是一致的，使用的都是 MASSSTORAGE 协议，但是目前技术上面，暂时还没办法实现插上 USB 连接电脑，同时显示 TF 卡和 SPIFLASH 的盘符功能，这里就分为两种操作，针对芯片说明：

- (1) 芯片一上电检测到 USB 连接电脑，则进入 SPIFLASH 的读卡器功能
- (2) 芯片上电没检测到 USB 连接电脑，进入正常工作模式。如果有 USB 连接至电脑，则进入 TF 卡读卡器模式。如果此时没有 TF 卡在线，则还是进入 SPIFLASH 的盘符模式。
- (3)

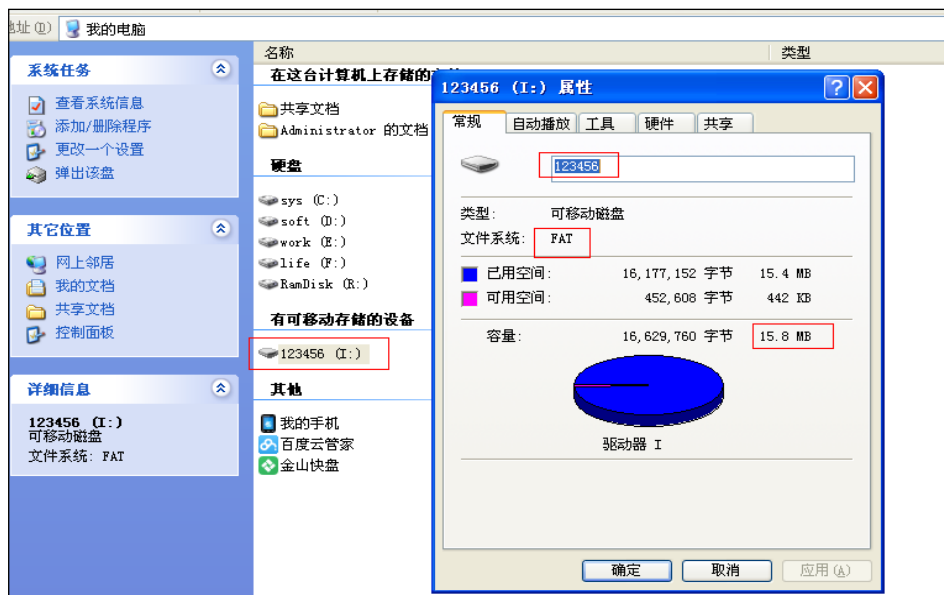


#### 4.5.1 USB 更新 SPIFLASH 的语音详细说明

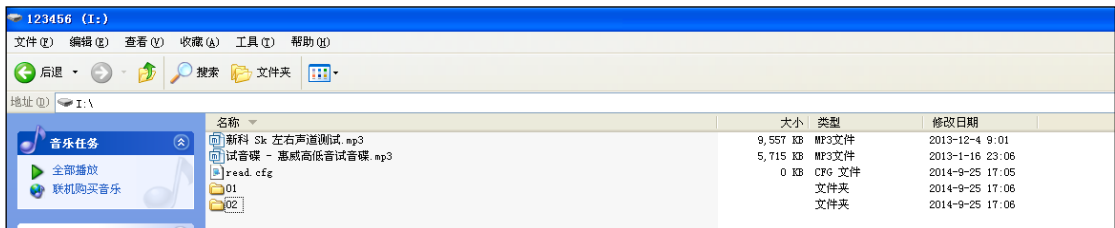
我们的芯片支持 USB 更新语音功能，方便、灵活。我们的优势如下

- ◆ 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、USB 连接上我们芯片之后,可以以 SPIFLASH 作为存储介质，如下图



- (1) 可以从上图看到 FLASH 的总容量为 15.8M 字节。已经使用的空间为 15.4M 字节。虚拟出来的设备的文件系统的为 FAT 格式。FAT 文件系统占的存储空间为 442K
- (2) 进入设备之后，如下图



