# 项目总结Phase1

## 目标描述

第一步： 开发一个应用程序。

用 VS2010 ， C 或 C++ 开发一个 USB 声卡的声音 loop。 即： 打开 USB 声卡， 读取声音数据， 然后输出， 即耳机里能听到自己 MIC 的声音。

延时越小越好。 建议用 DirectShow 和线程。 （提示： 声音数据包的长度， 是影响延时的一个因素。 一般 5ms 一个数据包 ）

## 情况简述

已经使用VS2010通过WinAPI编写了DirectShow程序，进行声卡的loop工作。延迟由最初的500ms以上已经减小至150ms。

但是并未找到数据包延迟的相关信息，最后延迟的优化主要是通过更改pin的缓存属性来实现的。而且测量延迟的方法同样需要一些指导。

## 详细情况

### DirectShow

在NT内核的Windows操作系统中，大多数多媒体程序都使用DirectShow模型编写。在Win7出现之前，DirectShow归属于DirectX SDK的一个分支。但是在Win7发行之后，DirectShow已经划归于Windows SDK的分支。

在DirectShow程序中，一个多媒体程序使用一个Graph来描述，其中的各种软件或者硬件设备都使用一个filter来表示。其中程序的工作流程都是使用逻辑连线来代表媒体的流向。

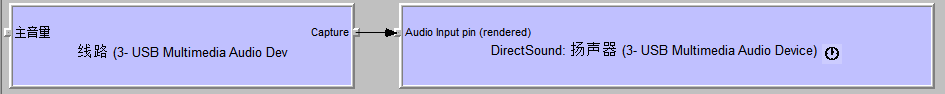


Figure 1 现有程序使用的Graph模型

现在使用以上的Graph来作为DirectShow程序的模型，但是在测试模型的时候发现一个重要的缺陷，就是延迟大概有500ms以上。能够明显感觉到延迟的存在。之后，通过更改Capture的pin上的缓存属性来改善延迟的状况。

