

上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

机械动力综合实验 II



闭口系统与开口系统 热力学第一定律的验证

学生姓名: 蒋如飞、张洵、高小博、张科、吴磊、刘述旺

指导教师: 施光林

学院(系): 机械与动力工程学院

目录

一、实验目的	1
二、实验原理	1
三、STM32F4 单片机简介	6
四、实验装置	10
五、实验步骤	10
六、程序设计	11
七、实验过程	11
八、实验结果与误差分析	13
九、实验感想	13

一、实验目的

1. 了解噪声的基本概念；
2. 了解工程中处理噪声的常规方法；
3. 掌握主动降噪的基本原理与方法；
4. 通过实验模拟主动降噪，分析降噪效果。

二、实验原理

被动式噪声控制和主动式噪声控制是目前最主要的两种噪音消除方式。一般的噪音消除方式采用被动式降噪，例如耳塞、耳罩等等。其原理是利用外部硬质材料和内部的填充材料，以隔离与吸收噪音的原理，阻挡外界的噪音进入耳朵，对高频噪音的降噪效果明显，这也是现在噪音抑制最常使用的方法。然而，这种降噪方式虽然实现容易，但对 800Hz 或更低频率的低频噪音而言，若要应用这种被动式减噪，需要负担大量的设备成本以及材料重量，此时被动式降噪方式失去了实际上的效益，并且被动式降噪的方式，在抑制噪音的同时，也会阻隔使用者谈话的声音，这也是被动式的主要缺陷之一。

ANC 原理是使用一颗控制器核心，进行适应性算法的运算，来输出与欲消除的噪音相位相反，频率和能量相同的声波，以达到降噪的目的；控制器核心常以数位信号处理器实现。与 PNC 不同的是，ANC 是以波形的重叠原理为基础。藉由噪音与抑制声波相结合，来造成破坏性的干涉而产生声波能量相消的效果，若抑制波的波形与欲消音的声波振幅相同，频率且反相时，则会得到降噪效果，如图 1 所示。



图 1 ANC 原理示意图

一般可将主动式噪音控制系统的架构分前馈式与反馈式：

前馈式结构，利用两个麦克风来接收外面的噪音，参考麦克风接受外噪音源信号，并经过适应性滤波器后，由扬声器输出与噪音相位相反之声波。再利用误差麦克风来接受残余的噪音讯号，调整适应性滤波器。直到达到预期值或噪音消除后停止，如图 2 所示。

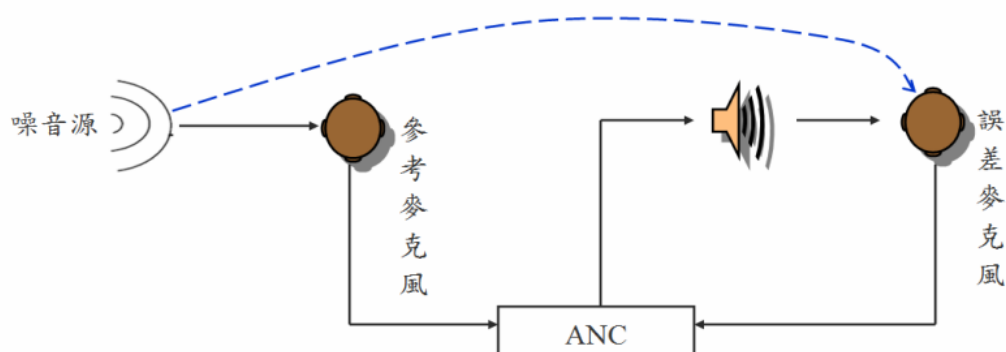


图 2 主动式噪音控制前馈式结构

而反馈式结构，仅使用一只误差麦克风。并将噪音与经由扬声器输出噪音来改变适应性滤波器，直到达到预期值或噪音消除后停止，如图 3 所示。由于反馈式的主动式噪音控制只需一个误差麦克风，可减少成本及达到轻量化目的。

