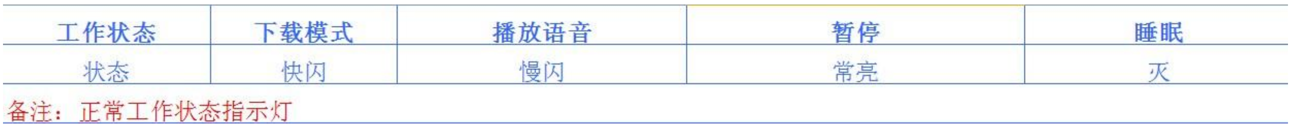
目 录

1, module hardware description

Module power supply modules the ideal working voltage as 4.2V。 So users are using 5V Power supply, it is recommended that thread a diode

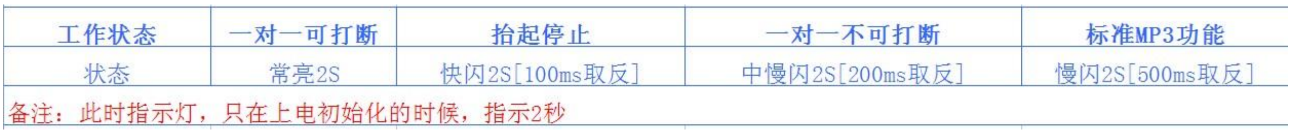
2, Led module [Power state]



工作状态	下载模式	播放语音	暂停	睡眠
状态	快闪	慢闪	常亮	灭

备注：正常工作状态指示灯

3, Led module [Work State]



工作状态	一对一可打断	抬起停止	一对一不可打断	标准MP3功能
状态	常亮2S	快闪2S[100ms取反]	中慢闪2S[200ms取反]	慢闪2S[500ms取反]

备注：此时指示灯，只在上电初始化的时候，指示2秒

4, Audio transmission description

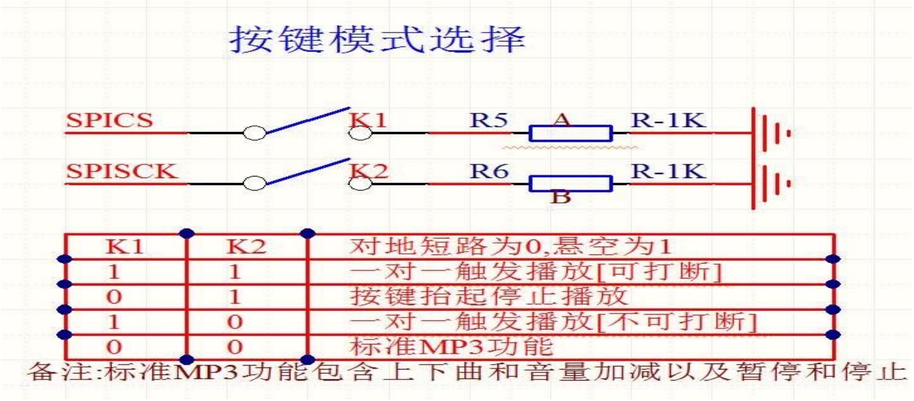
1. SPK1 和 SPK2 Received the trumpet two Swiss, regardless of polarity, note: only allow 2W Following Horn
2. Amplifier or headphones connected directly DAC－R 和 DAC－L Ends Note: common ground (GND)

5, Module debug instructions

(1), Our module is the default plug USB Line, enter download mode. When a user updates after your speech, triggered a IO , You can exit the download mode and return to normal working condition

(2), Modules before shipping, have been download and test the sound, before the customers get the module, do not hurry to download sound, first by way of small speakers or amplifier or headphones, plug USB (When the power supply), short circuit to ground triggers trigger sound, then change the sound,

Note that update voice browsing open is completed, once again updated, because after every update module memory wiped clean burning

