

# 项目三

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，程序需实现以下功能:
  - (1)编写用户界面供显示和用户交互;
  - (2)使用数字输出(DO)实现特定频率范围(小于 50Hz)的方波输出，可单次固定时间的输出，也可不间断地输出;
  - (3)可实时改变输出方波的频率;
  - (4)可开始、停止和继续输出方波;
  - (5)使用数字输入(DI)在用户界面上实时显示设定频率的方波(或正弦波)波形，其中，1 路通道表示开始/停止输出，2 路通道表示幅度(0~3V)，5 路通道表示频率(0~31Hz);
- 2、通过 ELVIS II+采集输出的方波信号，测量方波频率，并判断是否和要求一致;
- 3、通过 ELVIS II+输出外部控制信号，观察用户界面显示的方波，并判断是否和要求一致;
- 4、完成程序编码后，对程序进行测试，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;

## [报告目录]

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

**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) 确定采用 DO 输出方波的频率范围，并给出相应数据

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

实验内容:

1. 通过 ELVIS II+采集输出的方波信号, 测量方波频率, 并判断是否和要求一致;

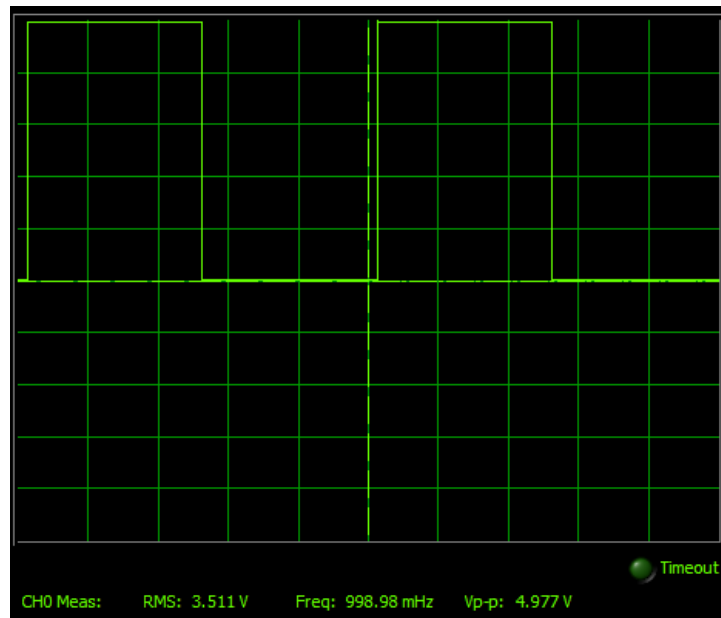


图 1-1 输入信号:方波; 频率: 1Hz; 幅值: 0~5V ; 偏置: 2.5V  
实际采集信号频率: 998.98mHz; Vp-p:4.977V

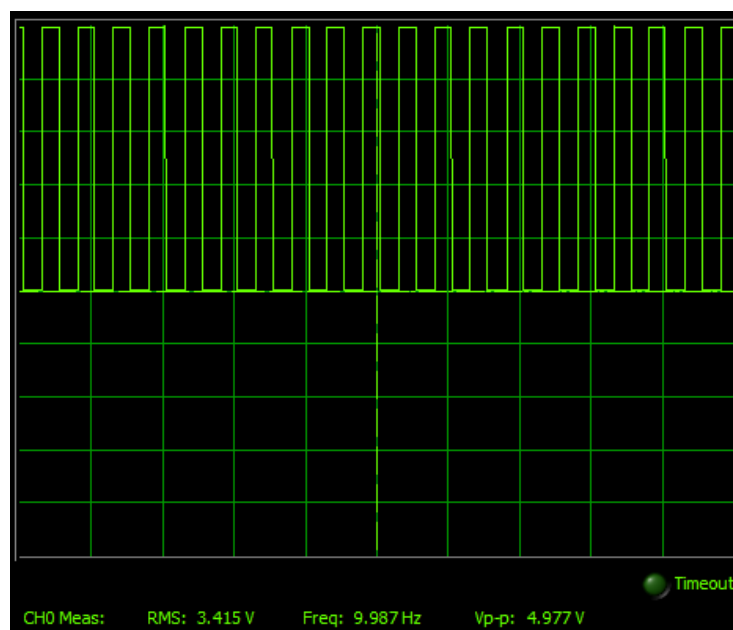


图 1-2 输入信号:方波; 频率: 10Hz; 幅值: 0~5V ; 偏置: 2.5V  
实际采集信号频率: 9.987Hz; Vp-p:4.977V

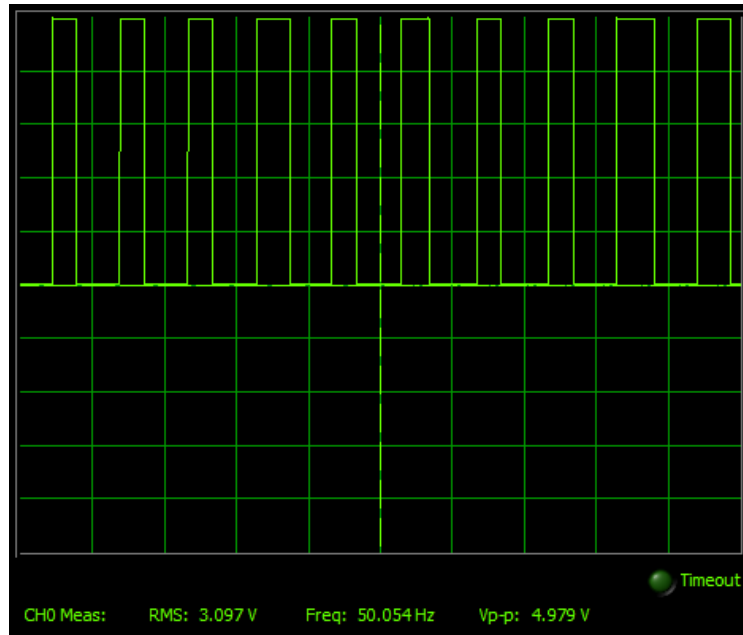


图 1-3 输入信号:方波; 频率: 50Hz; 幅值: 0~5V ; 偏置: 2.5V  
实际采集信号频率: 50.054Hz; Vp-p:4.979V

**2. 通过 ELVIS II+输出外部控制信号，观察用户界面显示的方波，并判断是否和要求一致;**

(第 0~4 路通道表示频率(0~31Hz) 第 5~6 路通道表示幅度(0~3V)，第 7 路通道表示开始/停止输出)