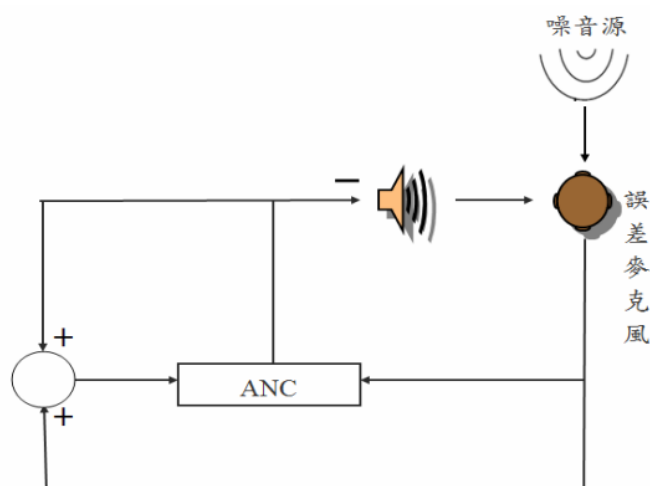


图 3 主动式噪音控制回馈式结构

主动式降噪在实际的系统中，输出信号需要经由 D/A 转换器，功率放大器和低通滤波器等电路送出，而由误差麦克风所接受的误差讯号也需要经由放大器、滤波器和 A/D 转换器送到处理器处理。除此之外扬声器至误差麦克风和噪音源也包含了声场中的转移函数，这些转移函数总括为 $S(z)$ ，也称作次级通道转移函数。如果直接使用 LMS 算法来降噪，而不考虑这些转移函数的影响，可能会使得系统变得不稳定。如图 5 所示 $S(z)$ 包含了扬声器至误差麦克风和噪音源的转移

函数，若考虑次级通道转移函数的影响，则误差信号如下，

$$\begin{aligned}
 e(n) &= d(n) - y'(n) \\
 &= d(n) - s(n) * y(n) \\
 &= d(n) - s(n) * [w(n)x(n)]
 \end{aligned} \tag{2.1}$$

使用最快下降法，对误差信号微分如下

$$\frac{\partial e^2}{\partial w} = 2e \cdot \frac{\partial e}{\partial w} \tag{2.2}$$

$$\frac{\partial e}{\partial w} = -s(n) * x(n) \tag{2.3}$$

将(2.1)式以及(2.3)式代入(2.2)式则调整滤波器系数的方法必须被修正为

$$w(n+1) = w(n) + ue(n)\{s(n) * x(n)\} \tag{2.4}$$

然后将(2.4)式改写如下

$$w(n+1) = w(n) + ue(n)x'(n) \tag{2.5}$$

在实际 ANC 应用上，S(n)是无法先被预知的，因此传统的 LMS 演算法必需被修正，而最为广泛使用的算法则为 Filtered-X LMS 演算法，在降噪系统执行前，先估计出次级通道转移函数如图 5 所示，使得输入信号 x(n) 经由第二路径函数而得的 d(n) 与经过估计函数得到的 y(n) 能达到一致，籍此来消除第二路径的影响。如图 6 所示，图中 S(Z) 为第二路径转移函数的估计函数，算法为：

