Eye record

First, the chip hardware description

1 , The chip's power supply

Optimal operating voltage of the chip 4.2V . So if the user is using 5V A power supply, a diode connected in series recommendations

2 Chip LED [Power Status]

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 Chip LED [operating 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 , An audio output Description

1 , SPK1 with SPK2 Swiss received two speakers, regardless of possible positive and negative, attention: only allow access 2W The following speaker

2 , Then the amplifier or headphones directly connected DAC - R with DAC - L Ends Note: common ground (ground power supply)

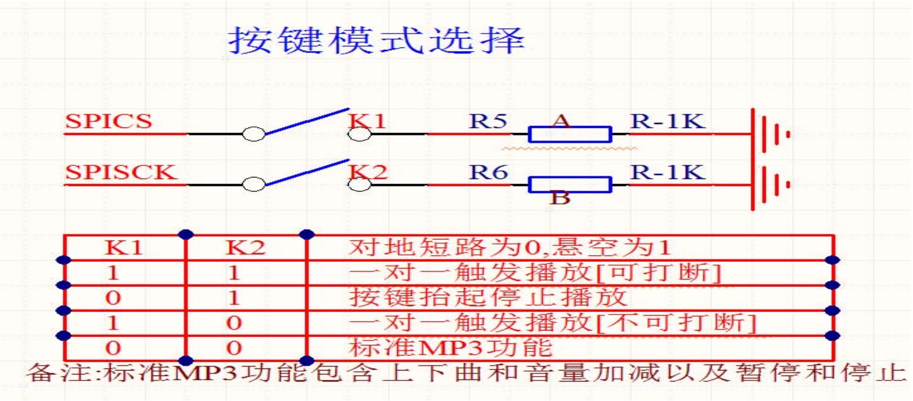
5 Chip debugging instructions

(1) , Our chip is the default plug USB Line, to enter download mode. When the user finishes updating voice, a trigger any IO

Mouth to exit download mode and resume normal working condition

(2) , Starting with chip debugging easy to difficult, from simple to complex first, and then transfer the keys to ensure the normal serial after serial debugging debugging normal hand,

and then transferred microcontroller, TF card no sound, the first computer via USB plug, see if I can read TF card letter



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

Serial debugging assistant test	Command sent [with parity]	Command sent [without parity]	Remark