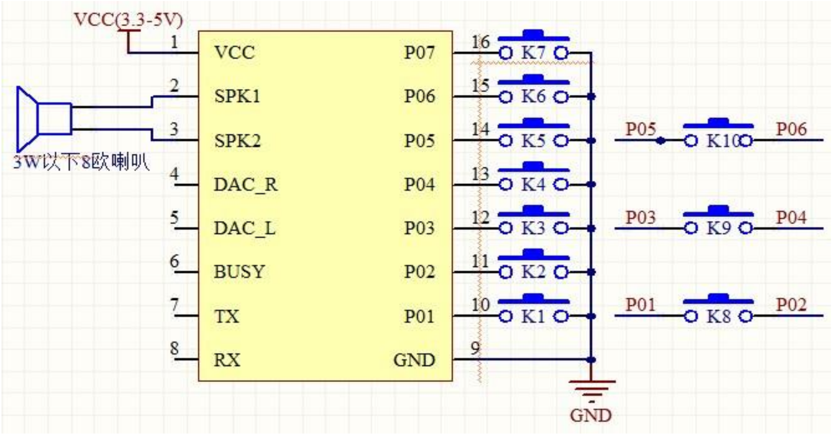


K1	K2	对地短路为0,悬空为1
1	1	一对一触发播放[可打断]
0	1	按键抬起停止播放
1	0	一对一触发播放[不可打断]
0	0	标准MP3功能

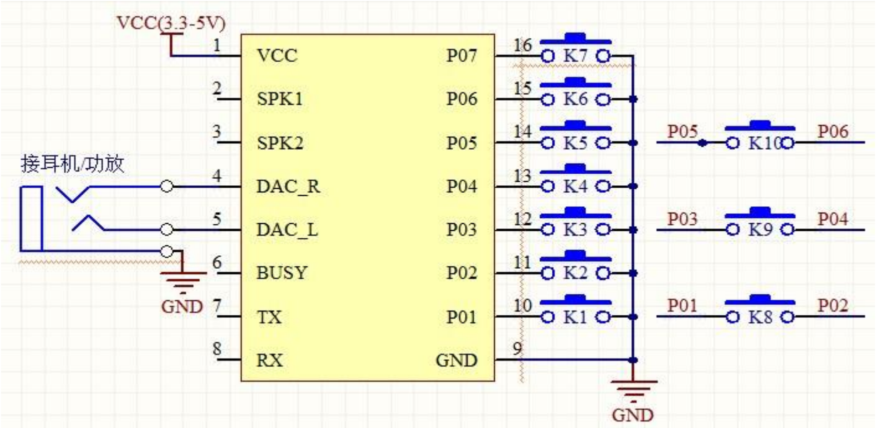
备注:标准MP3功能包含上下曲和音量加减以及暂停和停止

Second, module wiring diagram

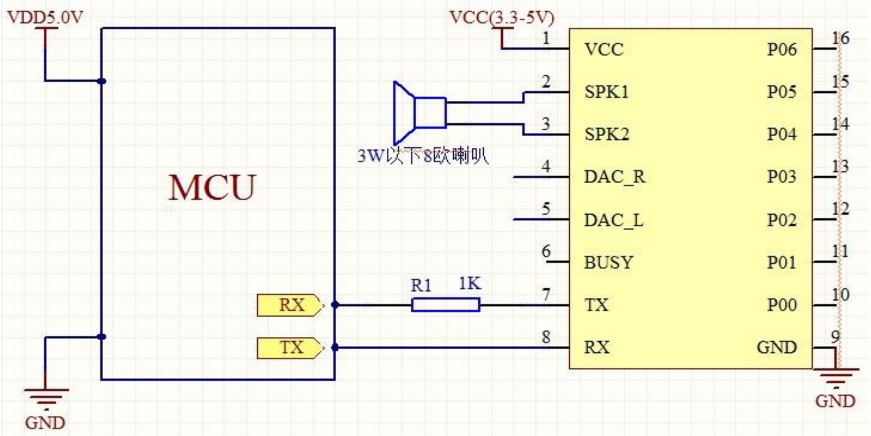
Press the key mode-trumpet



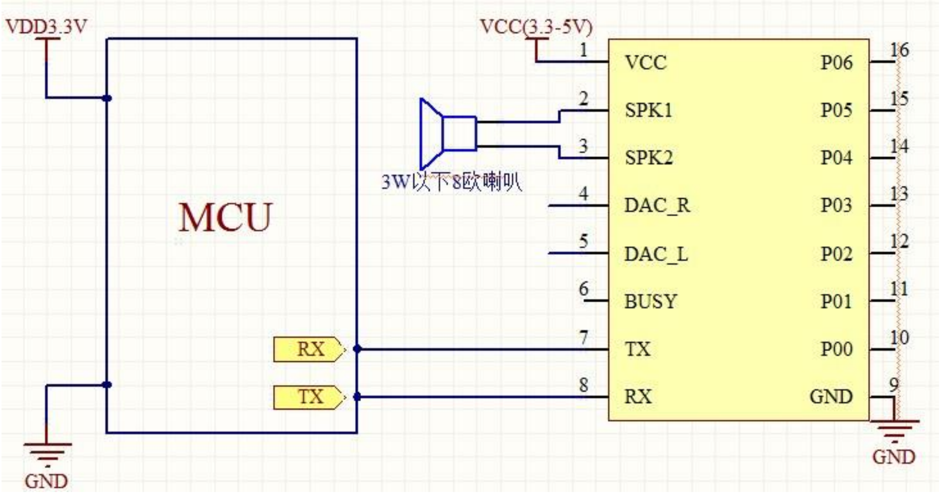
Press the key mode-amplifier or headphones