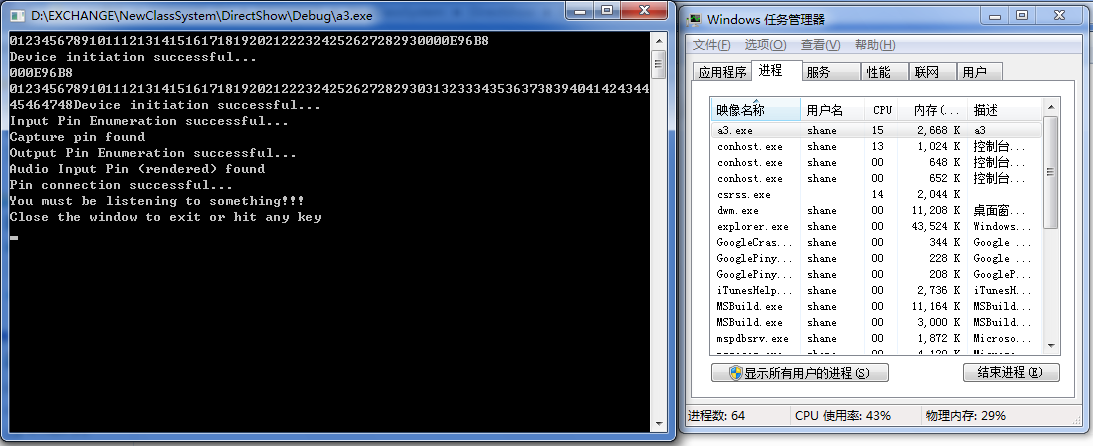


Figure 2 程序运行时的资源占用状况

程序的具体源码可以见相关的NewClassSystem的Github源。

### 双声道测试法

为了测试声卡的回放延迟，这里使用双声道测试法配合本地的Line In接口和录音软件进行精确的测量。测量系统图如下：



通过相关的分析软件分析录音样本得到的图形如下：

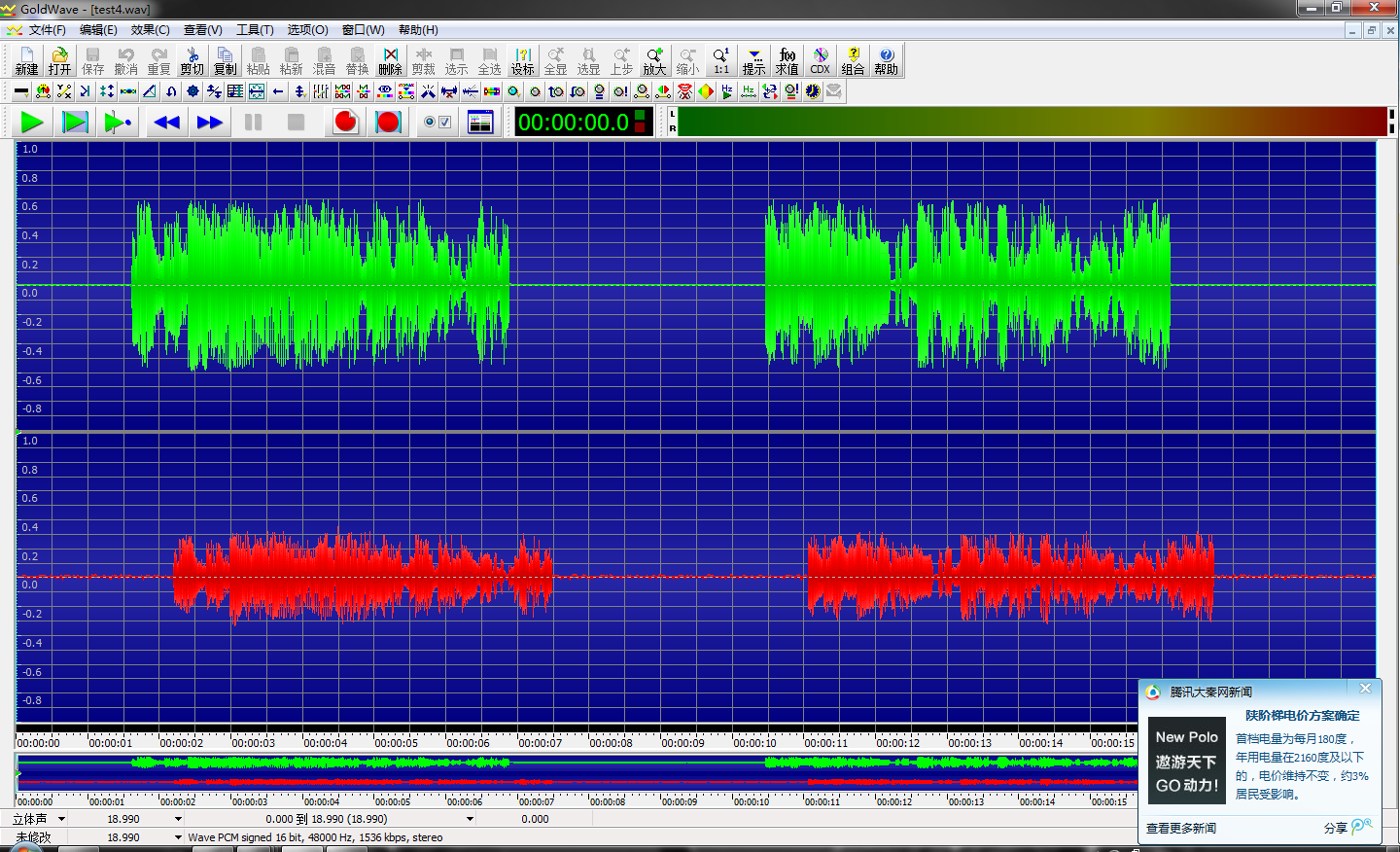


Figure 3 整体样本

