



广州市九芯电子科技有限公司

Guangzhou nine chip electron technology co.,Ltd

N9200 MP3 模块

网址: www.vic18.com



广州市九芯电子科技有限公司

GUANGZHOU NINE CHIP ELECTRON TECHNOLOGY CO.,. LTD.

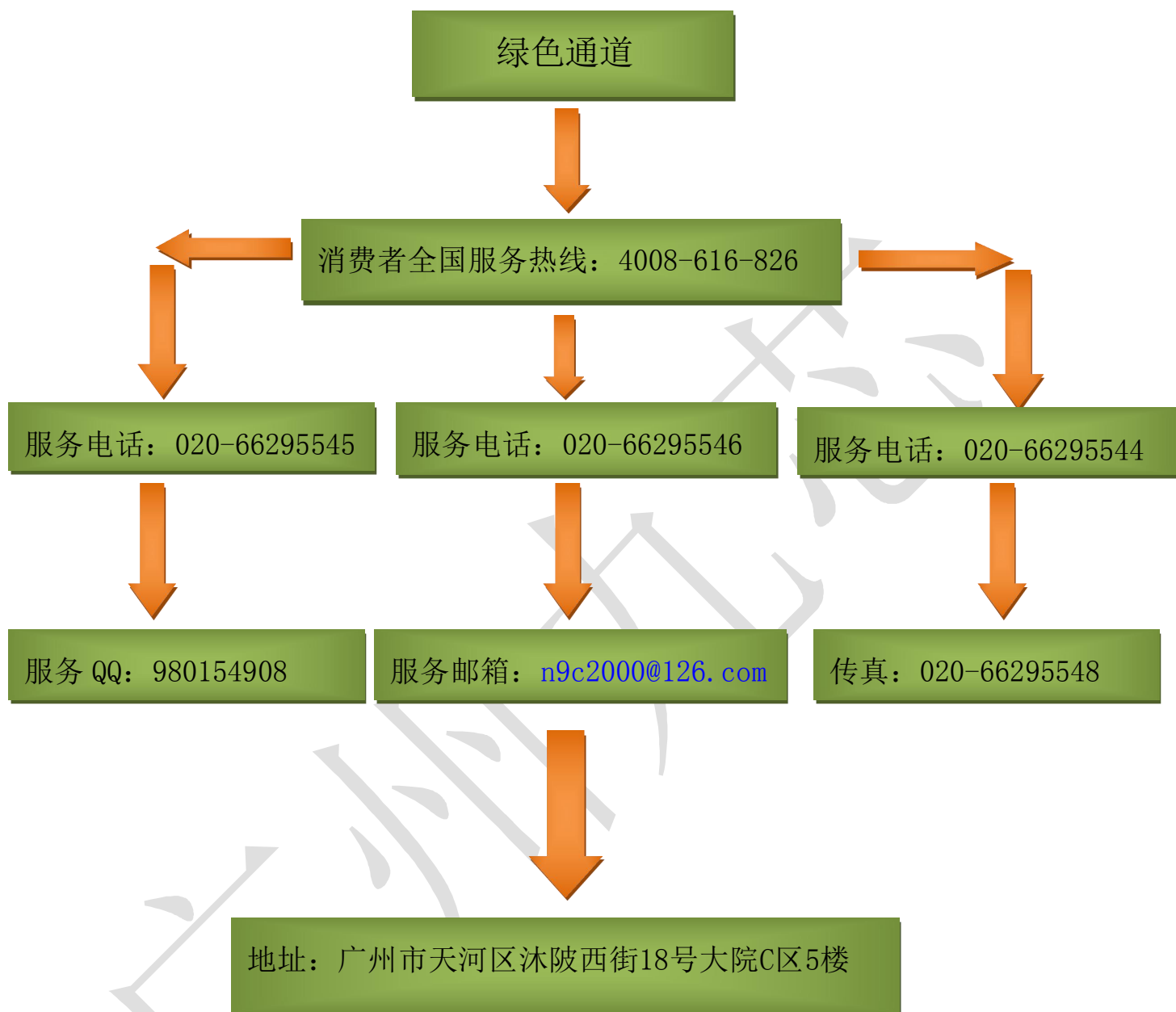
使用说明书

N9200 MP3 模块

(包括 N9200A-16P, N9200B-28P)

文档版本 V1.3

发布日期 2016. 4. 14



备注:

- 1、图片仅供参考, 请以实物为准, 九芯公司保留对产品外观及设计改进的权利, 恕不另行通知。
- 2、广州市九芯电子为客户提供全方位的技术支持, 客户可与就近的九芯电子办事或者代理机构联系, 也可直接与公司总部电话联系。



目录

1. 概述	
1.1. 简介	4
1.2 特征	4
1.3. 应用场合	5
2. 模块使用参数	6
2.1 模块结构图	6
2.2. 电路参数	6
3. 模块管脚图	7
4. 模块管脚说明	8
4.1 表一、N9201A-16P MP3 模块管脚说明	8
4.2表二、N9201B-28P MP3模块管脚图	9
4.3 硬件参数以及环境极限参数说明	10
5. 文件命名以及控制方式说明	10
5.1 ADKEY 按键控制	10
5.2 MCU 一线串口控制	12
5.3 文件排列与命名规则	13
5.3.1 一线串口时序	13
5.3.2 程序范例	14
5.4 UART 串口通信协议	15
5.4.1. 通信数据格式:	15
5.4.2. 容错要求:	16
5.4.3. 协议约定:	16
5.4.4 CMD 通信指令控制列表(16 进制)	17
6 模块内部电路图	24
7. 模块封装尺寸图	24
8. 文件夹路径格式说明及校验码的算法	25
8.1 文件夹路径要求	26
8.2 校验和计算方法	27
9 注意事项	27
9.1GPIO特性	28
9.2应用中的注意事项	28
9.3 串口操作	29
9.3.1 串口操作需要延时的注意事项	29
9.3.2 串口编程参考的说明	30
9.3.3 串口编程需要适当延时的注意点	
10.免责声明	30
10.1 开发预备知识	30
10.2 EMI 和 EMC	30
10.3 修改文档的权力	30
10.4 ESD 静电放点保护	30
11、历史版本	31



