

目 录

1. 概述	3
1.1 简介	3
1.2 功能	3
2. 模块的分类说明	3
2.1 支持文件系统的模块	4
2.1.1 MP3-TF-16P 模块(支持 TF 卡、U 盘播放)	4
2.1.2 MP3-FLASH-16P 模块(支持 USB 更新 SPIFLASH)	4
3. 串口的说明	5
3.1 校验的说明	5
3.1.1 校验的计算说明	5
3.1.2 校验和不带校验的兼容说明	6
3.1.3 校验代码的移植	6
3.1.4 MCU 的晶振选择	7
3.2 串口调试的说明	8
3.3 模块调试的步骤	8
3.4 串口的误差分析	10
1、首先明确串口的特点 UART	10
通用异步收发传输器 (UNIVERSAL ASYNCHRONOUS RECEIVER/TRANSMITTER), 通常称作 UART, 是一种异步收发传输器特点如下:	10
4. 播放的说明	11
4.1 指定曲目说明	11
4.2 USB 更新 TF 卡的说明	11
4.3 带有文件夹的设备, 文件物理排序说明	12
4.4 制作语音文件	13
5. 注意事项	14
5.1 模块供电说明	14
5.2 模块低功耗说明	14
5.3 TF 卡的供电走线说明	15
5.4 模块其它说明	15
5.4 常见的接口说明	16
5.5 常音频的杂音处理	19
5.5.1 DAC 输出注意事项	19
5.5.2 功放参考消除杂音	21
5.5.3 电源纹波	22

5.6 客户问题集锦23

1. 概述

我们所有的语音类的模块以及芯片，均属于自主开发，所以模块以及芯片的性能均是经过严格测试的，我们能保证产品大批量出货的性能一致性，以及个性化方案的定制。我们方案的优势在于操作简单、性能优越、资料齐全。本文档只争对模块的调试进行说明，以及一些注意事项，对芯片的性能不做说明。

1.1 简介

我们的语音类产品大体分类如下：

MP3-TF-16P	支持 TF 卡、U 盘播放	支持串口协议,自带3W 的功放
MP3-FLASH-16P	支持 SPI flash、并且通过 USB 直接更新语音	支持串口协议,自带3W 的功放
YX5200-24SS 芯片	MP3-TF-16P 主控芯片，支持 TF 卡、U 盘、SPI flash 播放	
YX5100-24SS 芯片	MP3-FLASH-16P 主控芯片，支持 SPI flash 播放	

1.2 功能

1、我们的 MP3-TF-16P 产品主要围绕的是以下功能，进行深度的剪裁和组合，已满足不同的需求

- 支持 FAT、FAT16、FAT32 文件系统。和 windows 的硬盘文件系统一样
- 支持 USB2.0、以及标准的 SPI 协议的 SD 卡驱动，芯片自己完成
- 支持传统的 MCU 的输入输出控制功能，并且有若干的管脚可供使用
- 支持 MP3、WAV 的硬解码，芯片自动完成，无需用户操心
- 支持工业级别的串口通信协议，丰富且又灵活的控制指令

2、根据以上这些功能，我们就组合出了两大类型的产品：

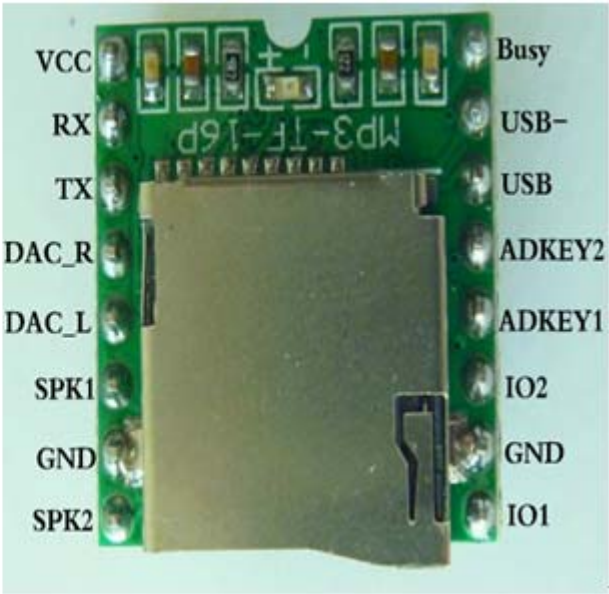
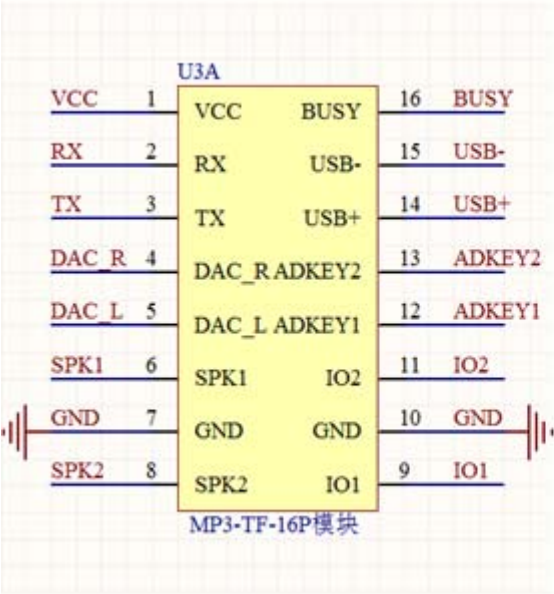
- 支持文件系统的 TF 卡、U 盘播放，以及 USB 更新 TF 卡的芯片方案以及模块，如：MP3-TF-16P 模块，以及 YX5200-24SS 芯片。
- 支持 USB 更新 SPIFLASH 语音的方案,如 MP3-FLASH-16P 模块，以及 YX5100-24SS 芯片方案。

