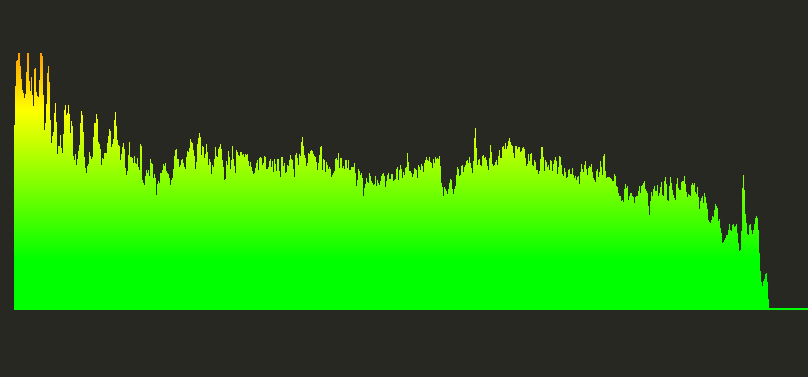
**音频频谱分析设备**

**项目需求**

1. 基于MSP432的在线频谱分析器。
2. 设备实时收集环境音频信息，在电脑上可以显示出实时的各频率分量，即FFT变换结果。
3. 音频收集器使用MSP432\_MKEDUii板上的话筒收集。
4. 采集的信息可以动态切换环境音频的采样速率，通过设备按钮或者电脑进行切换。
5. 利用MCU进行信息提取分析。
6. 频谱的展示需要使用刻度标明， 并且刻度随着采样率的改变而改变。

**需求分析**

1. 项目分为两个MCU端和展示端：
   * MCU端负责对环境音频的收集，并进行FFT分析。
   * 展示端负责接收控制频谱信息并展示。
2. FFT计算使用CMSIS库函数。
3. 需要选择或设计一套简单可扩展的协议进行串口通信。
4. 考虑到原型设计，展示效果的要求，可使用Web或者本地客户端的方式进行开发。
5. 实时数据展示可以不使用数据库。



**异常情况**

1. 无数据输入情况下应展示无数据输入。
2. 无法连接串口或串口连接中断时，应展示连接情况。

**测试方法**

1. 通过板载蜂鸣器发出指定频率声音，使MacrePhone检测到。
2. 通过产生向dma缓冲区产生指定频率正弦信号。
3. 通过信号发生器向

**项目设计**

**IO分配**

\* MSP432P401 \* ----------------- \* | P5.1/IO\_IN|<---- SW\_1 (Switch PWM) \* | P2.7/TA0.4|----> Beep (PWM) \* | P3.5/IO\_IN|<---- SW\_2 (Switch Sampling rate) \* | | \* | P1.2/UCA0RXD|<---- PC \* RST -| P1.3/UCA0TXD|----> PC \* | | \* | TA3.1|----> PWM (ADC Sampling Clock) \* | P4.3/A10|<---- Mic \* | | \* | |

其中SW*1 SW*2 Beep Mic设备由BOOSTXL-EDUMKII（扩展板）提供

**数据流**

MCU : Waves --> Mic --> ADC（DMA） --> MEM Buffer --> FFT --> uart（DMA）

PC : uart --> Display(Python Tkinter)

1. 通过ADC从Mic采集数据到数据后送给DMA进行缓存。
2. DMA完成后利用中断唤醒主线程执行FFT运算，存入缓冲区。
3. 接收到上位机请求数据指令。
4. 判断上位机是否要求此次数据，有则通过DMA向PC机发送数据。

**技术细节**

在进行程序编写方面存在一些问题，现进行一些总结。

**CCS 配置**

* 编译器选用ti提供的16.04版本gcc编译器, 并提供编译编译参数：\_\_MSP432P401X\_\_。
* 环境问题主要是编译库和静态链接库的配置问题: 需要在CCS配置项中提供相对SDK所在的CMSIS库头文件相对位置，以及静态链接库的位置。 用到的Grlib也需要添加与CMSIS类似的配置。

**ADC采样频率控制**

ADC采样时钟有多个可选项，这里由一个Timer\_A产生的PWM实现控制，通过修改TA3的CCR寄存器来控制采样频率和占空频率。

TA3时钟源配置为SMCLK，Timer\_A3->CCR[0]设置为SMCLK/sample\_frec，无需设置IO引脚。

**DMA与ADC结合的配置**

DMA使用Channel7，数据源设置为 ADC的MEM。 采用Ping-Pong模式，设置主副传输缓冲区，在一次中断之后切换主副缓冲区。

**FFT函数**

CMSIS的fft函数提供q15与q31两种，前者为数据16位后者为32位。

fft不考虑数据的采样率，或者说fft将两个采样数据之间的时间间隔认为下标间隔。

由于FFT函数的时间复杂度为nlog2(n)，即当处理的数据量增加一倍时，所需要的计算时间将远大于原来的一倍。 而采样数据的量与时间成线性关系，故当数据缓冲区间越小，花费的CPU时间越少，效率越高。

此外，fft结果的频率分辨率与缓冲区间长度有关，区间长度与分辨率为正比关系，长度越大，分辨率越大。 为了方便进行FFT运算，处理长度即采样缓冲区长度取2的整数次方。此处，fft长度为512。