1、概述

1.1 简介

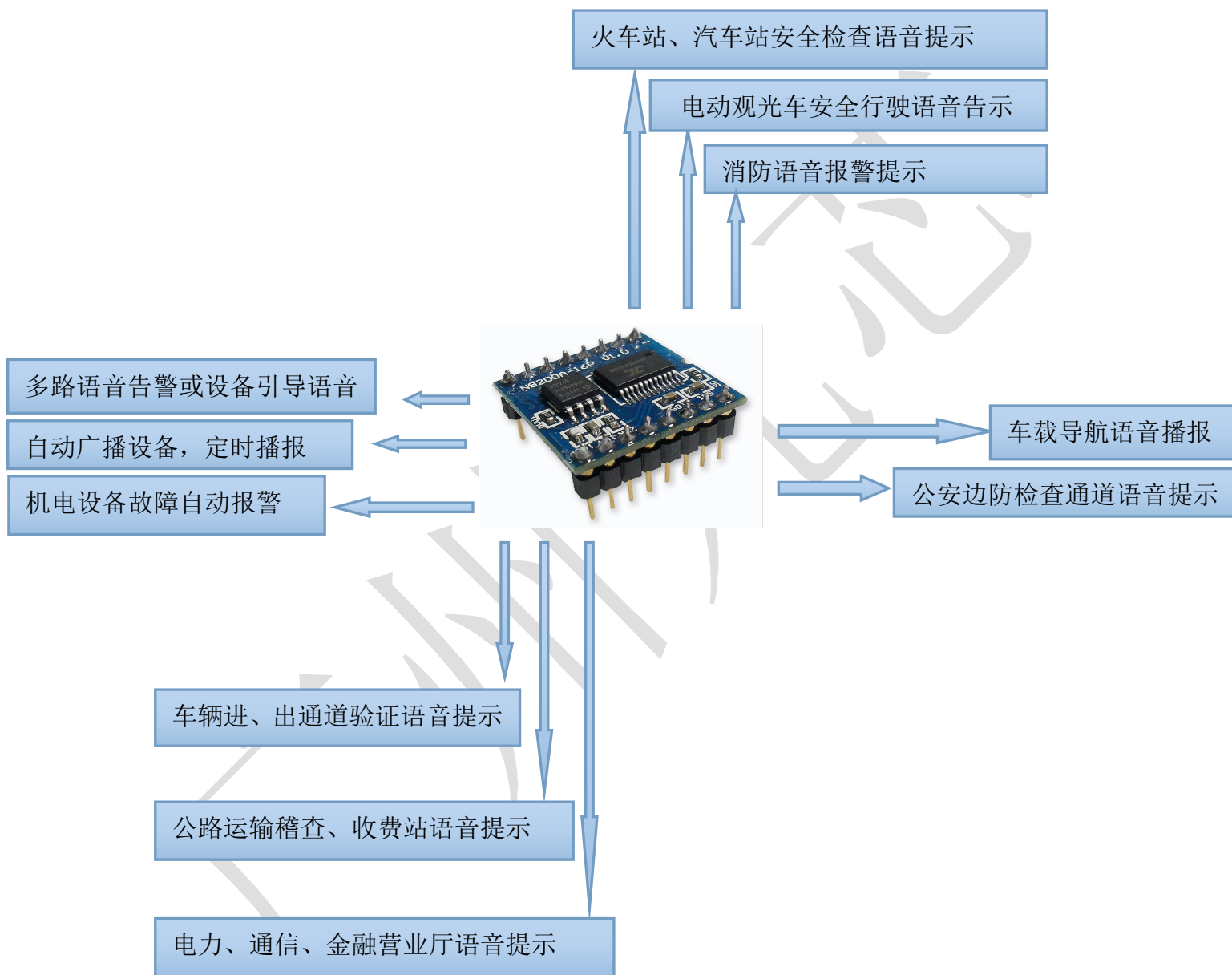
N9200B 是一个提供串口的 MP3 芯片，完美的集成了 MP3、WMV 的硬解码芯片。它包括了 N9200A 与 N9200B 音乐 MP3 模块，同时软件支持 TF 卡驱动，支持电脑直接更新 spi flash 的内容，支持 FAT16、FAT32 文件系统。通过简单的 UART 串口指令或一线串口指令即可完成播放指定的音乐，以及如何播放音乐等功能，无需繁琐的底层操作，音质优美，使用方便，稳定可靠是此款产品的最大特点。另外该芯片也是深度定制的产品，专为固定语音播放领域开发的低成本解决方案。

1.2 特征

- 支持采样率(KHz):8/11.025/12/16/22.05/24/32/44.1/48。音质优美，立体声。
- 24 位DAC 输出，内部采用DSP硬解码，非PWM输出，动态范围支持90dB，信噪比支持85dB
- 完全支持FAT16、FAT32 文件系统，最大支持32G的TF 卡，支持32G的U盘
- 多种控制模式，UART串口模式、一线串口模式、AD按键控制模式
- 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，支持跨盘符插播，支持插播提前结束
- 指定时间播放
- 指定盘符播放，指定曲目播放
- 30级音量可调，5种EQ可调（NORMAL—POP—ROCK—JAZZ—CLASSIC）
- 指定路径播放（支持中英文）
- 文件夹切换
- 指定时间段复读
- 支持电脑声卡控制，支持USBmass storage
- 支持触发不可以打断播放

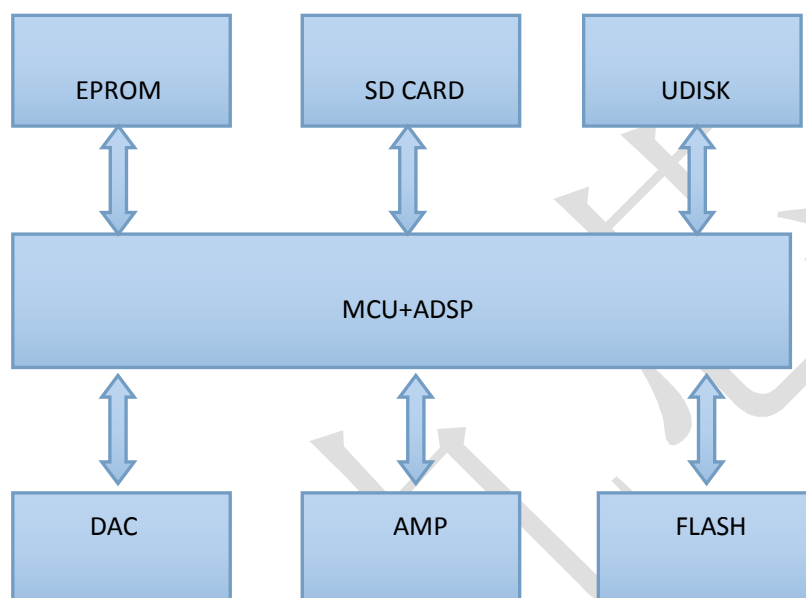


1.3 应用场合



2 模块使用参数

2.1 模块结构图



模块选用的是 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。

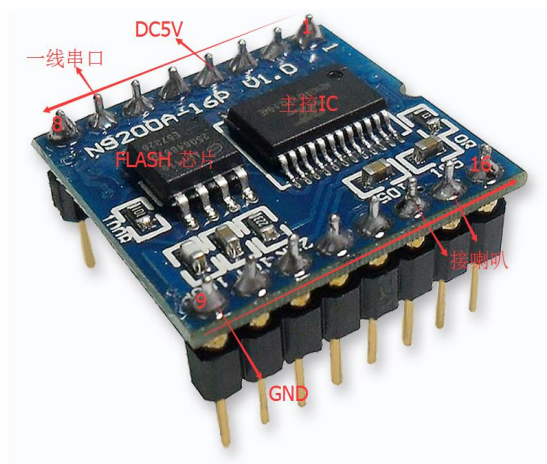
2.2. 电路参数

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 (KHZ) :8/11.025/12/16/22.05/24/32/44.1/48
	3、支持Normal、Jazz、Classic、Pop、Rock 等音效
UART 接口	标准串口，TTL 电平, 波特率可设
输入电压	供电在3.5V-5V 最佳为4.2V
I/O口电平	芯片I/O口电平是3.3V，所以MCU 5V供电控制时，I/O口中间需串1K电阻
额定电流	20ma
尺寸	标准的宽体DIP16/DIP28封装
喇叭功率	8欧/3瓦
工作温度	-40度~80度
湿度	5% ~ 95%

3. 模块管脚图

1			16
2	P26	P05	15
3	DACL	SPK	14
4	DACR	SPK	13
5	VDD	VDDIO	12
6	TXD	USBDP	11
7	RXD	USBDM	10
8	1_LINE	ADK	9
	BUSY	GND	
N9200A			