$$y(n) = \sum_{l=0}^{L-1} S'_l(n)x(n-l) \tag{2.6}$$

$$e(n) = d(n) - y(n) \tag{2.7}$$

$$s'_l(n+1) = s'_l(n) + ux(n-l)e(n), \quad l = 0, 1, \dots, L-1 \tag{2.8}$$

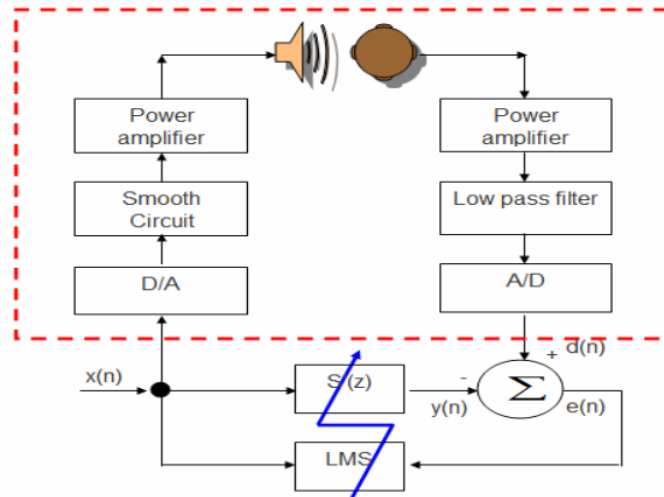


图 4 第二路径转移函数估计

而后，将 LMS 演算法结合次级通道转移估计函数，如图 6 所示，

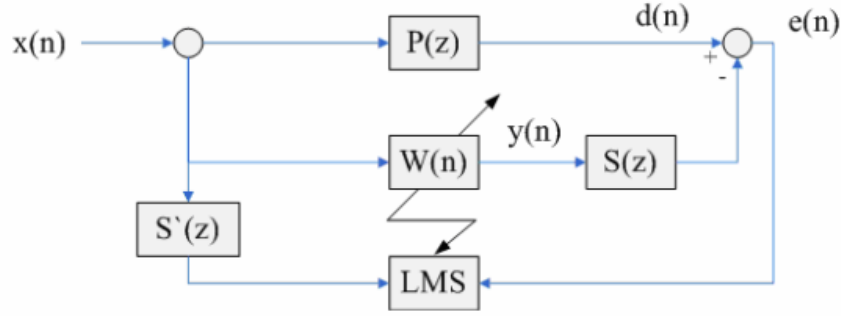


图 5 Filter-X LMS 演算法

而常用的 Filter-X LMS 演算法即可得之为

$$e(n) = d(n) - y'(n) \quad (2.9)$$

$$y(n) = \sum_{l=0}^{L-1} w(n)x(n-l) \quad (2.10)$$

$$x'(n) \equiv \sum_{m=0}^{M-1} s_m' x(n-m) \quad (2.11)$$

采用反馈方式的 ANC 系统如图 6 所示。x(n) 噪音信号为 e(n) 残余误差信号和 y(n) 输出信号回馈后计算求得。

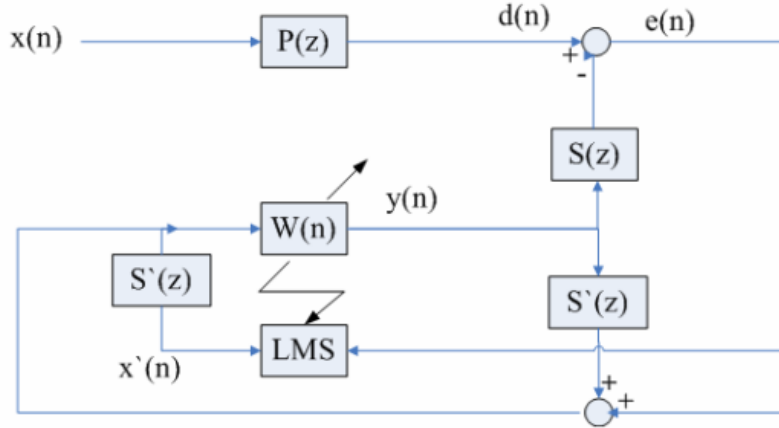


图 6 Feedback FXLMS 演算法

Feedback FXLMS 方式的 ANC 系统，整个算法如下：

$$x(n) \equiv d'(n) = e(n) + \sum_{m=0}^{M-1} s_m' y(n-m) \quad (2.12)$$

