

项目四

小组成员: 陈宇 516021910399

李萌 516021910153

[要求]

1、利用 DAQ Navi SDK 并结合 ELVIS II+, 选择一门语言进行编程, 程序需实现以下功能:

(1)集成项目一、二和三的功能模块, 形成一个统一界面供用户交互;

(2)可独立完成模拟输入并显示、模拟输出和方波输出的功能;

2、通过 ELVIS II+和 USB-4704 进行功能验证;

3、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;

4、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)界面使用说明(Readme);

5、自行录制三分钟短片, 介绍如何使用最终项目的开发程序;

[报告目录]

第一部分: 程序测试

第二部分: 利用 DAQ Navi SDK 编写用户界面

I、程序开发逻辑

II、界面使用说明(Readme)

III、测试中出现的问题及其解决方案

第一部分：程序测试

I、模拟输入并显示

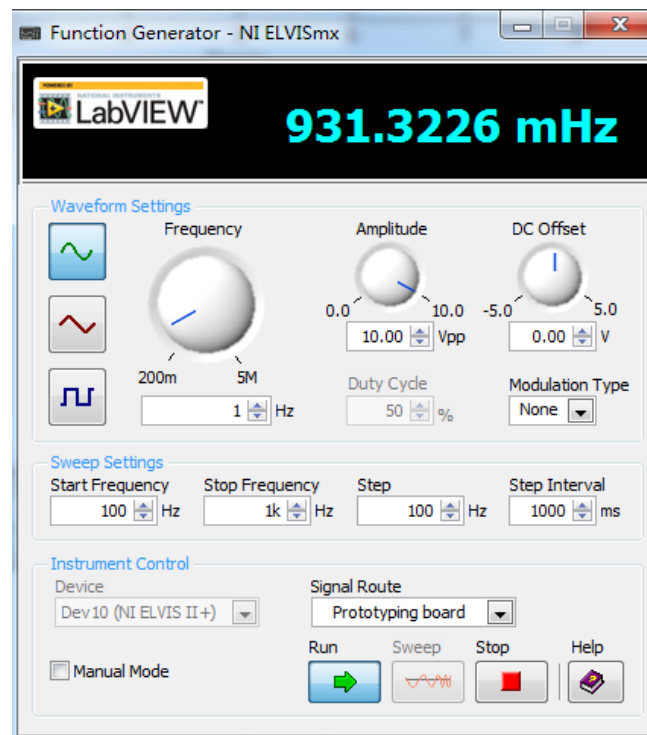


图 1-1 输出信号:正弦波; 频率: 1Hz; V_{p-p} :10V ;偏置: 0V‘