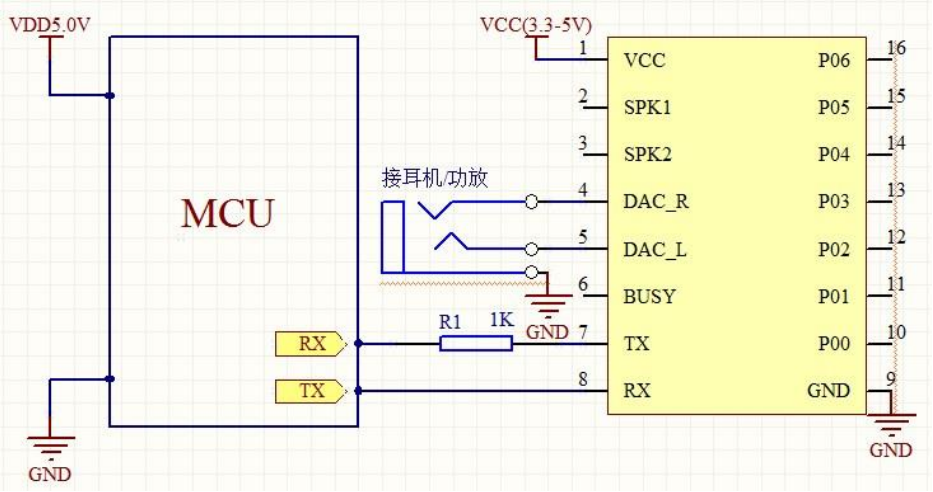
Single-chip microcomputer 5V-Trumpet



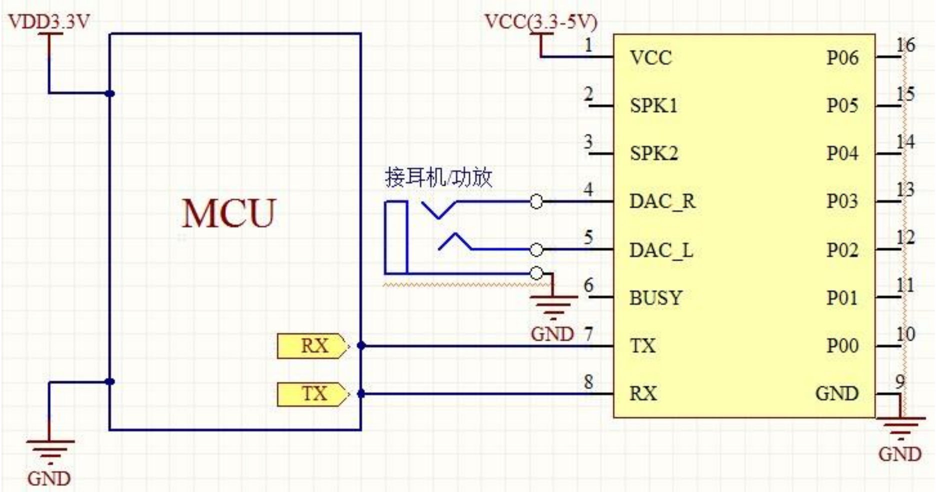
Single-chip microcomputer 3.3V-Trumpet



Single-chip microcomputer 5V-Headset or amplifier



Single-chip microcomputer 5V-Headset or amplifier



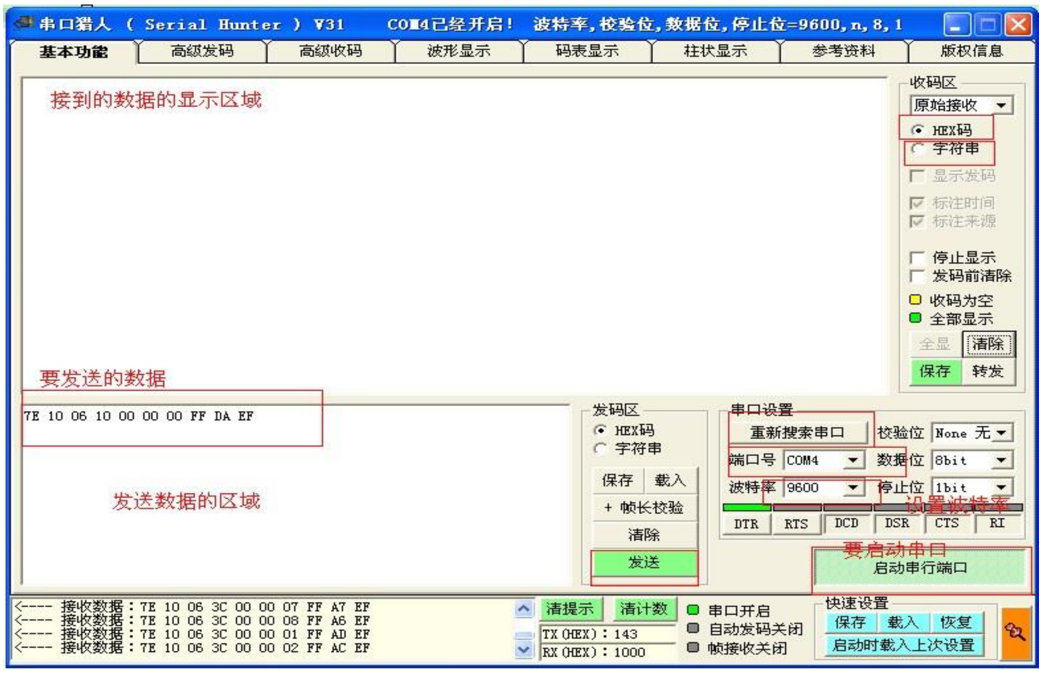
Testing serial debugging assistant	Send command[With checksum]	Send command[Without calibration]	Notes
[The next song]	7E FF 06 01 00 00 00 FE FA EF	7E FF 06 01 00 00 00 EF	
[Previous track]	7E FF 06 02 00 00 00 FE F9 EF	7E FF 06 02 00 00 00 EF	
[The specified track]	7E FF 06 03 00 00 01 FE F7 EF	7E FF 06 03 00 00 01 EF	Specifies the first play
	7E FF 06 03 00 00 02 FE F6 EF	7E FF 06 03 00 00 02 EF	Specifies the second song
	7E FF 06 03 00 00 0A FE EE EF	7E FF 06 03 00 00 0A EF	XAF10 首
[Specifies the volume]	7E FF 06 06 00 00 1E FE D7 EF	7E FF 06 06 00 00 1E EF	Specifies that the volume is30 级
[The specifiedEQ]	7E FF 06 07 00 00 01 FE F3 EF	7E FF 06 07 00 00 01 EF	Reservation
[Loop playlist]	7E FF 06 08 00 00 01 FE F2 EF	7E FF 06 08 00 00 01 EF	Loop the first
	7E FF 06 08 00 00 02 FE F1 EF	7E FF 06 08 00 00 02 EF	Loops II
	7E FF 06 08 00 00 0A FE E9 EF	7E FF 06 08 00 00 0A EF	Looping the tenth
[Specifies the playback device]	7E FF 06 09 00 00 01 FE F1 EF	7E FF 06 09 00 00 01 EF	Specifies the playbackFLASH
[Enter sleep mode]	7E FF 06 0A 00 00 00 FE F1 EF	7E FF 06 0A 00 00 00 EF	
[Wake up sleeping]	7E FF 06 0B 00 00 00 FE F0 EF	7E FF 06 0B 00 00 00 EF	
[Module reset]	7E FF 06 0C 00 00 00 FE EF EF	7E FF 06 0C 00 00 00 EF	
[Play]	7E FF 06 0D 00 00 00 FE EE EF	7E FF 06 0D 00 00 00 EF	
[Pause]	7E FF 06 0E 00 00 00 FE ED EF	7E FF 06 0E 00 00 00 EF	
To stop playback	7E FF 06 16 00 00 00 FE E5 EF	7E FF 06 16 00 00 00 EF	Stop software decoding