$$y(n) = \sum_{l=0}^{L-1} w(n)x(n-l) \quad (2.13)$$

$$x'(n) \equiv \sum_{m=0}^{M-1} s_m' x(n-m) \quad (2.14)$$

$$w(n+1) = w(n) + ux'(n-l)e(n), \quad l = 0, 1, \dots, L-1 \quad (2.15)$$

三、STM32F4 单片机简介

实验在 STM32F4 单片机上运行。STM32F4 是意法半导体公司推出的基于 ARM Cortex-M4 内核的一款单片机。STM32F4 采用了意法半导体的 NVM 工艺和 ART 加速器。STM32F4 系列单片机工作主频高达 186MHz，并且带有 DSP 和 FPU 指令，是 MCU 实时控制功能与 DSP 信号处理功能的完美结合。

STM32F407 是 STM32F4 系列中的一款单片机，不仅具备 STM32F4 系列的基本特性，还具有以下特点：

1. 高达 1M 的闪存，高达 192K 的 SRAM
2. 3 个 12 位 2.4M 的 AD 转换器，通道数高达 24 个
3. 2 个 12 位的 DA 转换器
4. 16 个数据流的 DMA 控制器，DMA 可以直接在内存和外设之间传输数据，无需 CPU 参与，大大降低了 CPU 的负荷
5. 多达 17 个计时器，多达 140 个 I/O 口
6. 多达 15 个通信接口，有三个 I2C 接口、4 个 USART 接口、2 个 UART 接口、3 个 SPI 接口、2 个 CAN 接口和一个 SDIO 接口。