可以很清晰的看到设备里面的文件，以及文件名称。可以像操作 U 盘或者读卡器一样操作 FLASH。只是速度会比他们慢。

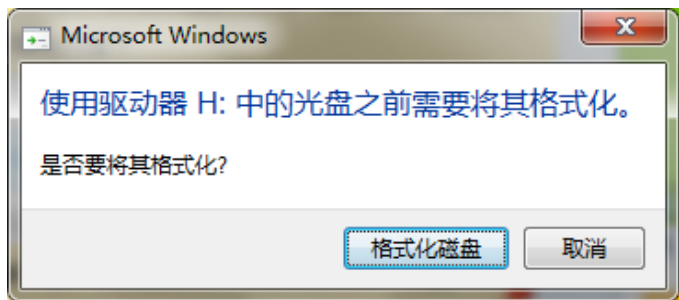
- (3) 无论用户使用多大容量的 FLASH，我们芯片都是支持的，并且内部已经做了自动识别，无需用户操心，用户也仅仅需要根据自己的需求来确定 FLASH 的容量和型号。
- (4) 目前经过我们反复的测试和验证，SPIFLASH 支持最大的容量为 16M 字节，对应型号 W25Q128。因为再大容量的 FLASH。由于技术的原因暂时还没有突破，后续会直接更新，请用户知晓。

## 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。

- ◆ 将新的空白的 FLASH 焊接在板子上面
- ◆ 通过 USB 接口对空白的 FLASH 进行枚举和格式化
- ◆ 格式化完毕，就可以像使用 U 盘一样使用
- ◆ 以 16M flash 操作为例，具体操作如下：

插入 USB 链接电脑



点击“格式化磁盘”出现如下：



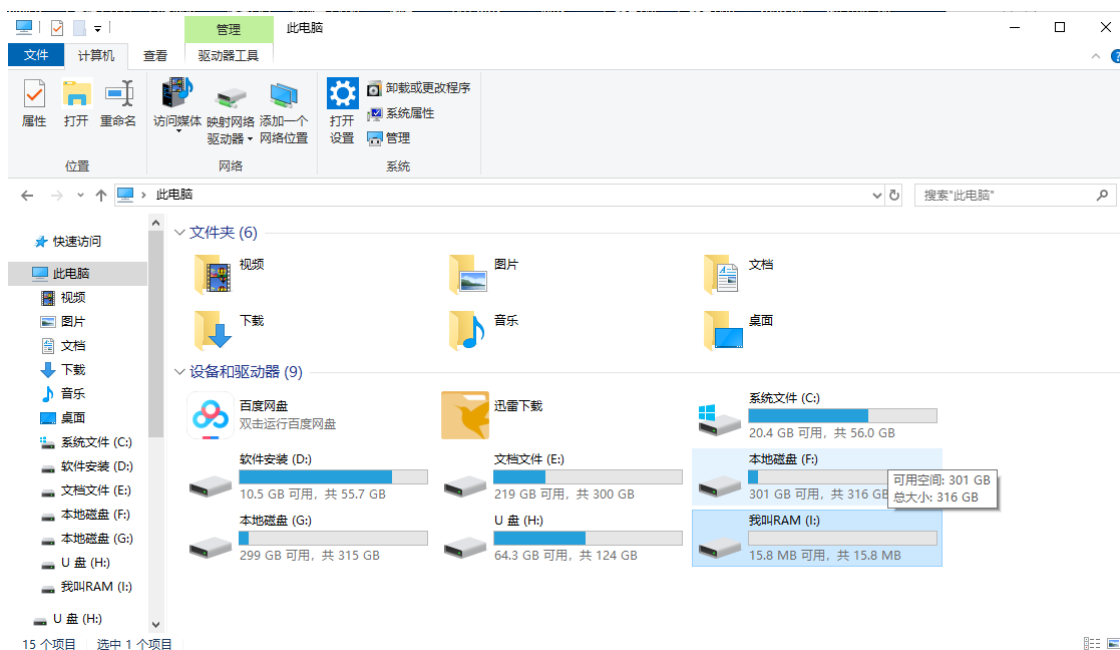
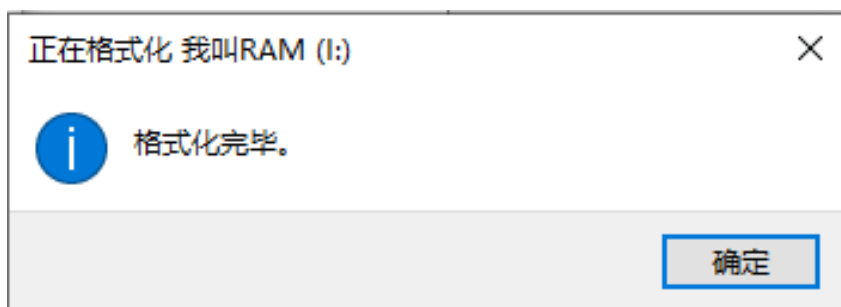
格式化 我叫RAM (I:)