为了能够准确测量回放误差，我们可以拉伸时间轴来进行比对最后得到最后的回放延迟：

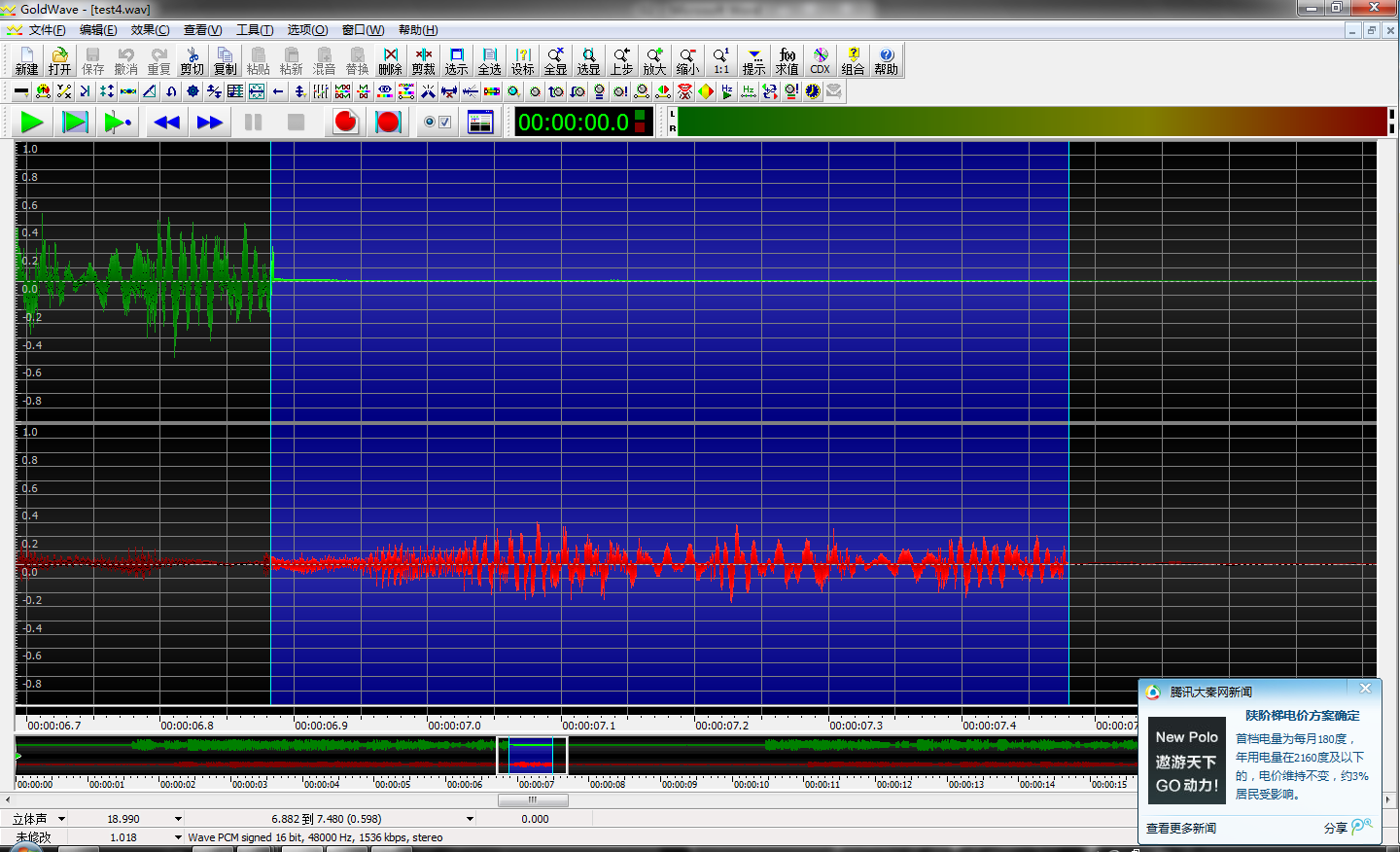


Figure 4 测量具体的延迟

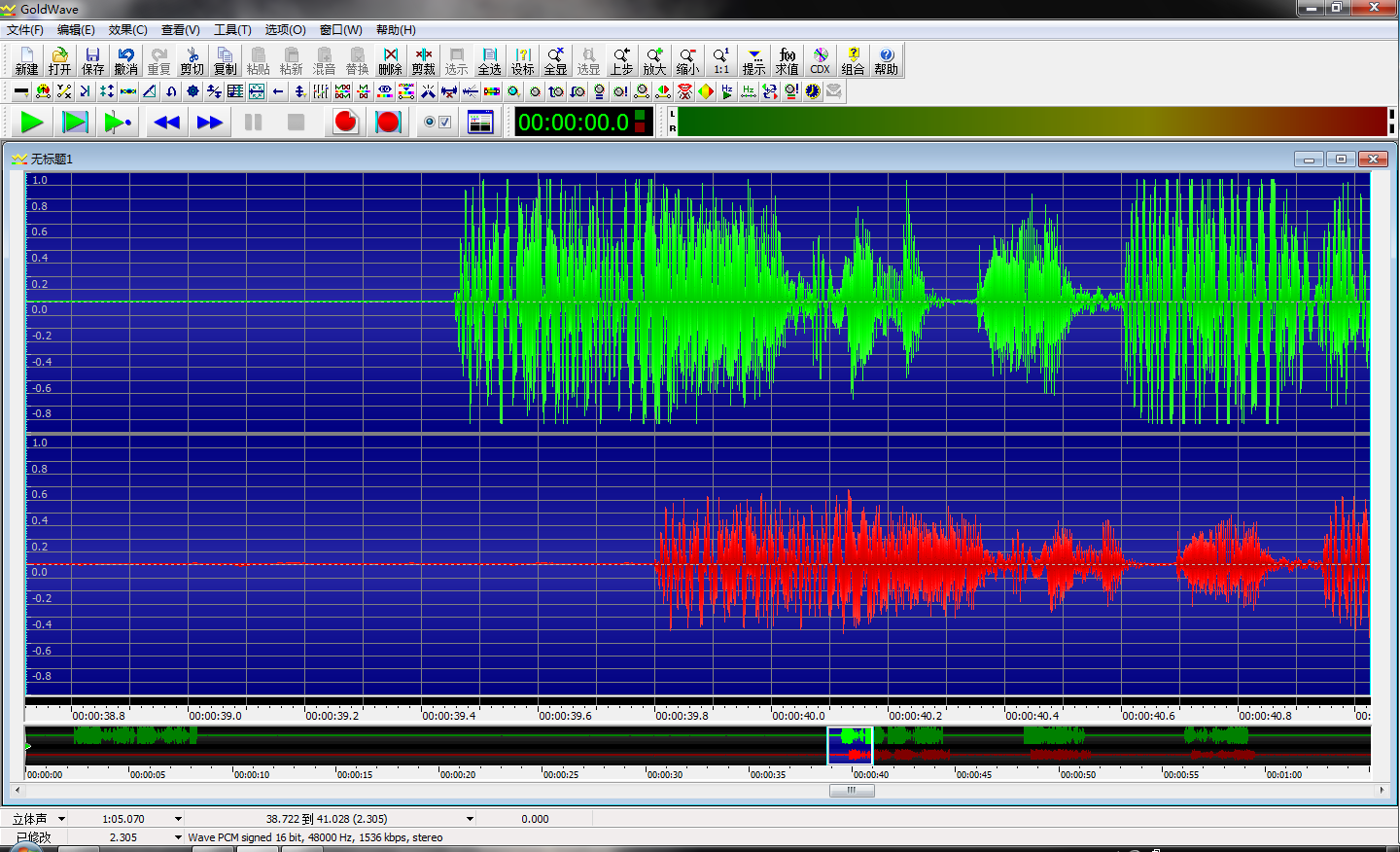
### 回放误差优化

在默认未优化的情况下，回放延迟大致保持在560ms的水平。查询相关资料，得到相关信息为，Windows默认在pin上预留500ms的缓存。根据相关资料，Audio Capture默会设置容纳0.5s媒体的缓存区，