Combination play	7E FF 09 21 00 05 01 02 03 04 FE C8 EF	7E FF 09 21 00 05 01 02 03 04 EF	Combination play5、1、2、3、4
Combination play	7E FF 0C 21 00 05 01 02 03 04 06 07 08 FE B0 EF	7E FF 0C 21 00 05 01 02 03 04 06 07 08 EF	Combination play5、1、2、3、4、 6、7、8
Play with volume	7E FF 06 22 00 1E 01 FE BA EF	7E FF 06 22 00 1E 01 EF	30Level volume play1 曲
	7E FF 06 22 00 0F 01 FE C9 EF	7E FF 06 22 00 0F 01 EF	15Level volume play1 曲
	7E FF 06 22 00 0F 02 FE C8 EF	7E FF 06 22 00 0F 02 EF	15Level volume play2 曲
Play it a specified folder tracks	7E FF 06 30 00 01 01 FE C9 EF	7E FF 06 30 00 01 01 EF	FOLDER1Subsection1 曲
	7E FF 06 30 00 02 01 FE C8 EF	7E FF 06 30 00 02 01 EF	FOLDER2Subsection1 曲
	7E FF 06 30 00 03 01 FE C7 EF	7E FF 06 30 00 03 01 EF	FOLDER3Subsection1 曲
	7E FF 06 30 00 03 03 FE C5 EF	7E FF 06 30 00 03 03 EF	FOLDER3Subsection3 曲
Specifies the folder tracks cycle sowing	7E FF 06 31 00 03 03 FE C4 EF	7E FF 06 31 00 03 03 EF	FOLDER3 第 3Music looping
	7E FF 06 31 00 01 03 FE C6 EF	7E FF 06 31 00 01 03 EF	FOLDER1Subsection3Song cycle sowing 放
Specify a folder loop	7E FF 06 32 00 03 01 FE C5 EF	7E FF 06 32 00 03 01 EF	FOLDER3Subsection1Song start looping

Query the current state	7E FF 06 42 00 00 00 FE B9 EF	7E FF 06 42 00 00 00 EF	
[Query volume]	7E FF 06 43 00 00 00 FE B8 EF	7E FF 06 43 00 00 00 EF	
[Query the currentEQ]	7E FF 06 44 00 00 00 FE B7 EF	7E FF 06 44 00 00 00 EF	
The current play mode	7E FF 06 47 00 00 00 FE B4 EF	7E FF 06 47 00 00 00 EF	
FLASHTotal number of files	7E FF 06 49 00 00 00 FE B2 EF	7E FF 06 49 00 00 00 EF	
FLASHCurrent folder track pointers	7E FF 06 4D 00 00 00 FE AE EF	7E FF 06 4D 00 00 00 EF	
Currently online folder	7E FF 06 60 00 00 00 FE 9B EF	7E FF 06 60 00 00 00 EF	Query online folder
Pointer to current folder	7E FF 06 61 00 00 00 FE 9A EF	7E FF 06 61 00 00 00 EF	Queries the current playback folder
Check the specified total number of folder tracks	7E FF 06 62 00 00 01 FE 98 EF	7E FF 06 62 00 00 01 EF	QueryFOLDER1Total files

Third, testing methods

1、 串口软件的操作



串口猎人 （ Serial Hunter ） V31 　COM4已经开启！ 波特率,校验位,数据位,停止位=9600,n,8,1

发码区：7E 10 06 10 00 00 00 FF DA EF

串口设置：端口号 COM4　波特率 9600　校验位 None 无　数据位 8bit　停止位 1bit

```
<---- 接收数据：7E 10 06 3C 00 00 07 FF A7 EF
<---- 接收数据：7E 10 06 3C 00 00 08 FF A6 EF
<---- 接收数据：7E 10 06 3C 00 00 01 FF AD EF
<---- 接收数据：7E 10 06 3C 00 00 02 FF AC EF
```

TX (HEX)：143
RX (HEX)：1000

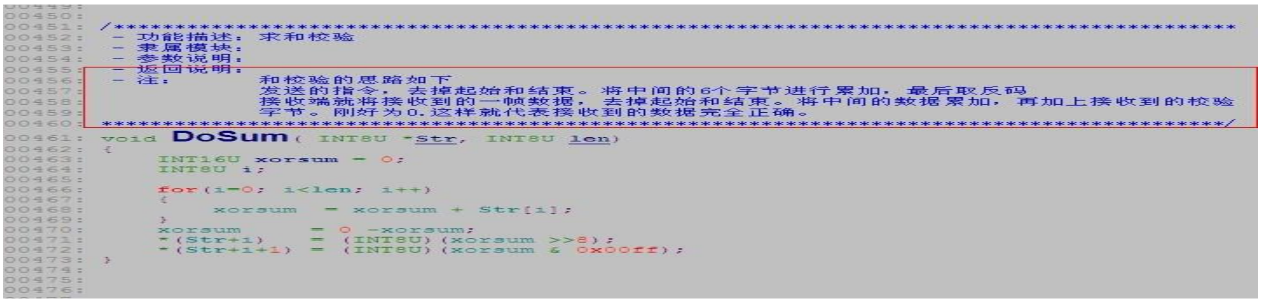
(1)、首先 Installation information”串口猎人”Software, 打开软件， 首先要搜索串口， 找 To refer to 定的端口之后， 指定“波特率”， 我们的模块默认的波特率为 9600 , The most 后就是“启动串行端口”， 这样软件就配置好了。这里有两个概 Reading required 要明确一下第一是“HEX 码”， 我们默认是这个， 这个是用来显 Readings 据的。所以必须设置这里第二是“字符串”， 这个是用来显示打 Printing 符的， 我们这里用不到。