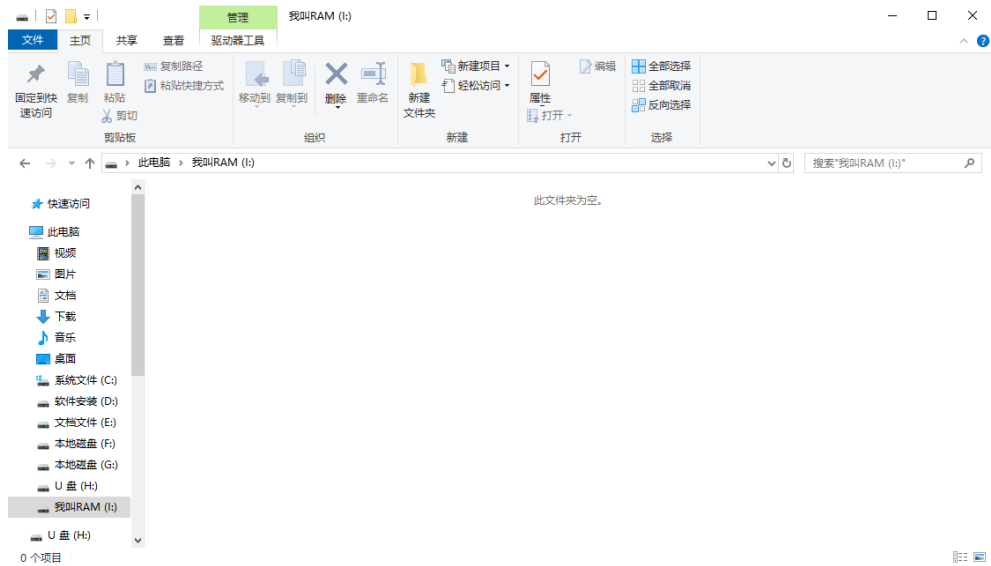


警告: 格式化将删除该磁盘上的所有数据。  
若想格式化该磁盘, 请单击“确定”。若想退出, 请单击“取消”。

确定

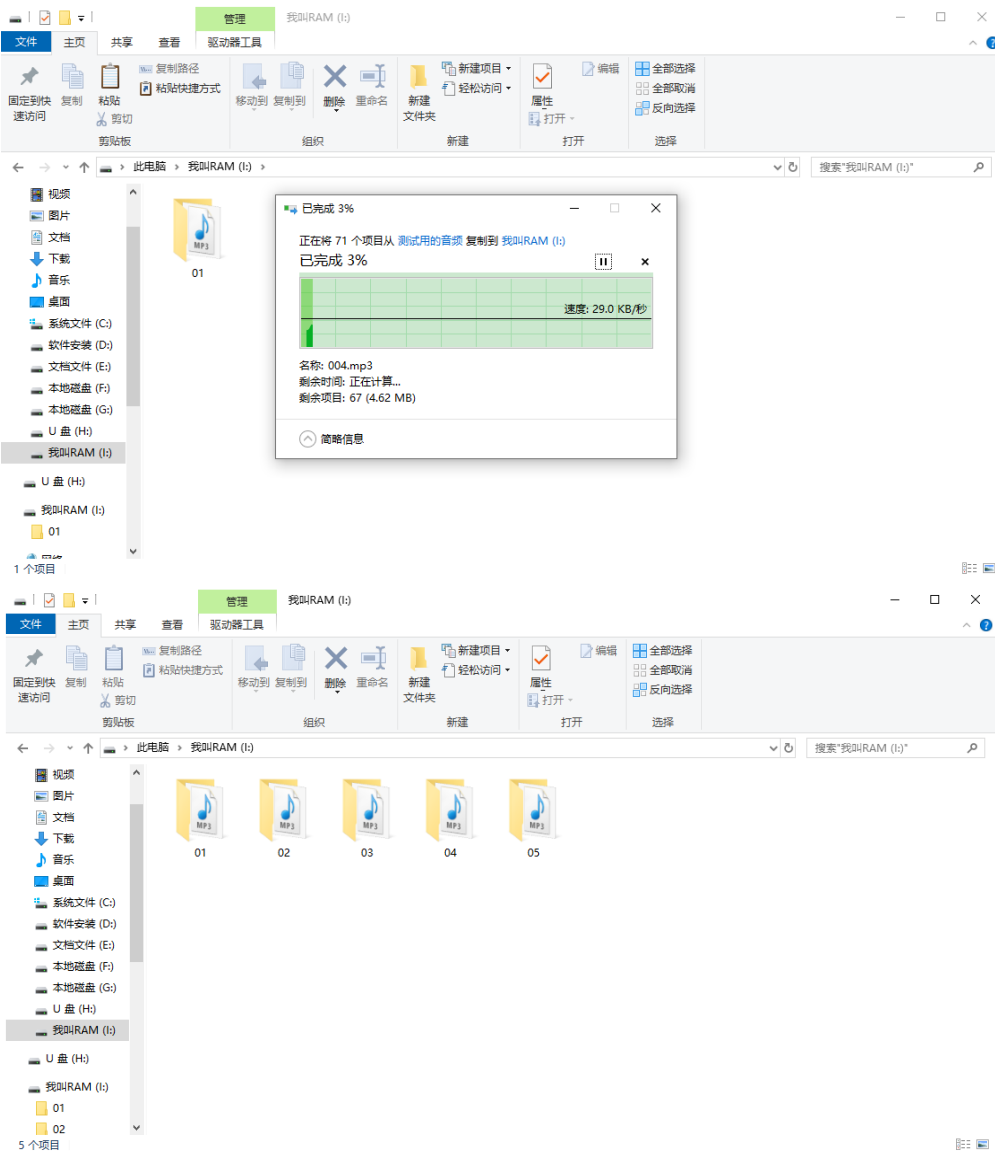
取消





如图所示，这样可以当做一个 U 盘正常使用。当 flash 第一次使用时需要格式化的操作，操作过一次后，后面若未对 flash 做格式化或其他删除动作，则不需要重复上面动作。

## 加载语音





如上图所示，当做 U 盘直接拖入语音即可。操作和 SD 卡，U 盘完全一致。

1、我们的方案目前最大支持 64M 字节的 FLASH。换算为 FLASH 一般的型号，如：W25Q512.但是市面上目前量产的 FLASH 为 32M 字节，也就是 W25Q256。请用户注意，并且封装还不是 SOP8 的。

型号	容量	封装
W25Q80	1Mbyte	SOP8L[宽体]
W25Q16	2Mbyte	SOP8L[宽体]
W25Q32	4Mbyte	SOP8L[宽体]
W25Q64	8Mbyte	SOP8L[宽体]
W25Q128	16Mbyte	SOP8L[宽体]
W25Q256	32Mbyte	SOP16[宽体]
W25Q512	64Mbyte	SOP16[宽体]