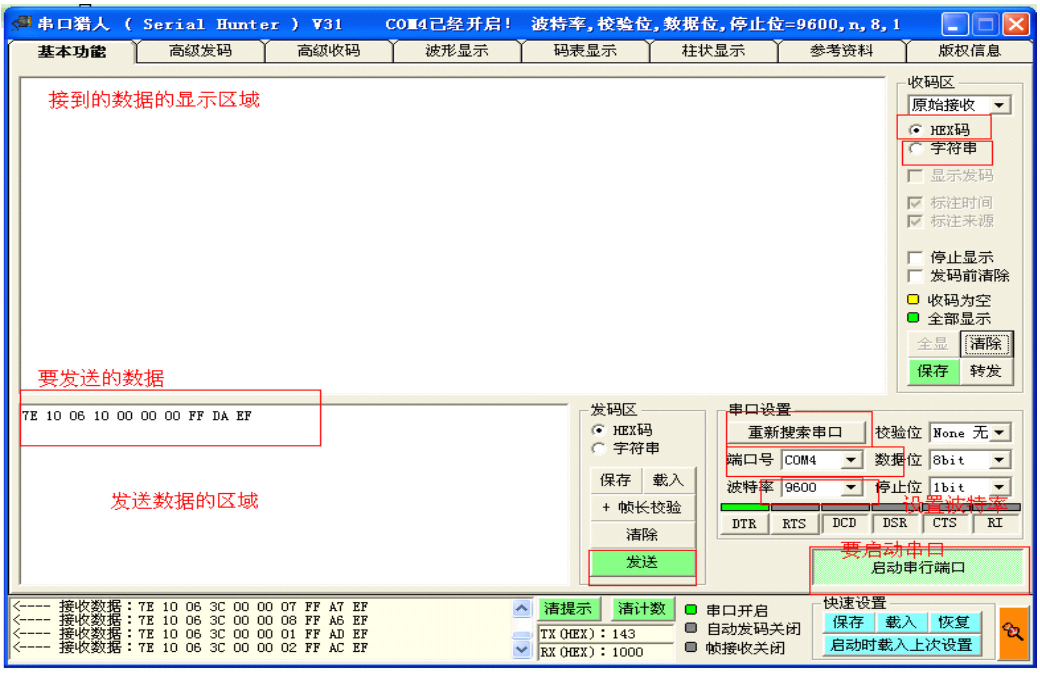
[next song]	7E FF 06 01 00 00 00 FE FA EF	7E FF 06 01 00 00 00 EF	
[On one]	7E FF 06 02 00 00 00 FE F9 EF	7E FF 06 02 00 00 00 EF	
[Specify the track]	7E FF 06 03 00 00 01 FE F7 EF	7E FF 06 03 00 00 01 EF	Specifies the first song to play
	7E FF 06 03 00 00 02 FE F6 EF	7E FF 06 03 00 00 02 EF	Specifies the second song
	7E FF 06 03 00 00 0A FE EE EF	7E FF 06 03 00 00 0A EF	Specify the first 10 first
Volume Up	7E FF 06 04 00 00 00 FE F7 EF	7E FF 06 04 00 00 00 EF	
Volume down	7E FF 06 05 00 00 00 FE F6 EF	7E FF 06 05 00 00 00 EF	
[Specifies the volume]	7E FF 06 06 00 00 1E FE D7 EF	7E FF 06 06 00 00 1E EF	As specified volume 30 level
[Designation EQ]	7E FF 06 07 00 00 01 FE F3 EF	7E FF 06 07 00 00 01 EF	Retention
[Loop Tracks]	7E FF 06 08 00 00 01 FE F2 EF	7E FF 06 08 00 00 01 EF	Playing the first cycle
	7E FF 06 08 00 00 02 FE F1 EF	7E FF 06 08 00 00 02 EF	The first play of the second cycle
	7E FF 06 08 00 00 0A FE E9 EF	7E FF 06 08 00 00 0A EF	The first loop tenth
	7E FF 06 08 00 01 01 FE F1 EF	7E FF 06 08 00 01 01 EF	Loop FLASH of FOLDER1 First song
	7E FF 06 08 00 02 01 FE F0 EF	7E FF 06 08 00 02 01 EF	Loop FLASH of FOLDER2 First song
[Specifies the playback device]	7E FF 06 09 00 00 01 FE F1 EF	7E FF 06 09 00 00 01 EF	Designated player UDISK
	7E FF 06 09 00 00 02 FE F0 EF	7E FF 06 09 00 00 02 EF	Designated player TF
	7E FF 06 09 00 00 04 FE EE EF	7E FF 06 09 00 00 04 EF	Designated player FLASH
[Into sleep mode]	7E FF 06 0A 00 00 00 FE F1 EF	7E FF 06 0A 00 00 00 EF	
[Wake-sleep]	7E FF 06 0B 00 00 00 FE F0 EF	7E FF 06 0B 00 00 00 EF	
[Chip 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	
[time out]	7E FF 06 0E 00 00 00 FE ED EF	7E FF 06 0E 00 00 00 EF	
[Specified folder filename]	7E FF 06 0F 00 01 01 FE EA EF	7E FF 06 0F 00 01 01 EF	Appointed as" 01 " Folder, track for" 001 '
	7E FF 06 0F 00 01 02 FE E9 EF	7E FF 06 0F 00 01 02 EF	Appointed as" 01 " Folder, track for" 002 '

stand by 1000 first	7E FF 06 14 00 10 FF FD D8 EF	7E FF 06 14 00 10 FF EF	Appointed as" 01 " Folder, track for" 0255 '
	7E FF 06 14 00 17 CF FE 01 EF	7E FF 06 14 00 17 CF EF	Appointed as" 01 " Folder, track for" 1999 '
	7E FF 06 14 00 C0 01 FE 26 EF	7E FF 06 14 00 C0 01 EF	Appointed as" 12 " Folder, track for" 0001 '
	7E FF 06 14 00 C0 FF FD 28 EF	7E FF 06 14 00 C0 FF EF	Appointed as" 12 " Folder, track for" 0255 '
	7E FF 06 14 00 C7 CF FD 51 EF	7E FF 06 14 00 C7 CF EF	Appointed as" 12 " Folder, track for" 1999 '
Stop play	7E FF 06 16 00 00 00 FE E5 EF	7E FF 06 16 00 00 00 EF	Stop decoding software
Specified folder loop	7E FF 06 17 00 02 00 FE E2 EF	7E FF 06 17 00 02 00 EF	Designation 02 Folders loop
	7E FF 06 17 00 01 00 FE E3 EF	7E FF 06 17 00 01 00 EF	Designation 01 Folders loop
Shuffle Playback	7E FF 06 18 00 00 00 FE E3 EF	7E FF 06 18 00 00 00 EF	Shuffle Playback
Single Loop	7E FF 06 19 00 00 00 FE E2 EF	7E FF 06 19 00 00 00 EF	Single loop open
	7E FF 06 19 00 00 01 FE E1 EF	7E FF 06 19 00 00 01 EF	Single closed loop
DAC setting	7E FF 06 1A 00 00 00 FE E1 EF	7E FF 06 1A 00 00 00 EF	open DAC
	7E FF 06 1A 00 00 01 FE E0 EF	7E FF 06 1A 00 00 01 EF	turn off DAC
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 Play 5 , 1 , 2 , 3 , 4
Combination Play	7E FF 0C 21 00 05 01 02 03 04 06 0708 FE B0 EF	7E FF 0C 21 00 05 01 02 03 04 06 07 08 EF	Combination Play 5 , 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	30 Level playing the first volume 1 song
	7E FF 06 22 00 0F 01 FE C9 EF	7E FF 06 22 00 0F 01 EF	15 Level playing the first volume 1 song
	7E FF 06 22 00 0F 02 FE C8 EF	7E FF 06 22 00 0F 02 EF	15 Level playing the first volume 2 song

