声音采集器实验报告[[1]](#footnote-1)

1. 实验要求

要求设计一个声音采集系统，通过拾音器采集音频信号，经过放大、加直流偏置、滤波后由单片机采样进行模数转换（AD）得到数字信号，通过串口发送给电脑进行播放。

1. 实验原理[[2]](#footnote-2)
2. 拾音器（咪头）

咪头是将声音信号转换为电信号的设备，这里选用电容式驻极体咪头。



图1.咪头实物图

驻极体话筒具有体积小、结构简单、电声性能好、价格低的特点，广泛用于盒式录音机、无线话筒及声控等电路中。属于最常用的电容话筒。由于输入和输出阻抗很高，所以要在这种话筒外壳内设置一个场效应管作为阻抗转换器，为此驻极体电容式话筒在工作时需要直流工作电压。

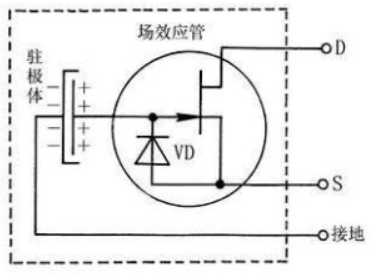


图2.咪头内部电气原理图

驻极体话筒的极化膜上在制作时被注入一定的永久电荷，当咪头检测到音频信号时，极化膜随着声音振动，两极板间距离随之变化，因为电荷量固定不变，两极板之间的电压差就随声音强弱而变化。由于这个电压变化十分微弱，需要用一个输入阻抗非常大的场效应管进行放大。

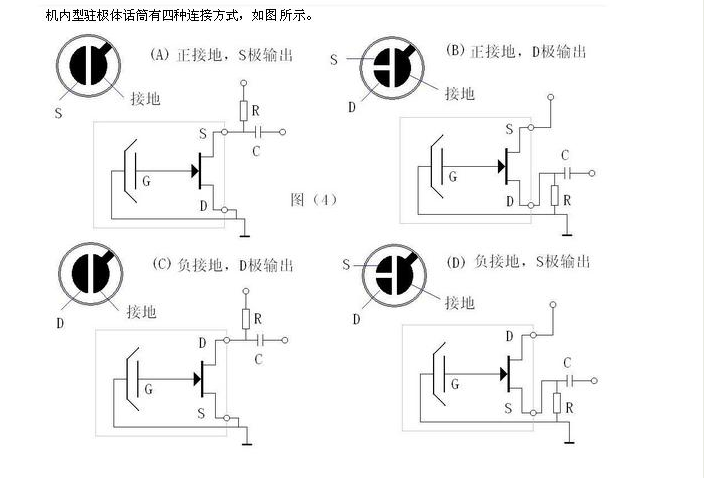


图3.咪头参考连接电路图

应用中，有图2所示四种连接电路，由于所购买的咪头中场效应管源极已经连接到外壳，所以只能选用（C）图，将咪头负极（连接到外壳的管脚）接地。另一极通过1KΩ电阻接5V电平，并通过1uF电容隔除直流成分。

1. 小信号放大

咪头输出电信号较小，约数十毫伏，且经过电容隔直后，信号既有正电压也有负电压，为了满足AD采样的要求，需要将小信号放大并增加直流偏置使电压范围在0~3.3V内。

本系统使用LM358双运放（5V单电源供电）对信号进行两级放大，每级获得约10倍的电压增益，同时在运放同相端用分压电阻提供直流偏置。为了调试方便，将第一级一个影响增益的电阻和一个影响直流偏置大小的电阻用电位计替代。

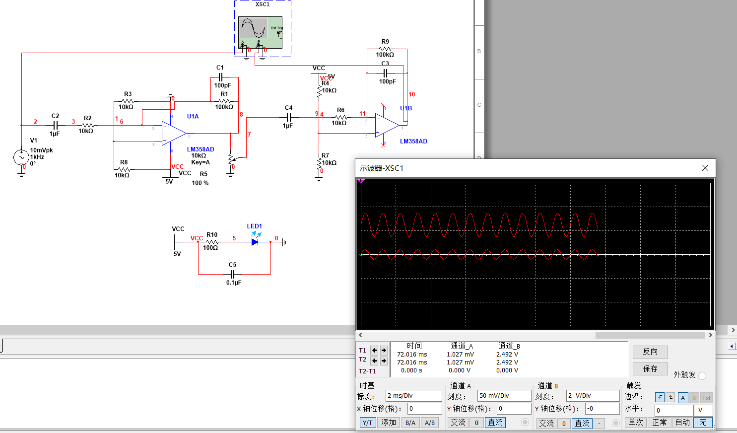


图4.小信号放大的Multisim仿真

1. 滤波

语音信号频域范围在300Hz到3400Hz之间，考虑到保护频带，在电话技术中为单个语音频率传输信道分配的带宽通常为4kHz。因此需要设计一个低通滤波器，其上截止频率在4KHz左右。

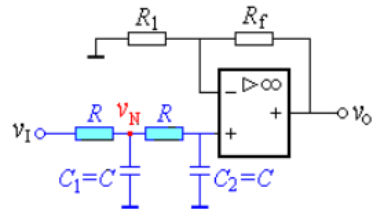


图5.简单二阶有源低通滤波器

图5中滤波器实际上就是把一个二阶无源RC滤波器连接到集成运放，由于运放输入端“虚断”的性质，后级负载对RC滤波器的影响可以忽略，同时反相端的两个电阻可以提供一定的增益。