2、我们的芯片支持自动识别 FLASH 的容量大小。所以用户无需关心，只需要按照自己的需求使用合适大小的 FLASH 即可。

3、因为空白的 FLASH 里面什么都没有，所以拿到空白的 FLASH 第一件事情，就需要对 FLASH 进行格式化。将文件系统的列表写入 FLASH 中。

4、等到格式化成功之后，再拔掉 USB，再插上 USB 之后，就可以进行 SPIFLASH 的读写了。另外不同容量的 SPIFLASH，格式化的所需要的时间长度是不一致的。也就是说，FLASH 的容量越大，格式化所需要的时间越长。

## 6. 芯片使用注意事项

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

### 6.1 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
VIL	Low-Level Input Voltage	-0.3	-	0.3*VDD	V	VDD=3.3V
VIH	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
VOL	Low-Level Output Voltage	–	–	0.33	V	VDD=3.3V
VOH	High-Level Output Voltage	2.7	–	–	V	VDD=3.3V

## 6.2 应用中的注意点

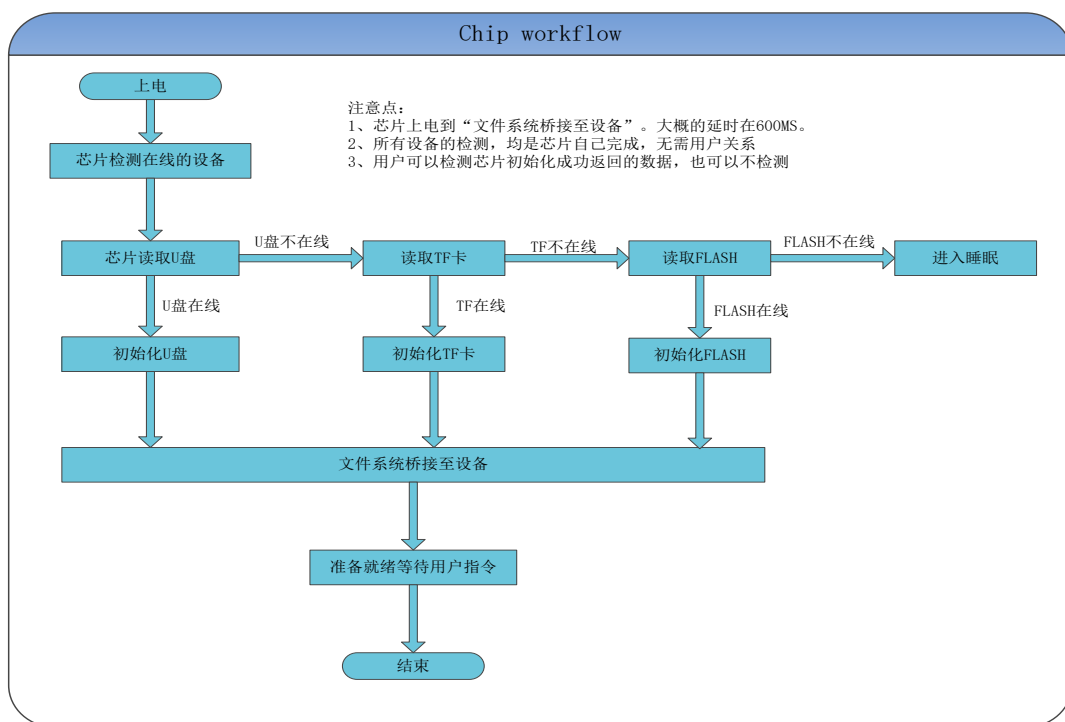
- (1) 芯片对外的接口均是 3.3V 的 TTL 电平，所以在硬件电路的设计中，请注意电平的转换问题。
- 另外在强干扰的环境中，请注意电磁兼容的一些保护措施，GPIO 采用光耦隔离，增加 TVS 等等。
- (2) 串口通信，在一般的使用环境下，注意好电平转换即可。如果强干扰环境，或者长距离的 RS485 应用，那么请注意信号的隔离，严格按照工业的标准设计通信电路。可以联系我们，我们提供设计参考。
- 我们支持音频文件的采样率最低为 8KHZ。也就是说低于 8KHZ 的音频文件是不支持的，不能正常解码播放。用户可以使用音频处理软件，提高音频文件的采样率来解决这个问题。
  - 芯片在睡眠状态的电流在 10MA 左右，功耗稍大。如果使用在低功耗场合，请用户控制模块或者芯片的供电。这样可以减小芯片的功耗。
  - 该芯片支持 MP3、WAV 二种主流的音频格式。