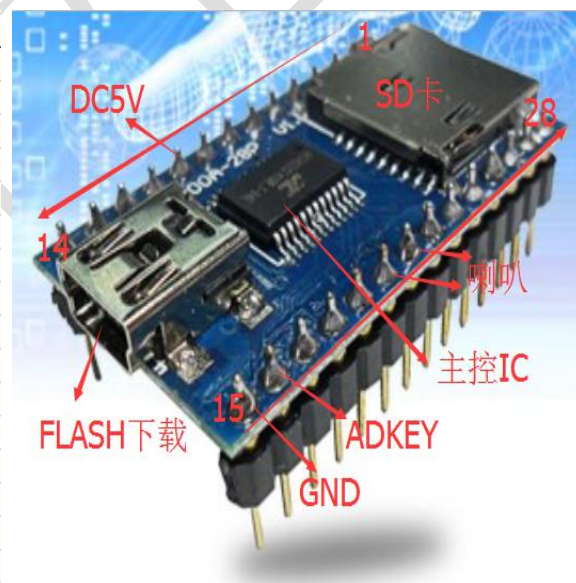
原理图 (1)



实物图 (1)

1	NC	NC	28
2	NC	NC	27
3	NC	NC	26
4	NC	NC	25
5	NC	NC	24
6	NC	NC	23
7	P27	NC	22
8	P26	P05	21
9	DACL	SPK	20
10	DACR	SPK	19
11	VDD	VDDIO	18
12	TXD	USBDP	17
13	RXD	USBDM	16
14	1_LINE	ADK	15
	BUSY	GND	
N9200B			

原理图 (2)



实物图 (2)



4. 模块管脚说明

4.1表一、N9200A-16P MP3模块管脚说明

引脚序号	引脚名称	功能描述	备注
1	P26	NC	NC
2	DACL	音频输出左声道	驱动耳机、外接功放
3	DACR	音频输出右声道	驱动耳机、外接功放
4	VDD	模块电源输入	3.3V—5V正电源输入
5	TXD	UART串行口	UART数据输出
6	RXD	UART串行口	UART数据输入
7	ONE_LINE	一线串口输入	一线串口MCU控制
8	BUSY	忙信号	有音频输出低，无音频输出高
9	GND	地	电源地
10	ADKEY	AD按键	参考AD按键电路，AD按键
11	USBDM	USB数据口	可以接USB连电脑或U盘
12	USBDP	USB数据口	可以接USB连电脑或U盘
13	VDDIO	3.3V电源输出	3.3V电源输出
14	SPK2	喇叭-	直接驱动3W/8Ω以下喇叭
15	SPK1	喇叭+	直接驱动3W/8Ω以下喇叭
16	P05	AD口	AD按键功能输入



1.2表二、N9200B-28P MP3模块管脚图

引脚序号	引脚名称	功能描述	备注
1	NC	无连接	
2	NC	无连接	
3	NC	无连接	
4	NC	无连接	
5	NC	无连接	
6	P27	无连接	
7	P26	无连接	
8	DACL	音频输出左声道	驱动耳机、外接功放
9	DACR	音频输出右声道	驱动耳机、外接功放
10	VDD	模块电源输入	3.3V—5V正电源输入
11	TXD	UART串行口	UART数据输出
12	RXD	UART串行口	UART数据输入
13	ONE_LINE	一线串口输入	MCU一线控制
14	BUSY	忙信号	有音频输出低，无音频输出高
15	GND	地	电源地
16	ADKEY	AD口	AD按键输入
17	USBDM	USB数据口	USB连电脑或U盘
18	USBDP	USB数据口	USB连电脑或U盘
19	VDDIO	3.3V电源输出	3.3V电源输出
20	SPK2	喇叭+	直接驱动3W/8R以下喇叭
21	SPK1	喇叭-	直接驱动3W/8R以下喇叭
22	P05	AD口	AD按键口
23	NC	无连接	
24	NC	无连接	
25	NC	无连接	
26	NC	无连接	
27	NC	无连接	
28	NC	无连接	



4.3 硬件参数以及环境极限参数说明

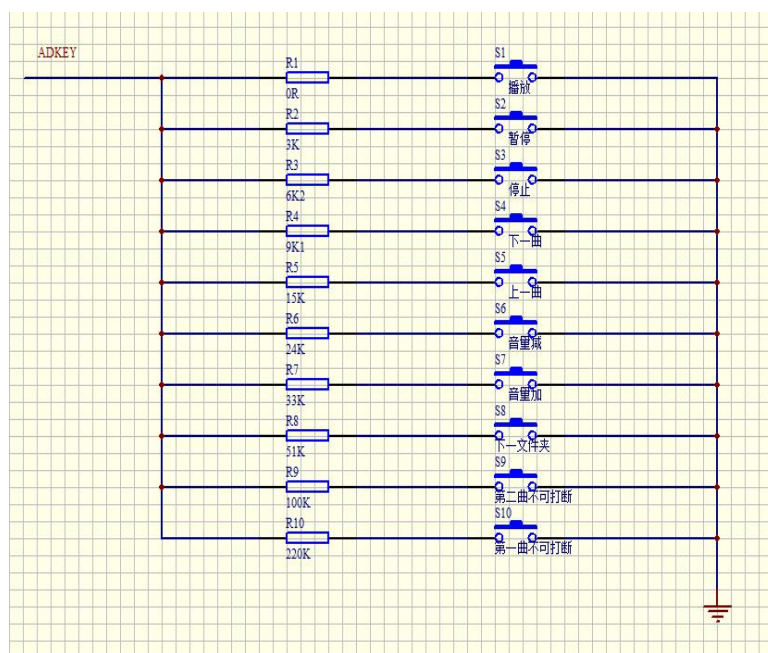
名称	参数说明
MP3文件格式	1. 支持所有比特率11172-3和ISO13813-3 layer3音频解码
	2. 采样率支持 (KHz): 8/11.025/12/16/22.05/24/32/44.1/48
	3. 支持Normal、Jazz、Classic、Pop、Rock等音效
USB接口	2.0标准
一线串口接口	自定义一线串口, 1: 3占空比, 300us: 900us到1000us: 3000us间
UART接口	标准串口, 9600波特率, TTL电平,
输入电压范围	DC3.3V—5V
额定电流	15mA (不带u盘)
工作温度	-40° C到80° C
工作湿度	5%—95%

5. 文件命名以及控制方式说明

5.1 ADKEY按键控制

模块我们采用的是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, 模块默认配置2 个AD 口, 10 个按键的阻值分配, 如果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)、ADKEY参考原理图



(2)、根据不同的电压实现不同的播放功能. (更新日期: 2016. 4. 13)

10个按键就取S1—S10, 电阻值就取R1—R10;

9个按键就取S1—S9, 电阻值就取R1—R9;

8个按键就取S1—S8, 电阻值就取R1—R8;

7个按键就取S1—S7; 电阻值就取R1—R7;

6个按键就取S1—S6; 电阻值就取R1—R6;

5个按键就取S1—S5; 电阻值就取R1—R5;

4个按键就取S1—S4; 电阻值就取R1—R4;

3个按键就取S1—S3; 电阻值就取R1—R3.