1. 所以一种方法是简单粗暴的减小缓存区的大小；
2. 另外一种就是通过相关的公式以及录制的媒体的性质，一方面减小线入媒体的音质，另一个方面减小缓存区的大小。

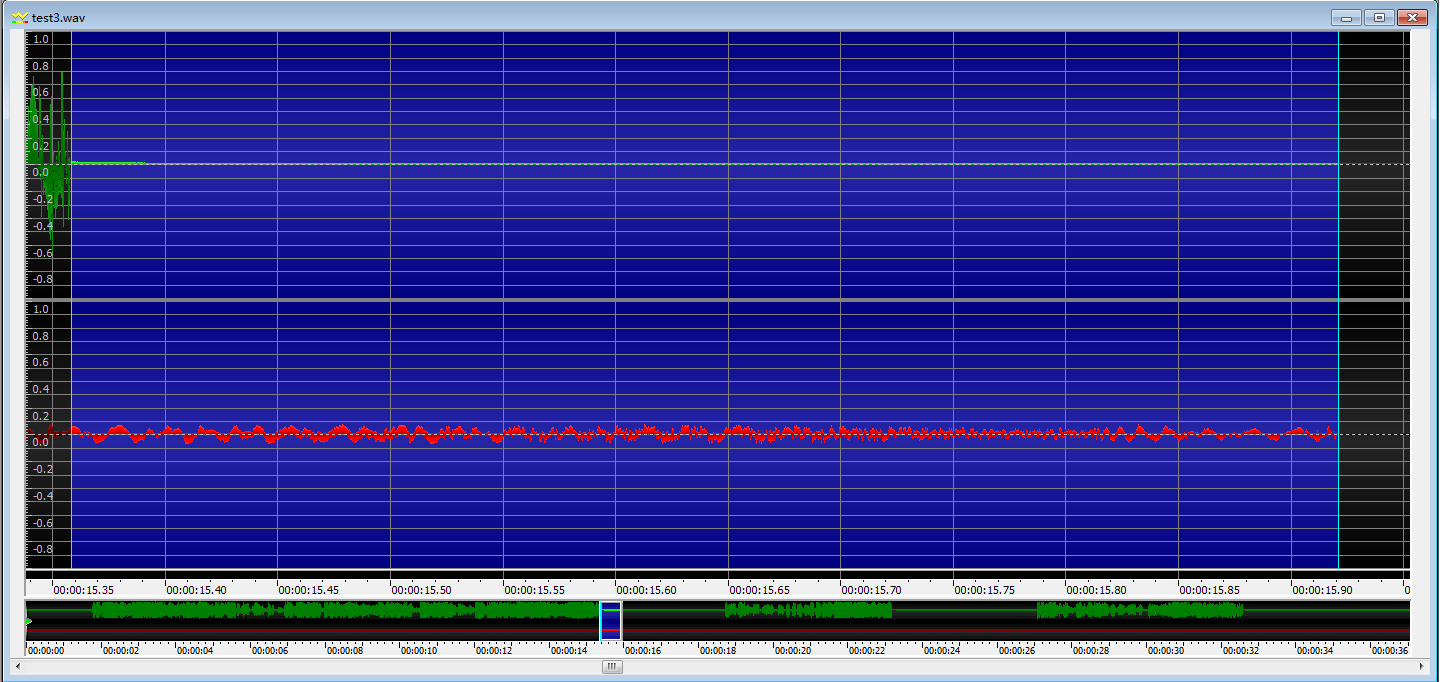
在这里尝试的是第一种方法。但是第二种方法也是完全可行的。在本次程序中已经能够成功调取Capture的媒体属性。

需要注意的是在使用方法IAMBufferNegotiation:: SuggestAllocatorProperties(&prop)的时候，务必要保证Capture接口未连接。



