**项目二**

515082910001 崔子君（报告，界面优化）

515082910004 朱璟然（绘制产生波形代码，测试）

5150829100010 肖潮（频率逻辑代码，测试）

一、程序开发及运行环境

程序开发环境：Visual Studio 2017

程序语言：C#

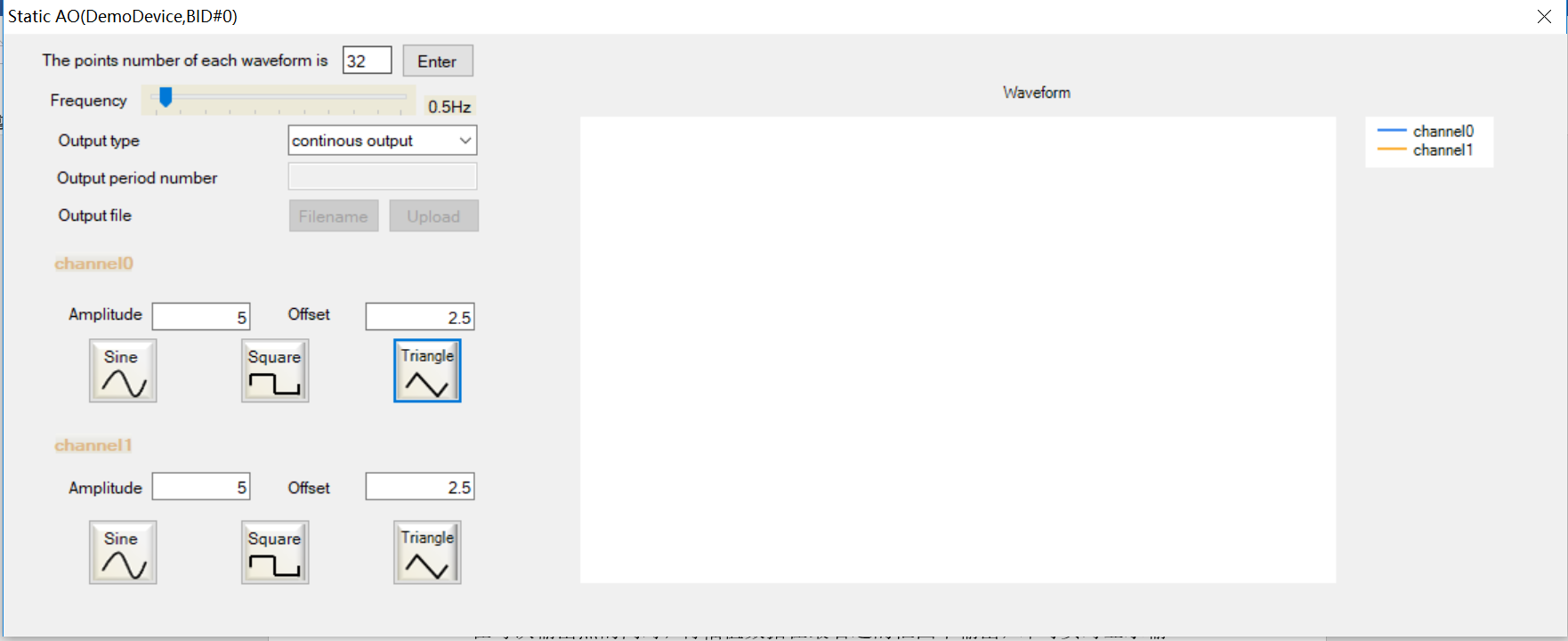
目标框架：.NET Framework 4.0

二、程序开发逻辑

按照项目要求依次阐述程序开发的逻辑。

1、编写用户界面供显示和用户交互。

程序界面如下：



从界面中可以看出，我们设计的为左半部分控制输出的操作端，右半部分则是用以显示输出波形和波形值。

控制端包括一个波形上的点数，波形频率值，输出类型，输出周期数，还有每个通道的生成波形形状和参数。