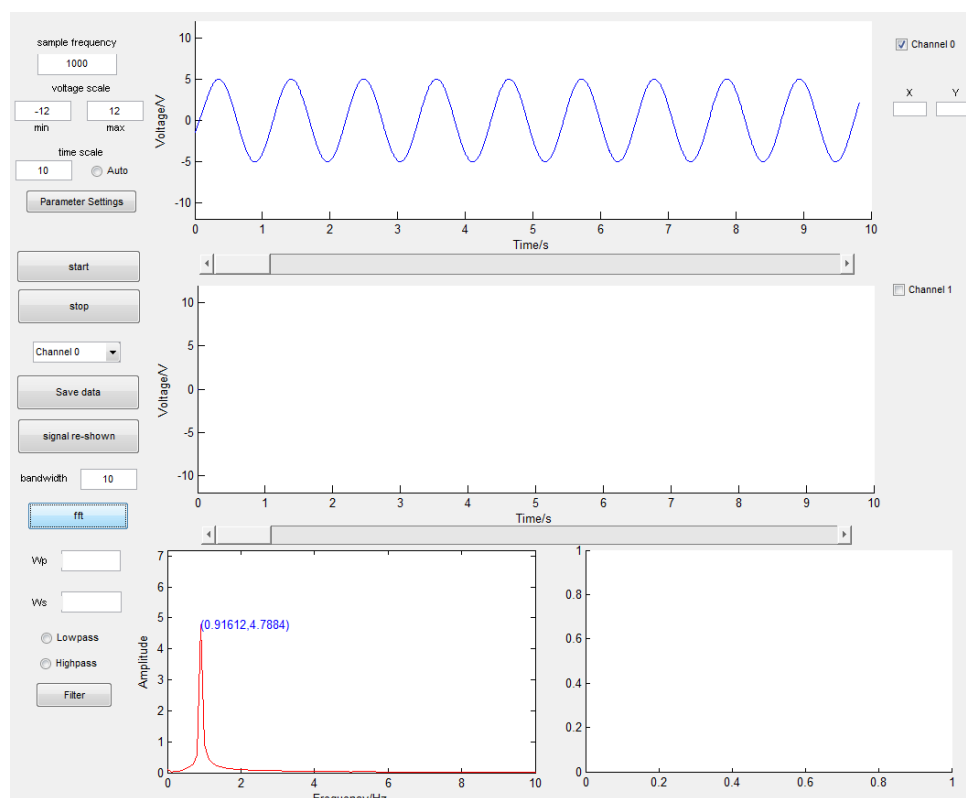


图 1-2 采集信号:正弦波; 测得频率:~1Hz; V_{p-p} :~10V

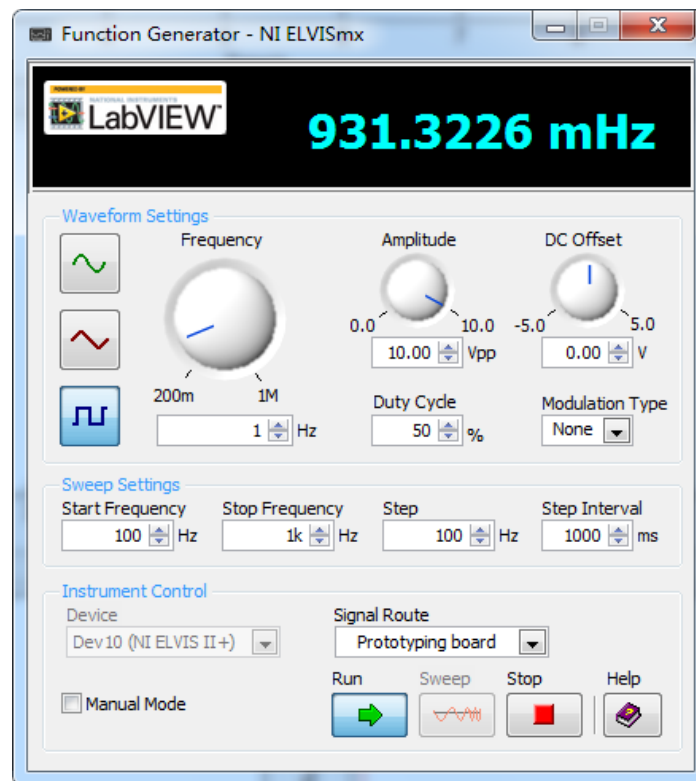


图 1-3 输出信号:方波; 频率: 1Hz; V_{p-p} :10V; 偏置: 0V

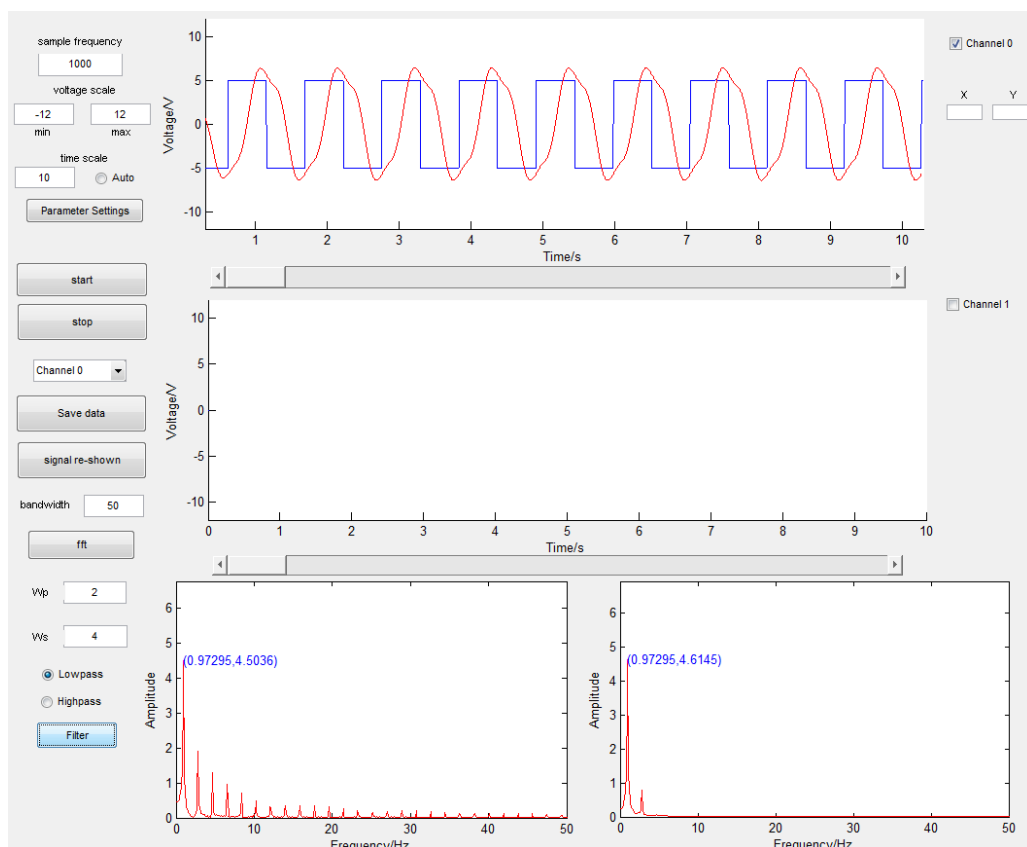


图 1-4 采集信号:方波; 测得主频率:~1Hz; V_{p-p} :~10V