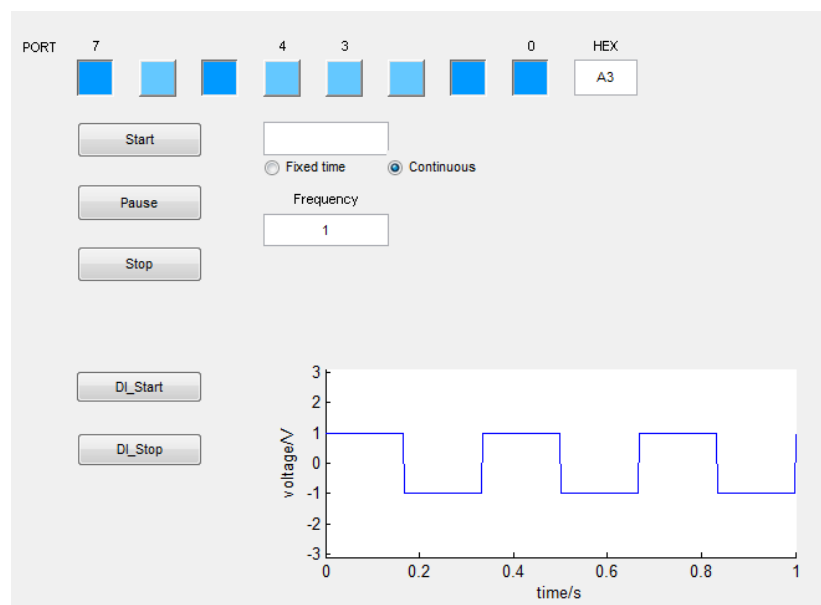


图 1-4 输出信号:方波; 频率: 3Hz; 幅值: 1V  
DI 信号频率: 一秒内约有 3 个周期, 频率约为 3Hz, 幅值为 1V

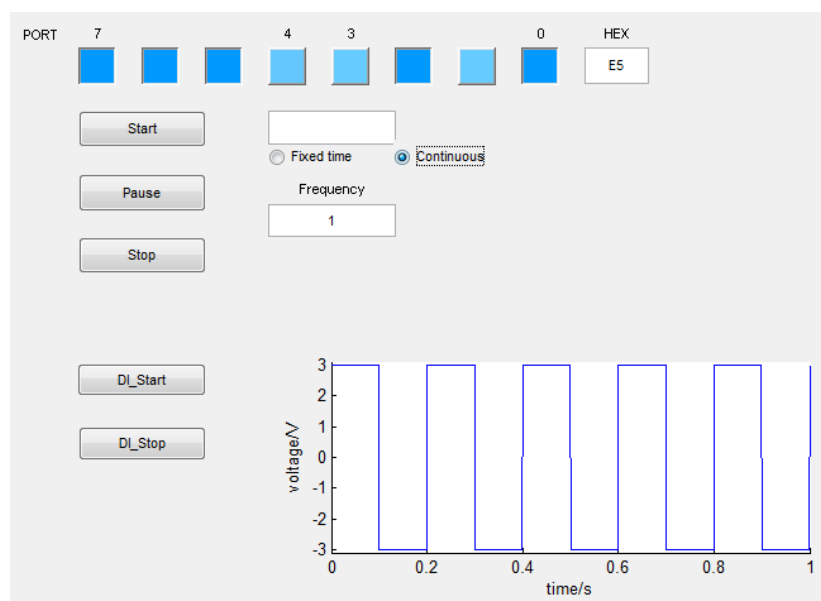


图 1-5 输出信号:方波; 频率: 5Hz; 幅值: 3V  
DI 信号频率: 一秒内约有 5 个周期, 频率约为 5Hz, 幅值为 3V

### 3. 固定时间输出信号

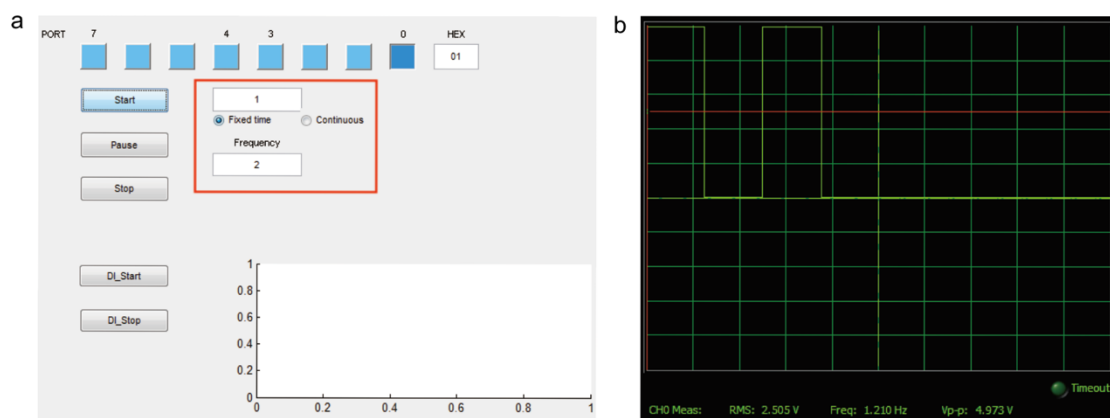


图 1-6 DO 信号: 方波; 频率: 2Hz; 固定时间: 2s  