关于FFT的详细解释参考：[FFT结果的物理意义](http://blog.sina.com.cn/s/blog_640029b301010xkv.html)

**DMA传输串口TX数据的配置**

DMA与UART结合也有相应的Channel，这个在Datasheet中可以查到。 由于从采样到fft均为16位整形数据，使用DMA传输大小和源指针增量为8位，将传输两倍Buffer\_len长度。 并且无需启用中断。

**主循环控制**

程序在进行完初始化操作后，将会循环调用PCM\_gotoLPM0()，该函数将使CPU进入Sleep模式以降低功耗。 在Sleep模式下，CPU不会进行任何操作，只有当触发一个中断操作的时候CPU会被唤醒。CPU被唤醒之后，将会执行FFT运算。

此时，由于中断源有ADC采样的DMA中断与串口RX数据中断两个。

故通过在采用中DMA断设置一个volatile sampling\_completed变量来保证CPU执行FFT时一次采样完成。

**串口传输设置**

当MSP432P401R-LanchPad通过板载调试器（XDS110）连接到电脑时，将会在PC上映射两个串口端口，通过设备管理器可以看到 板载串口使用

**串口传输指令**

1. 增加同步头“ACK=” 所有指令发送之前需传输同步头，设备只有在接受到同步头ACK=后，才会开始接受指令，以防止出现指令错位，出错的情况。
2. 同步头后跟的指令最长可以有六个字节。
3. 可接受的数据内容如下：
   * ACK=rfft :取一次当前fft结果，故数据频率刷新率由上位机确定。 设备接收到一个时，将在最近下一次fft采样计算完成时，向上位机回传数据。 数据为512字节，小端模式，每16位（short型）为一个单位点上的幅度值。 即，包含256个单位幅度值，第0个单位表示0Hz频率上的幅度值： 当采样率为8000时，第255个单位表示4kHz附近的幅度值。 当采样率为16000时，第255个单位表示8kHz附近的幅度值。 当采样率为32000时，第255个单位表示16kHz附近的幅度值。 第【1~255】个单位按线性则按上述线性排列。
   * ACK=pcmon :开启PCM实时采样回传。 当设备每采512个样本点（每个样本单位16位）之后，都会发，返回数据长度为1024个字节的数据，每个样本点占两个字节，小端模式。 例: fe ff 01 00 代表采到第一个样本为 -2,第二个样本为1。
   * ACK=pcmoff:关闭PCM实时采样回传。 当设备接收到关闭指令时，未发送完成的样本将继续发送，直到单次样本传送完成后停止。
   * ACK=s1 :设置设备采样率为8000Hz
   * ACK=s2 :设置设备采样率为16000Hz
   * ACK=s3 :设置设备采样率为32000Hz

**频谱分析软件-展示端介绍**

**软件功能**

1. 软件可以实时展示设备采集分析到的频谱数据。
2. 软件可以设置设备的采集速率。
3. 软件可以接收设备采集到的原始PCM信号，并保存至文件（**待测试**）。

**详细功能**

**频谱数据展示**

频谱数据通过类似柱状图来表示，横坐标为频率（最小值为0Hz，最大值为当前采样率的一半），纵坐标为幅度。刷新率为10Hz。

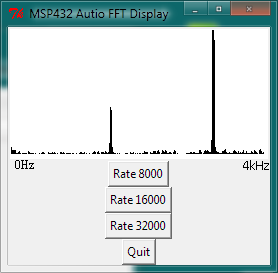
1. 横坐标-频谱数据的分辨率取决于设备使用fft分析时输入的样本数量，考虑到fft算法时间复杂度为nlog2(n)，输入样本数量为512，故输出的频谱数据长度为256个short型，即：
   * 当采样率为 8kHz时，每一条条竖线代表15.6Hz(4kHz/256)左右的频率的幅度
   * 当采样率为16kHz时，每一条条竖线代表31.25Hz(8kHz/256)左右的频率的幅度
   * 当采样率为32kHz时，每一条条竖线代表62.5Hz(16kHz/256)左右的频率的幅度
2. 纵坐标-幅度 其中0Hz为直流分量，具体含义参考fft算法结果。

**采样速率的设置**

软件频谱图下方有三个按钮，当点击不同的按钮时，将设置设备的采样率，实时生效。

**存储PCM数据**

软件可以接收到设备的原始PCM数据，并保存到指定路径下（**仍在测试**）



当蜂鸣器使用3.3khz的频率进行测试时，展示如图所示。

**详细设计**

软件使用Python编写，依赖PySerial(串口通信)和Tkinter(图形展示)扩展包。

软件主要维护两个类：

* **设备控制类（MSP432）** 该类维护一个由PySerial.open返回的一个串口设备句柄。通过该句柄，可以对串口发送和接收信息。 该类完成了对设备的RFFT请求，采样速率的设置，PCM数据传送开关接口的封装。
* **界面展示类** 该类继承了一个Tkinter的视图类，可以方便的添加一些视图控件，以及绑定一些具体事件，具体用法参见TKinter文档。