2. 模块的分类说明

- 支持文件系统的模块
- 支持 USB 更新 SPIFLASH 的模块

2.1 支持文件系统的模块

2.1.1 MP3-TF-16P 模块(支持 TF 卡、U 盘播放)



2.1.2 MP3-FLASH-16P 模块(支持 USB 更新 SPIFLASH)



3. 串口的说明



模块的串口为 3.3V 的 TTL 电平，所以默认的接口的电平为 3.3V。如果系统是 5V。那么建议在串口的对接接口 TX 和 RX 串联一个 1.5K 左右的电阻。这样足以满足一般的要求，如果应用于强电磁干扰的场合，请参考“注意事项”的说明。模块在 5V 和 3.3V 的系统中均正常的测试过，一切正常。均在采用的是直连的方式，并没有串 1K 的电阻。

3.1 校验的说明

增加校验是为了保证串口通信的严谨性，因为很多场合并不能完全保证，主机发送的每一位都能被正确的接收到，这个时候就需要通过“校验码”来判断是否收到的是正确的数据。

我们常见的网络协议，都是有无数的校验在里面，每一个数据包均有 CRC 校验码，校验的复杂度相比较我们而言要复杂很多，但是在传输的过程中，还是会出错，所以，校验的重要性就不言而喻了。我们强烈推荐用户使用校验。

3.1.1 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。

相加的过程: $FF + 06 + 01 + 00 + 00 + 00 = 0x0106$

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

3.1.2 校验和不带校验的兼容说明

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：

组合播放[不带校验]	7E FF 09 21 00 05 01 02 03 04 EF	播放5、1、2、3、4
组合播放[带校验]	7E FF 09 21 00 05 01 02 03 04 FE C8 EF	播放5、1、2、3、4

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。

3.1.3 校验代码的移植

我们这里的说明，争对的是用户的 MCU 给我们的模块发送控制指令

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]这个 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，模块发送数据给用户的 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。

3.1.4 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。我们的模块串口误差是允许在 3% 以内的

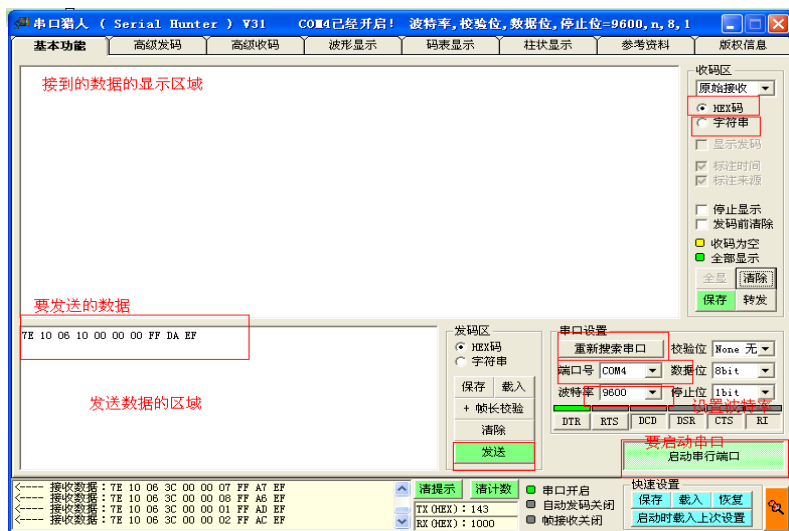
2、如果用户在 12M 的晶振时。首先要做如下判断，

(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 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

3.2 串口调试的说明



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”，我们的模块默认的波特率为 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

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。

(3)、软件配置 OK 之后，将需要的指令复制到发送区域即可。具体的指令请参照模块的数据手册

(4)、如果模块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，模块是不接受指令的,同时会返回相应的错误指令

3.3 模块调试的步骤

给予新手的小小建议[老手直接忽视]

对于一个新的产品，拿到手中进行测试，多多少少会有一些不适感，但是我们一定要有耐心，同时也要遵循合理的调试方法，这样不仅有助于我们更快的调试成功，同时也锻炼了我们产品开发过程中碰到问题的一种解决思路，这个才是最重要的。

我们给出的测试步骤基本如下，如果按照我们推荐的步骤，基本上都能完成调试的各种问题处理。

1、首先检查模块硬件是否正常。

模块拿到手中，我们不确定模块是否是好的，这个时候就需要简单的播放一下，来确定模块的好坏

(1)、购买了我们的模块测试底板，此时只需要插上 TF 卡或者 U 盘，按下测试底板的按键即可判定模块是否正常。正常的话，一定会有声音播放出来

(2)、如果没有购买我们的模块测试底板，可以自行按照我们的 AD 按键方式，引出按键来进行触发

播放。或者直接短接 ADKEY 和 GND 这两个管脚，再对模块进行上电，这样模块会循环播放 TF 卡或者 U 盘里面的所有文件，这个也是一个测试手段

(3)、如果购买的是我们的 MP3-FLASH-16P 系列模块，直接通过 Miniusb 线连接电脑，通过是否弹出下载窗口，来判断模块是否正常。

2、通过电脑的“串口调试助手[串口猎人]”对模块上电初始化的数据进行查看

(1)、我们的模块上电之后，如果一切正常，会返回初始化成功的数据。

(2)、根据模块的说明文档，可以很容易的看到，模块是否检测到 TF 在线或者 U 盘在线

3、短接串口板或者 USB 转串口板的正确性

(1)、如果模块工作一切正常，这时，我们还需要验证串口工具的是否正常

(2)、短接串口板或者 USB 转串口的 RX 和 TX 这两个脚，再通过串口调试软件发送数据，正常的话，就是发送什么数据，接收窗口就会显示接收到什么数据

(3)、如果这一步是正常的，就可以进行下一步操作了

4、发送一帧数据给模块，看是否播放声音

(1)、首先我们按照最简单的指令，就是发送“0x03 指令的指定曲目播放”功能，测试一下，如果前面的步骤都正确的话，这一步的效果就是一定能听到声音

3.4 串口的误差分析

1、首先明确串口的特点 UART

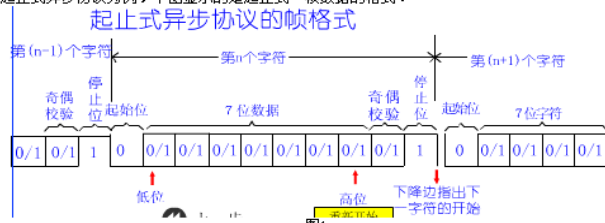
通用异步收发传输器（Universal Asynchronous Receiver/Transmitter），通常称作 UART，是一种异步收发传输器特点如下：

