AudioPro项目详细设计说明书

作者：牛一群

完成日期：2019年5月23日

签收人：

签收日期：

修改情况记录：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 修改批准人 | 修改人 | 修改日期 | 签收人 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

[1 引言 2](#_Toc9353307)

[1.1编写目的 2](#_Toc9353308)

[1.2范围 2](#_Toc9353309)

[1.3定义 2](#_Toc9353310)

[1.4参考资料 2](#_Toc9353311)

[2总体设计 2](#_Toc9353312)

[2.1需求规定 2](#_Toc9353313)

[2.2运行环境 2](#_Toc9353314)

[2.3基本设计概念和处理流程 2](#_Toc9353315)

[2.4结构 2](#_Toc9353316)

[2.5尚未解决的问题 2](#_Toc9353317)

[3接口设计 2](#_Toc9353318)

[3.1用户接口 3](#_Toc9353319)

[3.2内部接口 3](#_Toc9353320)

# 1 引言

## 1.1编写目的

* 为编码人员提供依据；
* 为修改、维护提供条件；
* 说明程序模块的设计考虑；

## 1.2范围

本说明书的预期读者为系统设计人员、软件开发人员、软件测试人员和项目评审人员。

## 1.3定义

## 1.4参考资料

Audacity源码；

# 2总体设计

## 2.1需求规定

* 如有界面，需用QT来开发界面；
* 功能函数要进行封装，便于后期打包成动态库或优化时最小代价修改代码；
* 关键代码加注释；
* 基于C、C++开发；
* 对拾音器采集到的音频进行进一步的优化和输出（或生产音频文件）；

## 2.2运行环境

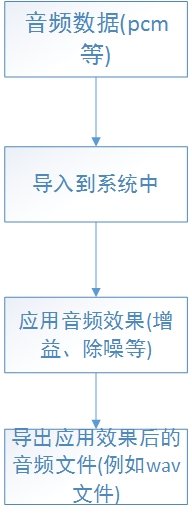
开发工具

* 平台：Windows10
* 开发工具：QtCreator4.8.0
* 开发库：Qt5.12.2

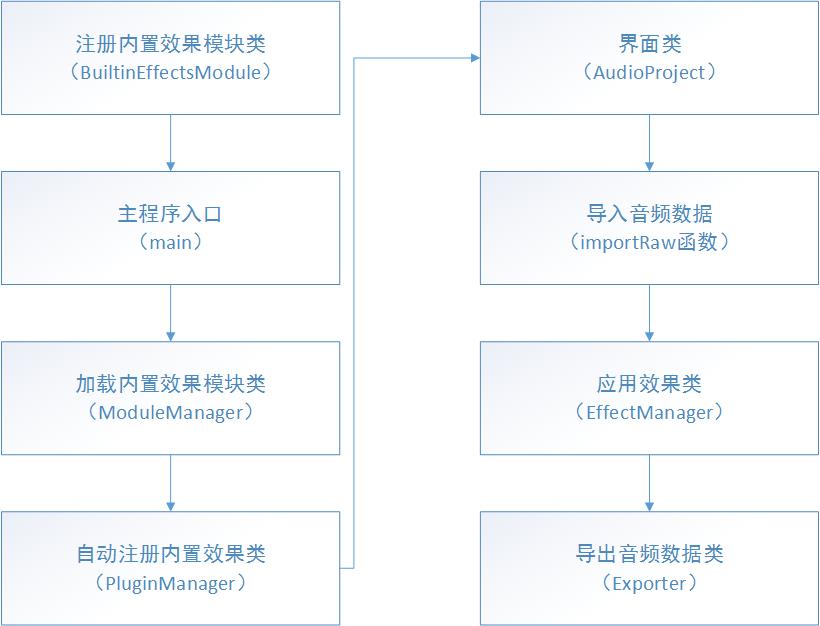
测试环境

* Windows10

## 2.3基本设计概念和处理流程



## 2.4结构



## 2.5尚未解决的问题

* 除噪效果未完成；
* 回音消除效果未完成；
* 混音效果未完成；
* 声音定级效果未完成；
* 回音消除和声音定级Audacity软件中并未有现成效果（后续版本不知道是否会增加）需要查找别的三方库来实现；

