

USB FM 收音机

前言

音乐，每个人都想听，你想一边写代码，一边欣赏好听的音乐吗？我最初的想法就是听好听的音乐而且不用交网费。这使我想到了 USB 的 FM 收音机，只要插上你的 USB 收音机，就可以实时的享受电波的味道。

一个多星期的折腾，终于成功了！而且用最低的成本，最简单的材料，制作成功了一个真正属于我的 FM 收音机。

这个 FM 收音机主芯片为 TEA5767，立体声高品质 FM 收音芯片，AD 采样 USB 数据传输均由 ATMEGA8 完成。器 USB 软核是公版的软核，不过已经经过我的改造。上位机界面是由 VC++ 编写，分别用到了 HID 的 API 及多媒体 API，还有一个很好用的上位机软件 LED 显示类。

传说 USB1.1 不支持等时传输(Isochronous)，但是实验发现是可以用等时传输的(参考 USB1.1 技术文档，是可以传输的)，这个发现让我惊喜不已，这样不是可以通过 AVR USB 来做语音传输了！

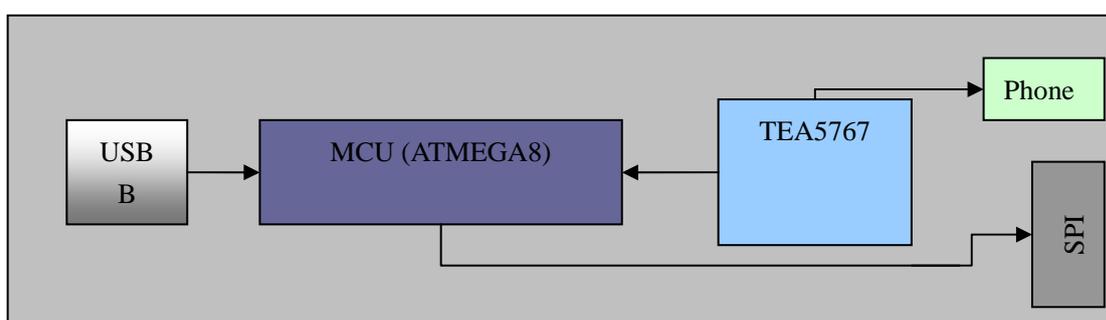
简单的计算可知，SUB1.1 传输速度是 1.5MHz/s，千万不要理解错了，数据传输速度的确是 1.5MHz/s，但是这其中有同步信号，握手信号，数据校验信号等，实际数据传输速率并没有那么多。严格理论计算，如果除去同步字段(SYNC)，令牌(Token)，数据不计算位填充，校验(CRC16)，一字节的延迟，那么可以达到 $(1.5M-9*8)Hz/s$ ，其速度

还是很可观的。实际上最大帧数据为 1023 个字节，可见数据量还是很大的。这样 USB1.1 做语音传输是完全可以的，而且余量还很大。如果是双声道，16 位，44.1KHz 采样，每帧平均数据量为 $44.1 \times (16/8) \times 2 = 176.4$ 个字节，完全在容许的范围内。所以可以用 USB1.1 传输。

实际考虑到 MCU(ATMEGA8)数据处理能力，暂时考虑单声道 (mono)，16 位，8KHz 采样率。限制其速度的一个重要原因是当 MCU 和 USB 主机传输数据时，MCU 不能采样数据，因为给 USB 主机传输数据时，一帧数据是连续的，中间不能被打断。所以我只考虑 8KHz 采样率。有一个备用方案，一枚 MCU 当 USB 数据传输用，另一枚 MCU 数据采集用，这样可以得到高品质的音频数据。