- (1)、传输的每一个字节，均有起始位和停止位，这样不会因为时钟的误差而导致每个字节的误差累积，也就是说传输 1 个字节的误差，和传输 100 个字节的误差基本一样
- (2)、理论上允许 4% 的误差，但是实际的应用应该保证误差在 3% 一下

2、异步通信的格式

1. 异步通信的特点及信息帧格式：

以起止式异步协议为例，下图显示的是起止式一帧数据的格式：



常见的串口通讯格式：

起始位 + 数据字节 + 校验位 + 结束位

3、我们的串口格式，我们采用的是硬件串口，所以误差会小于 1%。

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 \$0

起始位 + 数据 + 结束位

波特率: 9600bps 没有校验位

很显然，每 1S 能发送的最多字节数为 $9600 / 10[10 \text{ 位数据}] = 960 \text{ 字节/秒}$

误差的计算可以分两种，一种波特率的偏差[按照 9600bps]，另一种是每个字节的时间偏差

(1)、 $9600 \times 97\% \text{ ---- } 9600 (1+3\%)$ [也就是 9312 --- 9888]

(2)、接收一个字节需要的时间: $1000/960 = 1.04\text{ms}$

用户用 IO 模拟串口的话，就可以计算出自己的时间是否在 3% 的误差范围内

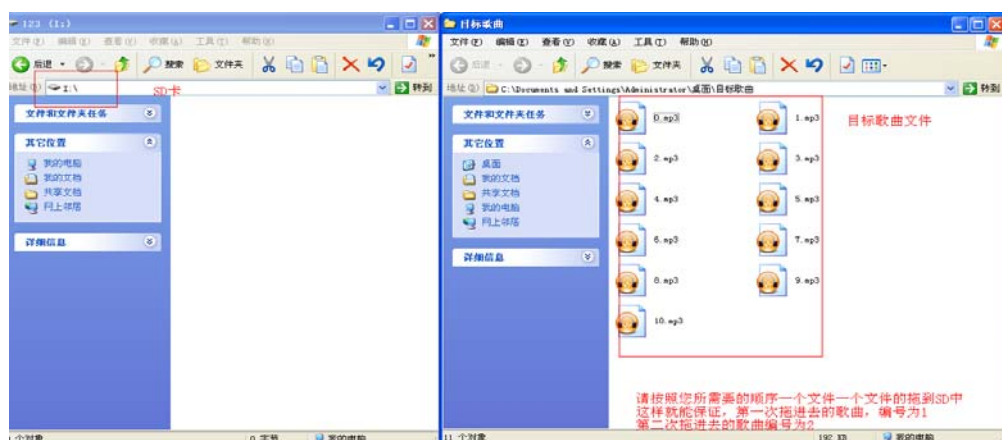
4. 播放的说明

4.1 指定曲目说明

- 指定曲目播放[按照物理顺序]
- 指定曲目播放[按照文件夹和文件名的名称]
- 指定曲目播放[按照文件名，必须在 MP3 文件夹下]