II、模拟输出

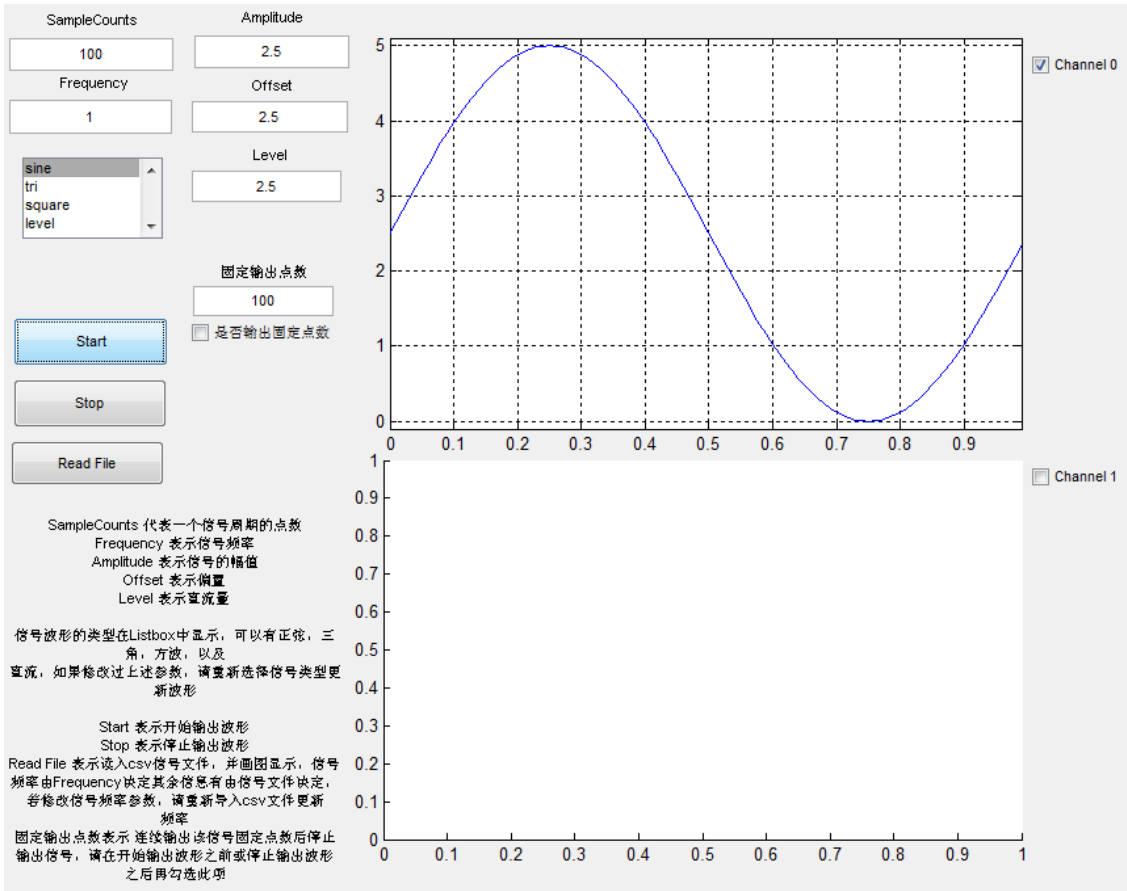


图 1-5 输出信号:正弦波; 频率: 1Hz; 每周期点数:100; 幅值: 0~5V; 偏置: 2.5V

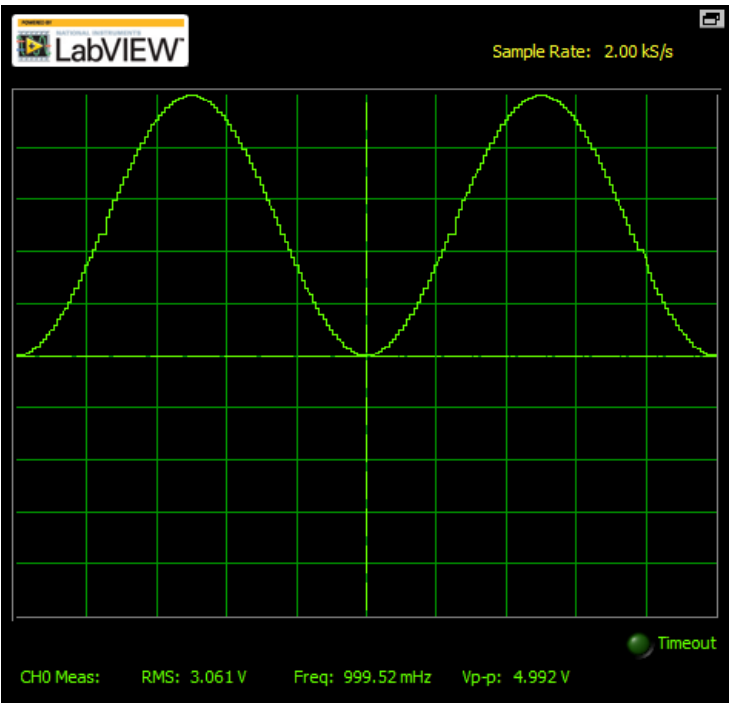


图 1-6 采集信号:正弦波; 测得频率:999.53mHz; Vp-p:4.992V

III、方波输出

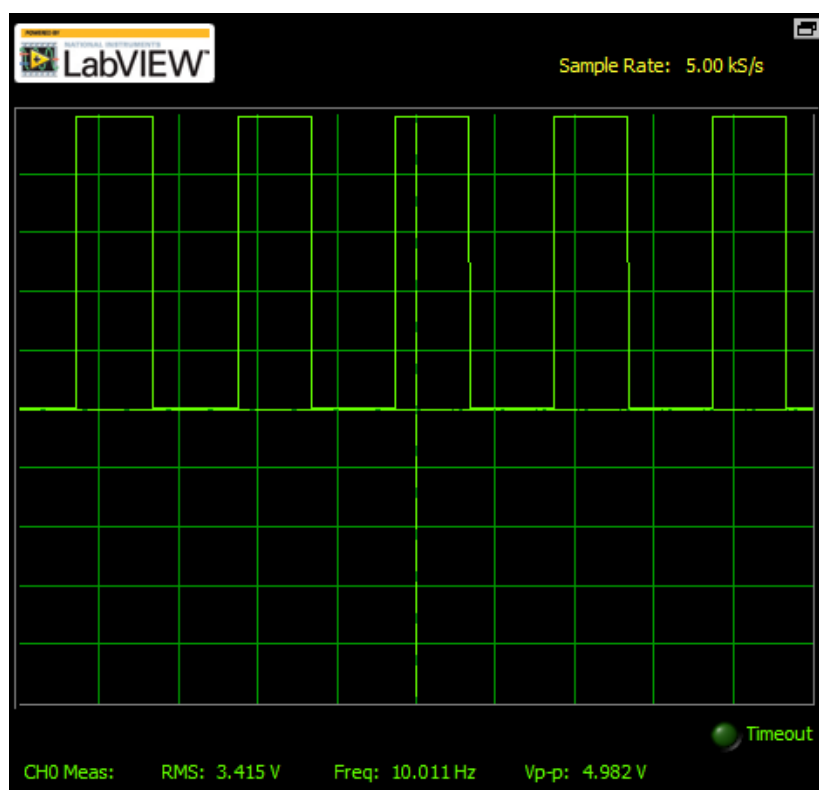


图 1-7 DO 测试. 输出数字信号: 频率: 1Hz