Query the current status	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	
[The current inquiry EQ]	7E FF 06 44 00 00 00 FE B7 EF	7E FF 06 44 00 00 00 EF	
U The total number of disk file	7E FF 06 47 00 00 00 FE B4 EF	7E FF 06 47 00 00 00 EF	The current total number of files equipment
TF The total number of files	7E FF 06 48 00 00 00 FE B3 EF	7E FF 06 48 00 00 00 EF	
FLASH The total number of files	7E FF 06 49 00 00 00 FE B2 EF	7E FF 06 49 00 00 00 EF	
U Disc current track	7E FF 06 4B 00 00 00 FE B0 EF	7E FF 06 4B 00 00 00 EF	Currently playing track
TF Current track	7E FF 06 4C 00 00 00 FE AF EF	7E FF 06 4C 00 00 00 EF	
FLASH Current Folder Tracks pointer	7E FF 06 4D 00 00 00 FE AE EF	7E FF 06 4D 00 00 00 EF	
Queries folder total number of tracks	7E FF 06 4E 00 00 01 FE AC EF	7E FF 06 4E 00 00 01 EF	Inquire 01 Folder or FOLDER1 The total number of tracks
Inquire TF or U Total disk file Number of folders	7E FF 06 4F 00 00 00 FE AC EF	7E FF 06 4F 00 00 00 EF	Only supports TF Card and U plate
The current folder pointer [FLASH]	7E FF 06 61 00 00 00 FE 9A EF	7E FF 06 61 00 00 00 EF	Query currently playing folder [branch hold FLASH]

Third, test methods

1 , Serial port software operation



```
7E 10 06 10 00 00 00 FF DA EF
```

校验位 None 无
端口号 COM4
数据位 8bit
波特率 9600
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
```

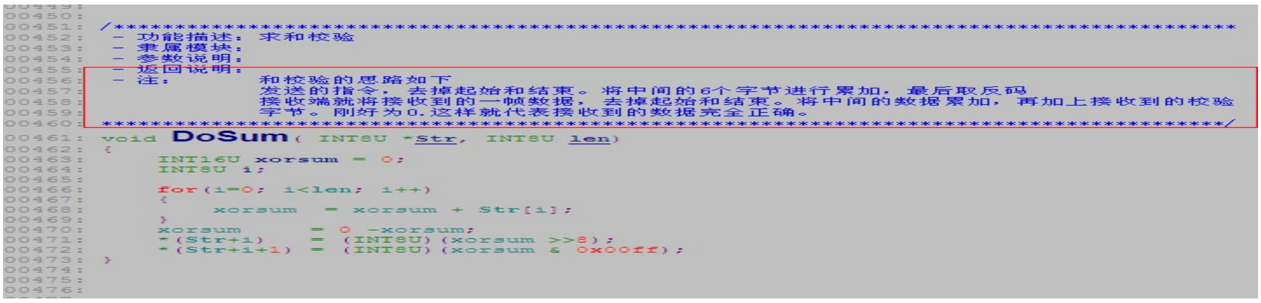
TX (HEX)：143
RX (HEX)：1000

(1) , First of all Installation information in the " Serial Hunter " Software, open the software, you must first search for the serial port, refer to find After a given port, specify "baud", our default baud rate for the module 9600 Most After that "Start Serial Port", so the software is configured. There are two concepts need The first is to define what " HEX Code ", this is our default, this is used to display the number of Data. Here is the second set must be "string", this word is used to display the print Breaks, we can not use here.

(3) Software Configuration OK Thereafter, the required instructions to copy the transmission region can. Specific instructions Please refer to the module data sheets

(4) , If the data module is not manual test command, then, calculate their own particular needs Note Meaning how two checksum bytes do not, then the module does not accept the instruction .

Calculated check code:



```
/*****************************************************************************************
 - 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 analogy 0000 0000 (0) = 00,100,100 (twenty four) + 11,011,011 (DB + 1)

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

Schedule 1-1 YX5300-24SS FLASH Capacity and the length of time of the audio conversion table :(Unit: S)

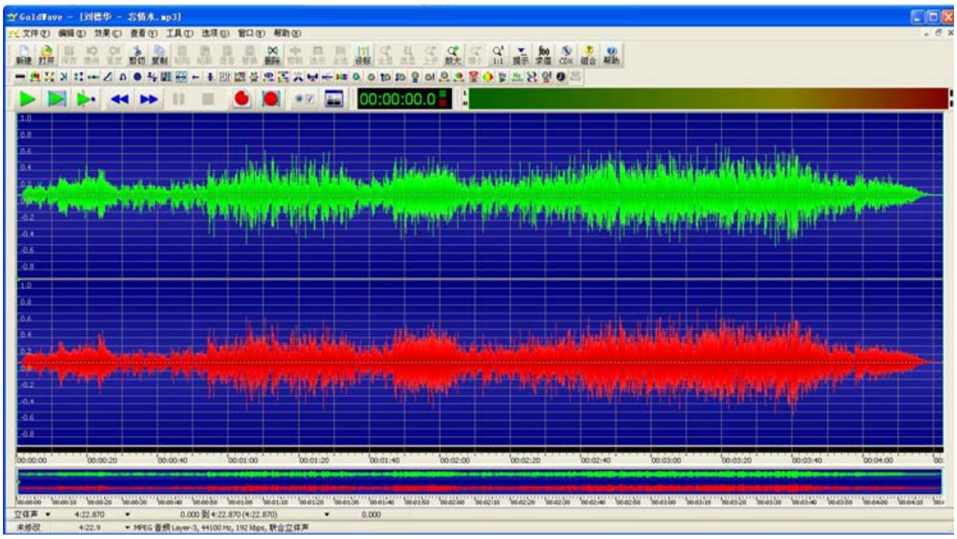
Capacity rate	4MBits	8MBits	16MBit	32MBit	64MBit	
16Kbps	252	505	1011	2022	4045	
24Kbps	163	327	654	1309	2618	
32Kbps	113	226	453	906	1812	
64Kbps	59	119	239	477	955	
96Kbps	41	81	162	325	651	
128Kbps	31	61	123	246	493	
160Kbps	twenty four	49	97	194	389	
192Kbps	20	40	81	161	323	
256Kbps	15	30	60	120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

Note: MP3 File size depends on the bit rate, regardless of the sample rate. Voice Broadcast recommended 16Kbps ~ 64Kbps , Music players recommended 32Kbps ~ 96Kbps .

Third, the rate of conversion method

1 To fight for SPIFLASH Small-capacity, stable characteristics, we have developed YX5300-24SS . Directly through mobile phones Microusb Update the voice line, but for the common MP3 Documents, mostly 4M Bytes or so, use SPIFLASH ,

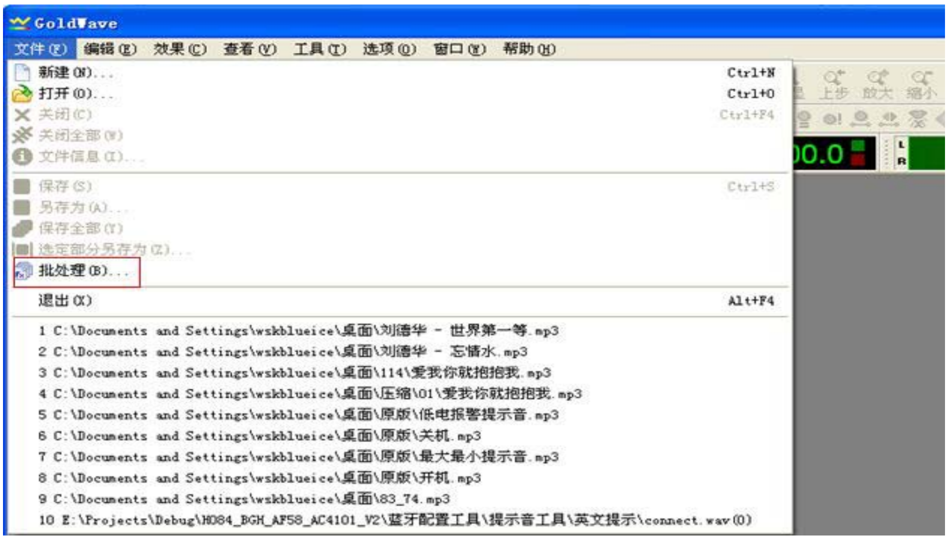
Space becomes very hard for. But we generally as a voice broadcast and tips occasions, do not need high sampling rate



