

项目二

小组成员: 陈宇 516021910399

李萌 516021910153

[要求]

- 1、利用 DAQ Navi SDK, 选择一门语言进行编程, 程序须实现以下功能:
 - (1) 编写用户界面供显示和用户交互;
 - (2) 生成输出波形的数据(从已有文件中读取或手动编写);
 - (3) 在界面上可显示输出波形;
 - (4) 可改变输出波形的频率和周期输出点数;
 - (5) 可单次固定数量的数据输出, 也可不间断地输出;
 - (6) 可开始、停止和继续输出信号;
- 2、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;
- 3、完成程序编码后, 对程序进行测试和调试, 记录所遇到的问题及如何处理;
- 4、通过测试后, 发布可执行文件, 并在第三方 PC 上进行功能展示;
- 5、项目报告包括以下内容:
 - (1) 程序开发逻辑;
 - (2) 总结测试中出现的问题及其解决方案;
 - (3) 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;
 - (4) 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;

[报告目录]

第一部分: ELVIS II+、USB-4704 及 DAQ Navi 实验结果与讨论

第二部分: 利用 DAQ Navi SDK 编写用户界面

I、程序开发逻辑及功能说明

II、测试中遇到的问题及其解决方法

III、讨论

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;
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

第一部分: ELVIS II+、USB-4704 及 DAQ Navi 实验结果与讨论

实验内容:

使用 MATLAB 程序产生信号, 并通过 USB-4704 的模拟输入端, ELVIS II+ 和 DAQ Navi 采集所产生信号

1、使用 MATLAB 产生正弦波信号

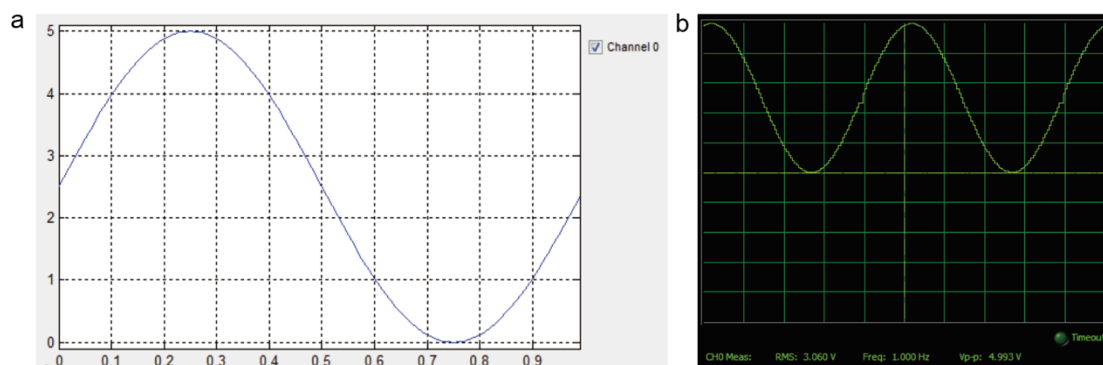


图 1-1 a) 输出信号:正弦波; 频率: 1Hz; 每周期点数: 100; 幅值: 0~5V ; 偏置: 2.5V

b)采集信号:正弦波; 测得频率: 1Hz; V_{p-p} :4.993V

2、使用 MATLAB 产生三角波信号

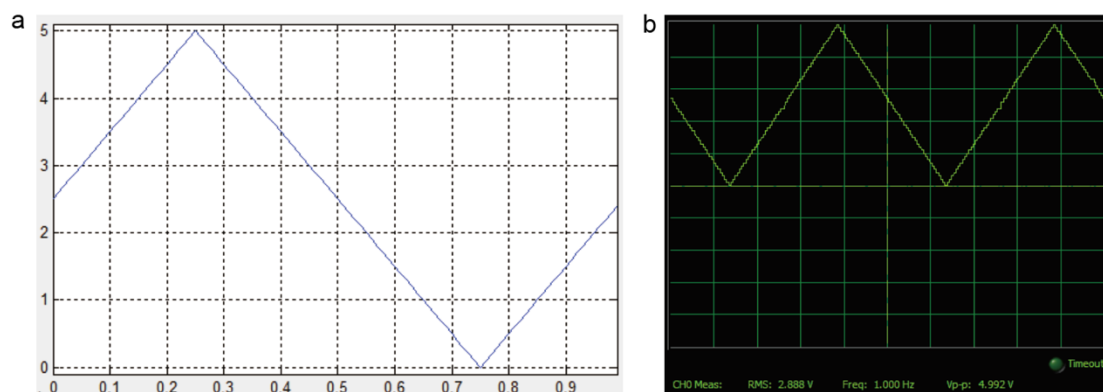


图 1-2 a) 输出信号: 三角波; 频率: 1Hz; 每周期点数: 100; 幅值: 0~5V ; 偏置: 2.5V

b)采集信号: 三角波; 测得频率: 1Hz; V_{p-p} :4.992V

3、使用 MATLAB 产生方波信号

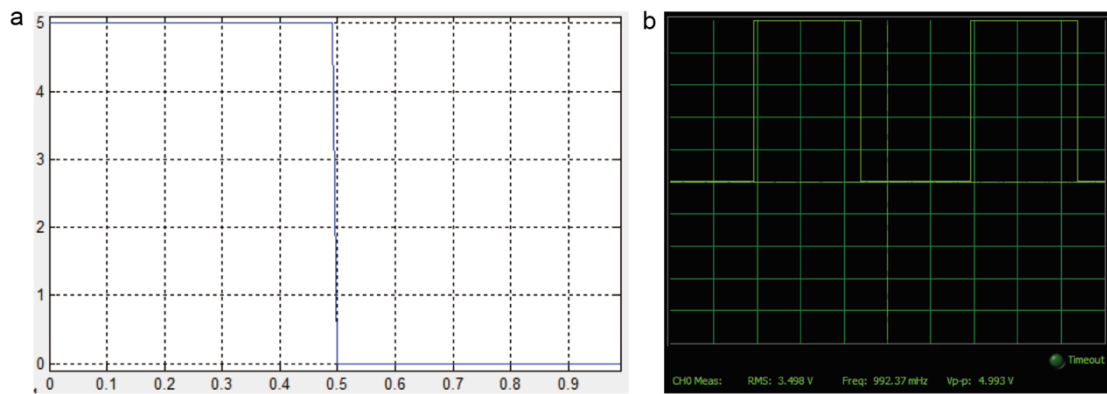


图 1-3 a) 输出信号:方波; 频率: 1Hz; 每周期点数: 100; 幅值: 0~5V; 偏置: 2.5V
b)采集信号:方波; 测得频率: 992.37mHz; Vp-p:4.993V