采集信号:方波; 测得频率:999.53mHz; Vp-p:4.982V

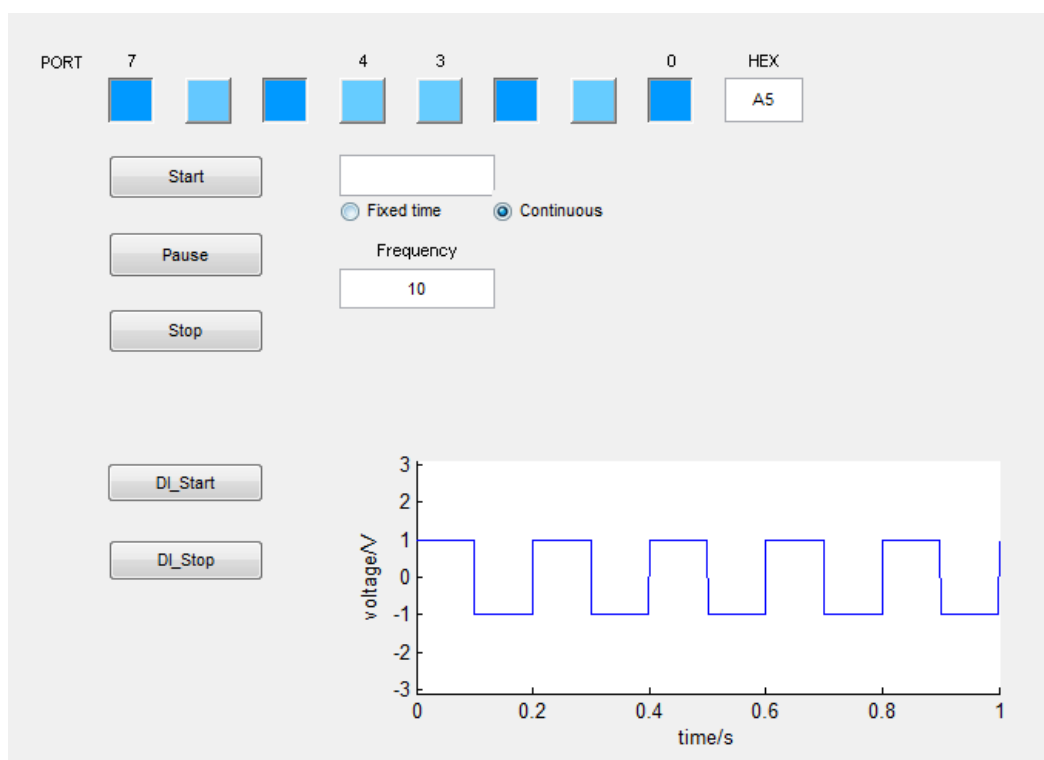


图 1-8 DI 测试. DI 检测到通道 7 开启, 幅值为 01 对应 1V, 频率为 00101 对应 5Hz, 画出对应信号如上图所示

第二部分： 利用 DAQ Navi SDK 编写用户界面

I、程序开发逻辑

程序开发逻辑图：

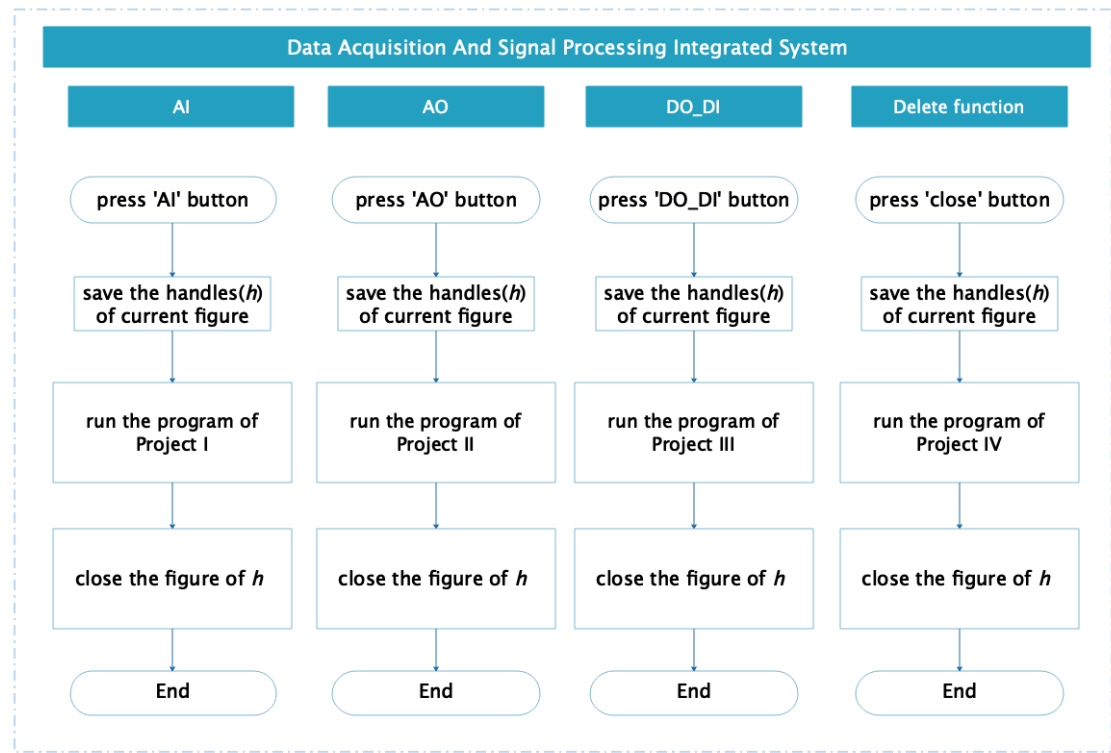


图 2-1 程序开发逻辑图

II、界面使用说明（Readme）

功能：

结合 ELVIS II+ 和研华 USB-4704，实现信号的模拟输入，模拟输出，数字输入与数字输出。

运行环境：

- 1) 在 Matlab 上运行 .m 文件（prj4.m）
- 2) 在安装了研华驱动后双击运行 .exe 文件

1.主界面

有三个按钮，从左到右依次为 AI、AO 和 DI_DO 模块，点击可进入相应模块接线：

1) AI 模块

用 ELVIS Functional Generator 产生波形，连接 EVIS 板上的 FCNG 口和 USB-4704 的 AI 0 或 AI 1 口（双通道），即可测试功能