## 6.3 串口注意事项点

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：

- 芯片上电的流程
- 串口编程参考的说明
- 串口操作需要延时的注意事项
- 串口协议的校验的说明
- 串口协议的校验的算法详解
- 外接 MCU 的晶振选择说明
- 芯片的播放说明[物理顺序等等]

### 6.3.1、芯片上电的工作流程图



- (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。

### 6.3.2、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

- (1) 第一部分是测试版的测试代码，相关的串口操作比较全面
- (2) 另一个是基本版，只是指定曲目的范例。请用户耐心消化

### 6.3.3、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。
- 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、0x14]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，会导致芯片的工作不正常。

### 6.3.4、校验的重要说明

1、针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：

指定曲目（带校验）	7E FF 06 08 00 00 01 FE F2 EF	根目录循环播放第一曲
指定曲目（无校验）	7E FF 06 08 00 00 01 EF	根目录循环播放第一曲

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，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。

### 校验的计算说明

发送的指令，去掉起始和结束。将中间的 6 个字节进行累加，最后取反再+1 得到校验码。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。

举例说明：

举个例子，如果我们播放下一曲，就需要发送：7E FF 06 01 00 00 00 FE FA EF

数据长度为 6，这 6 个字节是 [FF 06 01 00 00 00] 不计算起始、结束、和校验。校验字节为 FE FA。

**发送端得到校验码的过程：**

相加的过程： $FF+06+01+00+00+00 = 0x0106$

取反加 1 的过程： $0x0106$  取反 = FEF9

再用  $FEF9 + 1 = 0xFEFA$ ，和我们的校验值对比一下。

**接收端进行校验的过程：**

相减的过程： $0 - 0x0106 = 0xFEFA$ 。再和我们的结果比较一下。

最后可以参考一下我们给出的参考例程，或者直接移植我们给出的函数，就很容易理解。

### 6.3.6、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。

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 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。

### 6.3.7、指定播放的说明

- (1)、指定曲目播放[按照物理顺序]
- (2)、指定曲目播放[按照文件夹和文件名的名称]