(3), Soft 件配置 OK 之后 Will 需要的指令复制到发送区域 , That is, 可。 With 体的 Instructions 请参照模块的数据手册

(4), Such as 果模块的数据手册没有的测试指令的话 , Please 自行计算 , Especially 其需 To note 意的是校验和这两个字节如何计算不对的话， 模块是不接受指令

的 。

CheckCode is calculated by:



```
/*******************************************************************************
 - 功能描述： 求和校验
 - 隶属模块：
 - 参数说明：
 - 返回说明：
 - 注：          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);
}
```

0 = 24 + x The analogy 0000 0000（0）=0010 0100（24）+1101 1011（DB+1）

Appendix I:SPI-FLASH Capacity and the length of the audio table

Schedule 1-1 MP3－FLASH-16p Module FLASH Capacity and length of audio swap tables:(Work unit:S)

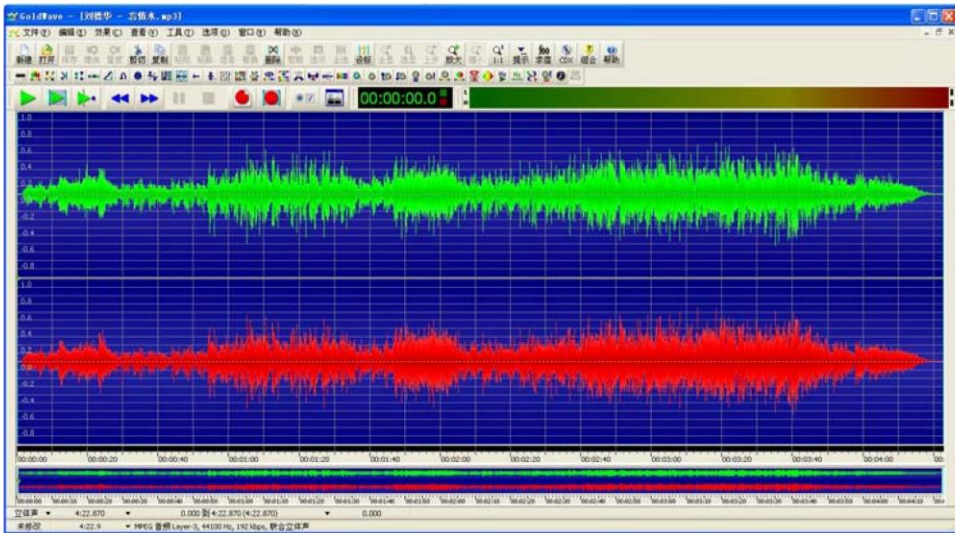
容量 码率	4MBits	8MBits	16MBit	32MBit	64MBit	
16Kbps	252	505	1011	2022	4045	
24Kbps	163	327	654	1309	2618	
32Kbps	113	226	453	906	1812	
64Kbps	59	119	239	477	955	
96Kbps	41	81	162	325	651	
128Kbps	31	61	123	246	493	
160Kbps	24	49	97	194	389	
192Kbps	20	40	81	161	323	
256Kbps	15	30	60	120	241	
320Kbps	11	23	47	95	191	

Note:MP3 File size depends on the bit rate, regardless of the sampling rate. Voice broadcast is recommended 16Kbps～64Kbps, Music recommended 32Kbps～96Kbps。

Third, the rate of conversion

To fightSPIFLASH The characteristics of small volume, stability, we have developedMP3－FLASH-16P Module. Directly through the phone'sMicrousb Line updates voice, but for the commonMP3 Files, mostly4M Bytes or so, useSPIFLASH,

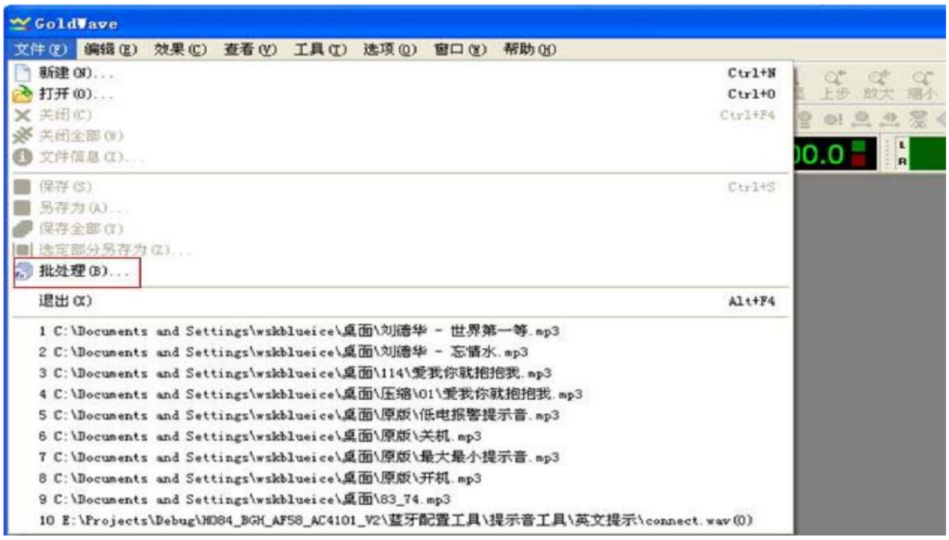
Space is very difficult. But as voice broadcast and prompt, do not need high sampling rates