2) AO 模块

用 ELVIS Oscilloscope 接收信号并显示，连接 EVIS 板上的 AI 口和 USB-4704 的 AO 0 或 AO 1（双通道），即可测试功能

3) DI_DO 模块

用 ELVIS Oscilloscope 接收信号并显示，连接连接 EVIS 板上的 AI 口和 USB-4704 的 DO 0（以单通道为例子，换成其他 DO 口亦可），可测试 DO 功能。

将 DO 口与 DI 口相连，通过界面修改 DO 的输出值，即可测试 DI 功能。



图 2-2 主界面

2. AI

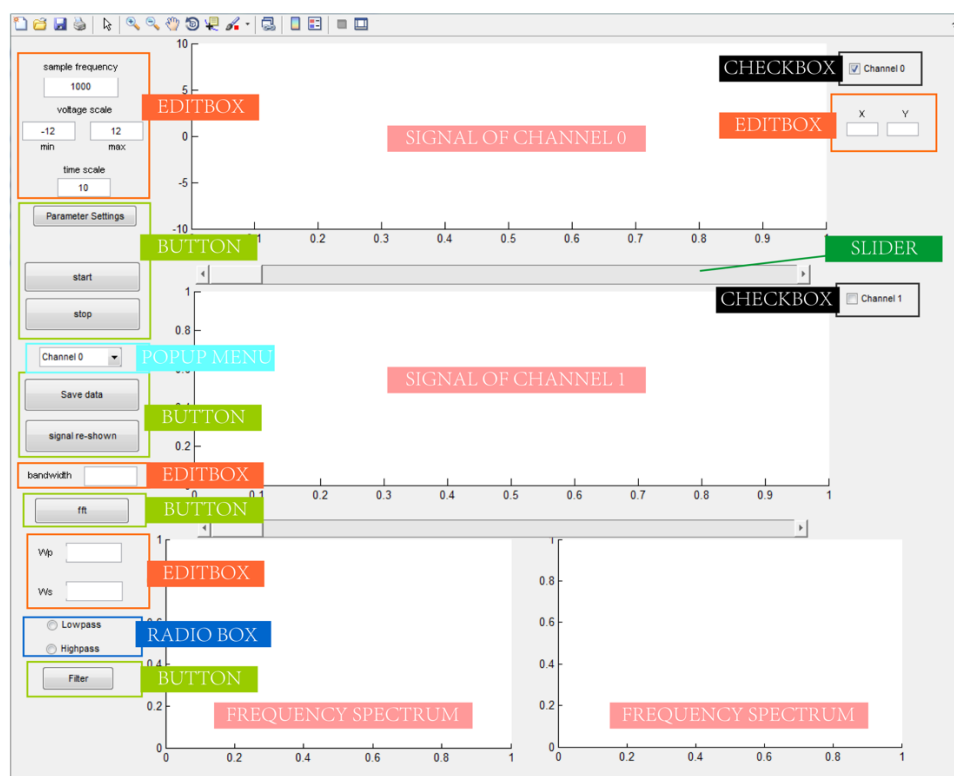


图 2-3 AI 界面

可以实现模拟输入，fft，滤波等功能。

可以实现两个通道的模拟输入，选择通道的 checkbox 即可开启相应通道。

1、模拟输入

1) 设定采样率 (sample frequency) (1-1000Hz)，电压范围 (V)和时间轴范围 (s)，点击 Parameter Settings

2) 点击 start，接收信号。信号波形将在对应通道的绘图区画出，可在接收过程中动态修改电压和时间的显示范围。

3) 点击 stop，停止采样。

*注：信号频率建议不超过 10Hz，采样率至少为信号频率的 10 倍

2、fft

1) 点击 fft，将模拟输入获得的信号进行 fft，并将频谱在界面左下方绘图区画出，幅值最高的波峰的坐标将在图中标出（频率和幅值）