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

详情请参考模块数据手册的“3.4.4 指定文件夹播放”章节

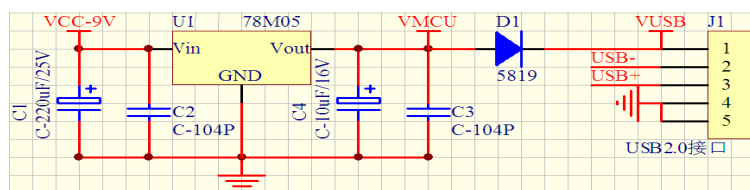
3、指定 MP3 文件夹里面的文件名播放

详情请参考模块数据手册的“3.4.5 指定 MP3 文件夹播放”章节

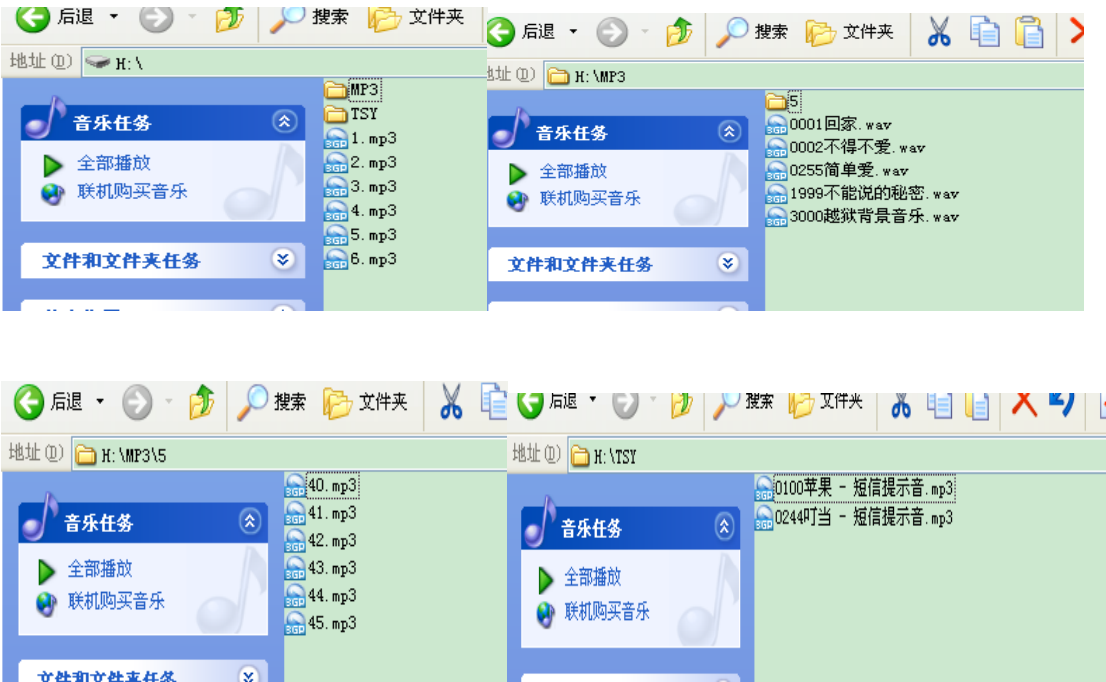
4.2 USB 更新 TF 卡的说明

1、我们的模块是支持读卡器功能和声卡功能，也就是通过 USB 接口可以在电脑上操作 TF 卡,更新文件等操作，十分的方便。前提是保证模块必须供电。直接插入 USB 线缆即可。

2、有些客户采用 7805 的输出端给 USB 供电，当 7805 的输入端无电源输入时，直接插入 USB 线时，由于 7805 的存在，会把电脑的 USB 端口电源拉低，导致不能正常工作，解决方案如下：



4.3 带有文件夹的设备，文件物理排序说明



- 1、首先格式化 TF 卡，依次放入 MP3 文件夹 --- TSY 文件夹 ---- [1.mp3~6.mp3]
 - 2、最后播放的顺序是，现播放根目录里面的[1.mp3~6.mp3]，再播放 MP3 文件夹里面的曲目，再播放“5 文件夹”的曲目，最后播放“TSY 文件夹”的曲目
- 文件夹里面的排序，也是按照拷贝先后顺序，进行编号的，如上图：

1. mp3	编号为1
6. mp3	编号为6
0001回家. mp3	编号为7
40. mp3	编号为12
0100苹果-短信提示音. mp3	编号为18

- 3、也就是说文件系统，会先扫描根目录，将根目录排好序，再排序文件夹里面的曲目
- 文件夹的先后，就是按照拷贝进去的先后顺序

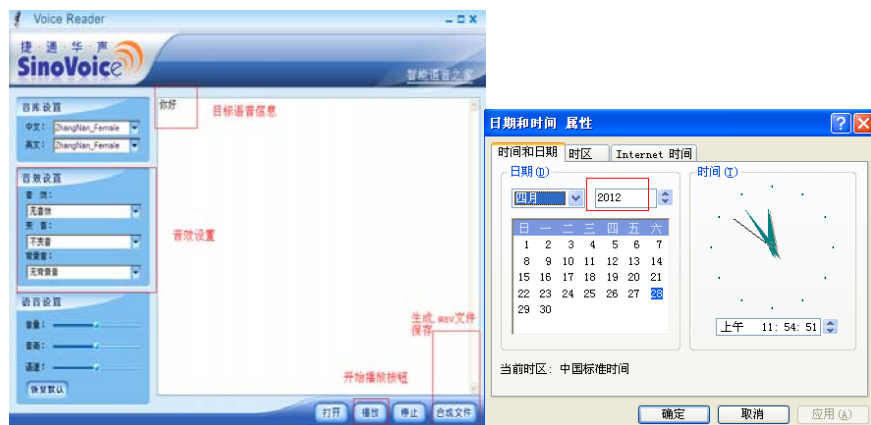
4.4 制作语音文件

- 1、原则上我们是不推荐客户自己制作音频文件的，因为我们的模块最大的特点就是播语音，并不能做到 TTS(文本转语音)。如果客户的语音文件比较小、并且变化频繁的话，我个人还是推荐使用现成 TTS 模块，虽然成本高，但是比较灵活
- 2、另外说一下 TTS，常用的 TTS 模块可供选择的太多，但是他们有二个缺点，就是语音播报比较生硬，音效不好、成本比较高。可能随着技术的发展，这些都会得到改善，但是目前还很难。现在的公交报站器，都是先用人录好音再播报的，所以听起来音效比较甜美。
- 3、我们自己制作语音文件，也是采用了 TTS 的软件来制作的，所以这样的应用环境，相对于 TTS 模块，我们能体现的就是成本优势了。下面看一下具体的制作方法

首先需要有一个软件，我们提供这款软件：



看一下软件的主界面：



软件里面需要输入的需要的语音信息“你好”，点击播放之后，就可以听到“你好”的语音。

如果有多个语音文件需要制作，就都按照这种方即可。

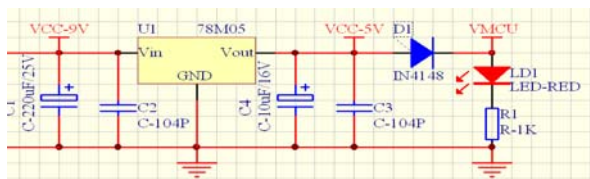
- 4、该软件不需要安装，直接解压就可以使用，但需要注意的是，需要查看软件的使用日期，因为我们现在使用的是官方提供的试用版本的软件，为了能够试用，如果过期就需要修改一下电脑的系统时间，见上图：

原本的时间是：20130428

5. 注意事项

5.1 模块供电说明

- 1、我们的模块，供电的范围是 3.3V--5V。一定不可以超过 5V，否则会造成模块的永久性损坏。
- 2、我们的模块，是音频类的产品，对电源的纹波是有要求的，建议用户最好使用线性电源[带变压器的电源]，或者使用 7805 之类的线性稳压芯片供电
- 3、参考电路如下