由于 STM32F407 功能十分多，整个系统的结构设计为总线矩阵的形式。如图 7 所示。

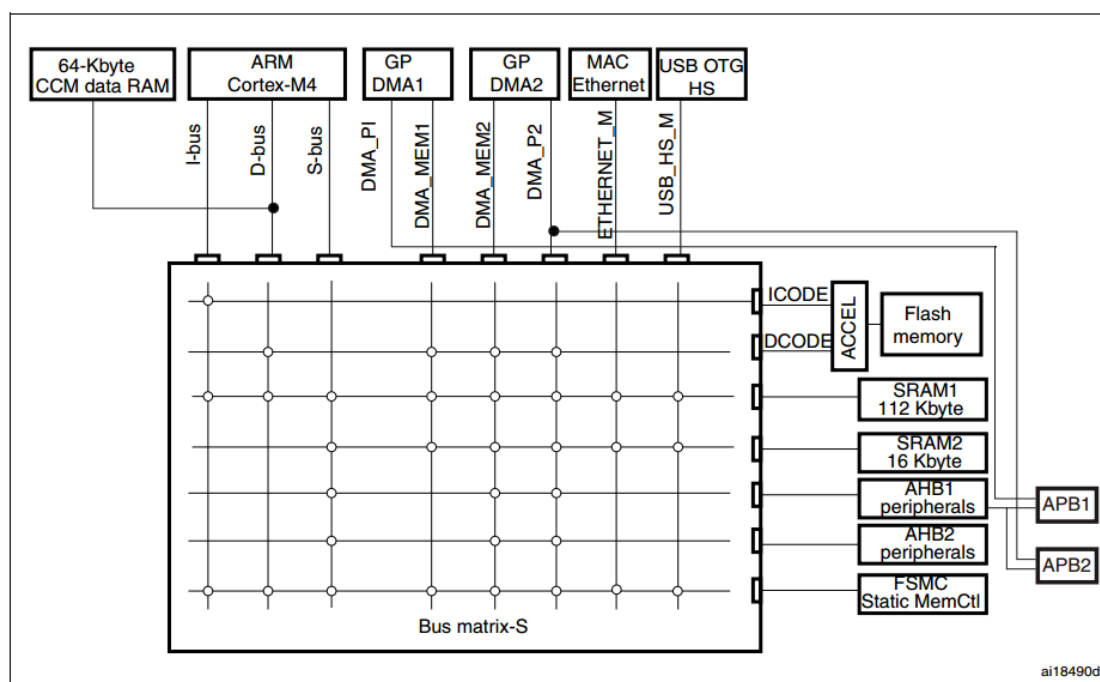


图 7 STM32F4 系统结构图

AHB (Advanced High performance Bus) 是高性能总线, 主要用于 CPU、DMA、DSP 等高性能模块之间的连接。APB (Advanced Peripheral Bus) 是外围总线, 主要用来连接一些外设, 由于外设的工作频率等要求较低, 将总线分开设计可以提高系统的可靠性、降低功耗。

STM32F4DISCOVERY 是意法半导体公司推出的基于 STM32F407VGT6 的一款开发板, 如图 8 所示。STM32F4DISCOVERY 具有以下特点:

1. 包含一个 ST-LINK/V2 嵌入式调试工具, 用户可以直接连接 USB 进行调试
2. 两个 MEMS 数字加速度计
3. 一个数字麦克风
4. 一个音频 DAC, D 类音频驱动和耳机接口, 无需外围电路即可驱动耳机
5. 一个 USB OTG micro-AB 接口, 可以插入 OTG 转接口读取 U 盘内的文件
6. 强大的官方支持, 有大量针对开发板的库和例程可以使用



图 8 STM32F4DISCOVERY 开发板

由于 STM32 系列单片机功能十分强大, 使用者不仅需要了解整个单片机的架构、外设和各种底层知识, 还需要配置各种软件功能。最早的方式是通过配置寄存器进行的, 但往往用户实现一个简单的功能需要大量的配置工作, 效率十分低下。随着单片机编程理念的发展, 渐渐出现了很多库函数帮助用户配置, 当使用者需要配置某个功能时只需要调用相应的库函数即可。

然而, 库函数虽然方便, 但仍然存在过于零散繁杂的问题。为了解决此问题, 意法半导体公司专门提供了一套硬件抽象库: HAL (hardware abstract layer)。硬件抽象库以功能为基本单元, 将整个单片机抽象为各种功能的集合, 为每个功能提供了一系列统一格式的配置函数, 降低了用户的学习成本。

更进一步的, 意法半导体公司推出了整套配置 STM32 系列单片机的工具: STM32CUBEMX 和 STM32CUBE。STM32CUBEMX 以图形化的方式配置 STM32, 用户在了解 STM32 的基本结构的基础上, 只需要点击相应的时钟和外设就可以完成单片机的初始化和基本功能的配置。

STM32CUBE 是一套庞大的代码，不仅包含 HAL 等各种基础库、RTOS 等各种高层次功能的中间件，还包含意法半导体公司推出的开发板的功能示例代码。其结构图如图 9 所示。