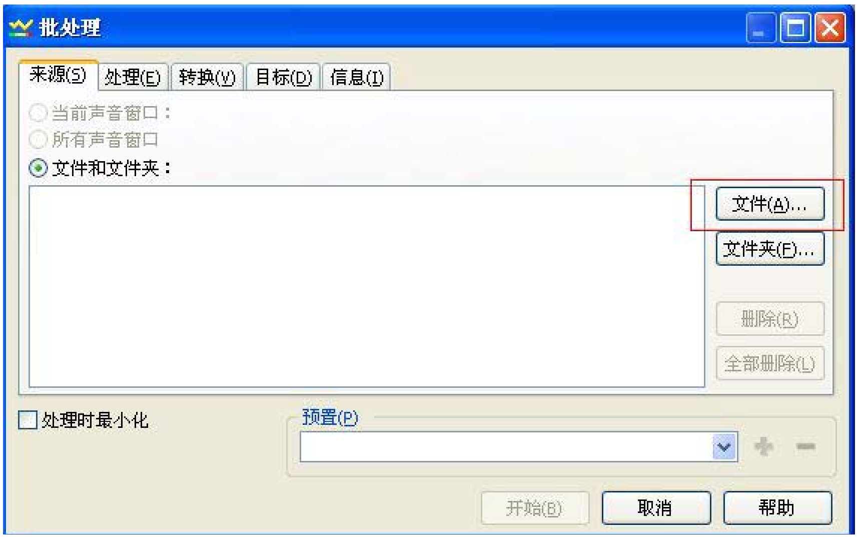
立体声 4:22.870 0.000 到 4:22.870 (4:22.870) 0.000
未修改 4:22.9 MPEG 音频 Layer-3, 44100 Hz, 192 kbps, 联合立体声

From the lower left corner of the chart, we can see "the world's first and so on. MP3 "Sampling rate of up to 44100HZ . Bit rate 256KBS . This parameter, indicating the current song sound quality is quite good, so take up 4.5M Space.

2 But actually we do not need such a high quality, then you can be compressed. As follows: Use " GoldWave "This software.

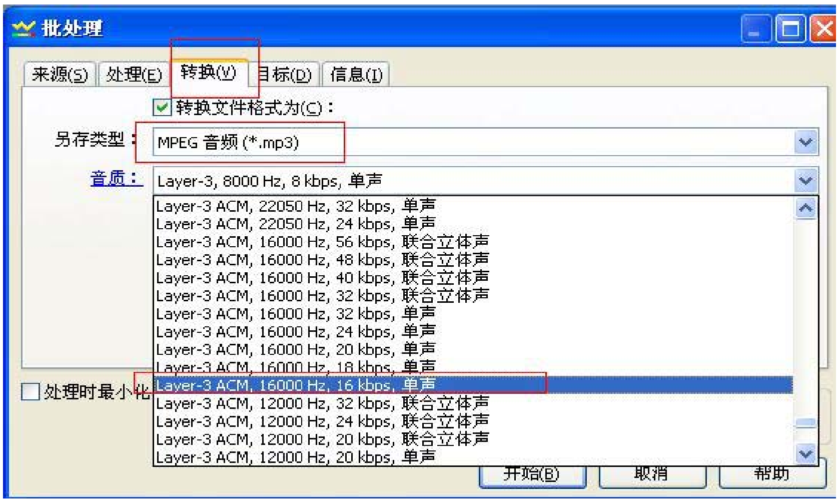

GoldWave
文件(F)
编辑(E)
效果(C)
查看(V)
工具(T)
选项(O)
窗口(W)
帮助(H)
新建(N)... Ctrl+N
打开(O)... Ctrl+O
关闭(C) Ctrl+F4
关闭全部(W)
文件信息(I)...
保存(S) Ctrl+S
另存为(A)...
保存全部(Y)
选定部分另存为(Z)...
批处理(B)...
退出(X) Alt+F4
1 C:\Documents and Settings\wskblueice\桌面\刘德华 - 世界第一等.mp3
2 C:\Documents and Settings\wskblueice\桌面\刘德华 - 忘情水.mp3
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)
上步
放大
缩小
00.0