4、使用 MATLAB 产生直流信号

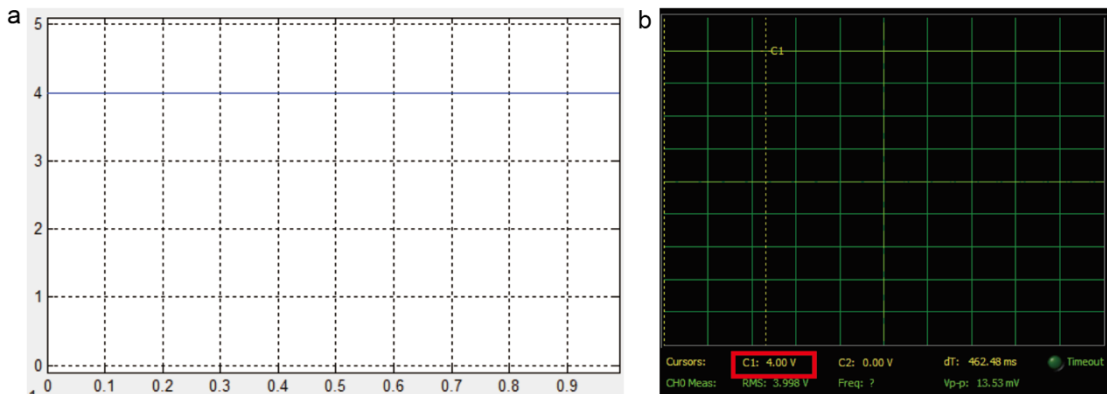


图 1-4 a) 输出信号:直流信号; 每周期点数: 100; 幅值: 4V
b)采集信号:直流信号; 幅值: 4V

5、使用 MATLAB 产生固定点数信号

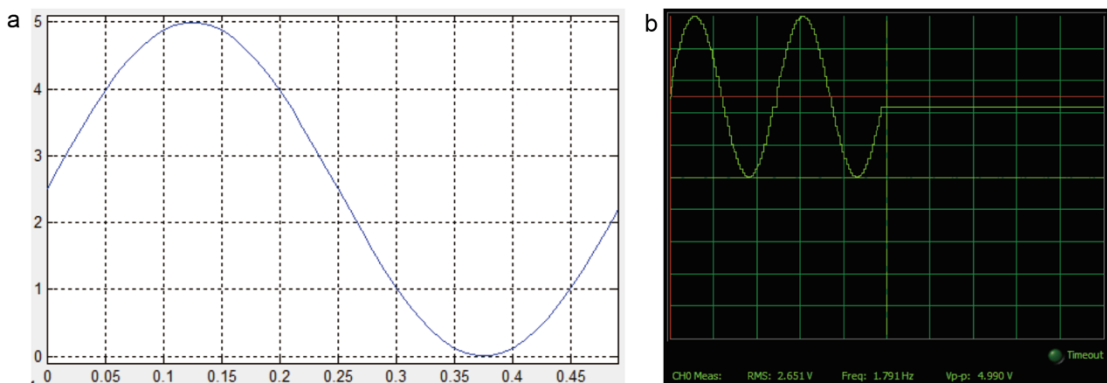


图 1-5 a) 输出信号:正弦波; 频率: 2Hz; 每周期点数: 50; 输出点数: 100; 幅值: 0~5V; 偏置: 2.5V
b)采集信号:正弦波; 测得频率: 1.791Hz; Vp-p:4.990V

6、使用 MATLAB 读取文件，产生对应信号

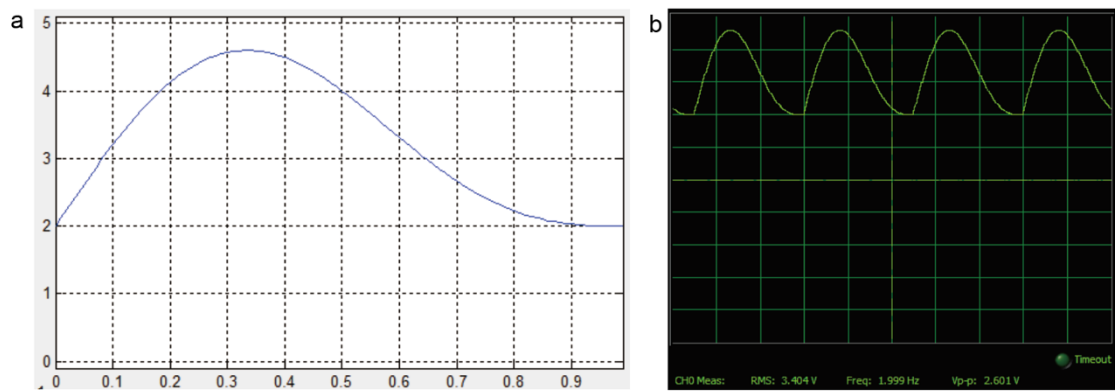


图 1-6 a) 输出信号, 频率: 2Hz;
b) 采集信号, 频率: 2Hz

第二部分: 利用 DAQ Navi SDK 编写用户界面

I、程序开发逻辑及功能说明

界面组成说明:

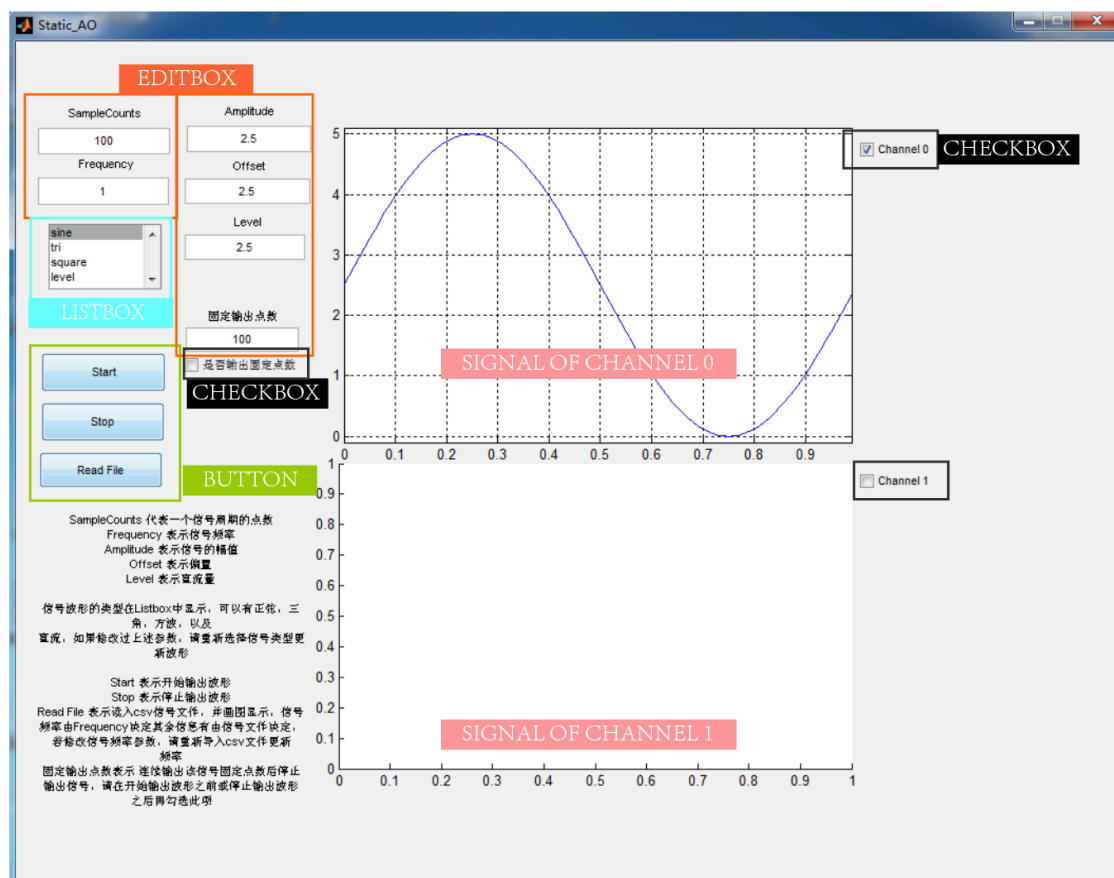


图 2-1 界面

EDIT BOX:

- 1) SampleCounts: 每周期点数
- 2) Frequency: 信号频率 (Hz)
- 3) Amplitude: 幅值范围 (V)
- 4) Offset: 偏置 (V)
- 5) Level: 直流信号的幅值 (V)
- 6) 固定输出点数: 设置输出的总点数

LISTBOX:

选择产生的信号, 可选正弦波、三角波、方波和直流信号; 点击后在 axes 画出信号波形

II、测试中遇到的问题及其解决方法

1. 设定频率和采集到的信号频率不一致

比如设定参数为 100point/period，频率为 3Hz，理论上采集到的数据应该频率为 3Hz；但实际情况是 3.33Hz，原因是 timer 的 Period 参数设定为 $1/(\text{周期点数} \times \text{频率})$ ， $\text{period} = 1/(100 \times 3) = 0.00333\text{s}$ ，而 matlab 中 timer 指挥保留到 3 位小数，即 $\text{period} = 0.003\text{s}$ ，所以得到的实际信号频率为 $1/\text{period} = 3.33\text{Hz}$

解决方法：

1) 在一个 timer callback function 中输出多个点，增加 timer 的 period，一定程度地减少信号的失真。

2) 设定计数器补偿，事先计算得到需要补偿的频率值，利用计时器定时触发额外的 timer callback function，从而补偿缺失的频率。

2. 测定固定输出频率功能是否正常运行

利用 elvis oscillation 中 Run Once 功能，再结合 trigger，type 设定为 edge，设定触发的电压值为信号启示电压，通过比较采集到的波形周期数跟设定点数之间的关系，即可证明功能是否正常运行。

III、讨论

1. 如何针对不同频率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

USB-4704 的 InstantAI 每秒最大输出点数为 1000points/s；而在 matlab 程序中，一次 timer callback function 就相当于输出一个点，也就是 1s 内最多执行 1000 次 timer callback function，其 period 最小值为 0.001s；根据 timer 的 period 设定方法， $\text{period} = 1/(\text{周期点数} \times \text{频率})$ ，由此可知信号设定的参数 周期点数*频率需要 <1000 ；

当然 1000 的情况是非常理想化，不考虑其他因素的，实际效果并不能达到 1000，测试结果显示，在 matalb 平台上运行程序时，周期点数*频率需要 <400 才能保证程序正常运行且信号不发生频率的失真。

设定周期输出点数需要考虑：

1) 波形的形状，如果需要比较平滑，比较接近理想情况的波形，那么输出点数就要比较高，从我们的观察结果来看，一个周期至少需要 50 个点才能保证波形的完整。

2) 信号采集时的采样率设定，如果信号采集的采样率比较高，那么周期点数也需要比较高，否则采集到的波形会出现大量的阶梯状信号，会出现波形的失真。

2. 分析输出信号频率范围

通过 tic/toc 测试可得，输出一个点电压需要的时间在 0.001-0.002s 之间，即输出信号理论最大频率为 500Hz；

但是我们设置 timer 的周期是 $1/(\text{周期点数} \times \text{频率})$ ，为了保证信号波形不失真，周期点数需要 ≥ 50 ，并且为了保证程序能够响应 Stop 按钮的命令（防止 matlab 所有的线程被占用），输出波形的频率需要 $\leq 8\text{Hz}$

如果输出信号在频率范围之外，可能会出现：

- 1) matlab 无法响应 stop 命令，timer 不能停止
- 2) 输出的信号频率不正确

可能的解决方法：

1) 修改 timer 的设定参数“busymode”，由“drop”改为“queue”，能够少量地补偿缺失的信号频率；

2) 修改 timer 的“TimerFcn”，即修改 timer 的 callback function，从原来每次输出一个点改为输出两个点；