采集信号: 持续时间 2s, 由于混杂直流信号, 频率计算为 1.210Hz

## 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:

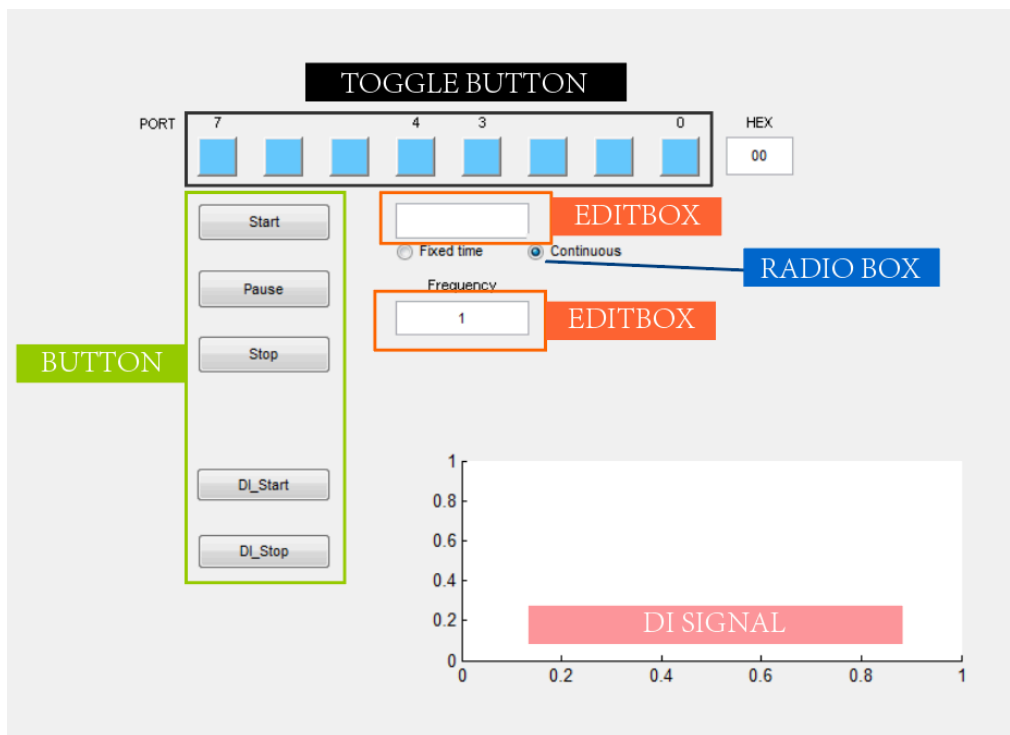


图 2-1 界面

#### EDIT BOX:

- 1) Fixed time: 信号固定输出的时间 (s)
- 2) Frequency: DO 输出信号的频率, 能够实时改变输出信号的频率

#### RADIO BOX:

- 1) Fixed time: 输出固定时间的信号