通过计算，当同相端用相同的电阻和相同的电容时，该滤波器的通带增益为

上截止频率为

考虑到电阻电容的经典取值，同相端选用1.5KΩ电阻和10nF电容，反相端选取两个1.5KΩ电阻。理论上，上截止频率为3.926KHz，增益为2。

1. 模数转换

本系统中需要将模拟声音波形转换成数字信号，便于单片机传输。

由于语音信号频带为4KHz，根据奈圭斯特采样定理，至少要8KHz的采样频率才能保证从样本中无失真恢复出原信号，否则信号在频域会有混叠。

从模拟信号到数字信号除了采样之外还需要量化，为了数据传输方便，本系统采用8bit量化精度，这样每一个字节就可以表示一个样本点，用串口传输时无需做样本之间的同步。

1. 单片机

本系统选用stm32f103c8t6作为系统的主控芯片。相比于51系列单片机，stm32具有更出众的性能，对比某些51单片机，甚至在成本上也更具优势。

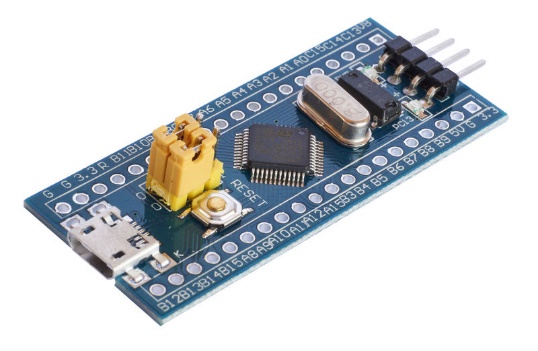


图6.stm32单片机（Blue Pill）

这款单片机最小系统板上集成了晶振和复位电路，通过5V管脚供电，引出3个USART串口，内置4个16位定时器，2个12位ADC。需要通过编程，开启一个定时器，每隔125us触发一次定时器中断，在中断服务函数中进行采样和AD转换，然后把采集到的数字信号通过串口发送给电脑。

1. 串口电平转换

由于单片机串口使用的是TTL电平，电脑上串口使用的是USB电平（有的电脑上还保留232串口），中间需要一个电平转换芯片，本系统选用CH340C，这款芯片可以将USART中的TX，RX转接到USB中的D+，D-从而实现电平转换，且相比于CH340G，这款芯片内置振荡电路，无需外接晶振和匹配电容。

1. 上位机

数据通过串口发送到PC机后，PC端需要有程序读出串口数据并合成音频文件。Python作为一种开源的编程语言，拥有大量的第三方库可以调用，其中包括实现底层串口通信的PySerial。

同时为了方便配置串口通信的相关参数，并实时显示音频波形，需要开发一个GUI程序，PyQt作为一个自由软件，可以免费的用于自由软件的开发，且可以开发跨平台的应用。另外，PyQt支持通过QtDesigner以拖拽控件的方式开发UI界面，较为方便高效。

1. 实验步骤
2. 设计原理图[[3]](#footnote-3)
3. 供电
4. 方案一

通过12V可充电锂电池供电，板载DC-DC（LM2596S）将电压降低至5V，给LM358和stm32最小系统板供电。

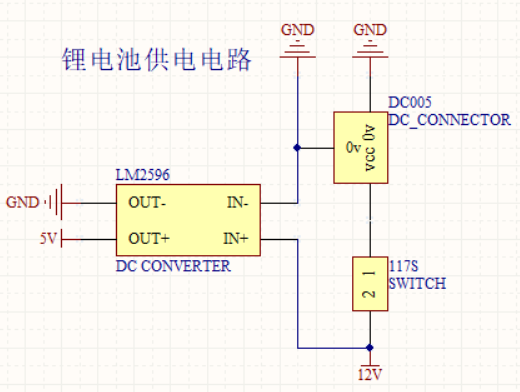


图7.锂电池供电电路

1. 方案二

通过USB-MicroUSB数据线从电脑获得供电，由于本系统所有元件所需电压都不超过5V，所以USB能够提供所需电压。

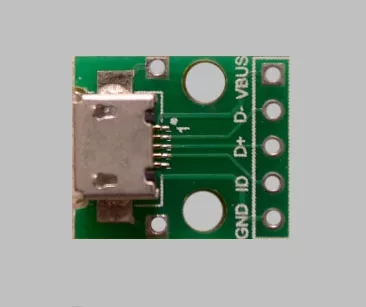
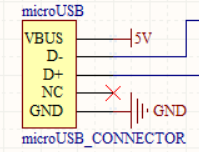


图8.USB供电

显然方案二的USB供电方案更加方便简单，成本也更加低廉，但最初考虑到电脑所能提供的驱动电流较小，担心USB供电无法驱动整个电路，所以同时使用了两套方案。

1. 信号调理
2. 声音信号转电信号

根据实验原理部分，咪头负极（外壳）接地，正极通过1KΩ电阻接5V电压，并经过1uF隔直电容后输出微弱电信号。

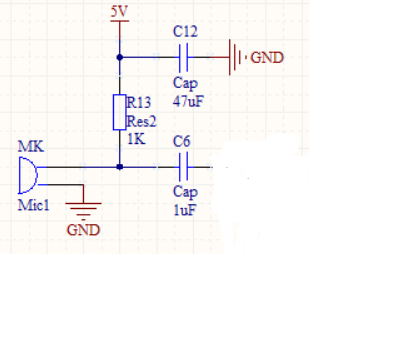


图9. 咪头工作电路

1. 小信号放大

集成运放通过负反馈很容易实现信号放大，本系统中通过两级放大，且第二级反馈电阻用电位计替代，使得增益可调。同时，因为咪头输出信号经过隔直后有正有负，而单片机采集范围为0~3.3V，所以需要添加直流偏置，偏置电压大小也通过电位计调节。（为了使偏置电压不至于过大烧坏单片机，这里用了一个电位计和一个定值电阻而不是只接一个电位计）