# 3接口设计

## 3.1用户接口

## 3.2内部接口

* 程序还没有进入main函数之前执行的宏展开（将创建内置效果模块的函数注册到模块管理对象中）。



* 程序主入口函数。



* 程序初始化函数（main中调用的第一个函数）加载内置效果模块和注册内置的效果。



* 创建内置效果模块对象并加载内置效果对象。



* 创建程序的主界面（和UI的交互全在这个类中完成）。

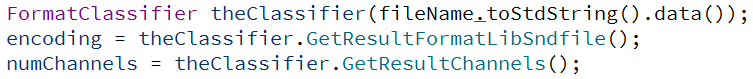


* 导入原始音频数据（pcm数据源）。





* 查看输入源音频支持的属性（例如：通道数和编码），以下代码块在importRaw函数中。



* 导入原始音频数据时显示的界面（包括导入时是支持立体声音或者导入的采样格式等）。

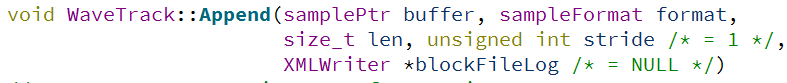


导入原始音频数据（pcm数据）时用到了libsndfile库。

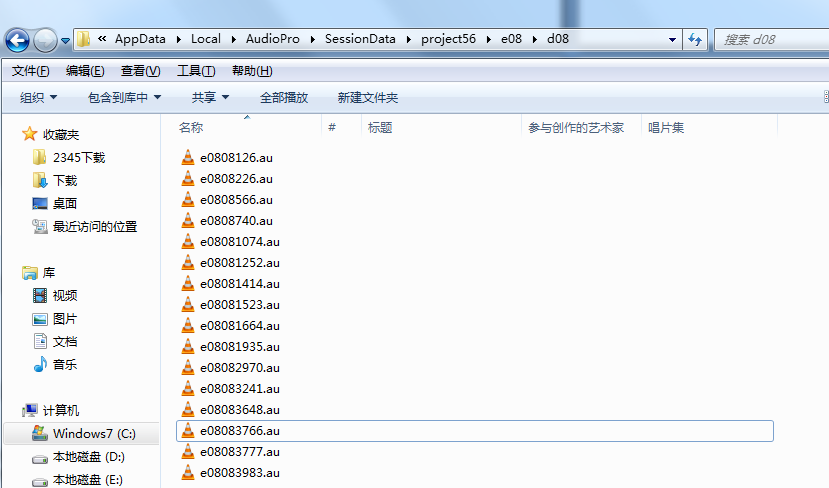
* 创建波形轨道（音频数据有几个通道这里边是抽象成了轨道，蛮好的，UI上表示的时候可以选择不同的表现形式，波形是比较常见的形式）。



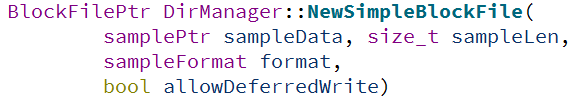
* 把原始音频数据添加到波形轨道抽象对象中（利用libsndfile读取pcm文件）。



* Audacity中专门把处理音频数据（原始字节数据）抽象了一个处理类（Sequence），这个类是实际操作导入的原始音频数据是怎么样以分块的形式保存到本地硬盘上的。（个人感觉如果时间不够的话可以不看里面的实现细节，如果想了解的比较透彻的话可以看下，会把pcm原始音频数据生成多个.au文件保存在指定的位置）。



创建本地的块文件（.au文件）。



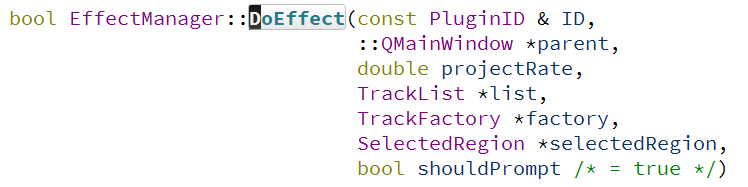
下面这个函数做的工作有点意思，它会在你指定的目录下生成二级子目录名字是随机的，看上面的那个生成.au的图，SessionData下边的目录。



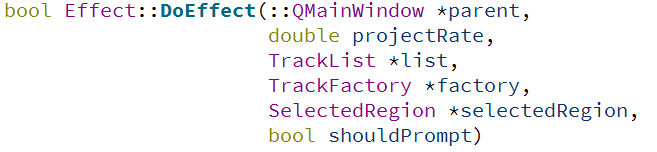
* 对已经加载的音频数据应用指定的效果（这里面改了一下和原始文件处理的流程不一样，原始处理流程比较大，感觉不太适用）。



这个函数中有一个计算结束时间的函数这个和原始的代码也有一些不一样，处理流程变了所以这里加上了计算音频数据结束时间的计算。



上面的函数中是最终确定的效果去执行对原始音频数据的修改。



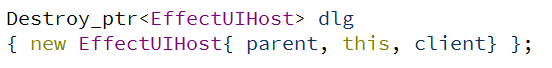
* 显示应用效果的界面。



* 创建应用效果的界面。



这个函数中的这两行创建一个通用的对话框，后面具体不同的效果想要什么控件往这个通用的对话框中添加就行。



它这个函数里有一个函数很有意思这是个纯虚函数在具体的效果类中添加自己需要的界面控件，剩下的界面的交互全是在自己的效果的界面事件中完成的，包括改变效果中的参数。

* 导出应用完效果后的音频数据。



* 导出音频数据处理类。



* 检查轨道和声道数是否匹配（比如是立体声音就会有左声道和右声道）。



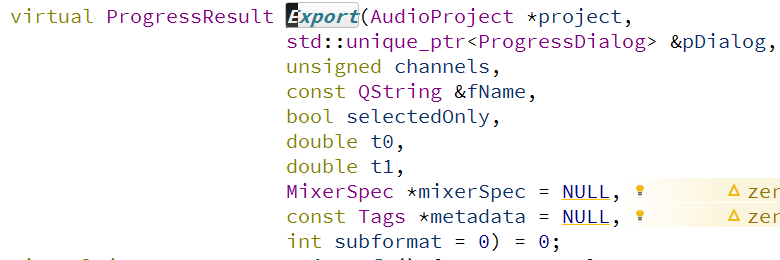
* 获取导出音频文件的名称（弹出对话框你自己来选择要把导出的音频文件放到哪里和取一个什么样的名字）。



* 这个函数会设置通道的数量，如果没有加这个函数会生成不了导出的文件。



这也是个虚函数，实际调用的时候会调用你具体导出的文件格式，我这里是导出pcm格式的文件所以会应用导出为pcm格式的处理对象。



处理数据（可以导出的大小和要导出的原始数据：原始字节）。