#### 500ms缓存延迟

未改之前的波形为



未改之前的延迟为



563ms

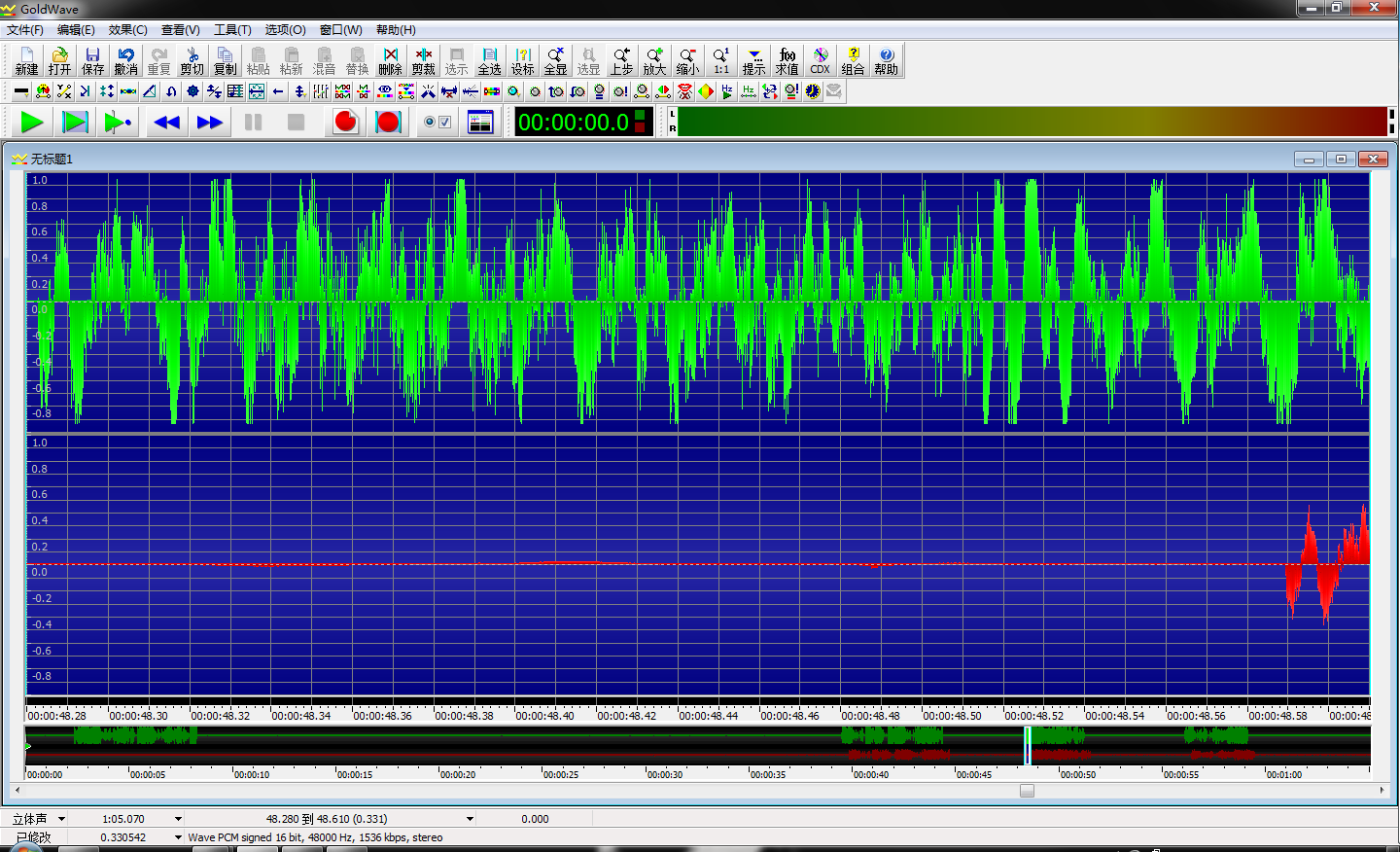
可以基本上确定系统延迟为50-70ms之间。

#### 250ms缓存延迟

当使用下列的Buffer属性优化之后，延迟得到改善。

|  |  |  |
| --- | --- | --- |
| cbAlign | 1 | 未知 |
| cbBuffer | 44100 | 缓存大小 |
| cbPrefix | 0 | 未知 |
| cBuffers | 4 | 未知 |