7805 后一级最好增加一个二极管，这里的 4148 其实并不是最合适的，因为 4148 的正向电流只有 500MA。如果我们后级的功放功率过大，会导致 4148 永久性损坏。这里选用 IN4001 或者 IN4007 才是最合适的。

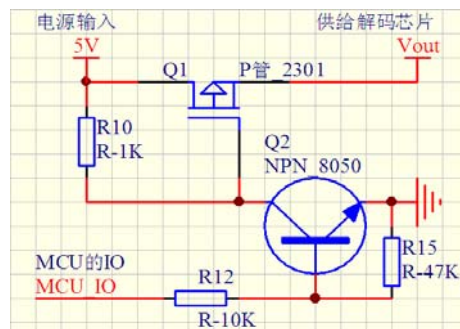
4、很多用户在应用过程中，往往很多供电是 12V 或者 9V，如果用户使用 7805 之类的线性稳压 IC 时，一定要注意芯片的发热，线性稳压 IC 的原理，基本上都是将多余的电压以热量的形式表现出去。举个例子，假如 12V 输入，经过 7805 之后，压差为 7V。假如后级的耗电为 200MA，那么 7805 产生的热量就是 1.4W，这个热量就很烫手了，会导致 7805 过热自保护，所以这样的硬件设计是很不合理的。

5、我们推荐的 8002 功放芯片，可以推动 3W 的喇叭，峰值电流可以达到 800MA。用户可以想象一下，如果是 12V 供电， $7 \times 0.8 = 5.6W$ 。这样的热量足够把手烫起泡了，7805 芯片最多工作半个小时后，就会自保护。

5.2 模块低功耗说明

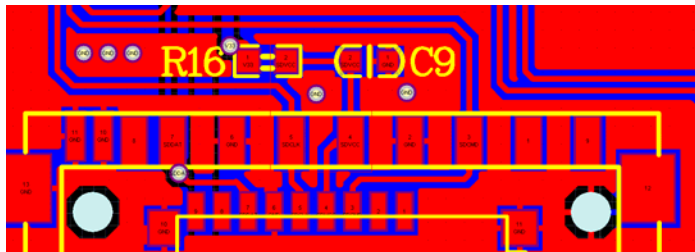
- 1、我们的模块，由于自身的低功耗功能设计得不是很合理，最小的待机电流都需要 1MA。
- 2、所以我们建议用户以控制模块供电的方式，来做到模块的低功耗，参考电路如下：

通过外部 MCU 的 IO 来控制模块的供电，这样可以做到不需要解码的时候，可以完全切断模块的供电，需要的时候再进行供电。



5.3 TF 卡的供电走线说明

1、用户在 TF 卡或者 SD 卡的供电走线方面一定要注意，如下图：



我们给出的参考原理图，TF 卡的供电，是 3.3V 经过 4R7，再经过一个 106 的瓷片电容，最后再给卡进行供电。

- (1)、4R7 的电阻是防止插入卡的时候，因为瞬间电流过大，导致芯片复位，如上图的 R16。
- (2)、106 的电容也是一定不能省的，原则上是越靠近卡座越好，如上图的 C9。
- (3)、注意走线，一定是先经过 4R7，再经过 106，最后到卡座的。

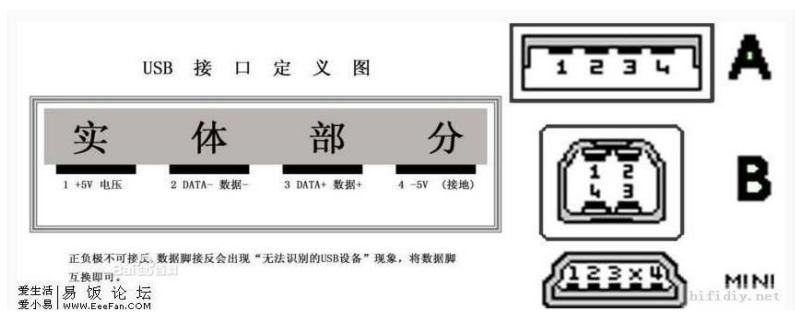
5.4 模块其它说明

- 1、模块支持 MP3、WAV 两种格式的歌曲播放，其它格式的歌曲暂时不支持。
- 2、只要 U 盘或者 SD 卡，里面有 MP3 或者 WAV 文件，就可以正常播放，不需要对 U 盘或者 SD 进行任何设置。客户有些是直接拿手机里面的 TF 卡做实验，建议要现格式化一下，否则会出来一些奇怪的现象。
- 3、如果需要接耳机的话，需要连接 3 根线，DAC 输出两根线，再加一根地，即可插入耳机。
- 4、如果需要接喇叭的话，MP3-TF-16P 直接按照模块的数据手册，接到指定管脚即可，不分极性。
5. 如果在锂电池供电的情况下，出现插 U 盘不播放的问题，应该是供电电压不足。
- 6、MP3-TF-16P 模块自带 3W 的功放，外接的喇叭负载不可超过 3W。否则会造成模块的功放部分永久损坏
- 7、如果直接外接 DAC 输出的话，请注意耳机的负载，默认是 16 欧姆的。负载过大，请不要长时间接入，否则造成模块的永久损坏。市面上的耳机默认是 16 欧姆或者 32 欧姆。包括手机或者 mp3 播放器配套的耳机，在这里均可直接使用。我们的模块是不是 PWM 输出，所以不可以直接驱动小喇叭，必须加功放。
- 8、很多系统中都是采用 12V 供电，后级采用 7805 进行稳压，再直接供给模块使用，这种方式是可行的，只要注意好 7805 的稳定输出，以及滤波，模块就可以正常稳定工作
- 9、如果用户使用低于 1G 的 TF 卡时，建议将卡格式化为 FAT 文件系统。因为市面上有些小容量的卡，对 FAT32 的文件系统支持不好。