(3)、不同的阻值所对应的功能不. (更新日期: 2016. 4. 13)

(0R) 所对应的功能有播放

(3K) 所对应的功能有暂停

(6K2) 所对应的功能有停止

(9K1) 所对应的功能有下一曲

(15K) 所对应的功能有上一曲

(24K) 所对应的功能有音量减

(33K) 所对应的功能有音量加

(51K) 所对应的功能有下一文件夹

(100K) 所对应的功能有第二曲不可打断

(220K) 所对应的功能有第一曲不可打断



5.2 MCU一线串口控制

N9200支持单片机模拟一线串口控制，该控制方式简单方便，适用于任何单片机。

九芯MP3一线串口十六进制指令表

曲目指令		音量指令		播放功能	
指令	功能	指令	功能	指令	功能
0X01	选择曲目1	0XD2	音量为0	0XF1	切换到U盘
0X02	选择曲目2	0XD3	音量为1	0XF2	切换到SD
0X03	选择曲目3	0XD4	音量为2	0XF3	切换到MP3模式
0X04	选择曲目4	0XD5	音量为3	0XF4	切换到flash模式
0X05	选择曲目5	0XD6	音量为4	0XF5	下一个EQ
0X06	选择曲目6	0XD7	音量为5	0XF6	全部循环
0X07	选择曲目7	0XD8	音量为6	0XF7	单曲循环
0X08	选择曲目8	0XD9	音量为7	0XF8	上一曲
0X09	选择曲目9	0XDA	音量为8	0XF9	下一曲
0X0A	选择曲目10	0XDB	音量为9	0XFA	播放
0X0B	选择曲目11	0XDC	音量为10	0XFB	暂停
0X0C	选择曲目12	0XDD	音量为11	0XFC	停止
0X0D	选择曲目13	0XDE	音量为12	0XFD	上一文件夹
0X0E	选择曲目14	0XDF	音量为13	0XFE	下一文件夹
0X0F	选择曲目15	0XE0	音量为14		
0X10	选择曲目16	0XE1	音量为15		
0X11	选择曲目17	0XE2	音量为16		
0X12	选择曲目18	0XE3	音量为17		
0X13	选择曲目19	0XE4	音量为18		
0X14	选择曲目20	0XE5	音量为19		
0X15	选择曲目21	0XE6	音量为20		
0X16	选择曲目22	0XE7	音量为21		
0X17	选择曲目23	0XE8	音量为22		
0X18	选择曲目24	0XE9	音量为23		
0X19	选择曲目25	0XEA	音量为24		
.....	0XEB	音量为25		
.....	0XEC	音量为26		
.....	0XED	音量为27		
0XCC	选择曲目204	0XEE	音量为28		
0XCD	选择曲目205	0XEF	音量为29		
0XCE	选择曲目206	0XF0	音量为30		
0XCF	选择曲目207				

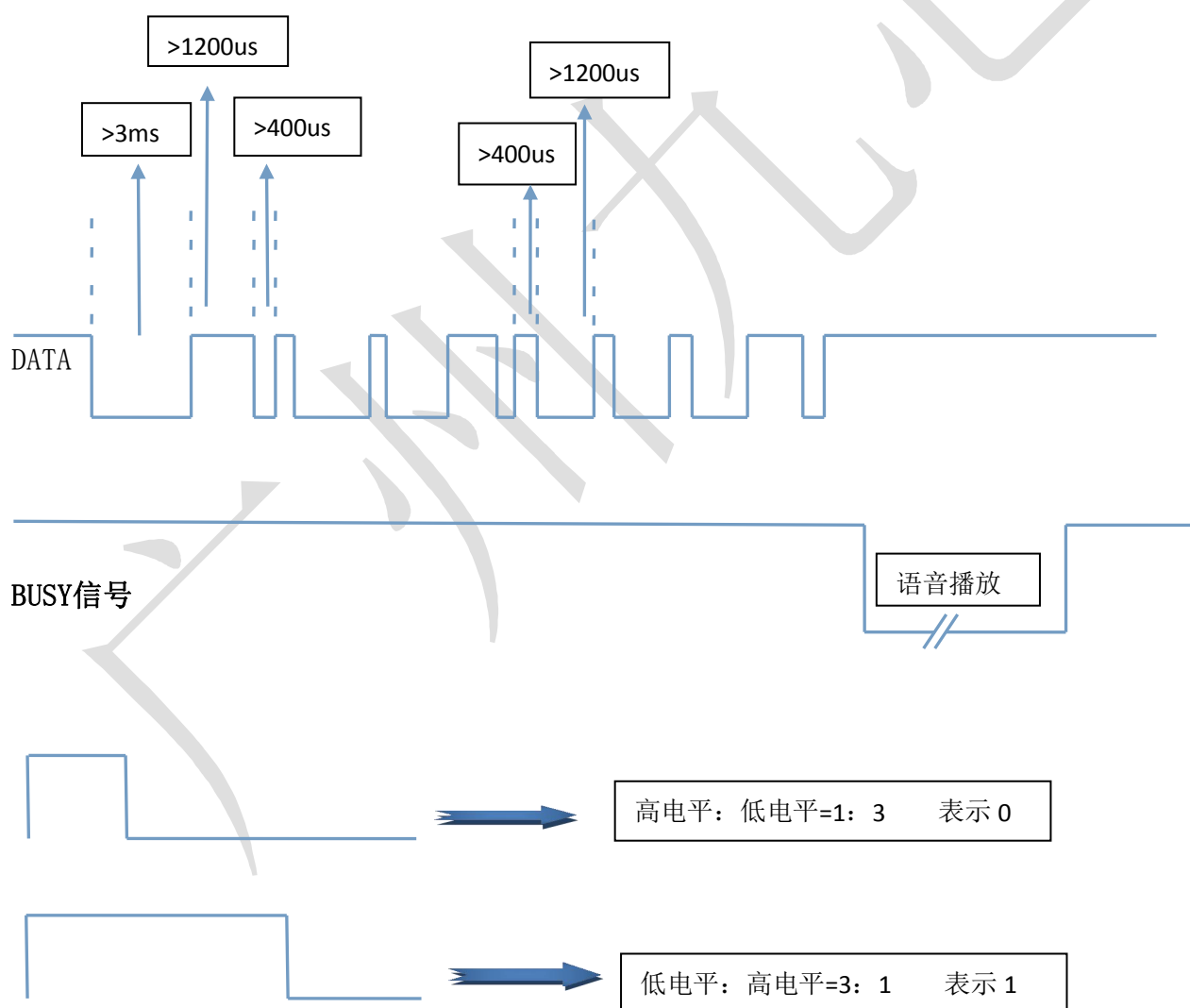
备注：最大可以控制播放到0XCF地址，207首。

5.3 文件排列与命名规则

1. 文件命名必须以MP3后缀结尾
2. 文件的顺序是按拷贝入存储设备中的先后顺序排序的。

5.3.1 一线串口时序

一线串口MCU只需要一根线就可以控制了，由头码和占空比组码，**先发低位**，头码拉低3ms以上，平时保持高。总共8位。时序占空比约为**1: 3**。时序范围介于**300us: 900us**到**1000us: 3000us**之间。误差容许10%。





5.3.2程序范例

(MCU: PIC16F57 晶振: 4MHz) (更新日期: 2016.4.14)

```
Sda=0;                /*数据拉低*/
Wait(300);             /*等待 3ms 以上*/
For(i=0;i<8;i++)       /*8 位循环*/
{
    Sda=1;             /*数据拉高*/
    If(addr&1)          /*如果地址低位拉高*/
    {
        Wait(15);      /*等待 1200us 以上*/
        Sda=0;         /*数据拉低*/
        Wait(5);       /*等待 400us 以上*/
    }
    Else
    {
        Wait(5);       /*等待 400us 以上*/
        Sda=0;         /*数据拉低*/
        Wait(15);      /*等待 1200us 以上*/
    }
    }Addr>>=1;         /*地址右移一位*/
} Sda=1;              /*数据拉高*/
```



5.4 UART串口通信协议

N9200 系列内置标准 UART 异步串行接口。属于 TTL 电平接口, 可通过 MAX232 芯片转换成 RS232 电平。模块通信采用全双工串口通信, 波特率为 9600, 数据位: 8, 停止位 1 位, 检验位无。设置如下图:

(1) 串口设置:



5.4.1 通信数据格式

格式:【CMD】【CheckCMD】【Len】【Data1】【Data2】 Data _n …… 【checksum】		
CMD	命令码	命令码, 接收的命令类型
Check CMD	命令码校验码	命令码的校验码, 即命令码的反码, 用来验证命令码
Len	数据长度	整段数据中所有数据 data 的长度, 即数据总字节个数
Data ₁	数据 1	该命令中所带的第 1 个字节的数据
Data ₂	数据 2	该命令中所带的第 2 个字节的数据
……		……
Data _n	数据 n	该命令中所带的第 n 个字节的数据
Checksum	校验和	数据的校验和, 取 16 位校验和的低 8 位, 下面简称 SM

示例: 比如发送指定播放当前盘符下第 8 首歌曲。那就发送:

04	FB	03	06	00	08	10
CMD 命令	命令反码	数据长度	3 位数据		校验和	



数据可以为 1 个，也可以为多个，主要根据 CMD 命令下规则，对应什么数据。

5.4.2. 容错要求:

接收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，因此必须实时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，则强制丢弃多接受的指令字节，重新同步指令接收。

5.4.3. 协议约定:

- 新曲目播放时会发送曲目序号。这里的序号为全盘统一编号的序号，如果要区分父目录和子目录可以查询路径和路径深度等信息。
- 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。
- 协议中所有的数据都表示 HEX(十六进制)。
- SM 校验和表示所有字节之和（不包括校验和本身），计算出的 16 位校验和的低 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，上电默认为单曲停止。
- 音量默认为最大，可以通过相关指令来查询或设置音量大小。



5.4.4 CMD通信指令控制列表（16进制）

1. 模式控制：（指令：03）

◆ MODE 说明：MUSIC:00 SPIFLASH:01

◆ 查询当前模式（00）：

指令	反码	数据长度	数据 1	校验和
03	FC	01	00	00

返回：03 FC 02 00 MODE SM

◆ 切换模式（01）：

指令	反码	数据长度	数据 1	数据 2	校验和
03	FC	02	01	MODE	SM

返回：无

（如果要切换的模式和当前模式相同则不处理，切换过去都处于停止状态，要发播放指令才播放）

例如：切换到 MUSIC：

指令	反码	数据长度	数据 1	数据 2	校验和
03	FC	02	01	00	02

切换到 SPIFLASH：

指令	反码	数据长度	数据 1	数据 2	校验和
03	FC	02	01	01	03

2. 播放时间控制：（指令：05）

◆ 总播放时间查询（00）：

指令	反码	数据长度	数据 1	校验和
05	FA	01	00	00

返回：05 FA 04 00 时分秒 SM

◆ 当前播放时间发送（01）：

指令	反码	数据长度	数据 1	数据 2	数据 3	数据 4	校验和
05	FA	04	01	时	分	秒	SM



◆ 发送播放时间开(02) :

指令	反码	数据长度	数据 1	校验和
05	FA	01	02	02

返回: 当前时间

◆ 发送播放时间关(03) :

指令	反码	数据长度	数据 1	校验和
05	FA	01	03	03

返回: 无

3. 播放控制: (指令: 04) 下面 SM 代表校验和

◆ 返回的播放状态代码含义: 00 已停止 01 播放中 02 已暂停

◆ 返回设备定义代码 : 00 U 盘 01 SD 02 SPI FLASH

◆ 查询播放状态 (00) :

指令	反码	数据长度	数据 1	校验和
04	FB	01	00	00

返回: 04 FB 02 00 【播放状态代码】 SM

◆ 播放 (01) :

指令	反码	数据长度	数据 1	校验和
04	FB	01	01	01

返回: 04 FB 02 00 【播放状态代码】 SM

◆ 暂停 (02) :

指令	反码	数据长度	数据 1	校验和
04	FB	01	02	02

返回: 04 FB 02 00 【播放状态代码】 SM

◆ 停止 (03) :

指令	反码	数据长度	数据 1	校验和
04	FB	01	03	03

返回: 04 FB 02 00 【播放状态代码】 SM

◆ 上一曲 (04) :

指令	反码	数据长度	数据 1	校验和
04	FB	01	04	04



返回: 无

◆ 下一曲 (05) :

指令	反码	数据长度	数据 1	校验和
04	FB	01	05	05

返回: 无

◆ 指定曲目 (06) :

指令	反码	数据长度	数据 1	数据 2	数据 3	校验和
04	FB	03	06	曲目高位	曲目低位	SM

返回: 无

备注: 曲目一共有高低 2 个字节表示, 如果第 6 首, 就是 00 06.

$$00\ 06 = 0 \times 16^3 + 0 \times 16^2 + 0 \times 16^1 + 6 \times 16^0 = 6 \quad (\text{即第 6 首})$$

例如:

04 FB 03 06 00 08 10 指定播放当前盘符第 8 首, 曲目数从范围 1—65535, 十六进制表示为 0000H 到 FFFFH.

$$00\ 08 = 0 \times 16^3 + 0 \times 16^2 + 0 \times 16^1 + 8 \times 16^0 = 8 \quad (\text{即第 8 首})$$

◆ 当前盘符指定路径播放 (07): 04 FB LEN 07 路径 SM (更新日期: 2016. 4. 13)

返回: 无

例如:

/周华健/有没有那么一首歌会让你想起我.MP3

文件名超过 4 个汉字用*代替

/周华健 /有没有*MP3 (周华健后面空两格)

取模数据为: 2FD6DCBBAABDA120202FD3D0C3BBD3D02A4D5033 (这些数据是路径)

指令	反码	数据长度	数据	路径	校验和
04	FB	15 (LEN)	07	取模的数据	SM

比如: 04 FB 15 07 2F D6 DC BB AA BD A1 20 20 2F D3 D0 C3 BB D3 D0 2A 4D 50 33 EC

特别声明: 代码中每两个空一小格





数据长度

15 (LEN) 这个是怎样得来的呢?

是这样子的: 15 (LEN) = 路径 + 1

数据

说明: 等式中的 +1 就是这个 07, 意思就是说 1 个数据长度;

等式中的路径就是 2FD6DCBBAABDA120202FD3D0C3BBD3D02A4D5033 (这些数据是路径)

2F	D6	DC	BB	AA	BD	A1	20	20	2F	D3	D0	C3	BB	D3	D0	2A	4D	50	33
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

路径里有 20 个数据 (每两个字母或者数字或者数字与字母结合在一起组成一个数据)

15 (LEN) = 21, 这里的 21 是十进制, 所以把 21 转化为十六进制

十进制转化十六进制方法:

$$\begin{array}{r}
 16 \overline{) 21} \\
 \underline{16} \\
 5 \text{ (余)} \\
 16 \overline{) 1} \\
 \underline{16} \\
 0 \text{ (余)}
 \end{array}$$

由下向上读为 15, 所以 15 是这样来的。

(详见路径格式说明)

◆ 查询当前在线设备数 (08) :

指令	反码	数据长度	数据 1	校验和
04	FB	01	08	08

返回: 04 FB 02 08 在线设备数目 SM

◆ 查询当前播放设备 (09) :

指令	反码	数据长度	数据 1	校验和
04	FB	01	09	09

返回: 04 FB 02 09 盘符 SM

◆ 切换到 U 盘 (0A) :



指令	反码	数据长度	数据 1	校验和
04	FB	01	0A	0A

返回: 无

◆ 切换到 SD (0B) :