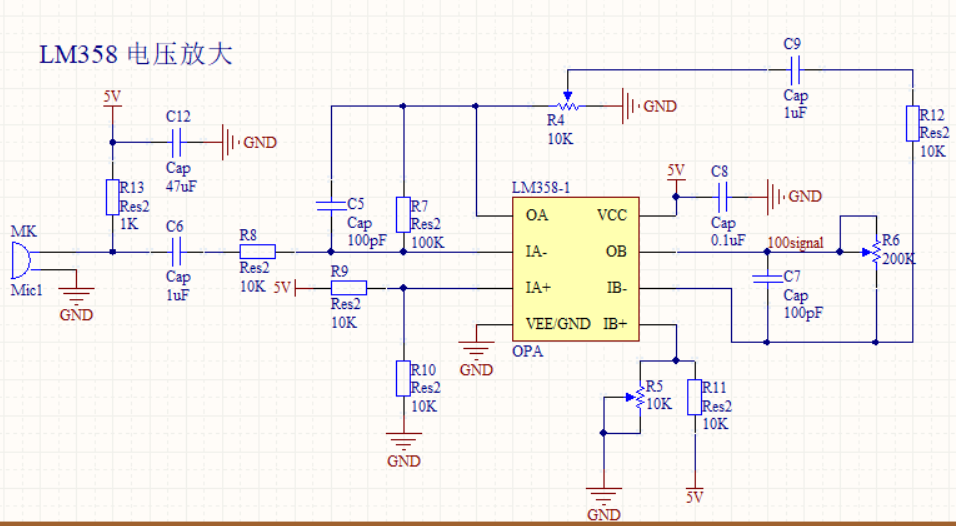


图10.小信号放大

1. 滤波

由于后面需要对模拟信号进行采样，采样频率为8KHz，为了防止频域混叠，采样率要大于信号最高频率两倍。而且，一般语音信号频率范围在300~3400KHz之间，滤除更高频率分量也有助于提高信噪比。

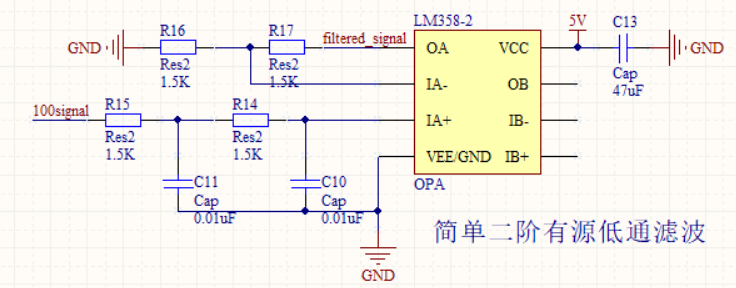


图11.滤波电路

1. 限幅

由于单片机AD采样电压范围有限，为0~3.3V，电压过高可能会将单片机烧坏，因此在采样管脚接两个肖特基二极管进行限压。实际中使用BAT54S，内置两个肖特基二极管，单个导通电压约为0.2V。并在AD采样管脚串联1KΩ电阻，增加输入阻抗。

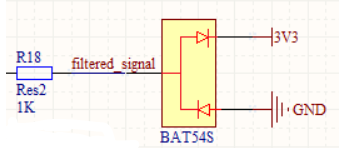


图12.限幅电路

1. 串口电平转换

若要通过USB数据线实现单片机和上位机之间的通信，需要将单片机串口的TTL电平转换为电脑的USB电平。本系统通过CH340C电平转换芯片实现。

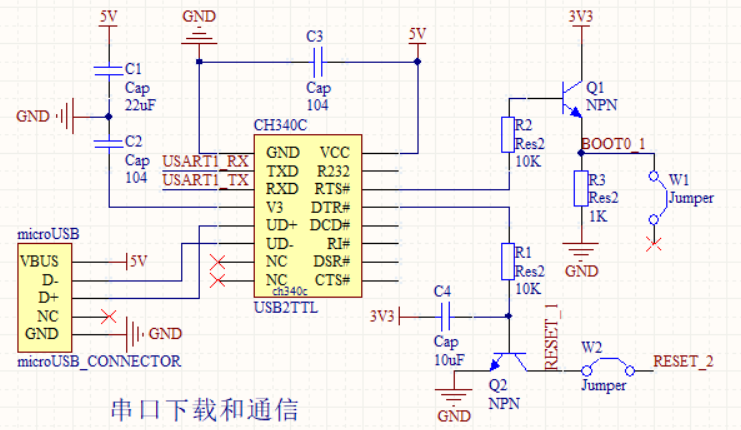


图13.电平转换

1. 仿真[[4]](#footnote-4)