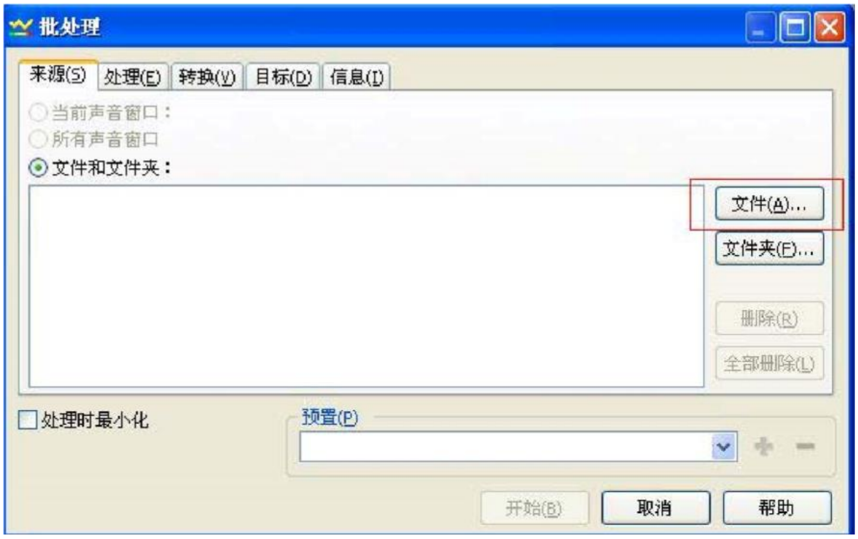
From the top left corner of the chart, we can see“The world's first. MP3”The sampling rate of up to44100HZ。 Bit rate 256KBS。 This parameter describes the current song sound quality is quite good, so take a4.5M Space.

But actually we do not need such a high sound quality, can be compressed. Are as follows:

Use“GoldWave”This piece of software.


GoldWave
文件(F) 编辑(E) 效果(C) 查看(V) 工具(T) 选项(O) 窗口(W) 帮助(H)
新建(N)... Ctrl+N
打开(O)... Ctrl+O
关闭(C) Ctrl+F4
关闭全部(W)
文件信息(I)...
保存(S) Ctrl+S
另存为(A)...
保存全部(Y)
选定部分另存为(Z)...
批处理(B)...
退出(X) Alt+F4
1 C:\Documents and Settings\wskblueice\桌面\刘德华 - 世界第一等.mp3
2 C:\Documents and Settings\wskblueice\桌面\刘德华 - 忘情水.mp3
3 C:\Documents and Settings\wskblueice\桌面\114\爱我你就抱抱我.mp3
4 C:\Documents and Settings\wskblueice\桌面\压缩\01\爱我你就抱抱我.mp3
5 C:\Documents and Settings\wskblueice\桌面\原版\低电报警提示音.mp3
6 C:\Documents and Settings\wskblueice\桌面\原版\关机.mp3
7 C:\Documents and Settings\wskblueice\桌面\原版\最大最小提示音.mp3
8 C:\Documents and Settings\wskblueice\桌面\原版\开机.mp3
9 C:\Documents and Settings\wskblueice\桌面\83_74.mp3
10 E:\Projects\Debug\H084_BGH_AF58_AC4101_V2\蓝牙配置工具\提示音工具\英文提示\connect.wav(0)
00.0

Click on batch processing,



批处理

来源(S) 处理(E) 转换(V) 目标(D) 信息(I)

○ 当前声音窗口：
○ 所有声音窗口
● 文件和文件夹：

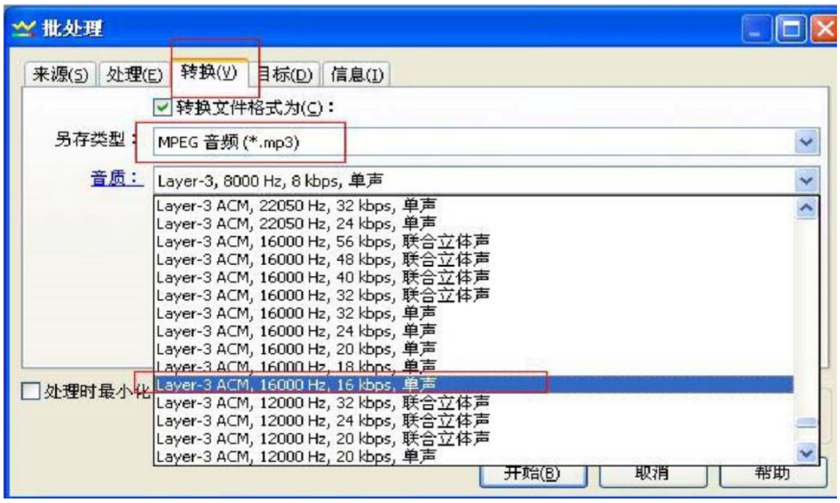
文件(A)...
文件夹(F)...
删除(R)
全部删除(L)

□ 处理时最小化

预置(P)

开始(B) 取消 帮助

Add a file



批处理

来源(S) 处理(E) 转换(V) 目标(D) 信息(I)

☑ 转换文件格式为(C)：

另存类型： MPEG 音频 (*.mp3)