显示端显示左边控制端输入参数的波形情况。

2、生成输出波形的数据（从已有文件中读取或手动编写）。

在每次输出波形的同时，将幅值和直流偏移等数据在右边的框图中与波形同时输出，即可实时显示输出波形数据。

3、在界面上可显示输出波形。

手动编写：根据控制端的参数设定，通过描点计算，绘制出相应的静态波形图。

4、可改变输出波形的频率和周期输出点数。

通过拉动Frequency轴和手动输入不同的周期输出点数，可改变输出波形的频率和周期输出点数，这是通过改变内部计时函数的周期来控制循环时间来实现这个功能。

5、可单次固定数量的数据输出，也可不间断地输出；

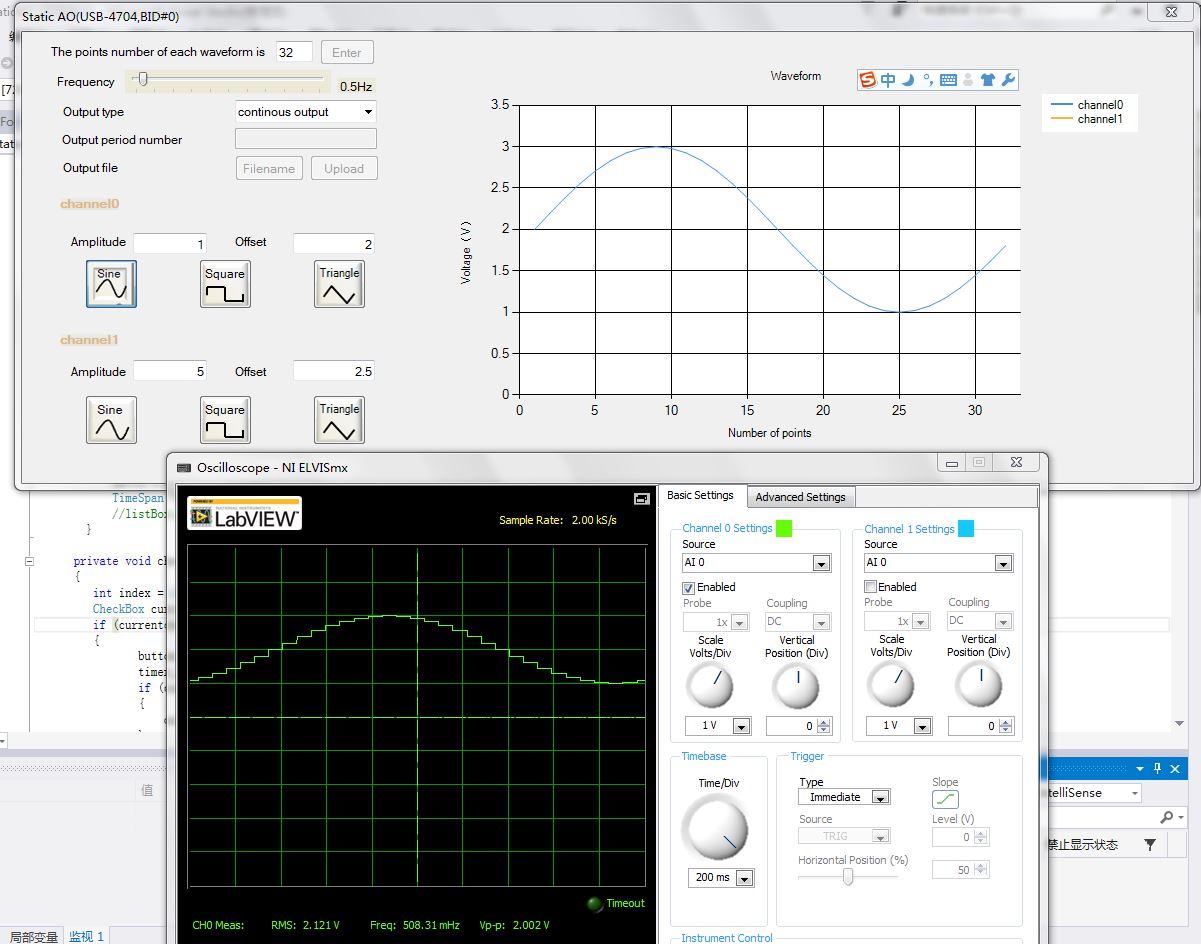
在continuous output连续输出模式下，用户单击波形按钮后开始输出，再次单击停止输出。