为了保证信号调理电路原理上没有问题，先在Multisim中进行软件仿真，使用1KHz，20mVpp的信号模拟咪头输出电压，得到如下仿真结果。

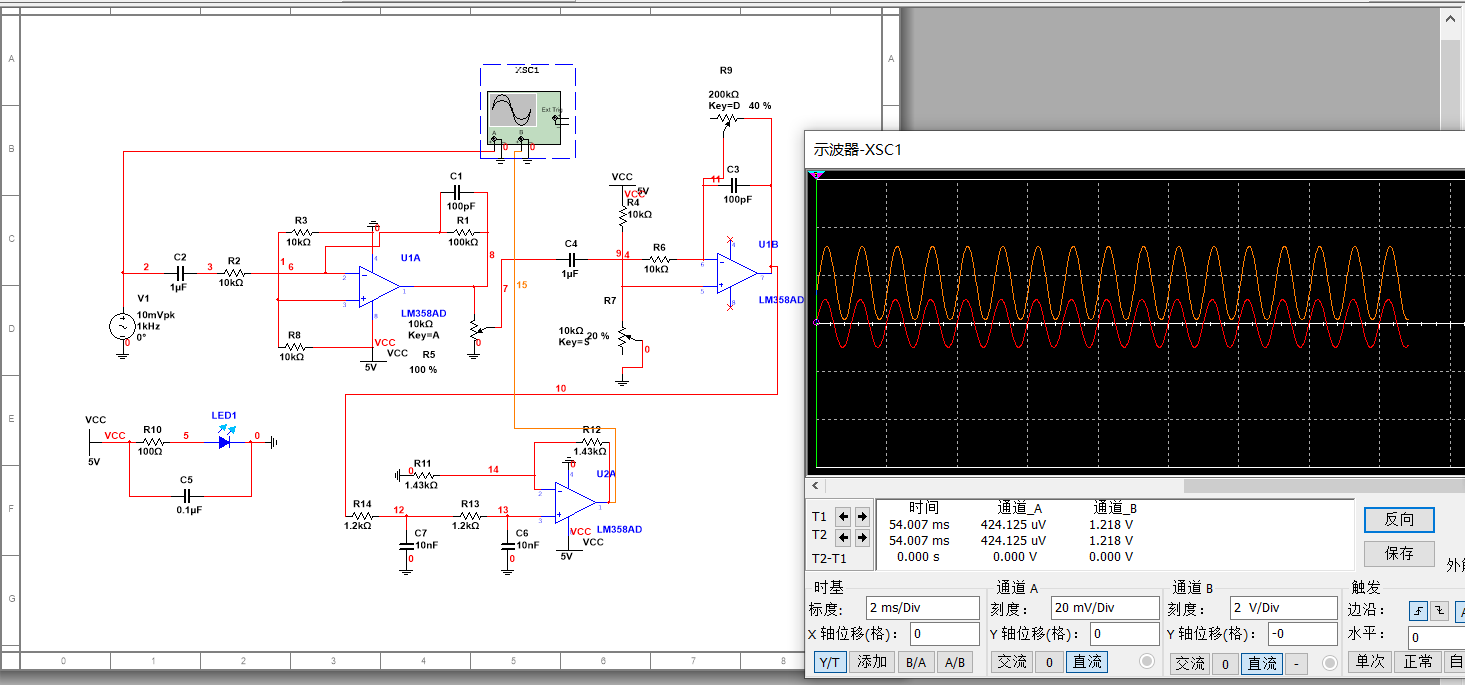


图14.Multisim仿真

1. 设计并焊接PCB板

综合考虑成本，实现复杂度等因素，设计10cm\*10cm双面PCB板。除了从原理图导出元件封装和网表连接之外，另加几个焊盘，分别连接到GND，第一级和第二级放大电路输出端，滤波输出端，AD采样管脚上，以便后续调试。