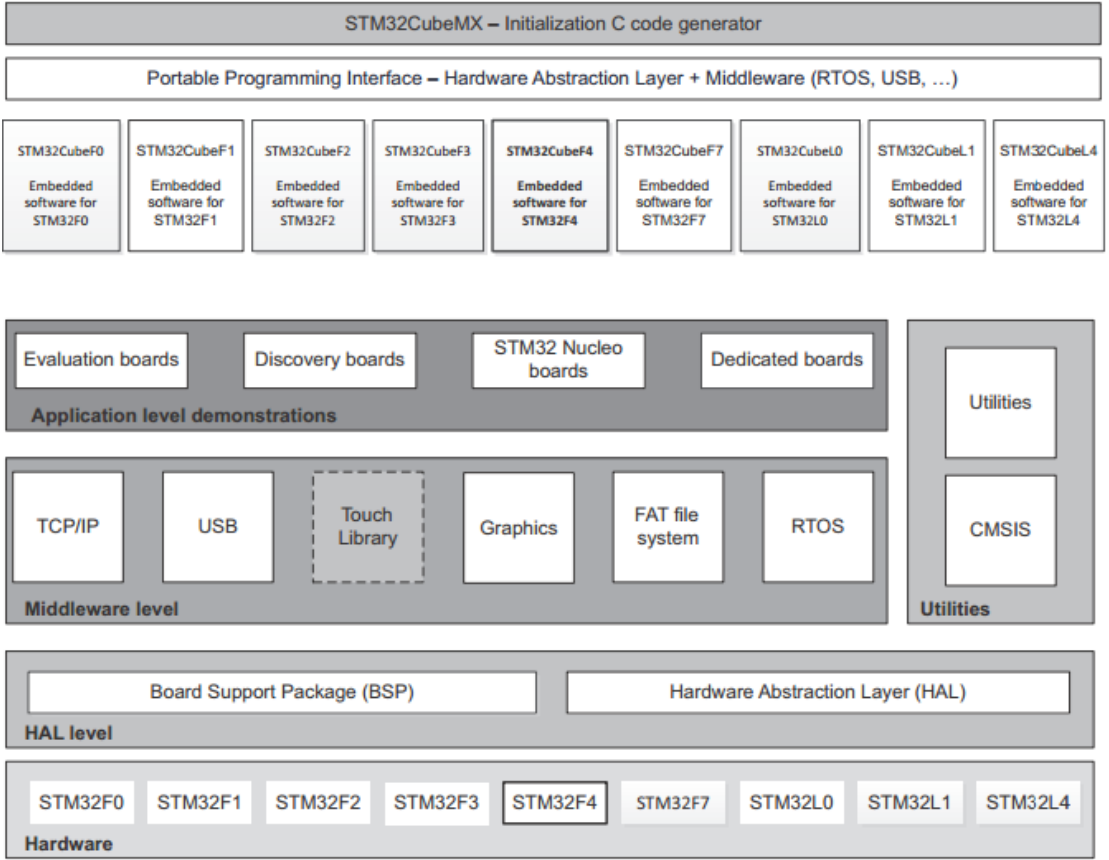


图 9 STM32CUBE 结构图

STM32CubeF4 为 STM32F4DISCOVERY 开发板提供了一个声音录放的示例工程，由于在本次实验中需要实时采集音频并播放算出来的降噪声波，因此可以参考示例工程来进行功能的开发与测试。示例工程示意图如图 10 所示。