Click Batch,



○ 当前声音窗口：
○ 所有声音窗口
● 文件和文件夹：
□ 处理时最小化

add files



☑ 转换文件格式为(C)：

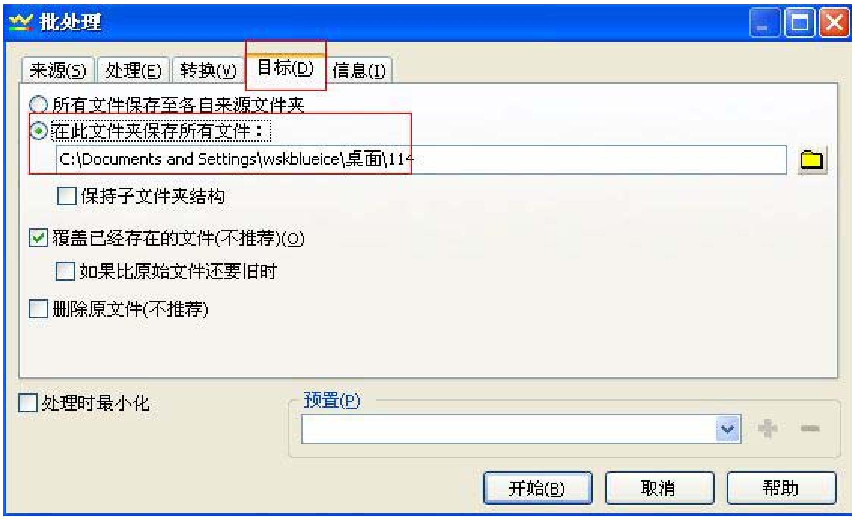
另存类型： MPEG 音频 (*.mp3)