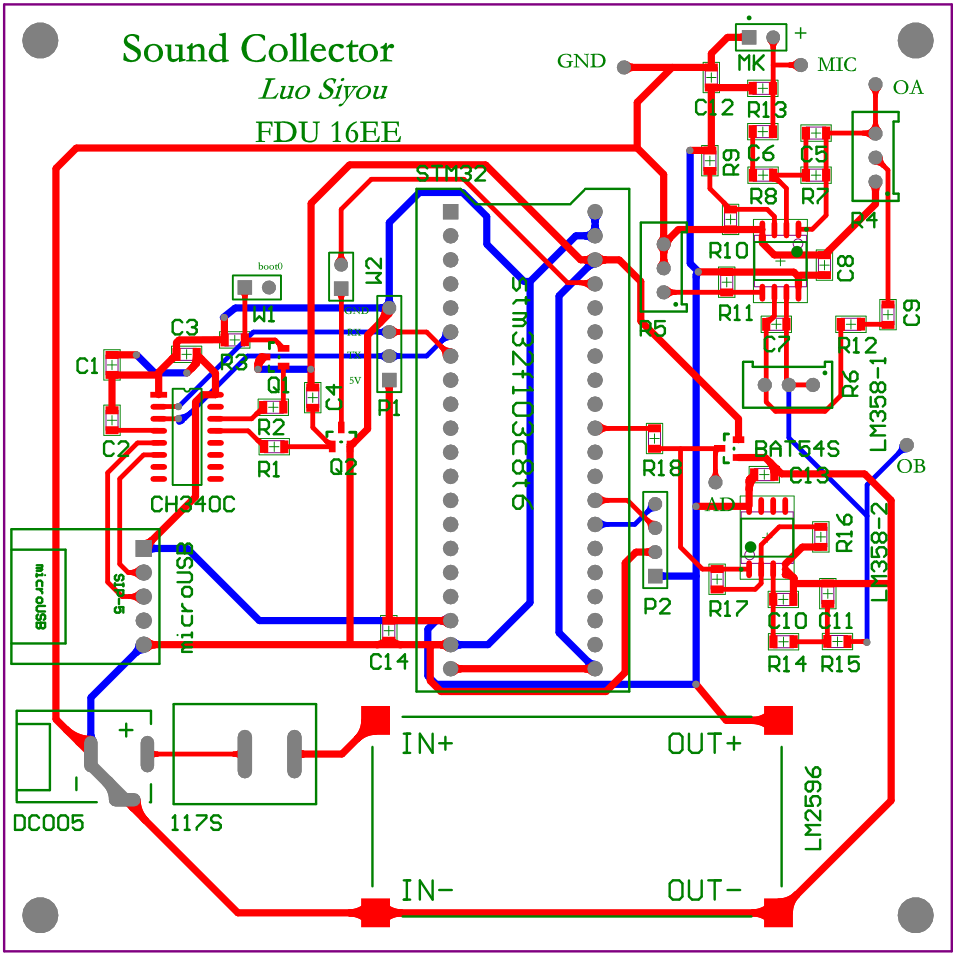


图15.PCB板图

生产出的PCB板以及焊接后的成品如下图。