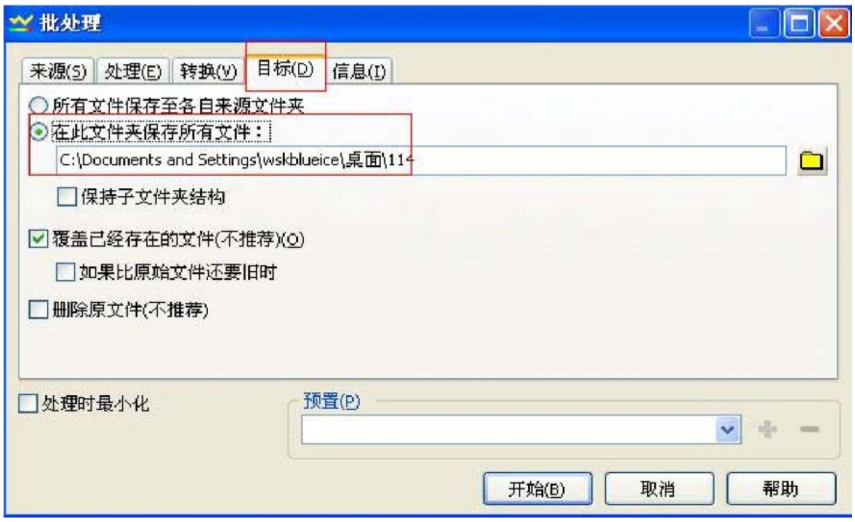
音质： Layer-3, 8000 Hz, 8 kbps, 单声

```
Layer-3 ACM, 22050 Hz, 32 kbps, 单声
Layer-3 ACM, 22050 Hz, 24 kbps, 单声
Layer-3 ACM, 16000 Hz, 56 kbps, 联合立体声
Layer-3 ACM, 16000 Hz, 48 kbps, 联合立体声
Layer-3 ACM, 16000 Hz, 40 kbps, 联合立体声
Layer-3 ACM, 16000 Hz, 32 kbps, 联合立体声
Layer-3 ACM, 16000 Hz, 32 kbps, 单声
Layer-3 ACM, 16000 Hz, 24 kbps, 单声
Layer-3 ACM, 16000 Hz, 20 kbps, 单声
Layer-3 ACM, 16000 Hz, 18 kbps, 单声
Layer-3 ACM, 16000 Hz, 16 kbps, 单声
Layer-3 ACM, 12000 Hz, 32 kbps, 联合立体声
Layer-3 ACM, 12000 Hz, 24 kbps, 联合立体声
Layer-3 ACM, 12000 Hz, 20 kbps, 联合立体声
Layer-3 ACM, 12000 Hz, 20 kbps, 单声
```

□ 处理时最小化

开始(B) 取消 帮助

Choose“Conversion”, Set the sampling rate for16000KHZAnd bit rate for16KBS。



批处理

来源(S) 处理(E) 转换(V) 目标(D) 信息(I)

○ 所有文件保存至各自来源文件夹
● 在此文件夹保存所有文件：
C:\Documents and Settings\wskblueice\桌面\114

□ 保持子文件夹结构

☑ 覆盖已经存在的文件(不推荐)(O)

□ 如果比原始文件还要旧时

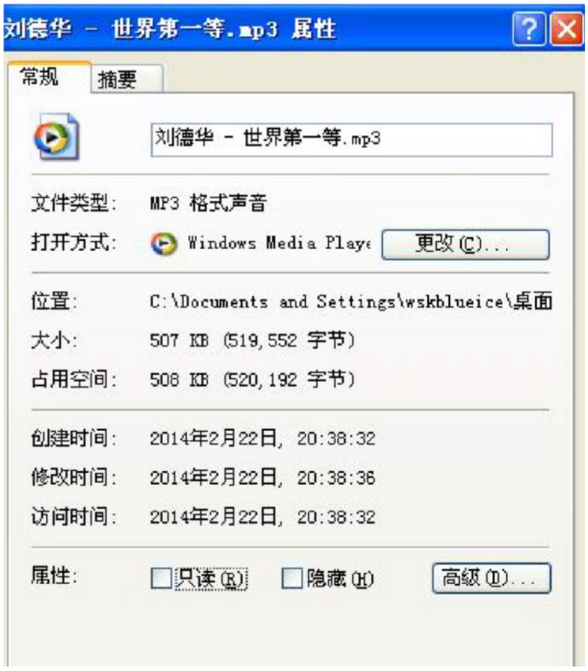
□ 删除原文件(不推荐)

□ 处理时最小化

预置(P)

开始(B) 取消 帮助

Then the path Specifies the converted files can be



刘德华 - 世界第一等.mp3

文件类型: MP3 格式声音

打开方式: Windows Media Playe

位置: C:\Documents and Settings\wskblueice\桌面

大小: 507 KB (519,552 字节)

占用空间: 508 KB (520,192 字节)