5.4 常见的接口说明

- USB2.0 接口
- miniUSB 接口
- microUSB 接口
- 音频线和音频座
- TF 卡座

1、USB2.0 接口

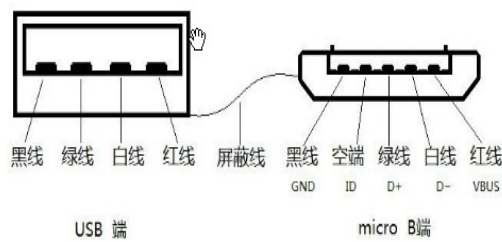


2、miniUSB 和 microUSB 接口



3、micro usb 接口

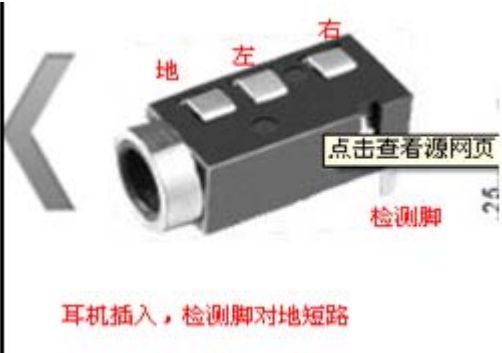
手机 micro USB 数据线 接线图



Pin	名称	线的颜色	描述
1	VBUS	Red (红)	电源正 5 V
2	D-	White (白)	数据线负
3	D+	Green (绿)	数据线正
4	ID	none (无)	分为A和B两种接口 A：与地线相连 B：不与地线相连
5	GND	Black (黑)	信号地线

对于用户使用 Microusb 或者 Miniusb 的时候，一定要弄清楚，这根 usb 线是否为充电线，因为有的线，里面其实只有两根线，就只有电源和地。那么用在我们的 MP3-FLASH-16P 等模块上就是不合适的。

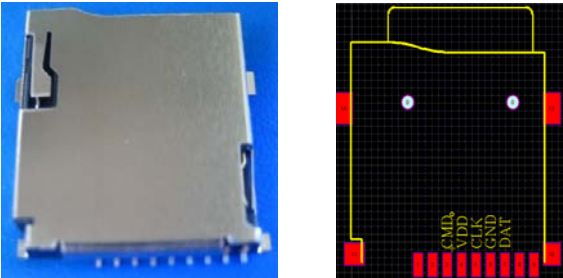
4、音频线和音频座子



5、TF 卡座

- 外焊自弹式
- 外焊非自弹式
- 内焊固定非自弹式

5.1、外焊自弹式



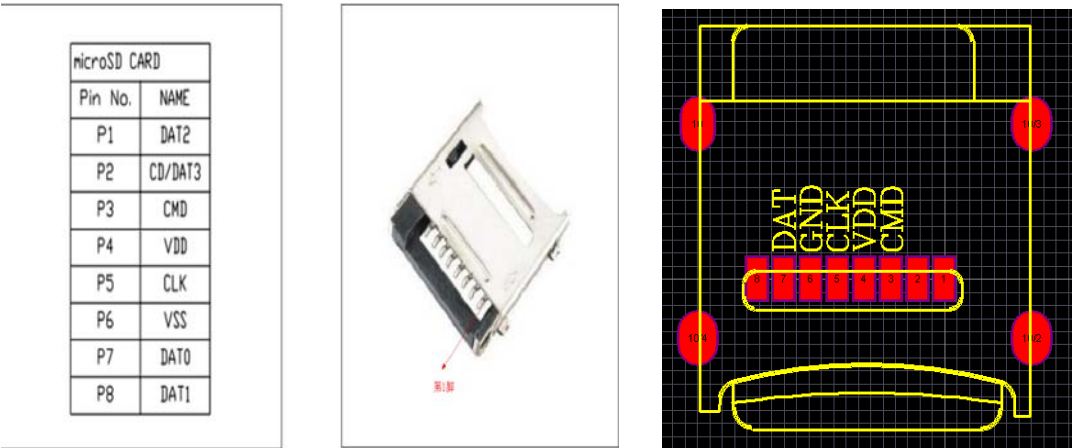
最左边的为第一脚。这种卡座最方便，这种卡带 GPIO 检测，也带 CLK 检测，还可以用 CMD 检测。使用最为广泛，且单价成本最低的卡槽

5.2、外焊自弹式



这种卡带 GPIO 检测，也带 CLK 检测，还可以用 CMD 检测。

5.3、内焊固定非自弹式



这种卡不带 GPIO 检测，也不带 CLK 检测，只能采用 CMD 检测。此卡座的特点是内焊式，可以完全的固定好 TF 卡。最好用于工业方面

5.5 常音频的杂音处理

音频的杂音主要在以下三点：

- 1、芯片 DAC 输出的杂音
- 2、功放芯片避免杂音
- 3、电源纹波产生的杂音

5.5.1 DAC 输出注意事项

音频输出比较麻烦的地方，就是出声音和关声音的时候，有一些杂音。好的电路和软件都是在想办法避免这些杂音。因为 DAC 的输出开启和关闭，再加上后级的电容充放电，出现杂音是非常正常的，我们只需要本着一个原则，就是尽量避开这些杂音，或者将这些杂音掩盖，让人耳听不到即可。这个杂音的具体表现就是“po”的一声

通常音频的输出分为两种接口，一种是 PWM 输出，一种是 DAC 输出

2. PWM 输出

这个基本上是低端的语音芯片常用的接口，他的好处就是可以直接推动 0.5W 的喇叭，而不需要外接功放
缺点就是音质差，因为这个 PWM 输出就像我们的 MCU 的 IO 口模拟输出，采样率是很低的