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
```

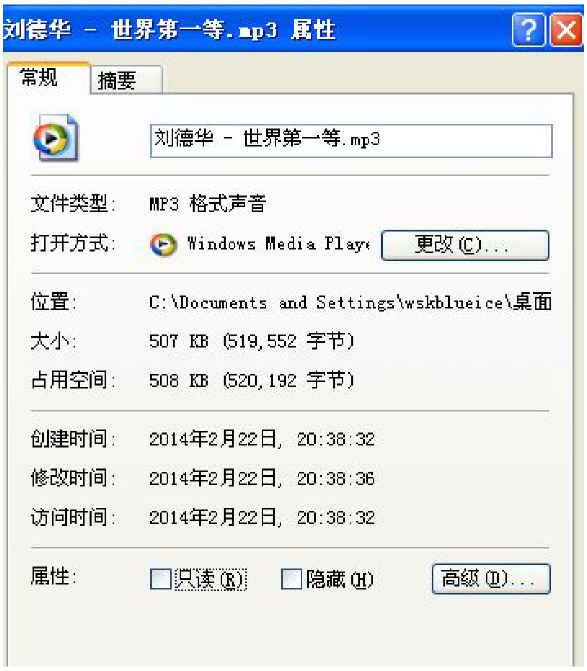
□ 处理时最小化

Select "conversion" Set the sampling rate 16000KHZ Bit rate 16KBS .



○ 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

Path to store files after conversion can then specify what



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 become 507K A. Is one such step

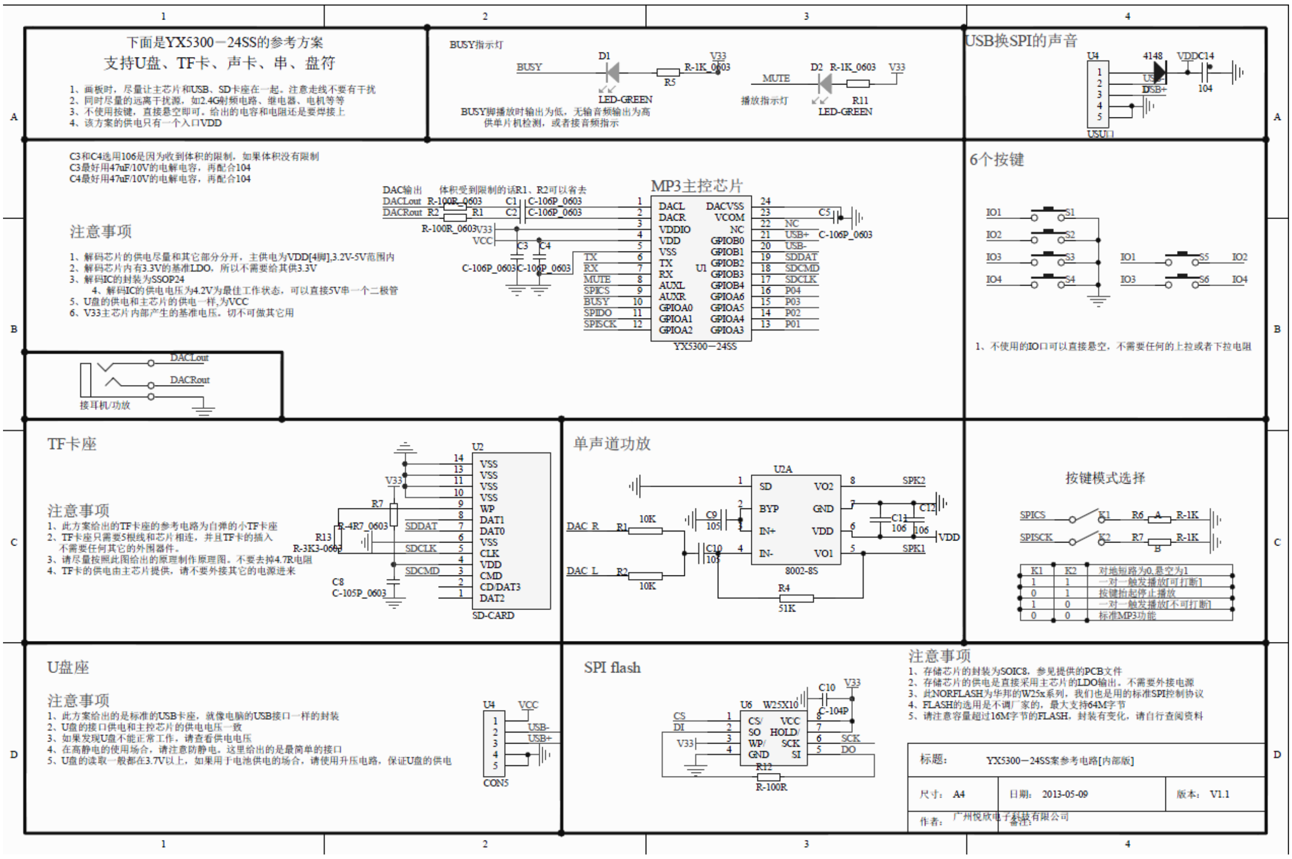
Remarks:

1 ,if wav File, you can also use this software to convert MP3 2 , After the effect of the conversion, users can listen to it in the computer above effect, playing computer above effect, and the effect of our players is the same chip

3 If that is not good sound quality, it can be appropriate to increase the sampling rate and bit rate of these two parameters. You can try it yourself

4 If you need to modify the source volume, and trim audio, you can use this software

Fourth, the chip's schematics



下面是YX5300－24SS的参考方案
支持U盘、TF卡、声卡、串、盘符

1、画板时，尽量让主芯片和USB、SD卡座在一起。注意走线不要有干扰
2、同时尽量的远离干扰源，如2.4G射频电路、继电器、电机等等
3、不使用按键，直接悬空即可。给出的电容和电阻还是要焊接上
4、该方案的供电只有一个入口VDD

BUSY脚播放时输出为低，无输音频输出为高
供单片机检测，或者接音频指示

C3和C4选用106是因为收到体积的限制，如果体积没有限制
C3最好用47uF/10V的电解电容，再配合104
C4最好用47uF/10V的电解电容，再配合104

体积受到限制的话R1、R2可以省去

注意事项

1、解码芯片的供电尽量和其它部分分开，主供电为VDD[4脚],3.2V-5V范围内
2、解码芯片内有3.3V的基准LDO，所以不需要给其供3.3V
3、解码IC的封装为SSOP24
4、解码IC的供电电压为4.2V为最佳工作状态，可以直接5V串一个二极管
5、U盘的供电和主芯片的供电一样,为VCC
6、V33主芯片内部产生的基准电压。切不可做其它用

1、不使用的IO口可以直接悬空，不需要任何的上拉或者下拉电阻

注意事项

1、此方案给出的TF卡座的参考电路为自弹的小TF卡座
2、TF卡座只需要5根线和芯片相连，并且TF卡的插入
不需要任何其它的外围器件。
3、请尽量按照此图给出的原理制作原理图。不要去掉4.7R电阻
4、TF卡的供电由主芯片提供，请不要外接其它的电源进来

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

注意事项

1、此方案给出的是标准的USB卡座，就像电脑的USB接口一样的封装
2、U盘的接口供电和主控芯片的供电电压一致
3、如果发现U盘不能正常工作，请查看供电电压
4、在高静电的使用场合，请注意防静电。这里给出的是最简单的接口
5、U盘的读取一般都在3.7V以上，如果用于电池供电的场合，请使用升压电路，保证U盘的供电

注意事项

1、存储芯片的封装为SOIC8，参见提供的PCB文件
2、存储芯片的供电是直接采用主芯片的LDO输出。不需要外接电源
3、此NORFLASH为华邦的W25x系列，我们也是用的标准SPI控制协议
4、FLASH的选用是不调厂家的，最大支持64M字节
5、请注意容量超过16M字节的FLASH，封装有变化，请自行查阅资料

标题： YX5300－24SS案参考电路[内部版]		
尺寸： A4	日期： 2013-05-09	版本： V1.1
作者： 广州悦欣电子科技有限公司	备注：	

four, Program Example

Program Example: Serial designated player