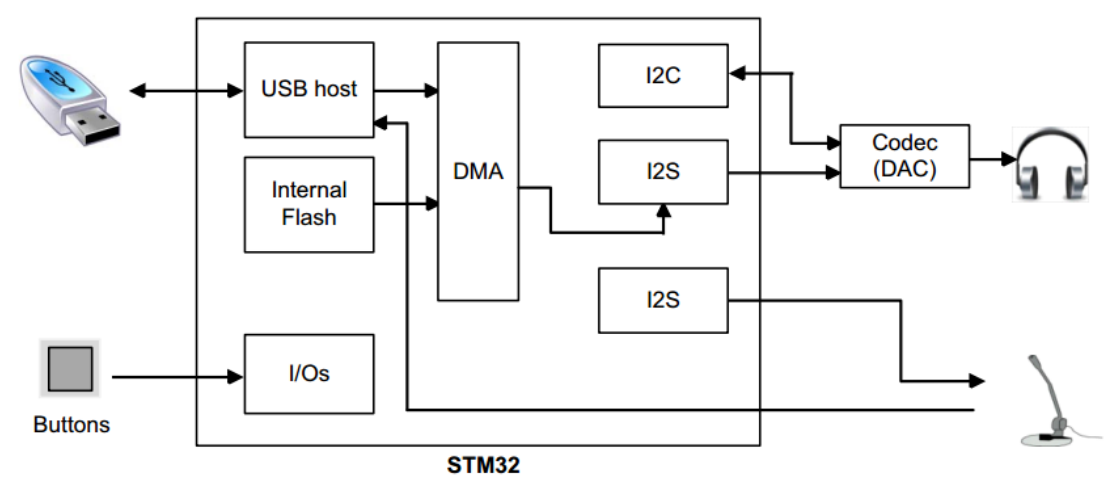


图 10 示例工程示意图

四、实验装置

如图 11 所示，实验仪器由开发板、扬声器、误差麦克风构成，组成一个基本的负反馈系统。U 盘是用来跑示例工程的，在改写完后的降噪实验中并没有用到。



图 11 实验装置图

五、实验步骤

1. 运行示例工程的代码，观察录放功能的操作和效果
2. 分析示例工程的代码，将示例工程的 USB 与逻辑控制相关代码删除，改写代码为实时录音播放，查看播放效果以确认改写成功
3. 添加核心的 FXLMS 算法代码与次级通道识别算法，实时输出麦克风测量到的声音误差，确认算法确实有降噪的效果
4. 满足要求后，结束程序，拆除实验平台，整理实验设备；
5. 整理相关实验图片和数据，进行数据分析；
6. 分析实验误差，得出结论并撰写实验报告。

六、程序设计

程序的工程图如图 12 所示。程序使用了 STM32CubeF4 提供的 HAL 库和板级支持包，使用了 DMA 功能传输麦克风和扬声器的数据。由于在 DMA 传输数据的时候数据不能被更改，因此程序中使用了双缓冲的机制连续地完成数据的传输：DMA 传输一半的数据后触发中断，此时这一半的缓冲可以被程序修改；同理，传输完下一半后反过来，程序修改后半段缓冲。

缓冲的大小决定声音的延迟，过大的缓冲会导致声音延迟太久，难以形成负反馈。在本次实验中，采样频率设置的 8K，缓冲区大小设置为 256。

为了降低运算的负荷，FXLMS 算法中使用的数据都是 16 位无符号整型。滤波器的大小设计为缓冲区的一半，这样可以保证滤波只用在数据收到一半时只进行一次，这样也可以降低运算量，保证计算的实时性。

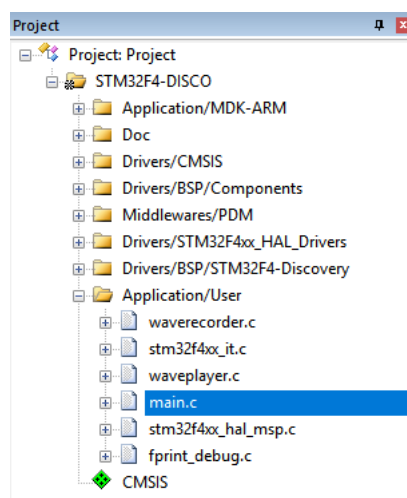


图 12 程序工程图

七、实验过程

1. 将示例工程提供的 wav 文件拷入 U 盘，使用 OTG 线连接到开发板的 micro-USB 接口上。将示例工程用 keil5 编译并下载到开发板，按下开发板上黑色的重

- 置按钮。实验中可以听到音乐播放，表明示例工程的播放功能完好。
2. 使用手机在麦克风周围播放音乐。再次按下重置按钮，之后立刻按下蓝色的用户按钮开启录音功能。开发板将录音 30s，之后播放录音。实验中可以听到完全相同的音乐，表明示例工程的录音功能完好
 3. 将示例工程的 USB 与按钮控制相关代码删除，改写为实时录放的代码。实验中在麦克风周围播放音乐，从耳机可以听到音乐，但是声音明显从女声变为男声，音乐的频率变低。由于水平和精力所限，多次检查后无法找到原因，考虑到降噪过程中会估计次级通道纠正麦克风采样到的声音，因此仍能进行下一步的实验。
 4. 使用随机数生成噪声，将耳机对准麦克风播放。使用 LMS 算法估计次级通道传递函数，10S 后保存得到的参数。
 5. 使用 FXLMS 算法降噪。将扬声器拿开，将麦克风测到的误差输出到 keil5 debug 窗口。之后再将扬声器放到麦克风周围，再次将误差输出到 debug 窗口。两次的误差情况如图 13 所示。