具体延迟见下图：



起始点位48.283

结束点位48.589

延迟为307ms

延迟有了较大的改观



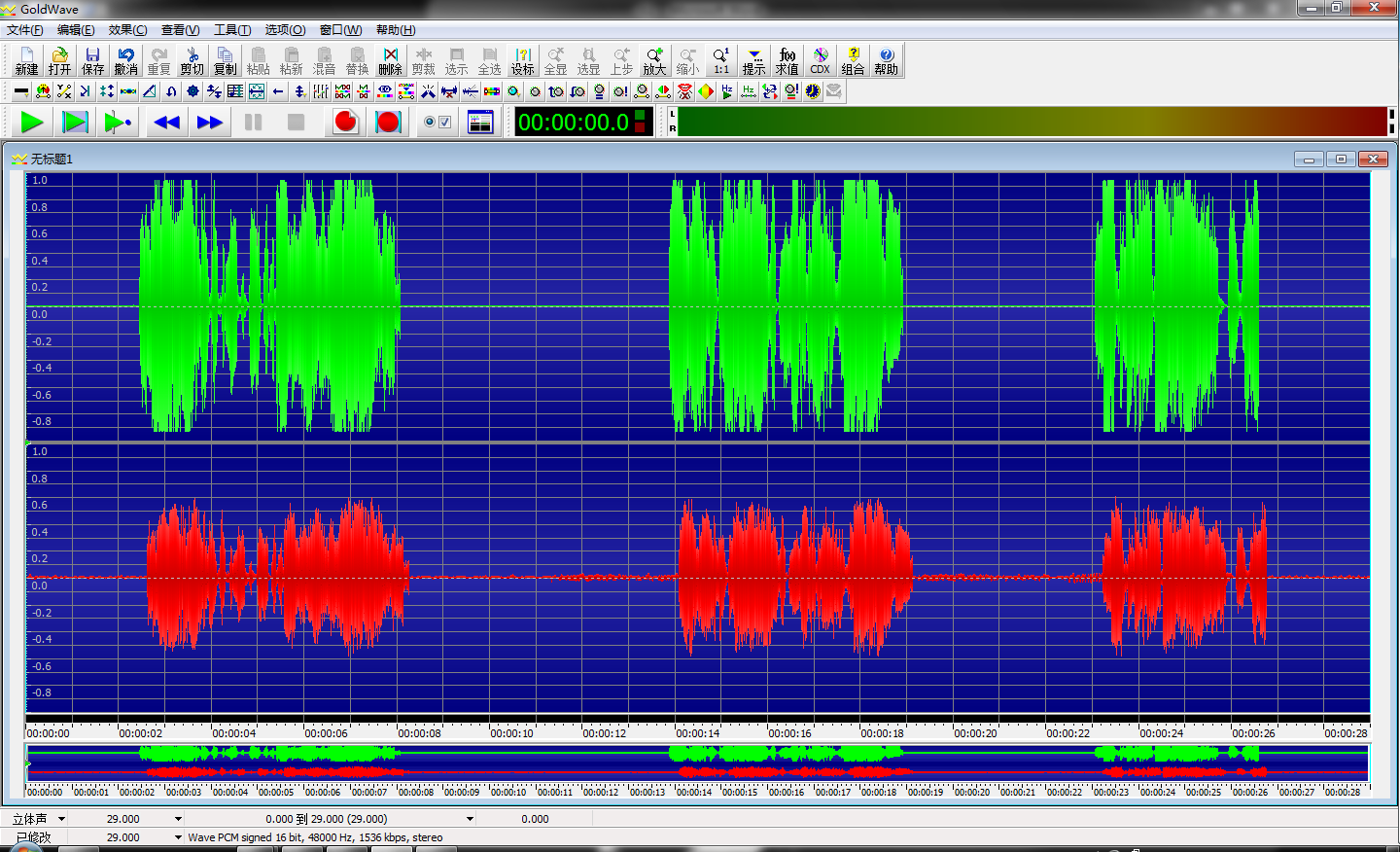
文件为test5.wav

#### 25ms缓存延迟

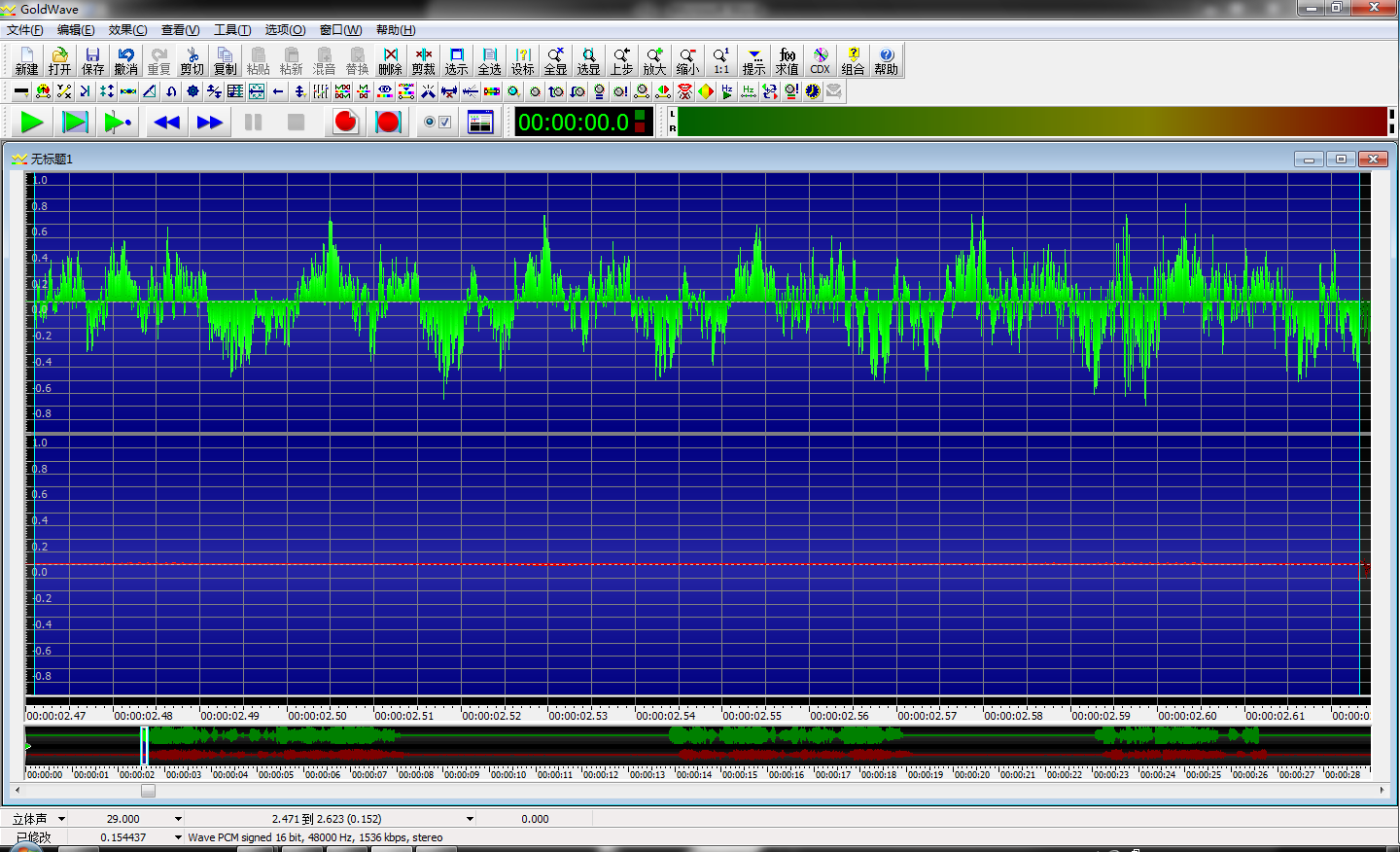
这次使用的配置为：

|  |  |  |
| --- | --- | --- |
| cbAlign | 1 | 未知 |
| cbBuffer | 4410 | 缓存大小 |
| cbPrefix | 0 | 未知 |
| cBuffers | 4 | 未知 |