指令	反码	数据长度	数据 1	校验和
04	FB	01	0B	0B

返回: 无

◆ 切换到 FLASH (0C) :

指令	反码	数据长度	数据 1	校验和
04	FB	01	0C	0C

返回: 无

◆ 查询总曲目 (0D) :

指令	反码	数据长度	数据 1	校验和
04	FB	01	0D	0D

返回: 04 FB 03 0D 总曲目高位 总曲目低位 SM

◆ 查询当前曲目 (0E) :

指令	反码	数据长度	数据 1	校验和
04	FB	01	0E	0E

返回: 04 FB 03 0E 曲目高位 曲目低位 SM

◆ 上一文件夹 (12) :

指令	反码	数据长度	数据 1	校验和
04	FB	01	12	12

返回: 在播放时返回曲目序号

◆ 下一文件夹 (13) : 04 FB 01 13 13

指令	反码	数据长度	数据 1	校验和
04	FB	01	13	13

返回: 在播放时返回曲目序号

结束播放 (14): 04 FB 01 14 14

指令	反码	数据长度	数据 1	校验和
04	FB	01	14	14

返回: 无

短文件名获取 (15):



指令	反码	数据长度	数据 1	校验和
04	FB	01	15	15

返回: 无

选曲不播放(16): 04 FB 03 16 曲目高 曲目低 SM

获取目录首曲目序号(17):

指令	反码	数据长度	数据 1	校验和
04	FB	01	17	17

返回: 04 FB 03 17 曲目高 曲目低 SM

◆ 查询当前目录下总文件数(18):

指令	反码	数据长度	数据 1	校验和
04	FB	01	18	18

返回: 04 FB 03 18 曲目高 曲目低 SM

4. 音量控制:(指令: 06)

◆ 音量最大为 30, 音量开机默认为最大, 音量范围 0 到 30 级可调。

◆ 查询音量(00):

指令	反码	数据长度	数据 1	校验和
06	F9	01	00	00

返回: 06 F9 02 00 VOL SM

◆ 音量设置(01):

指令	反码	数据长度	数据 1	数据 2	校验和
06	F9	02	01	VOL	SM

返回: 无

例如:

06 F9 02 01 14 16 设置音量为 20 级, 14 是 16 进制, 代表音量 20 级。

$1 \times 16^1 + 4 \times 16^0 = 20$ (级)

◆ 音量加(02):

指令	反码	数据长度	数据 1	校验和
06	F9	01	02	SM

返回: 无

◆ 音量减(03):

指令	反码	数据长度	数据 1	校验和
06	F9	01	03	03



返回: 无

5. 复读控制: (指令: 08)

◆ 指定时间复读(00) :

指令	反码	数据长度	数据 1	数据 2	数据 3	数据 4	数据 5	校验和
08	F7	05	00	起始分	起始秒	结束分	结束秒	SM

返回: 无

例如: 08F70500020602202E 从 2 分 06 秒开始到 2 分 32 秒复读

◆ 结束复读(01) :

指令	反码	数据长度	数据 1	校验和
08	F7	01	01	01

返回: 无

执行结束命令后此次复读无效, 或执行其它操作也可以结束复读.

6. 插播控制: (指令: 09)

◆ 指定曲目插播(00) :

指令	反码	数据长度	数据 1	数据 2	数据 3	数据 4	校验和
09	F6	04	00	盘符	曲目高	曲目低	SM

返回: 无

例如: 09 F6 04 00 01 00 04 08 , 表示插播 SD 卡的第 4 首。

◆ 指定路径插播(01): 09 F6 长度 01 盘符 路径 SM

指令	反码	数据长度	数据 1	数据 2	数据 3	校验和
09	F6	长度	01	盘符	路径	SM

说明: 长度=路径长度+2

◆ 结束插播(01) : 用结束播放指令

返回: 无

7. 循环模式控制: (指令: 0B)

◆ 查询循环模式(00) :

指令	反码	数据长度	数据 1	校验和
0B	F4	01	00	00

返回: 0B F4 02 00 循环模式 SM

◆ 设置循环模式(01) :

指令	反码	数据长度	数据 1	数据 2	校验和
0B	F4	02	01	循环模式	SM



◆ 设置为单曲循环 : 0B F4 02 01 01 03

指令	反码	数据长度	数据 1	数据 2	校验和
0B	F4	02	01	01	03

下面列举循环播放模式代码对应的功能:

- ◆ 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 按顺序循环播放+当前文件夹内曲目。
- ◆ 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。
- ◆ 文件夹顺序播放 : 06 顺序播放当前文件夹内曲目, 播放完停止。
- ◆ 文件夹随机播放 : 07 随机播放当前文件夹内曲目。

8. 睡眠指令 (0C)

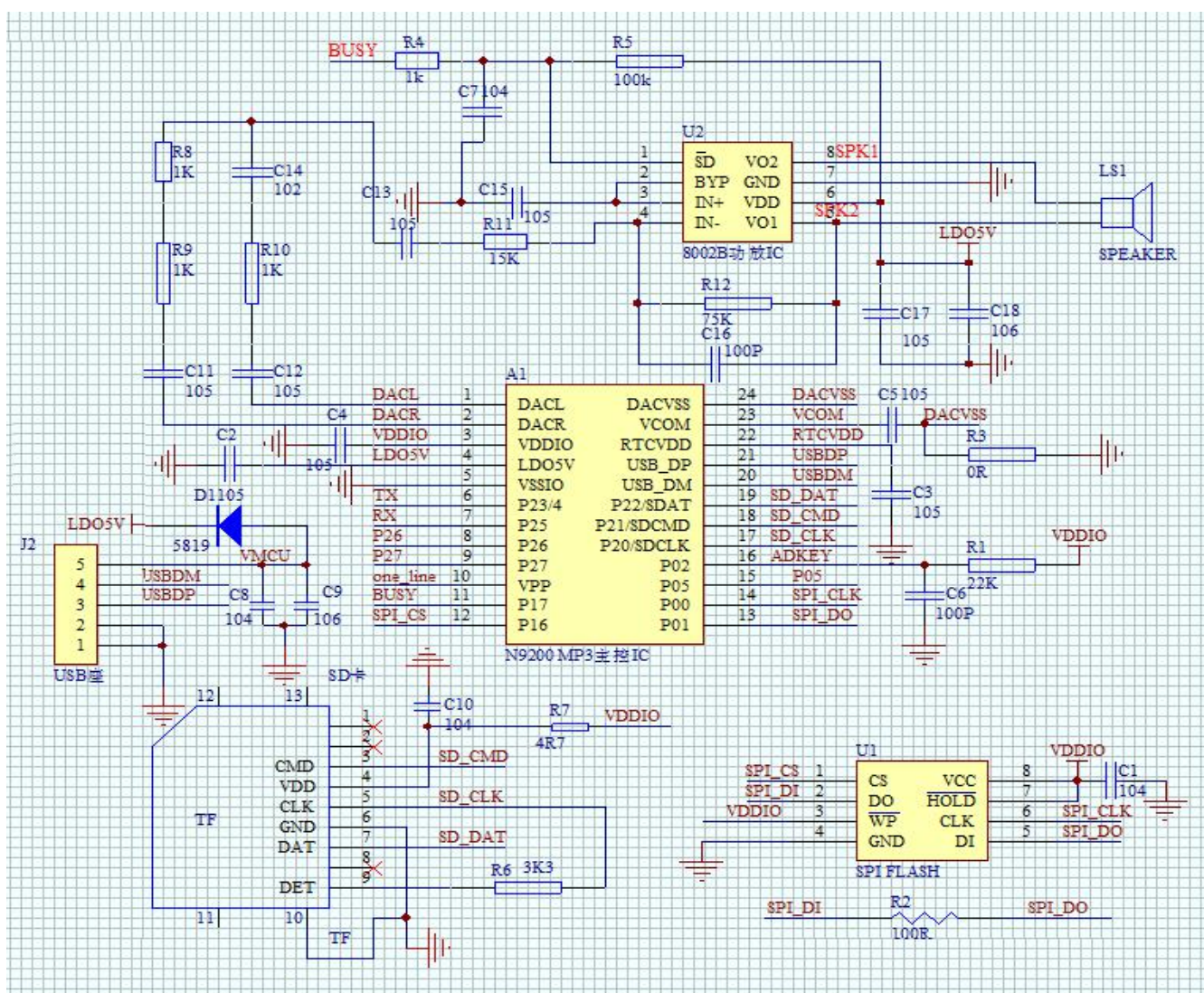
9. 返还错误信息: (指令: AA)