创建时间: 2014年2月22日, 20:38:32

修改时间: 2014年2月22日, 20:38:36

访问时间: 2014年2月22日, 20:38:32

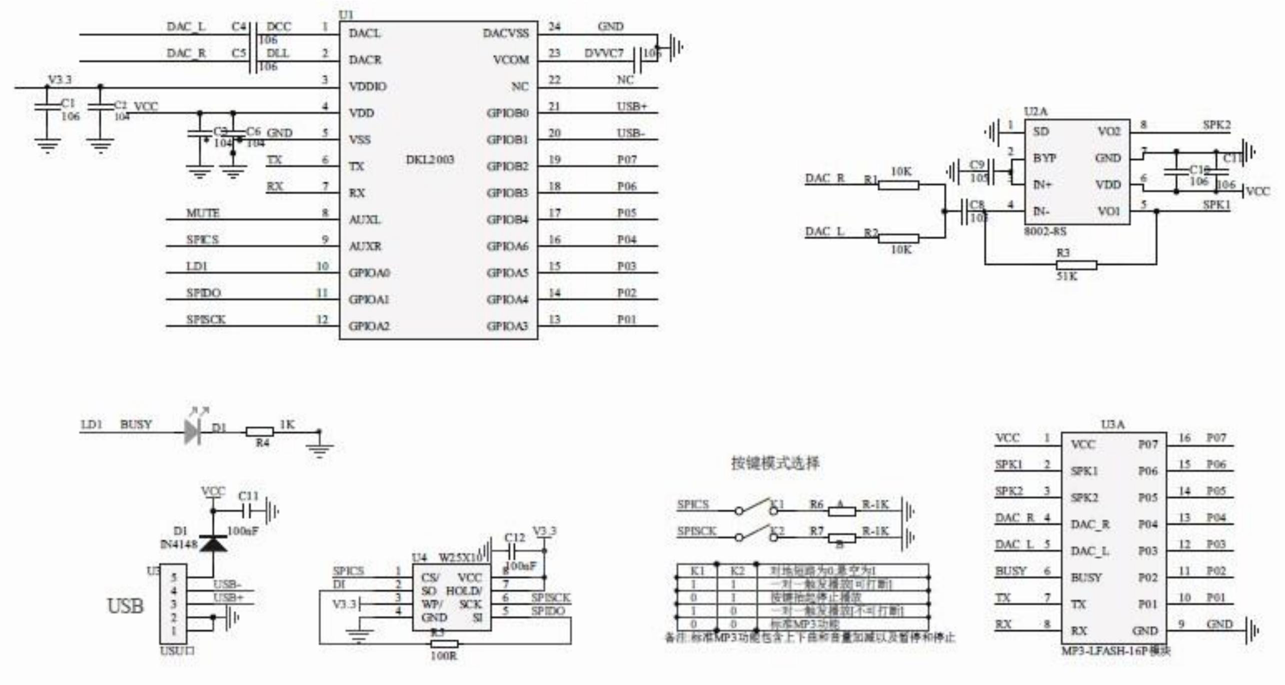
属性: □只读(R) □隐藏(H)

After compression,4.5M Files into 507K . Step is

Notes:

1. If it iswav File, you can also use this software convertMP3
2. After the transition effects, the user can directly on the computer listen for a minute effect, play computer effects, and We play is consistent with the results of the module
3. If sound quality is bad, you can increase the sample rate and bit rate of these two parameters. You can try it for yourself
4. If you need to modify the volume of the audio source, and trim an audio source, you can use this software

Four, module schematics



K1	K2	对地短路为0,悬空为1
1	1	一对一触发播放[可打断]
0	1	按键抬起停止播放
1	0	一对一触发播放[不可打断]
0	0	标准MP3功能

备注:标准MP3功能包含上下曲和音量加减以及暂停和停止

FourProgramming examples

Program example: serial port named

```
/*******************************************************************************
Function: towards a chip named the first song and the second song, the basic procedures for user testing

Date      : The year 2013-05-06

Operating environment:STC   Crystal:11.0592M     Baud rate: 9600

Notes    : In General, science and technology 51 Development Board debug OK --- STC89C516RD+

1, The testing program must be a module or chip equipment online in the programme, such as U Plates,TF Card,FLASH

*******************************************************************************/ #include "REG52.h"


#define COMM_BAUD_RATE 9600 //Serial baud rate

#define OSC_FREQ 11059200 //Run Crystal: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-- );

}


/*******************************************************************************

Function description: Serial port 1 Initialize

Note:      Set to 9600 Baud rate

*************************************/ void Serial_init(void)

{

        TMOD = 0x20; //Set T1 Baud rate generator

        SCON = 0x50; 0101,0000 8Data bits, No parity

        PCON = 0x00; PCON=0;

        TH1=256-(OSC_FREQ/COMM_BAUD_RATE/32/12);//Set to 9600 Baud rate

        TL1=256-(OSC_FREQ/COMM_BAUD_RATE/32/12);

  TR1 = 1;                                //Timer 1 Open it

  REN = 1;                                //Serial port 1 Receive enabled

  ES = 1;                                 //Serial port 1 Interrupt enable

}

void Uart_PutByte(INT8U ch)

{

  SBUF = ch; while(! TI){;}

  TI = 0;

}
```

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

Function description: Serial port sends out commands[Including control and query]

Parameter description: CMD:Control instructions, consult the instruction list also includes queries related instruction            feedback:Do
you need an answer[0:Without answering,1:Need an answer]

              data:Transmission parameters

********************************************************************************/ void SendCmd(INT8U len)

{

    INT8U i = 0 ;

    Uart_PutByte(0x7E); //Start     for(i=0; i<len; i++)//Data

    {

                            Uart_PutByte(Send_buf[i]) ;

    }

    Uart_PutByte(0xEF) ;//End

}


/********************************************************************************

Function: checksum

And calibration of thought are as follows:

    Send instructions, remove the start and end. The intermediate 6 Bytes added to finally take back yards. Receiver will receive the frame,
remove the start and end. Data accumulated in the Middle, plus receive parity bytes. Just as 0.On behalf of received data is completely
correct.

/ 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; //Reserved bytes

    Send_buf[1] = 0x06; //Length

    Send_buf[2] = CMD; //Control commands     Send_buf[3] = feedback;//Whether you need feedback     Send_buf[4] = (INT8U)(dat >>
8);//datah

    Send_buf[5] = (INT8U)(dat); datal

    DoSum(&Send_buf[0],6); //Check
    SendCmd(8); //Send this frame data
}


void main()
{
                Serial_init() ;//Initialization of the serial register setting
    Uart_SendCMD(0x03 , 0 , 0x01) ;//Play the first song
    Delay_Ms(1000) ;//Delay probably 6S
    Uart_SendCMD(0x03 , 0 , 0x02) ;//Play next song
    Delay_Ms(1000) ;//Delay probably 6S
    Uart_SendCMD(0x03 , 0 , 0x04) ;//Play the fourth song     while(1) ;
}
```