- 2) Continuous: 连续输出信号

#### BUTTON:

- 1) Start: 开始输出信号
- 2) Pause: 暂停输出信号
- 3) Stop: 停止输出信号
- 4) DI\_Start: 开始输出 DO 信号到 USB-4707, 并采集 DI 信号, 将信号波形画出

5) DI\_Stop: 停止输出 DO 信号, 停止采集 DI 信号.

### TOGGLE BUTTON:

能够改变 DO 8 个通道的输出值

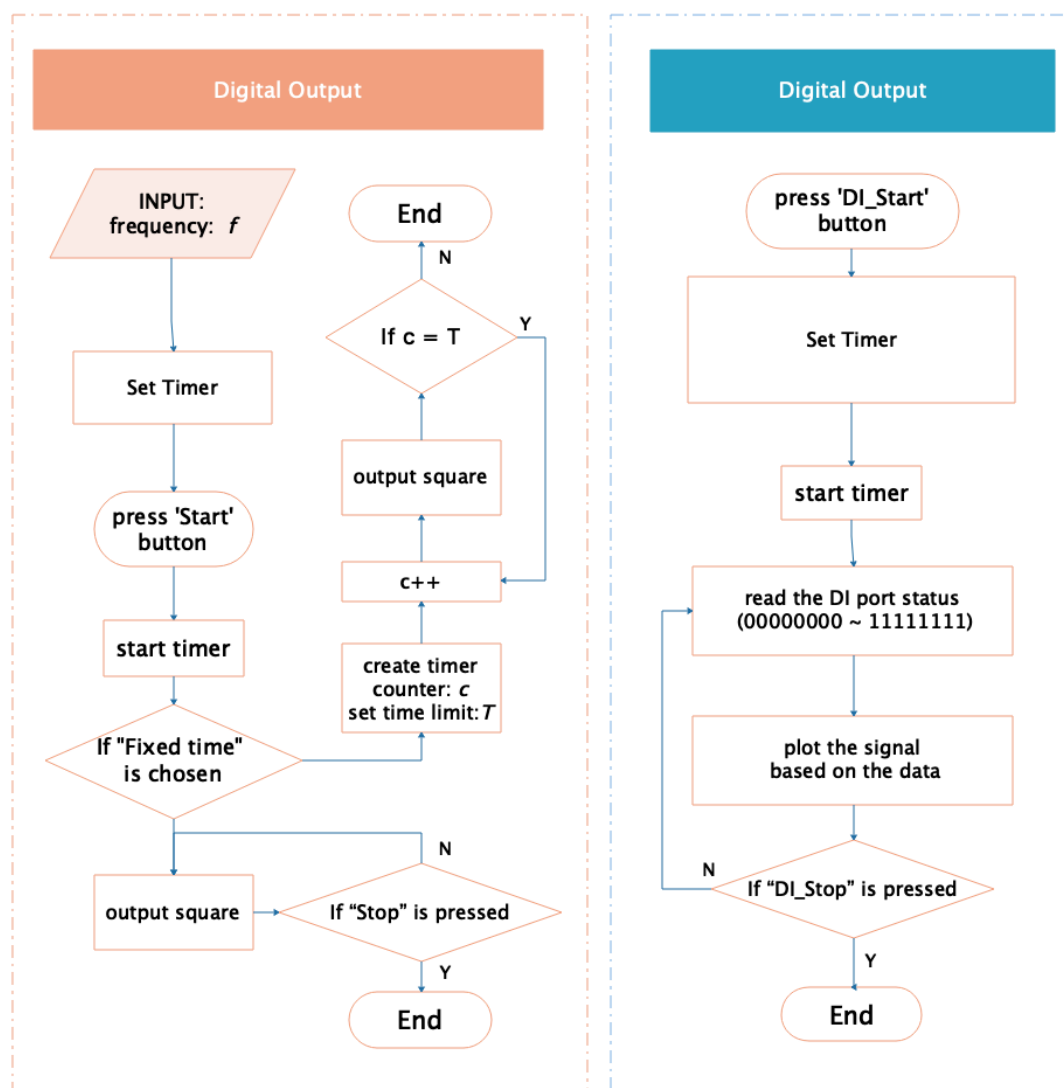
为了测试, 我们把 DI 的 8 个通道接到了 DO 上, 用 DO 进行编码, DI 进行解码