- ◆ AA 55 02 FF 00 00 串口接收数据错误
- ◆ AA 55 02 FF 02 02 批定的盘符找不到
- ◆ AA 55 02 FF 03 03 没有可以播放盘符。
- ◆ AA 55 02 FF 04 04 文件播放错误, 如文件找不到等

10. 返还设备插拔状态信息: (指令: BB)

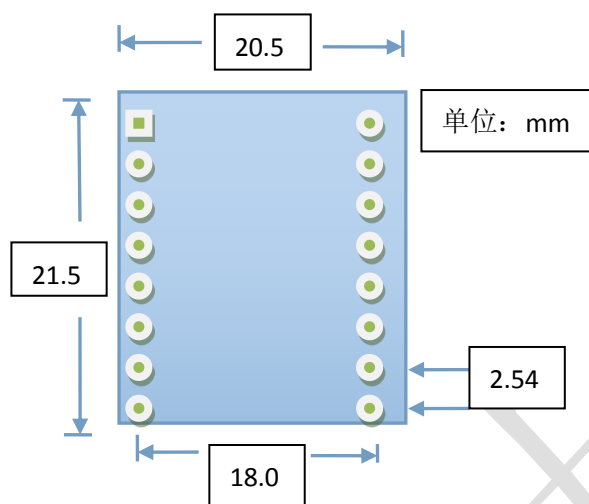
- ◆ BB 44 01 00 00 U 盘插入
- ◆ BB 44 01 01 01 U 盘拔出
- ◆ BB 44 01 02 02 SD 卡插入
- ◆ BB 44 01 03 03 SD 卡拔出

6. 模块内部电路图

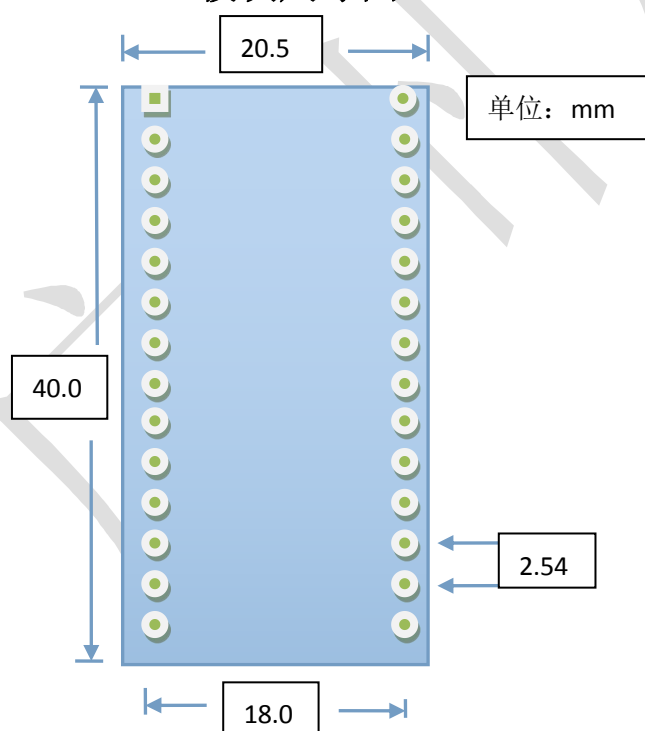


备注：模块内部 ADKEY 已有 22K 上拉电阻的，如果用户自行布板，ADKEY 必须有 22K 上拉电阻。

7. 模块封装尺寸图



N9200A 模块尺寸图



N9200B 模块尺寸图

8. 文件夹路径格式说明及校验码的算法

8.1 文件夹路径要求

- 文件夹名字为 8 个字节, 超过 8 个字节取前 8 字节, 不够 8 字节的用空格补充, 8 个字节就是四个汉字或 8 个字母。
- 文件名也为 8 个字节, 不够的可以用? 或者* 表示, 例如 ABC????? 表示开头的三个字符为 abc 的文件, abc* 也表示开头的 3 个字符为 abc 的文件。
- 不管盘符里面的文件夹或文件名是否是大写字母, 在编写程序时, 文件夹和文件的名字必须为大写字母或数字。
- 一个汉字占用两个字节, 一个空格或字符都是占用一个字节。

例如: 指定文件夹下歌曲

"/背景 /*???", /*背景文件夹下的所有文件, 背景后面 4 个空格*/


"/MODE????MP3", /*根目录下以 mode 开头的 MP3 文件*/

/周华健/难念的经 MP3

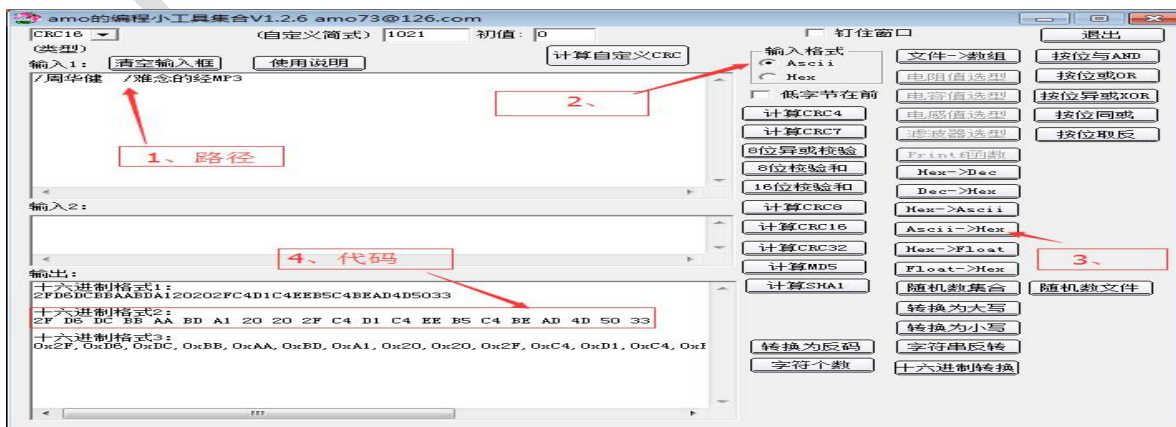
路径数据获取:

- ◆ 推荐工具: amo 的编程小工具集合 V1.2.6

例如: "/周华健 /难念的经 MP3", 输入格式要选择 Ascii, 在输入项里输入对应的路

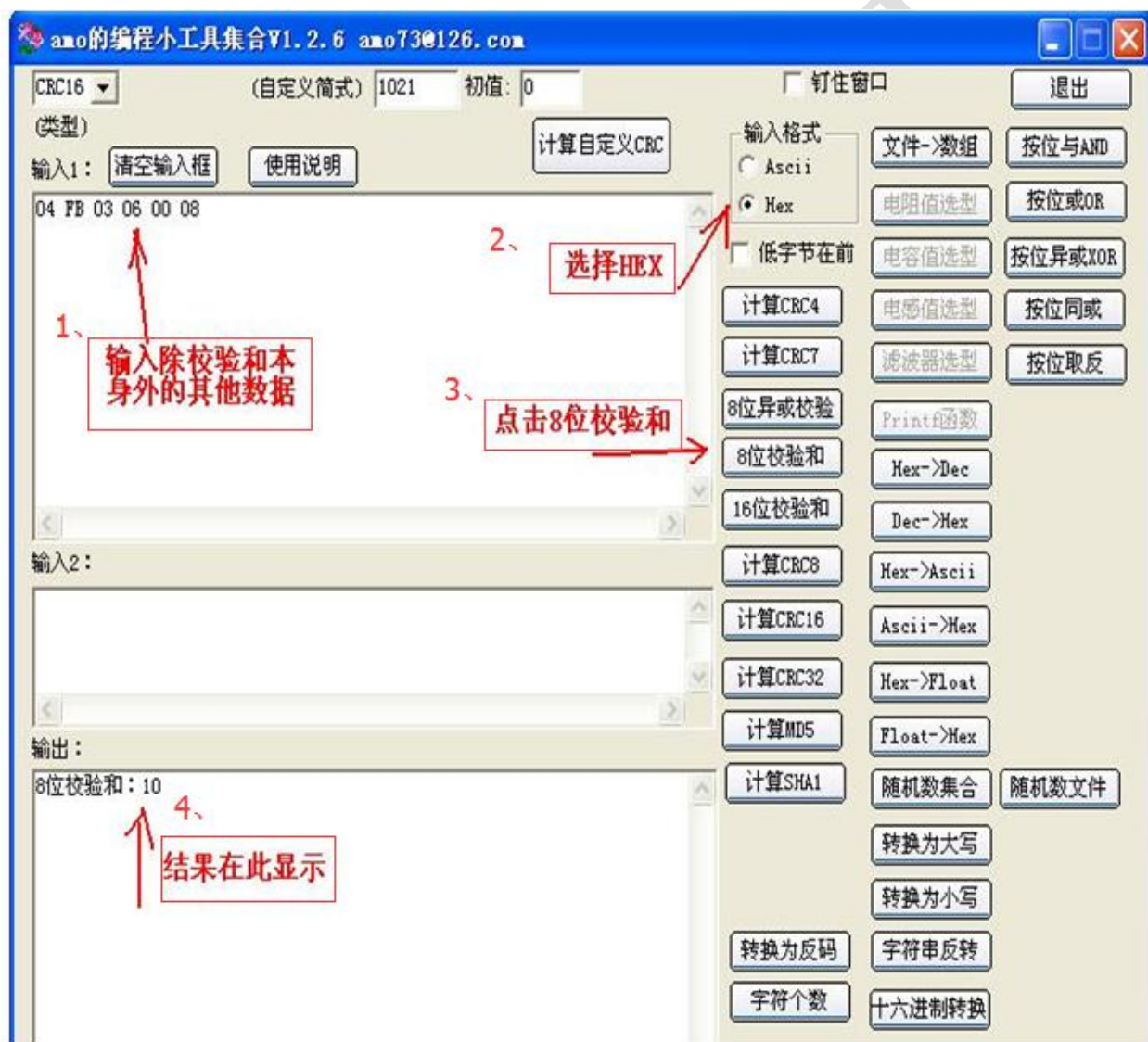
径。选择  就可以得到对应的数据。(周华健后面空两格)

2F D6 DC BB AA BD A1 20 20 2F C4 D1 C4 EE B5 C4 BE AD 4D 50 33



8.2 校验和计算方法

本 UART 数据协议中校验位采样的是校验和，即将不包括校验和本身的其他字节数据累加得出的 16 位校验和，然后只取其中低 8 位。推荐使用 amo 的编程小工具集合 V1.2.6 如发送 04 FB 03 06 00 08 10 指定播放当前盘符第 8 首，那么 04 FB 03 06 00 08 将这 6 个字节的数据进行校验和，然后取低 8 位，得出校验和是如下。





9 注意事项

模块的使用方法，关键的地方做如下说明：

- ◆ 模块的GPIO特性
- ◆ 应用中的注意事项
- ◆ 串口编程部分的注意

9.1 GPIO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=3.3
Voh	High-level output voltage	2.7	—	—	V	VDD=3.3

9.2 应用中的注意事项

- 1、N9200 模块对外的接口均是 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，请注意模块的供电，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、模块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, 请用户控制模块或者芯片的供电。这样可以减小芯片的功耗。

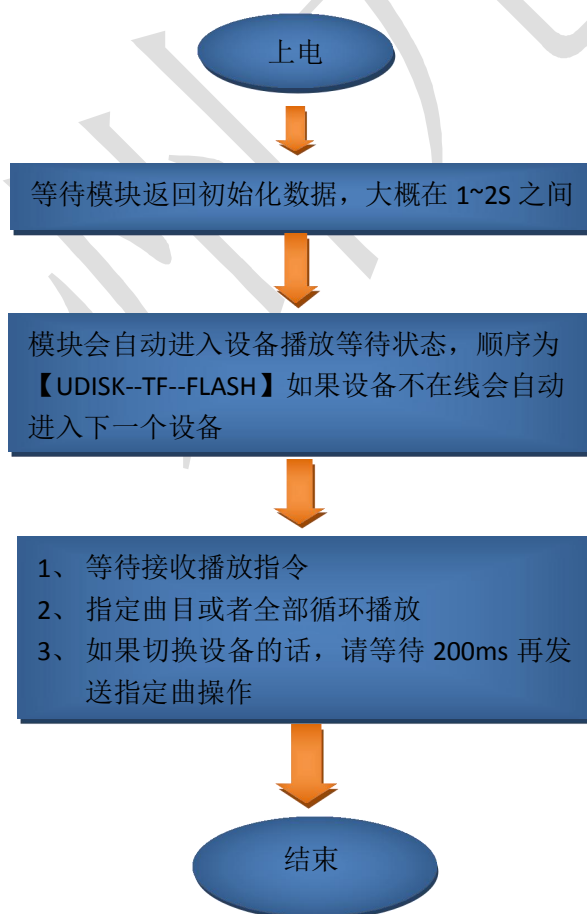
6、该模块支持 MP3、WAV、WMA 三种主流的音频格式。但是默认发货烧录的软件支持的是 MP3、WAV 这两种格式。如果特别需要支持 WMA 格式的需求, 请提前说明

7、我们的模块支持 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。

9.3 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

9.3.1 串口操作需要延时的注意事项





9.3.2 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

9.3.3 串口编程需要适当延时的注意点

- 1、模块上电之后,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、因为模块自带文件系统,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

10. 免责声明

10.1 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。

10.2 EMI 和 EMC

模块机械结构决定了其 EMI 性能必然与一体化电路设计有所差异。模块的 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。

模块的 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。我们将努力完善模块的电磁兼容特性,但不对用户最终应用产品 EMC 性能提供任何保证。

10.3 修改文档的权力

本公司有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

10.4 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。



11、历史版本

版本号	日期	修改记录
V1.0	2014.06.01	初次编制
V1.1	2014.12.25	修改了一线串口部分的操作
V1.2	2015.08.22	对 UART 串口的增加详细解释和说明
V1.3	2016.04.14	突出重点, 每种功能都有详细的解说