硬件

硬件大致结构如图一所示。其结构简单，可低成本制作。



图一 硬件大致结构

硬件制做是需要注意电平的匹配，USB 数据线是 3.6V 左右，而 USB 供电电压是 5.0V，要特别注意。

软件

实际上 USB FM 收音机主要还是在软件的编写上，不管是 USB 设

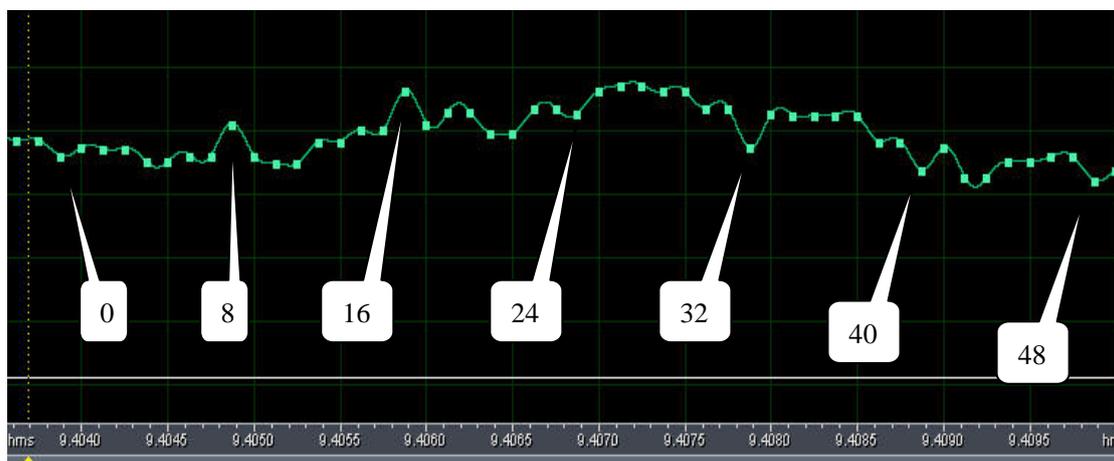
备，还是 USB 主机程序的编写。

软件调试

软件的调试很重要，对错误的排查很有帮助。

对于 MCU 端的是用串口向电脑发送数据，尤其对不支持 JTAG 调试的 MCU 来说很重要。AVR 串口调试在 USB 软核中可以调用，在 usbdrv 文件夹中有个头文件 oddebug.h，其中的宏 DEBUG_LEVEL 是定义调试级别的，同时也是调试开关，为 0 表示不启用调试，注意在主函数 (main) 中要对串口进行初始化 odDebugInit()，上位机串口软件推荐使用 SUDT AccessPort，当然也可以用其他的，这个软件有个好处是它存储空间较大，可以接收到较多的数据。

另外，对音频数据分析可以用 cool edit 2.1 软件，此软件可以分析音频数据流，并以图形化的形式呈现出来。调试试听过程中，发现声音总有种怪怪的金属音混在其中，听起来总感觉不舒服。刚开始怀疑是 AD 前端的滤波问题，加了截止频率很低的滤波器之后还是没有解决此问题，最后用 cool edit 软件进行音频数据分析才发现，原来



图二 音频数据分析

如图二所示，可以发现，图中所指示的离散的点 0, 8, 16, 24, 32, 40, 48 应该统统向右移动 8 个点，即现在 8 的位置数据应该是 0 位置的。最后发现有个数据装载时装错位置了。

VC++编程时 F5 调试就不用多说了。

USB FM 收音机设备部分

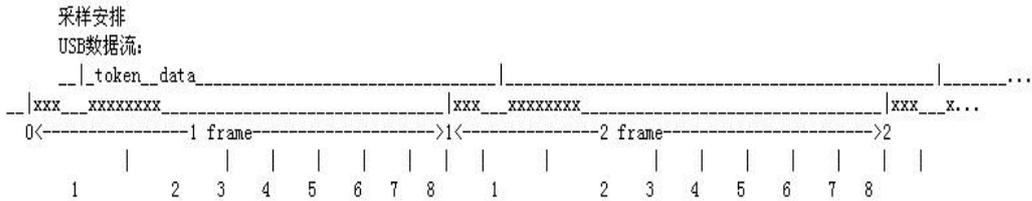
在编写 USB FM 收音机时还真走了不少弯路，比如枚举 AUDIO 时 IT(input terminal)与 OT(output terminal)中加入了 FU(feature unit)诸如静音，音量控制等后，发现在上位机找不到 USB Audio 输出设备了！可以双击音量控制，在选项->属性中可以找到 Audio 输出设备。

最后终于找到了，原来在描述符 **Class-specific AS General Interface Descriptor** 中有一个 bTerminalLink 即数据输出链接的端口号没改，在 OT 上，而是在原来的 ID 号上，即现在的 FU 上，所以主机链接不到数据。修改为 TO 的 ID 号上后，正常工作了。