```
/ ************************************************** ****************************
  -  Function: electrical chip are designated music playing the first and second curved, basic program for the user to test
  -  date : 2013-05-06
  -  Operating environment: STC   Crystal: 11.0592M    Baud rate: 9600
  -  Remark : In Pu in science and technology 51 Development board debugging OK --- STC89C516RD + 1 The test procedure is a module or chip
program must have equipment line, such as U plate, TF card, FLASH
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * /
#include "REG52.h"

#define COMM_BAUD_RATE 9600                            // Baud rate
#define OSC_FREQ                        11,059,200         // Crystal Run: 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--);}

/ ************************************************** ****************************
  -  Description: Serial 1 initialization
  -  Note:                 Set as 9600 Baud Rate
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * /
void Serial_init (void) {TMOD

= 0x20;                                                    // Set up T1 Baud Rate Generator.
       SCON = 0x50;                                       // 0101,0000 8 Data bits, no parity
       PCON = 0x00;                                       // PCON = 0;
       TH1 = 256- (OSC_FREQ / COMM_BAUD_RATE / 32/12); // Set as 9600 Baud Rate
       TL1 = 256- (OSC_FREQ / COMM_BAUD_RATE / 32/12); TR1
                       = 1;                            // Timer 1 turn on
        REN             = 1;                           // Serial ports 1 Receive enable
        ES              = 1;                           // Serial ports 1 Interrupt Enable
} Void Uart_PutByte (INT8U ch) {


        SBUF = ch; while

     (TI!) {;} TI = 0;}



/ ************************************************** ****************************
  -  Functional Description: Serial transmission command out [including control and query]
  -  Parameter Description: CMD: Represents control instructions, please refer to the instruction list, also includes instructions related queries
                           feedback: Need to answer [ 0: No reply, 1: Need to answer]
                           data: Parameter transfer
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * /
```

```
void SendCmd (INT8U len) {

        INT8U i = 0;
        Uart_PutByte (0x7E); // Starting
        for (i = 0; i <len; i ++) // data
        { Uart_PutByte (Send_buf [i]);} Uart_PutByte


        (0xEF); // End
}

/ ****************************************************** ********************************
  -  Description: checksum
  -  And verify the following idea:
        Instruction sent to remove start and end. The middle 6 Bytes are accumulated, the last inversion code. The receiving end will be received frame of data,
start and end removed. Intermediate accumulation data, plus parity bytes received. Just as 0. This represents the received data is 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 instruction
        Send_buf [3] = feedback; // The need for feedback
        Send_buf [4] = (INT8U) (dat >> 8); // datah Send_buf
        [5] = (INT8U) (dat);                             // datal
        DoSum (& Send_buf [0], 6);                       // check
        SendCmd (8);                  // This data frame transmission
}

void main ()
{Serial_init (); // Serial register initial setting

        Uart_SendCMD (0x03, 0, 0x01); // Playing the first
        Delay_Ms (1000); // Probably delay 6S Uart_SendCMD (0x03, 0, 0x02); // The
        first play of the second
        Delay_Ms (1000); // Probably delay 6S Uart_SendCMD (0x03, 0, 0x04); // The
        first play of the fourth
        while (1);}
```