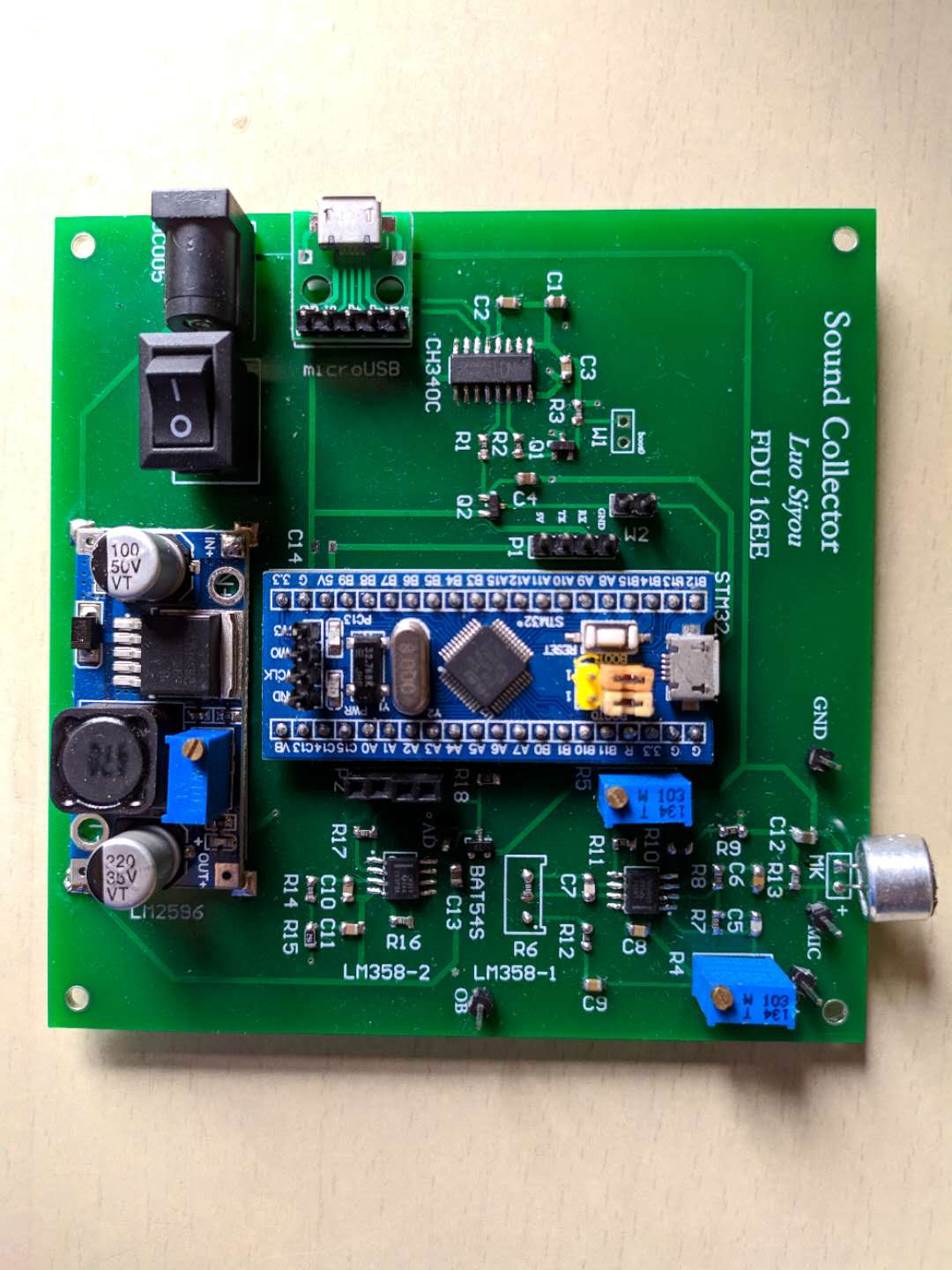
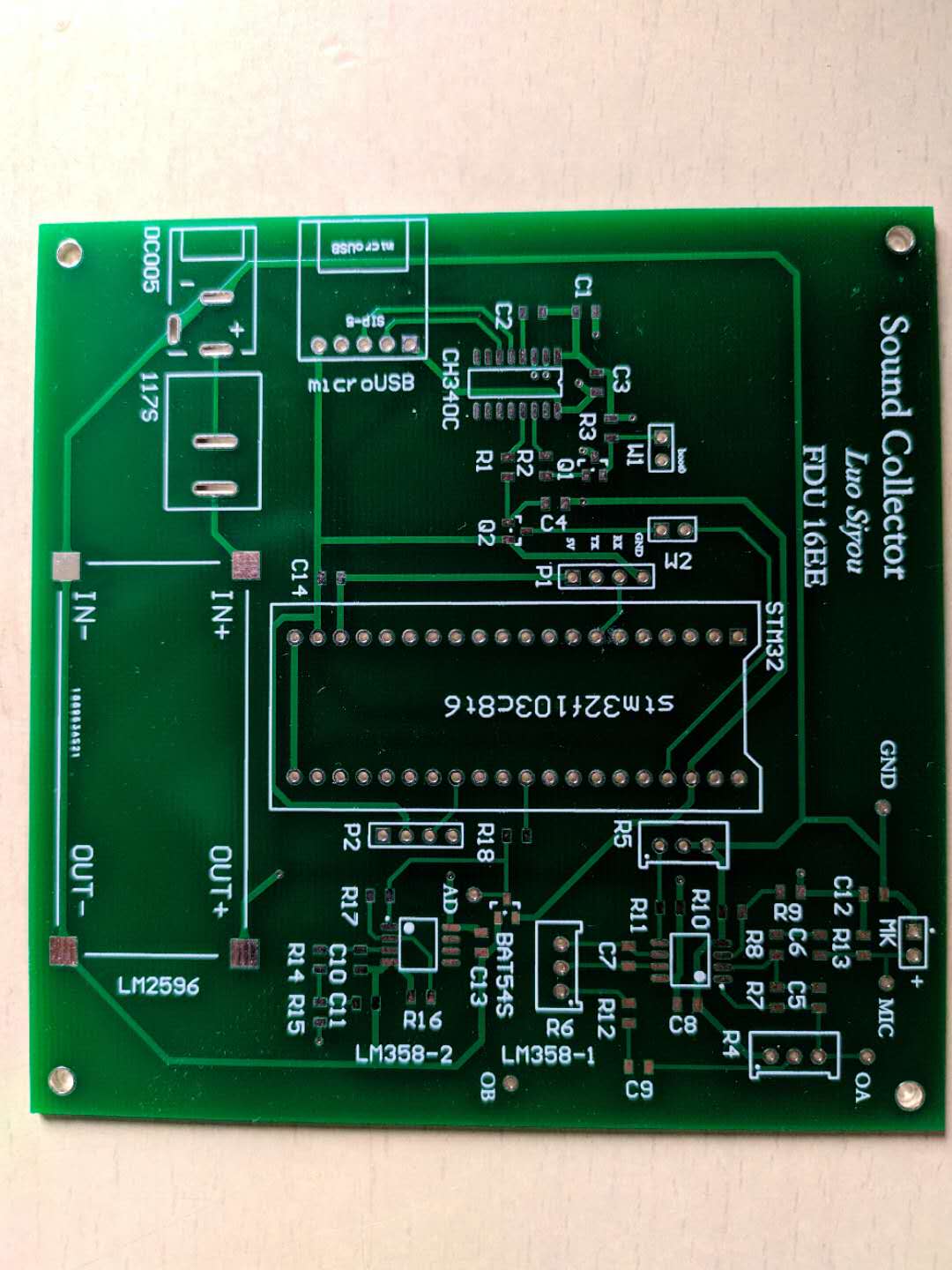