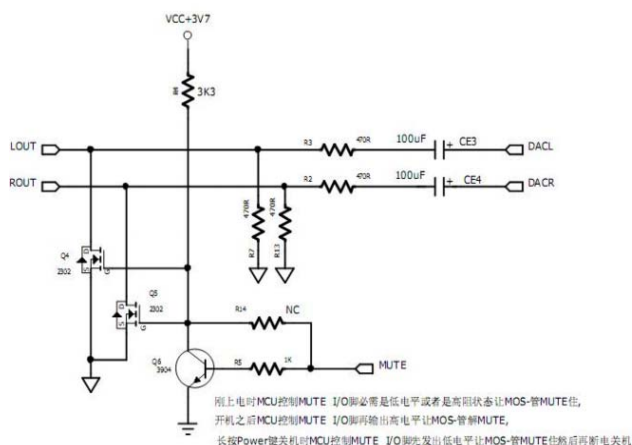
3. DAC 输出

这个是主流的音频输出，就像我们的手机耳机接口、平板等等，都是 DAC 输出，因为专业的音频解码芯片都是 DAC 输出，并且 DAC 的位数越高，音质也就越好，我们的 DAC 是 24 位，音质是有保障的
劣势，就是不能直接推动喇叭，而需要增加功放芯片，但是可以直接推动耳机

1、只要是 DAC 输出，均会有这样一个问题，就是芯片的 DAC 打开，会有一个瞬间的“po”音，这个是没办法避免的，但是可以有方法让人耳听不到这个声音。解决方如下：

4. 在 DAC 的输出增加静音电路

5. 增加一个功放芯片



(1)、上图采用的是增加一个三极管、2 个 mos 制作的静音电路，用 MCU 的一个 IO 口作为控制脚。上电的时候，

将 MUTE 脚拉低或者高阻，这样两个 MOS 就导通，这样 DAC 输出就没有声音，等到 DAC 初始化完毕之后，再将 MUTE 置高，这样 DAC 就打开了，就可以输出声音。这样做的目的就是为了让入耳听不到这个 po 音

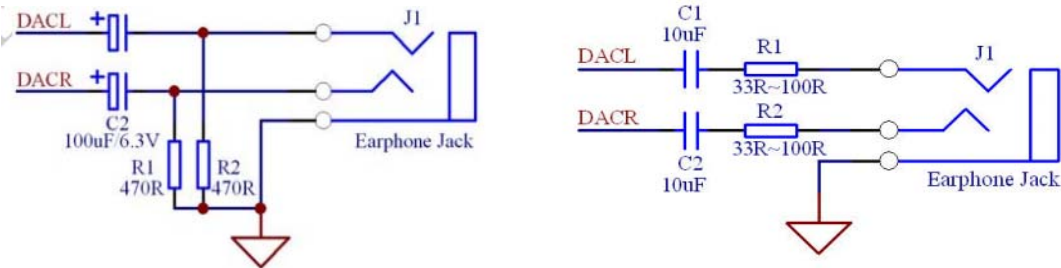
(2)、方案二就是增加一个音频功放，像 CS8002 等等，因为这些芯片都有一个静音脚。方法是，在芯片上电的时候，就让功放静音。等到 DAC 初始化完成，并且播放音乐的时候，再将功放打开，这样就避开了 DAC 开启的“po”音，或者说将这个 po 音复合到音乐中去了，人耳感觉不到

市面上的几乎所有音箱都是采用这两种方法中的一种

2、DAC 输出的两种参考电路，两种电路均是可行的，但是用电解电容的话，一些 po 音是比较难调试出一个好的效果，我们推荐第二种电路，也就是下图的有图

6. 用电解电容隔直

7. 用瓷片电容+电阻隔直



(1)、左边的图，DAC 输出经过电解电容隔直，电容 C1、C2 将决定低音的效果。电容越大，低音效果越重，而 R1、R2 是为了消除耳机插入的瞬间“啪啪声”

(2)、右边的图，DAC 输出隔直电容为 10uF，R1、R2 可加强低音的效果，也是我们推荐的电路。

5.5.2 功放参考消除杂音

功放的种类太多了，什么数字功放，A 类、B 类、D 类、AB 类等等，但实际上很多时候对音质要求不是超级高的场合，压根就不需要考虑这些，我们这里主推的还是 CS8002，简单、便宜。性价比非常之高，但是大功率的场合还需要选择其它的，譬如：TDA 系列。下面我们就说两点。

8. 电路详解

9. 注意点

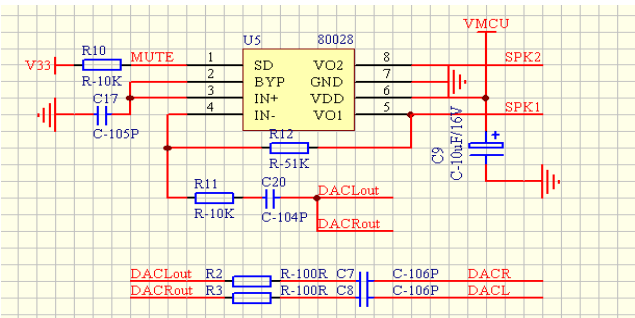
一、概述

TC8002D 是一颗带关断模式的音频功放 IC。在 5V 输入电压下工作时，负载（3Ω）上的平均功率为 3W，且失真度不超过 10%。而对于手提设备而言，当 VDD 作用于关断端时，TC8002D 将会进入关断模式，此时的功耗极低，IQ 仅为 0.6uA。

TC8002D 是专为大功率、高保真的应用场合所设计的音频功放 IC。所需外围元件少且在 2.0V~5.5V 的输入电压下即可工作。

二、功能特点

- 无需输出耦合电容或外部缓冲电路。
- 待机电流 0.6uA。
- 稳定的增益输出。
- 外部增益设置。
- 封装形式：SOP8、SOP8-PP、DIP8、MSOP8。



管脚排列图	序号	名称	类型	说明
	1	SHUTDOWN	I	关断端口
	2	BYPASS	I/O	电压基准端
	3	+IN	I	正向输入端
	4	-IN	I	反向输入端
	5	VO1	O	音量输出端 1
	6	VDD	POWER	电源端
	7	GND	POWER	接地端
	8	VO2	O	音量输出端 2