最后得到的设置缓存大小为4408，应该是由于内存对齐的问题。本次测试的输出波形为：



最终延迟为150ms





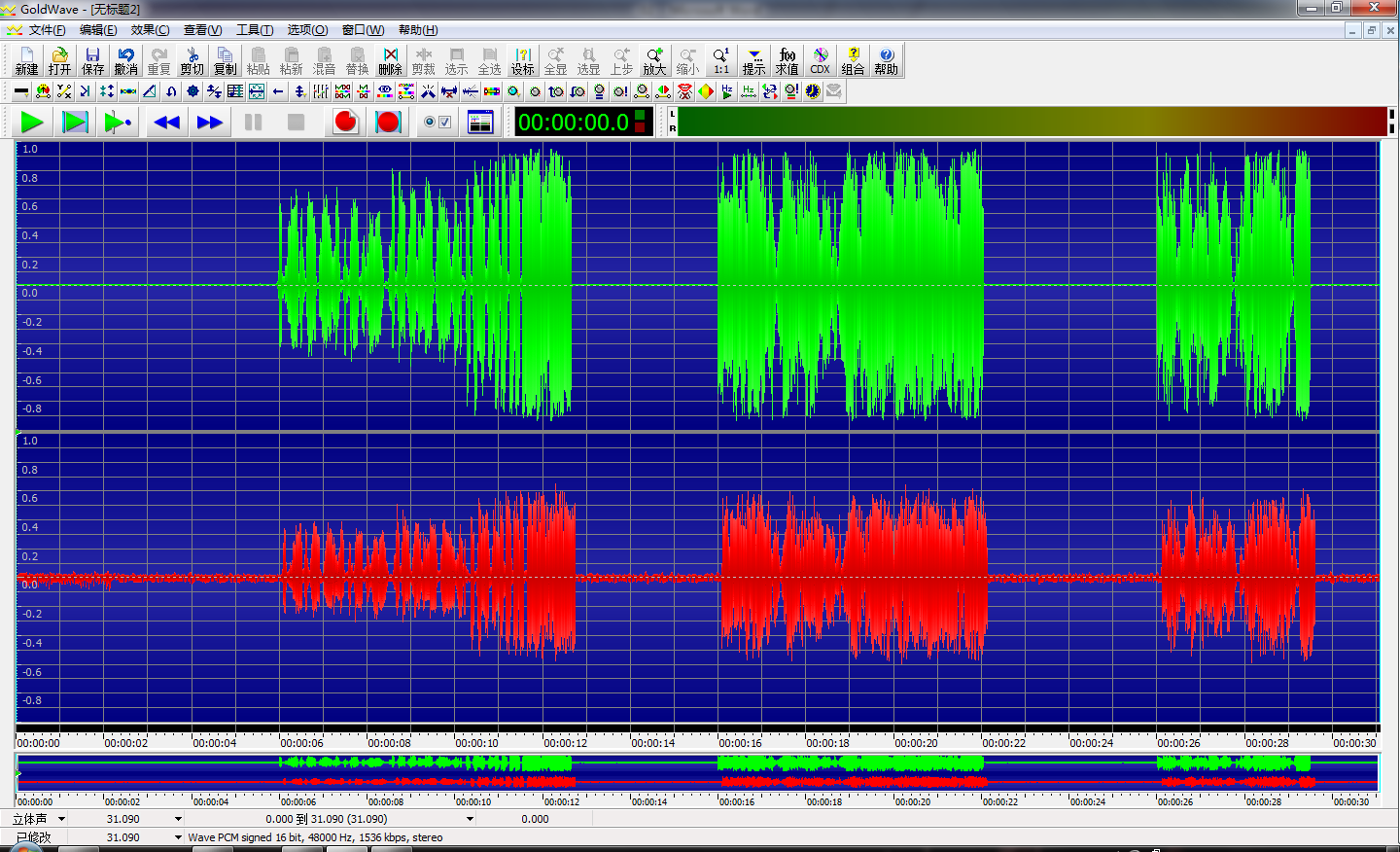
测量文件最后为test6.wav

#### 回返延迟测量后有如下的关系

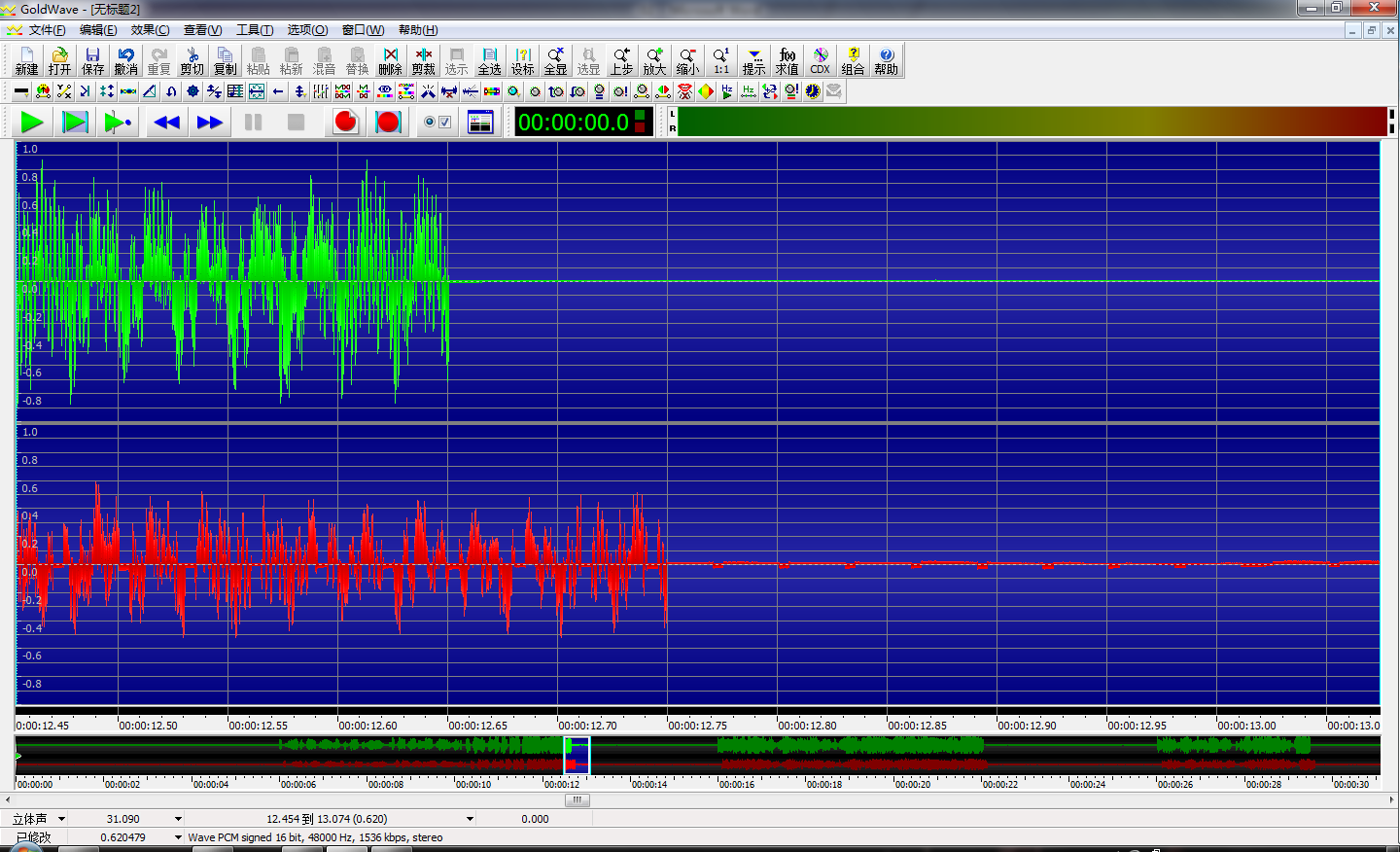
#### 对于缓存块个数的研究

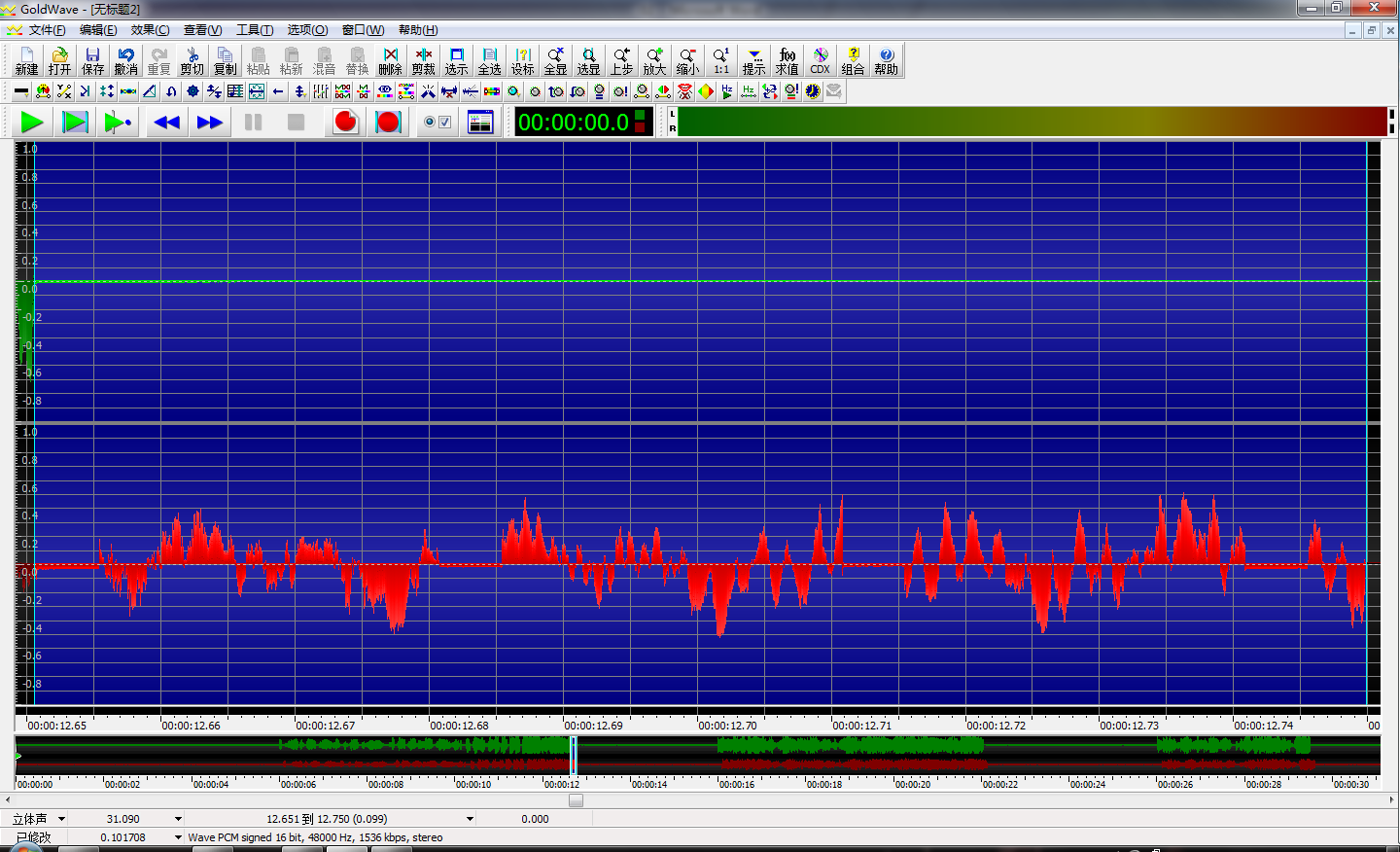
在缺省情况下，缓存块格式为4块。

改成1块之后，音质明显受损。之后进行具体的测量，看在1个缓存块的情况小到底延迟的表现能有多少。



延迟有所改善，但是音质损失严重







延迟为99ms。

文件为test7.wav。

## 遗留问题

1. 还未找到控制数据包延迟的相关方法；
2. 延迟测量方法精度是否能够接受；
3. 如果精度能够接受，如何进一步降低延迟。