图16.PCB实物图

1. 单片机编程[[5]](#footnote-5)

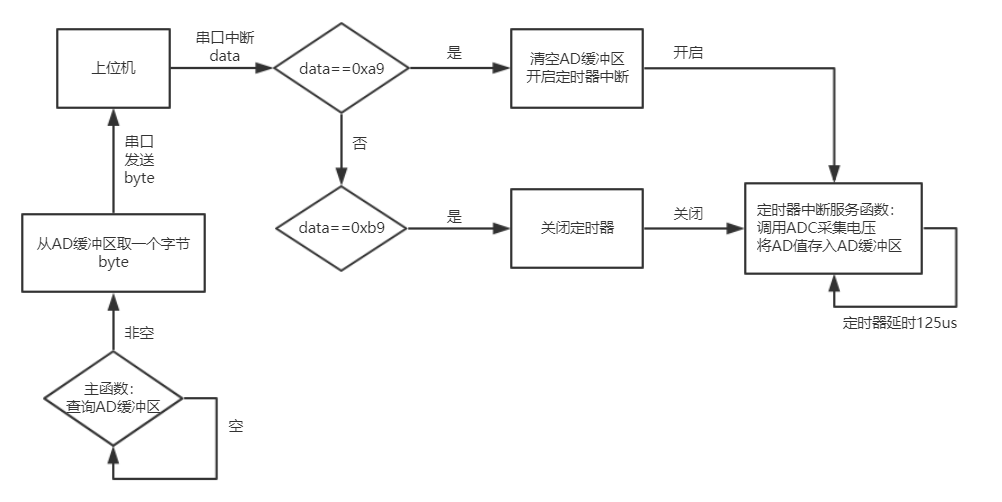


图17.单片机程序框图

为保证采样频率固定为8KHz，配置一个定时器TIMER4，其计数周期为125us，每隔125us触发一次定时器中断，在中断函数中进行AD采样。

为了避免在不需要采样的时候浪费计算资源，单片机可以接收串口指令，当接收到开始采样指令时才打开定时器，在中断函数中进行采样，当接收到停止采样指令时，就关闭定时器。

根据中断服务函数中不要执行太多程序的原则，构建一个AD缓冲区，其实就是一个环形队列，在定时器中断中只负责将AD采样的值放入AD缓冲区，然后在主函数中负责查询缓冲区，并将缓冲区内的数据通过串口发送个上位机。