第 0~4 路通道表示频率(0~31Hz);

第 5~6 路通道表示幅度(0~3V);

第 7 路通道表示开始/停止输出.

程序开发逻辑图:



Notes:  
1、'Stop' is actually an interrupt.

图 2-2 程序开发逻辑图

## II、测试中遇到的问题及其解决方法

### 1) Timer Callback Function 的运行时间会变化

我们的程序中，每一次 timer callback function 就输出一个电压点；

使用 tic/toc 测量一次 timer callback function 所需要的时间，发现这个时间会变化；timer 的触发的 frequency 为 1Hz 时，测得的时间是 7ms，触发的 frequency 为 50Hz 时，测得的时间是 4ms；

在不同的 CPU 占用率以及线程占用率的情况下，Timer Callback Fun 所需要的时间不一样，使得理论最大输出方波频率的计算有一定的疑惑。

### 2) DO 方波输出频率超过 50Hz 后会出现界面点击无响应的情况

DO 方波输出频率为 50Hz，timer 的 period 为 20ms，而根据 tic/toc 测得的数据，timer callback function 运行只需要 4ms，理论上线程并没有占满，但还是出现了界面点击无响应的情况；

跟老师讨论后，怀疑是因为 matlab 界面上点击按钮的判定条件为按下—保持—松开，整个过程所需要的时间必然超过 20ms，而只要其中一环因为 timer 的原因没有被检测到，那么点击的过程就相当于失效了；也就是说当 timer 的 period 太短的时候，GUI 界面是有可能没有反映的。

### 3) DO 输出过程中修改输出方波频率的方法

由于 Timer 在运行中是不能修改 period 的，所以如果想随时修改 timer 的 period，需要读取 timer 的“running”参数，如果该参数为 on，则需要先暂停 timer，再进行修改，否则，直接修改 period 即可。

### 4) 关于 handles 和 hObject 在函数中的参数传递问题

IMHO，即使是在同一个 GUI callback 函数中，只要使用过 Guidata (hobject, handles) 这条语句，handles 就会发生改变；这时候，就不要用 handles 去传递参数，而是用 hObject，然后在函数中使用 handles=guidata (hobject) 来读取修改后的 handles 即可。

## III、讨论

### 1) 确定采用 DO 输出方波的频率范围，并给出相应数据

根据 tic/toc 测得的数据，timer 的触发频率在 1-50Hz 范围内变化时，Timer callback function 的运行时间从 7ms 变到了 4ms，但是我并不能测到这段时间的极值是多少，因为触发频率大于 50Hz 时，程序就已经无法响应了。

所以根据所测得的数据，理论上能够输出的最大频率是 250Hz，如果 timer callback function 的运行时间为 4ms 的话。