1、上图的电路中，关键的参数就是 C17 和 C20.这两个电容，建议用户按照电路给出的参数，否则会有一些 po 音或者断音，另外 R11 和 R12 是用来调整功放的放大倍数，计算方法就是 $A=2* (R12/R11)$ 。另外在空间允许的情况下，电源的 C9 可以采用大一些，譬如 100uF 的电解电容。R10 的存在是为了芯片上电的瞬间是静音的，这样可以减小杂音

2、用户在使用此功放的时候，要自己摸索一下，何时开启功放，何时关闭功放，这个也是为了避免一些“po”音

3、针对一些没有静音控制脚的功放芯片，有两种解决办法，一种是通过 mos 管控制芯片的供电，第二种就是通过控制信号源，不需要声音的时候，通过 mos 管直接拉到地，需要声音就断开 mos 管的控制。同样需要一个 MCU 的 GPIO 进行控制

5.5.3 电源纹波

测试音频电路的底噪最好的办法是在设备[TF卡或者U盘]中放一段“静音”文件，正常的情况下，我们听到的应该只有“功放的白噪声”，就不会有其它的声音了

[白噪声：指的是没有规律，声音很小，很空旷的那种声音]

(1)、如果电源不干净，出现的效果就是“丝丝丝丝”或者“吱吱吱吱”这样有固定频率，很规则的声音。这个就说明电源不干净，需要对电源进行相关的处理。这个地方是很考验电路的掌控水平的

(2)、因为在常规的应用中，我们可能都会用到 7805 等等这样的线性稳压 IC。电源当然会比较干净，效果也会比较好

(3)、如果在一些电流较大的情况下，线性稳压芯片就不适用了

这个时候就需要 DCDC。这里就会有一些噪声产品，也就是电源的纹波，这里需要用心处理好，网上可以参考的办法也有很多，我们也就不详细说明了

如果我们想测试功放是否失真，可以在设备中放一个 1KHZ 的正弦波文件进行播放。可以很明显的区分，是否失真

一般的功放应用都会选择失真的，如果不失真的话，功放的声音就比较小，可以在一定的范围内失真，这个是允许的。具体失真到什么程度，需要用户自己去测试和调整

5.6 客户问题集锦

1、上电电源指示灯不亮

==>注意查看供电接口是否正确。是否是接在 4.2V 供电端口了，如果电源接口正确，指示灯还是不亮，看一下工作指示灯是否闪烁，如果都不工作，说明要么电源有问题，要么模块有问题。--- 针对 MP3-TF-16P 模块

2、电源指示灯和工作指示灯均正常，但是没有声音

==>检查一下 DAC 输出线是否连接正确，再拔卡插卡试试。--- 针对 MP3-TF-16P 模块

3、通过串口发送的命令，模块没有反应

==>先根据手册里面提供的测试命令，测试一下串口软件的配置，确定这些配置 OK 之后，再自己指定一些命令，如果命令没有被执行，并且也没有反馈信息出来。就说明模块 RX 管脚没有连接好，或者模块损坏。

4、上电之后，没有收到模块发出的初始化数据

==>首先检查串口软件的设置是否正确，波特率是否为 9600，和电脑的接线是否牢固。再检查模块和电脑的串口连线是否交叉，必须是 RX--TX TX--RX 。

5、MP3-TF-16P 外接喇叭之后，声音偏小

==>我们的 MP3-TF-16P 和 MP3-FLASH-16P 模块都是上电默认音量最大的，也就是 30 级。MP3-TF-16P 最大支持 4 欧姆、3W 的喇叭，音量实际上是不小的。市面上的很多手持的小音箱的声音大小其实和我们的一样。唯一的区别就是，别人的喇叭有完善的模具进行环绕，而单独接一个空喇叭，声音自然会小，用户可以按照右图的方式握住喇叭，会发现声音会变大很多。所以为什么喇叭都需要一个音腔



6、完全按照我们的给出的 STC 的例程，下载进去之后，模块就是没反应

==>首先检查您的 MCU 的晶振是否为 11.0592.因为我们给出的例程，芯片是通过 11.0592 来设置波特率的再检查串口是否为交叉连接。还可以通过电脑的调试助手来查看模块的工作是否正常。

7、通过串口发送播放指令，模块可以工作，但是有很大的杂音

==>有的朋友使用的是 avr 单片机，将串口设置为“推挽输出”，此时就会出现播放有很大的杂音问题。建议用户将 MCU 的串口设置为弱驱。

8、我们使用你们的方案，产品时要出口至欧美的，MP3 专利费我们交了吗

- (1)、MP3 芯片专利已经过期了.尤其是在中国，另外基本上 80%的国家的 MP3 解码专利都已经到期了。除开小部分国家，如意大利等等，用户可以自行到专利网进行查询。
- (2)、如果产品出口至需要 MP3 专利的国家，请让客户自行寻找代理商，缴纳一定的费用。因为我们的芯片是不包含 MP3 的专利费的。
- (3)、我们的芯片已经热销至中东、欧洲等国家。目前为止还没有因为 MP3 专利问题产生矛盾，所以请用户朋友自行斟酌，

9、使用 MP3-TF-16P 模块时，发现 TF 卡里面的音源，前面的一截，大概 50MS，都没有播放出来

==>我们的 MP3-TF-16P 模块使用的功放，是受主控控制的，播放时，功放打开，暂停或者停止时，功放关闭。由于开始播放时，功放的打开是需要一定的时间，所以会出现音源的前面一点点播放不出来。解决办法有两种

(1)、可以把我们的 MP3-TF-16P 模块的功放芯片的第一脚，直接跳线到地。不建议

(2)、可以在需要播放的音源前端，添加一个 500MS 的静音，可以使用“Adobe Audition CS5.5.rar”或者“HA_GoldWave566_HZ.rar”软件进行处理。可以在我们的百度云分享地址直接下载这些工具软件。

。