另外在枚举 USB 复合设备(USB composite device)时，设备描述符的 USB_CFG_DEVICE_CLASS 与 USB_CFG_DEVICE_SUBCLASS 别忘了修改为 0，在 usbconfig.h 文件中。

另外如果 USB 枚举不成功，还会导致电脑蓝屏，如修改 Audio 数据采用位数，当时我把 16 位改为 8 位就导致蓝屏，Audio 数据接口数不符合实际接口时也有可能导致蓝屏。

数据采样分析如下



数据流如上图所示

sample_rate:8KHz=0.125ms=1500cyc

Frame_Cyc=12M/1K=12000, Token_Cyc=interrupt(1)+sync(8)+pid(8)+addr(7)+endport(4)+crc5(5)+ep0(2)=35*8=280

Data_Cyc=interrupt(1)+sync(8)+pid(8)+data(16*8)+crc16(16)+ep0(2)=163*8=1304<1500

所以在中断中的时间最长为数据阶段1304cyc, 即1304/12M=0.11ms<sample_rate所以想到一个方案:

在USB中断中当收到USB的IN中断后, 当采样完token后随即开启ADC转换定时器(同时读转换值), 这样当USB设备接着在中断发送数据(data)后ADC转换已经结束, 但是8KHz的中断还没有发生, 所以这样采样是有效的。满足8KHz采样的要求。为防止数据发送冲突采样用乒乓操作。

HID 的 Report 描述符有软件 Dt. exe 可以在 SUB 网站上取得。注意如果要得到 HID 的 Report 描述符需另存为.h 的文件。

主机的 Set Report 请求的数据阶段在 USB 软核中是被屏蔽了的, 自动回 ACK, 而在 FM 收音机中, 我需要这些数据来控制 FM 收音机。所以我修改了其软核, 在 usbdrv.c 中的函数 static void usbProcessRx(uchar *data, uchar len) 中

```

加入了下面
    #if USB_CFG_SET_DATA
        replyLen = usbFunctionSet(data, len); //接收数据
    #endif
的一段程序

```

这样就可以完成 Report 数据的接收。

采样加入的一段程序为

```

:luojunzhou add
    ldi    cnt, 0           :{1}
    sts    SampCnt, cnt    :{2}, SampCnt=0
    ldi    cnt, 0x24       :{1}
    out    TCNT1L, cnt     :{1}, Set count
    ldi    cnt, 0xFA       :{1}
    out    TCNT1H, cnt     :{1}, Set count
    ldi    cnt, ((1<<ADEN) | (1<<ADSC) | 0x04)
    out    ADCSRA, cnt     :{1} Start
    in     cnt, ADCL       :{1}
    sts    UsbIntSampl, cnt :{2}, UsbIntSampl=ADCL
    in     cnt, ADCH       :{1}
    sts    UsbIntSamph, cnt :{2}, UsbIntSamph=ADCH

```

分别为

程序改编, 主要是data in (endport1) token之后, 数据处理如下

- 1, SampCnt = 0://数组采样指针为1, 这里没有用指针, 是因为指针加在中断中, 有可能溢出。
- 2, 采样数据值UsbIntSamph, UsbIntSampl
- 3, ADC-->ADCSRA |= (1<<ADSC); //开始新的转换
- 4, Timer-->TCNT0 = 0x45; //set count

USB FM 收音机主机部分

主机程序主要分为三个部分

USB 音频部分，主要是用多媒体(Windows Multimedia)，其主要调用的库为 winmm.lib，头文件为 mmsystem.h，

HID 类用 DDK，库为 hid.lib，
setupapi.lib，头文件为右图所示

```
//DDK headers for HID and SetupAPI  
extern "C" {  
#include "hidsdi.h"  
}  
#include "setupapi.h"
```

注意：用 HID 类 CreateFile

时有时会返回-1，但是你的的确确在设备管理器中看到了你的 HID 类的枚举 HID 人体输入设备的图标，这说明你的驱动没更新，这时你可以卸载该设备，然后重新连接该设备，重新枚举该设备，重新装载驱动就可以了。

软件界面用了一个 CStaticCounter 类，这个类可以产生一个 LED 计数器，这正好适合 FM 收音机的频率选择界面，这个类有个很好的地方是可以通过鼠标的拖动，点击来操作，非常方便。