1. 上位机开发[[6]](#footnote-6)
2. UI界面

在QtDesigner中通过拖拽控件的形式构建上位机UI界面，主要包括串口参数配置框，串口指令发送按钮，接收数据显示（可配置为数值显示或实时波形绘制）。

绘制好UI界面后存为demo.ui文件，然后还需要通过pyuic工具将UI界面转换为python代码。具体指令为：

* pyuic4 –o Ui\_demo.py demo.ui

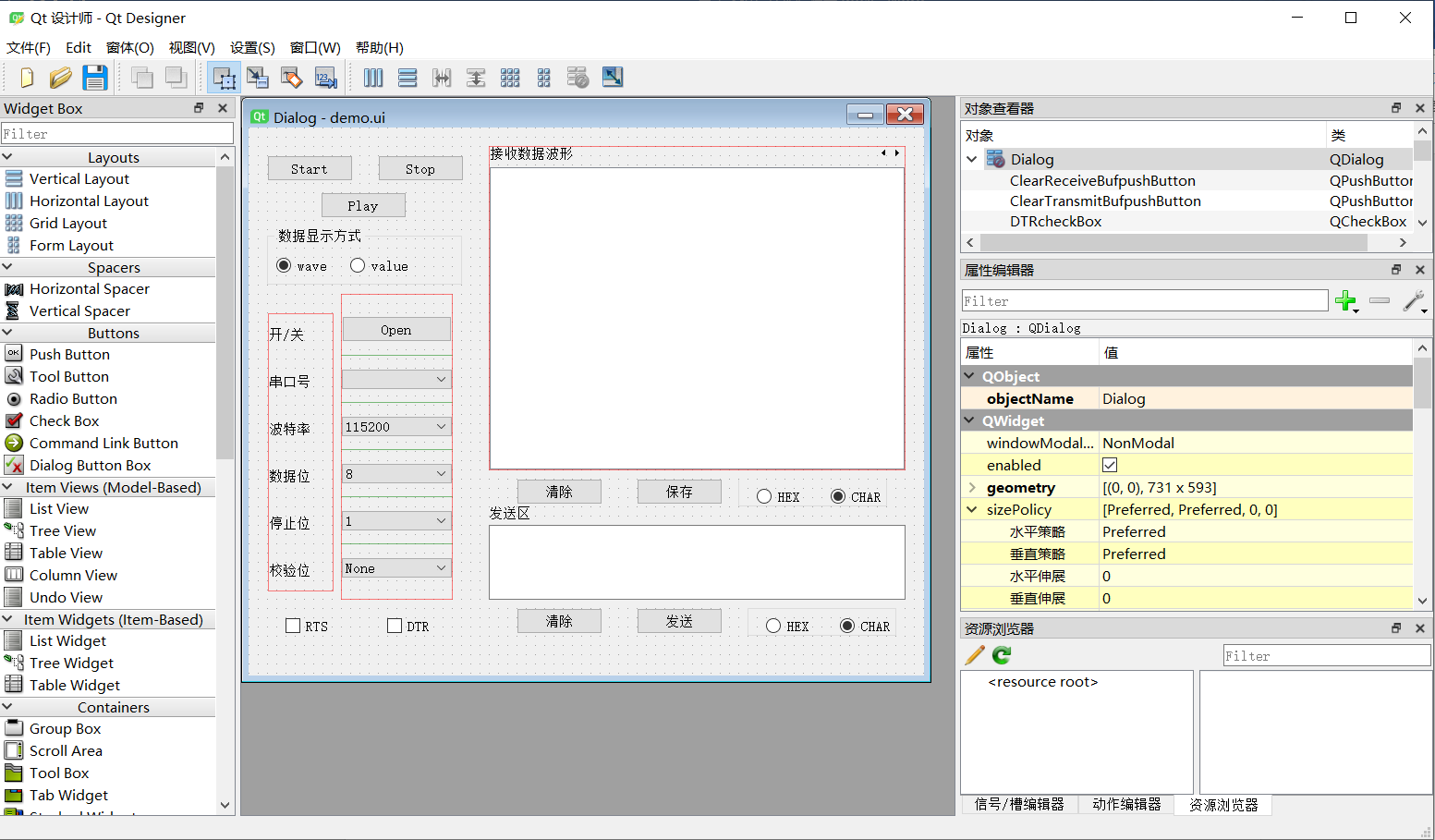


图18.UI界面开发

1. 功能实现

从UI界面自动转换成的python代码只包含一个窗体类，需要将其实例化，然后将各个控件所触发的信号连接到槽函数，在槽函数中操作底层通信。

底层通信需要控制串口，这就需要调用pyserial库，并配置好串口通信的波特率等参数。

为了实时观察AD数据，在上位机内部嵌入一个绘图窗口，将串口数据以波形图的形式显示。

完成波形采集后，需要将串口数据转转换成音频文件，这里调用python自带的wave库，合成.wav文件，点击上位机中的Play按钮后即可播放。

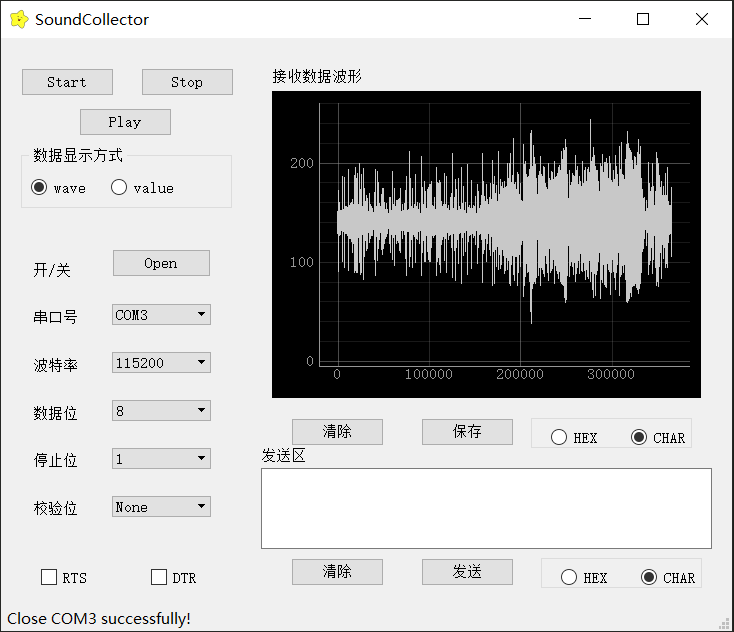


图19.上位机

1. 实验结果[[7]](#footnote-7)

最后的成果展示包括三个部分，PCB电路，上位机软件，音频播放。演示时，用手机播放音乐，将咪头靠近手机扬声器采集音频。经过调试，合成的音频在播放时会有部分杂音，但仍可听清歌词及音调。（双击图标查看音频效果）



为了更具体地描述音频质量，在较为安静的环境中进行对比试验，第一次不播放任何音频，采集一段环境噪音[[8]](#footnote-8)；第二次播放音乐，采集一段带噪信号[[9]](#footnote-9)。分别计算其功率，并求得信噪比[[10]](#footnote-10)约为11.23 dB.

1. 实验分析

经过本次实验，完成了从原理到验证的整个过程，总结出以下经验：

1. 如果不确定电路的实际参数，在设计电路板时就要考虑到调试的方便，比如用电位计作为可调电阻来调节信号增益和直流偏置大小。如果输出结果和预想的不一样，还要逐级检查信号，甚至于在电路焊接时就因该每焊接好一级电路就验证该级输出是否正确，否则到了最后出错时可能难以排查。为了方便检查每级信号，需要将各级信号引出一个排针，方便示波器测试。另外，在检查调试前一级电路时不希望对后一级产生意料之外的影响，可以在两级电路之间串联一个0Ω电阻，在调试的时候可以将其拆下断开。
2. 单电源供电

在很多电路设计中，LM358使用的是正负双电源供电，但这样无疑使得供电电路更加复杂，也更容易出错。实际上LM358双运放是可以直接使用5V单电源供电的，这样整个电路的供电就可以统一起来，只用一个锂电池或者直接用USB供电就可以了。

1. 在最初设计时为了保证系统能正确工作，考虑了较多因素，比如为了方便调试信号增益使用了电位计；使用了两套供电方案；单片机和上位机通信采用了串口蓝牙模块无线传输和USB数据线传输两套方案等。这样就使得整个系统较为臃肿，PCB板也过大。如果后续要优化，可以按照不同的需求进行适当裁剪，所有电位计也最好用定值电阻替代。

1. 所有设计资料（包括程序源码等）已上传到本人GitHub仓库：https://github.com/LiRengithub/SoundCollector [↑](#footnote-ref-1)
2. 所有参考资料存储在SoundCollector\Reference以及readme文件中 [↑](#footnote-ref-2)
3. 整个电路系统的原理图已导出成pdf存储在：SoundCollector\AltiumDesignerProject\Project Outputs for SoundCollector\SoundCollector.pdf [↑](#footnote-ref-3)
4. 电路仿真相关资料存储在：SoundCollector\MultisimProject [↑](#footnote-ref-4)
5. 单片机程序存储在：SoundCollector\KeilProject [↑](#footnote-ref-5)
6. 上位机程序存储在：SoundCollector\Scripts\QtProject [↑](#footnote-ref-6)
7. 合成的音频可以在采集时播放，或者使用系统音乐播放器播放SoundCollector\Scripts\QtProject\ .tempwav.wav，这是一个隐藏文件，正常情况下不可见 [↑](#footnote-ref-7)
8. 环境噪音存储在SoundCollector\Scripts\QtProject\noise.npy [↑](#footnote-ref-8)
9. 带噪信号存储在SoundCollector\Scripts\QtProject\signal.npy [↑](#footnote-ref-9)
10. 信噪比计算程序存储在SoundCollector\Scripts\QtProject\SNR.py [↑](#footnote-ref-10)