在assigned output period固定输出周期模式下，用户可输入周期，在输出相应周期图形后自动停止输出。

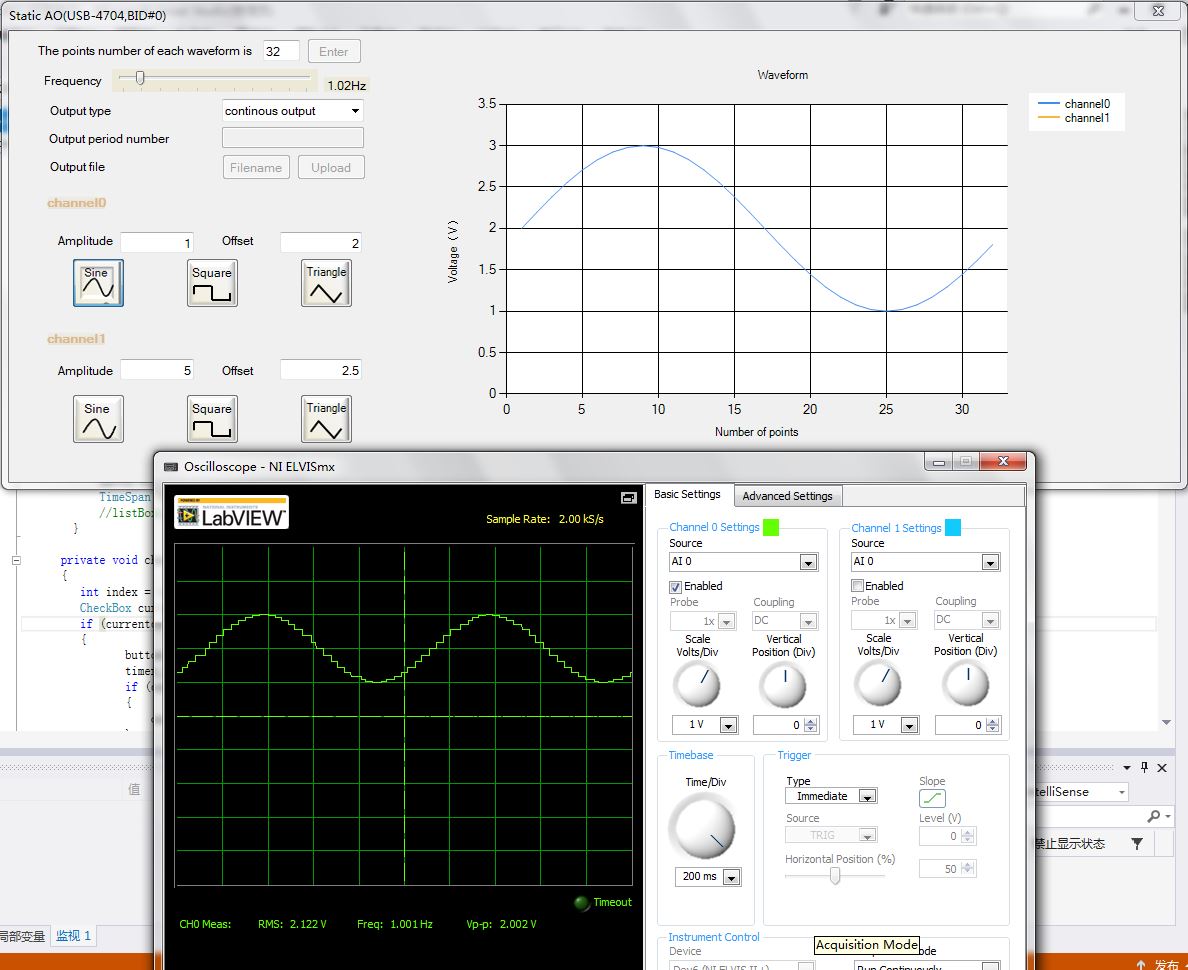
6、可开始、停止和继续输出信号。

通过start/pause按钮的切换，来暂停与继续信号的输出。

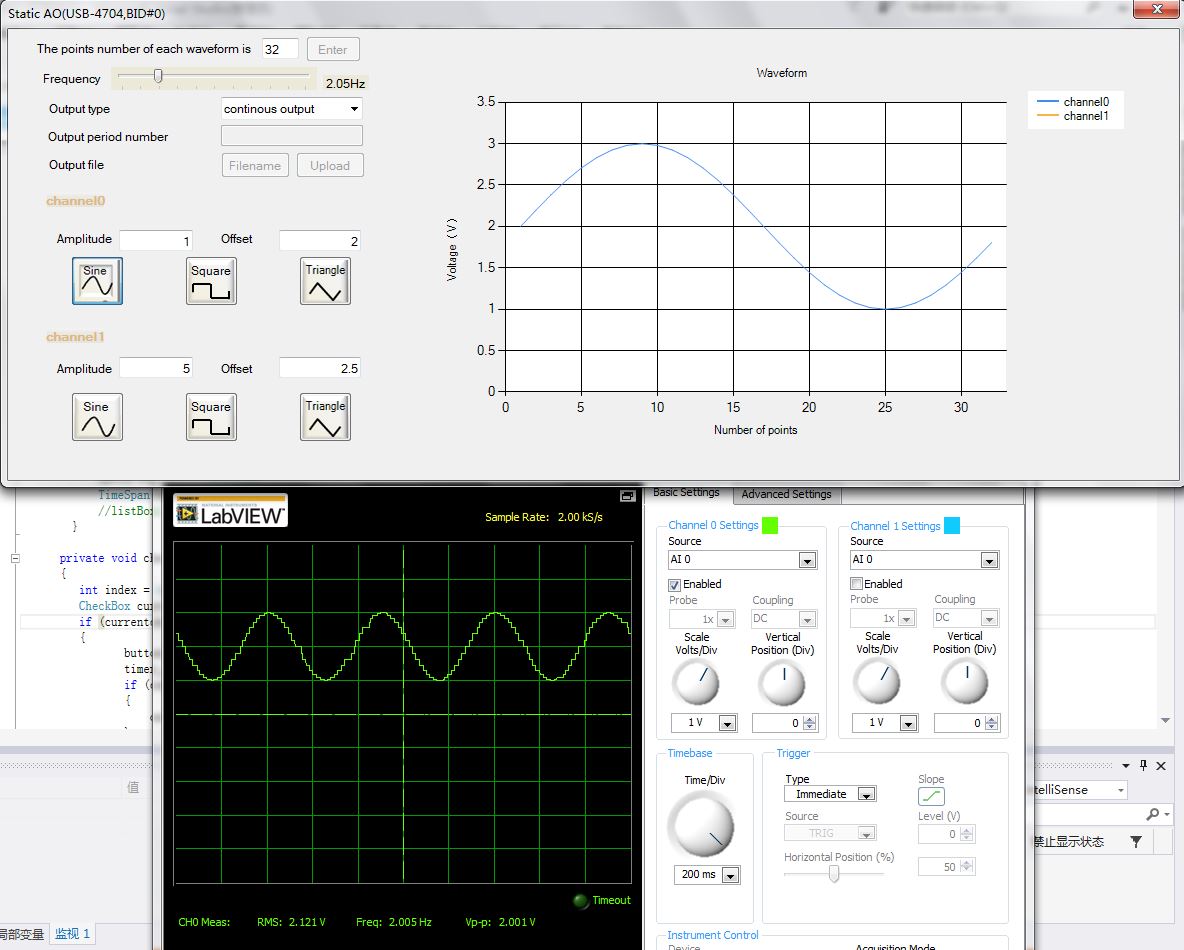
三、通过NI ELVIS II+采集输出信号，对比输出波形和采集波形，并记录结果。



F = 0.5Hz.



F = 1Hz.



F = 2Hz.

从上述的输出波形和采集波形的对比图来看，采集波形的效果总体符合输出波形的特征，数据也匹配。采集到的波的频率成分符合发生波形的调控参数。

四、测试中存在问题及解决方案

1、当输出信号频率较高时，显示的波形频率不准确。

由于信号内部计时器的误差，在输出信号频率和周期内信号点数较高时，即需要更高的采样频率，由此产生的相对误差会增大，导致显示的波形频率会低于要求的频率。因此我们应尽可能避免计时间隔较长或者在计时间隔较短时提供修正。

2、当改变频率以及周期内点数从而间接改变time interval时，信号的输出频率会有频移导致不准确，我们一开始没有很明确的找到这之间的关系。

后来在老师的指导下，我们利用C#的时钟函数，对信号的写入过程进行了计时，并将此过程进行了校正，从而得到了正确的频率值。

3、计算机信号只能输出0~5V的信号值，所以我们将输入信号值的大小自动调整为0~5V之间。当用户输入了不同的offset和magnitude的时候，我们通过内置函数自动调整信号值。

五、讨论与思考

1、如何针对不同频率的信号设置合适的周期输出点数，并分析设置周期输出点数时考虑的因素。

经过测试，周期内信号点数为32，信号频率为2Hz及以下时输出准确，若想达到更高的频率，则需要减少周期内信号点数。但同时要保证每周期至少有10个点，这样才能使波形较为完整。

2、分析 USB-4704的模拟输出功能可输出信号的频率范围，若输出信号在该范围外，会出现哪些问题，并探讨可能的解决方案

周期内信号点数为32时，USB-4704 的模拟输出功能可输出信号应小于2Hz。当输出信号高于10Hz时，显示的波形频率不准确，这与采样频率的大小也有关，此时应该减少周期内信号点数或减少输出信号频率。