#### 1、按照物理顺序指定曲目播放：

需要按照下图的方法往 TF 卡中拷贝歌曲。先拷贝进设备的，编号为第一首，以此类推。

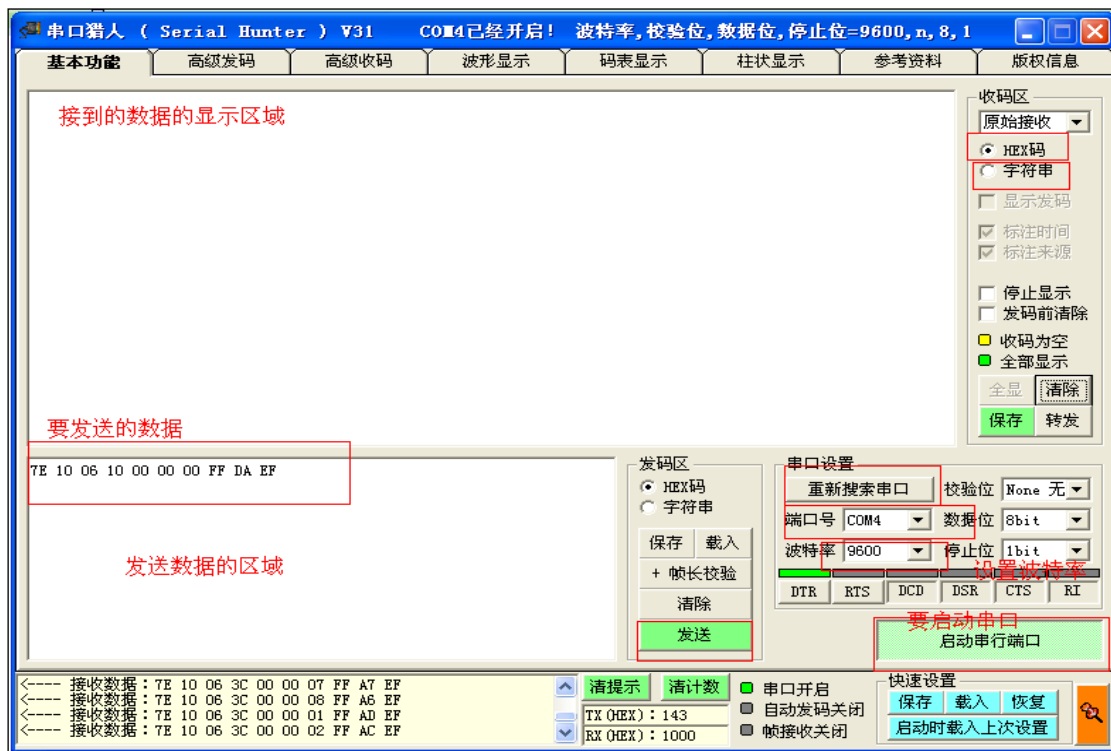
#### 2、指定文件夹和文件名播放

详情请参考芯片数据手册的“3.4.5 指定文件夹播放”章节

#### 3、指定 MP3 文件夹里面的文件名播放

详情请参考芯片数据手册的“3.4.5 指定 MP3 文件夹播放”章节

### 6.3.8、串口调试说明



- 1、首先现在网上下载一个软件”串口猎人”，为什么选择这个，因为它有自动搜索串口的功能，十分好用。打开之后的主界面如上图所示，可以看到红色部分，依次设置即可。

- 2、打开软件，首先要搜索串口，找到指定的端口之后，指定“波特率”，我们的芯片默认的波特率为 9600，最后就是“启动串行端口”，这样软件就配置好了。这里有两个概念需要明确一下。
- 3、第一是“HEX 码”，我们默认是这个，这个是用来显示数据的。所以必须设置这里第二是“字符串”，这个是用来显示打印字符的，我们这里用不到。
- 4、软件配置 OK 之后，将需要的指令复制到发送区域即可，具体的指令请参照芯片的数据手册。
- 5、如果芯片的数据手册没有的测试指令的话，请自行计算，尤其需要注意的是“校验和”这两个字节，如何计算不对的话，芯片是不接受指令的，同时会返回相应的错误指令。

### 6.3.9、校验代码的移植

我们这里的说明，争对的是用户的 MCU 给我们的芯片发送控制指令

#### 1、发送端 —— 用户的 MCU

```
/******
```

- 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 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
}
```

DoSum(&Send\_buf[0],6); 这里是指对 Send\_buf[0]---Send\_buf[5]这 6 个字节进行校验的算法生成校验的两个字节，并且存储于 Send\_buf[6]和 Send\_buf[7]中

```
/******
```

- 功能描述： 求和校验
- 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); //得到高字节
*(Str+i+1) = (INT8U) (xorsum & 0x00ff); //得到低字节
}
```