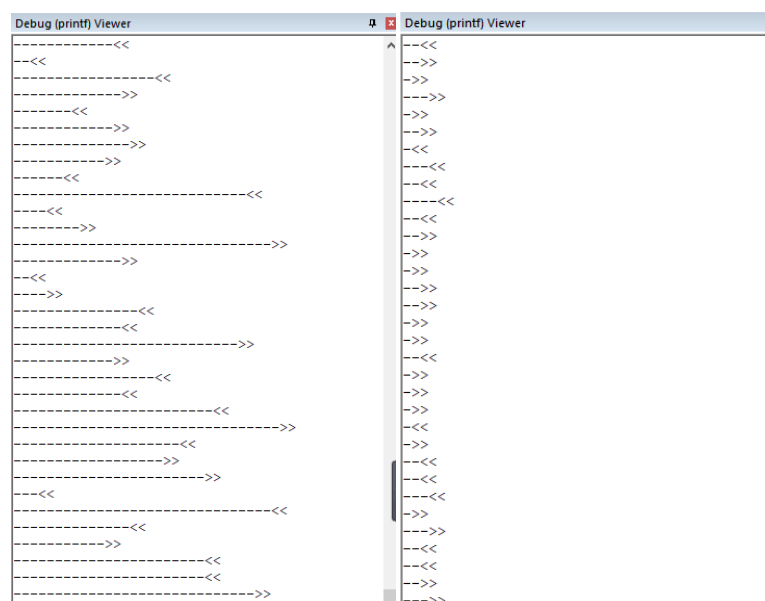


图 13 降噪前后噪声误差图

八、实验结果与误差分析

本次实验中实现了基于单麦克风反馈的 FXLMS 算法的对宽带噪音的降噪。实验结果显示大约能降低 70% 的噪声，误差分析如下：

1. 次级通道传递函数估计误差。实验中并没有类似耳机的固定和包围外壳，扬声器到麦克风的次级通道参数可能经常变化。
2. FXLMS 算法本身的缺陷。由于 FXLMS 算法是迭代算法，根据误差每次改变一点滤波器的值以找到噪音最小值，当目标存在多个极小值的时候，FXLMS 算法很可能会陷入局部最优解。实验中听到扬声器的输出接近白噪声，而且使用市面上的降噪耳机也发现有底噪的问题，说明 FXLMS 算法很容易陷入白噪声的境况。
3. 由于水平和精力所限，算法的各个参数和类型并没有精细的调整。

九、实验感想

在本实验中，我们很幸运地找到了意法半导体公司推出的针对音频功能的开发板，免去了大量硬件的开发工作。但是，由于 STM32 系列单片机功能强大，学习起来很困难，而且不仅仅需要对单片机了解，还要对 C 语言编程、各种外设功能都要熟悉，因此我们花费了大量的时间学习 STM32 单片机的开发。

使用单片机实现降噪系统是必要的，因为降噪产品大多数都是基于嵌入式系统的，不可能使用 PC 来构建降噪系统。通过对 STM32 单片机的学习，我们了解到了实验与产品的差距是巨大的，即使学会了算法，要在嵌入式系统中实现也是困难重重，要考虑到尺寸、功耗、内存、运算速度等各种各样的因素。

我们最终的结果是不完美的，可以从三个方面改进：

1. 制作一个真正的耳机，将麦克风扬声器固定好位置，制作专门的 PCB 板
2. 仔细调整 FXLMS 算法的各项参数，甚至寻找更好的算法解决落入局部最优解的问题
3. 使用专门的工具进行量化的降噪测试