## 2、接收端，芯片发送数据给用户的 MCU

```
If(一帧数据接收完毕)
{
    for(i=0; i<*(pi+1); i++) //这里 pi 指向的是接收缓冲区，*(pi+1)是获取数据长度。
    {
        xorsum = xorsum + pi[i] ; //将接受到的数据进行累加
    }
    xorsum1 = ((u16)((*(pi+i)<<8) | (*(pi+i+1))); //这里是接收到的校验字节，16 位
    xorsum = xorsum + xorsum1; //将接收到的校验字节和自己算的校验数据相加
    if(!xorsum)
    {
        Uart_Task(pi); //串口处理--对接收到的指令进行处理
    }
    else
    {
        ErrorStatus = ERROR_CheckSError ; //接收校验出错
        //校验码出错之后的处理
    }
}
```

3、用户无论使用的是什么 MCU，这两个函数均可以平行的移植到自己的程序中。

# 7、PC 端串口调试指令举例

用户可以通过电脑端的串口软件，对功放板进行测试。我们的功放板串口为 TTL 电平，请注意电平转换

- 控制指令
- 查询参数指令

## 7.1 、控制指令

功能说明	发送的命令	备注
下一首	7E FF 06 01 00 00 00 EF	
上一首	7E FF 06 02 00 00 00 EF	

指定曲目	7E FF 06 03 00 00 01 EF	指定第一首播放
	7E FF 06 03 00 00 02 EF	指定第二首
	7E FF 06 03 00 00 0A EF	指定第 10 首
音量加	7E FF 06 04 00 00 00 EF	
音量减	7E FF 06 05 00 00 00 EF	
指定音量	7E FF 06 06 00 00 1E EF	指定音量为 30 级
循环播放	7E FF 06 08 00 00 01 EF	循环播放第一首
	7E FF 06 08 00 00 02 EF	循环播放第二首
	7E FF 06 08 00 00 0A EF	循环播放第十首
设备复位	7E FF 06 0C 00 00 00 EF	
播放	7E FF 06 0D 00 00 00 EF	
暂停	7E FF 06 0E 00 00 00 EF	
指定文件夹文件名播放	7E FF 06 0F 00 01 01 EF	01 文件夹的 001 曲目
	7E FF 06 0F 00 02 01 EF	02 文件夹的 001 曲目
	7E FF 06 0F 00 01 02 EF	01 文件夹的 002 曲目
停止播放	7E FF 06 16 00 00 00 EF	停止
文件夹循环	7E FF 06 17 00 00 02 EF	02 文件夹循环
	7E FF 06 17 00 00 01 EF	01 文件夹循环
单曲循环	7E FF 06 19 00 00 00 EF	单曲循环播放
	7E FF 06 19 00 00 01 EF	单曲循环关闭
循环播放全部	7E FF 06 11 00 00 01 EF	开始循环播放全部
	7E FF 06 11 00 00 00 EF	停止循环播放全部

## 7.2 、查询指令

功能说明	发送的命令[不带校验]	备注
查询在线的设备	7E FF 06 3F 00 00 00 EF	
查询当前状态	7E FF 06 42 00 00 00 EF	
[查询音量]	7E FF 06 43 00 00 00 EF	查询当前音量级别
总曲目数	7E FF 06 49 00 00 00 EF	查询当前设备的总曲目数
当前曲目	7E FF 06 4D 00 00 00 EF	查询当前播放的曲目

## 8、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

悦欣电子能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。

## 9、厂家联系方式

产品使用中如遇到问题或对我们的产品有任何的疑问，请及时联系我们：

公司名称：广州悦欣电子科技有限公司

联系电话：020-36776060

手机号码：18925103846（微信同号）

Q Q号码：3229894586

淘宝店铺：[https //mk020.taobao.com](https://mk020.taobao.com)

公司邮箱：[lur@yxin18.com](mailto:lur@yxin18.com)

公司网址：[www.yxin18.com](http://www.yxin18.com)

公司地址：广东省广州市花都区商业大道334号

## 10、文件版本信息

版本号	文档说明	修改日期
V1	初期建档	2021-03-12
V2	增加按键配置功能、修改错误	2021-11