2) 可在 bandwidth 中输入频率范围，调整频谱绘制范围（不输入时默认显示范围是采样率的一半）

3、滤波：用 Butterworth 滤波器进行低通/高通滤波

1) 在 W_p 栏输入通带截止频率， W_s 栏输入阻带截止频率（低通需 $W_p < W_s$ ，高通需 $W_p > W_s$ ）

2) 选择 Lowpass（低通滤波）或 Highpass（高通滤波）

3) 点击 Filter，进行滤波。滤波后频谱将绘制在界面右下的绘图区，对应时域信号将用红色绘制在对应通道绘图区

* 由于 Butterworth 滤波器本身的限制， W_p 和 W_s 的选取需结合信号频谱考虑（但 w_p 与 w_s 过于接近时，不能生成对应的滤波器；相差过远时，滤波效果不一定好）

4、保存信号文件

1) 在 popup menu 选择通道

2) 点击 Save data，信号各点对应的时间和电压将保存为.csv 文件

5、读取信号文件

1) 点击 signal re-shown，在弹出窗口选择文件

2) 对应信号将在绘图区画出，可通过拖动绘图区的 sidebar 查看全部信号波形

6、显示波形坐标

1) 在通道对应绘图区点击，界面右方的 x, y 栏会显示距离点击位置最近的信号坐标

3.AO

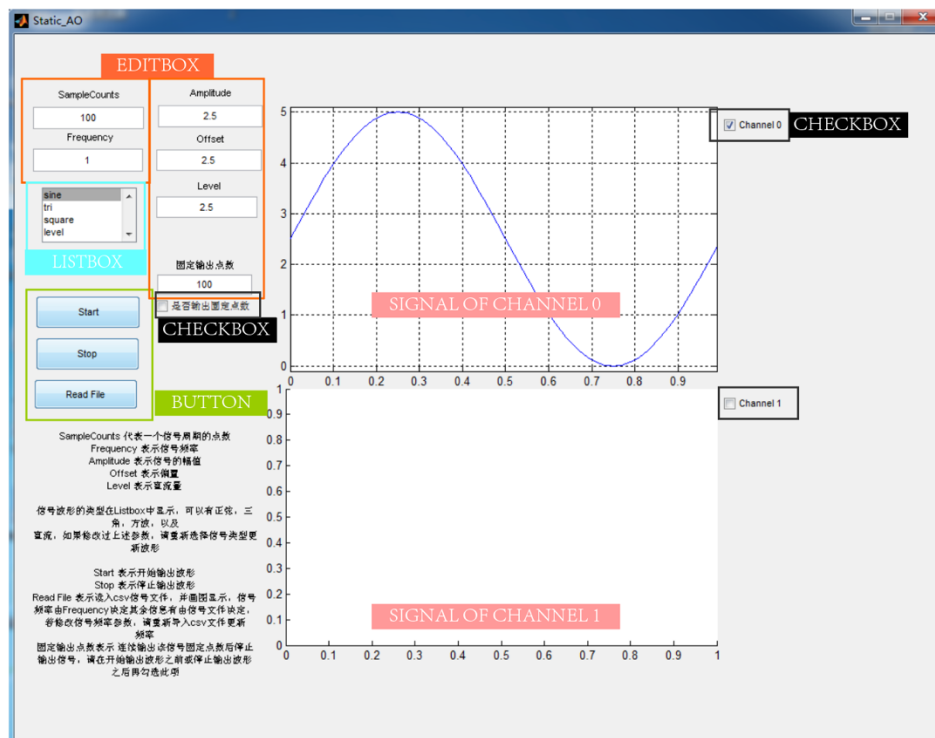


图 2-4 AO 界面

可以实现正弦波、三角波、方波和直流信号的模拟输出。

可以输出事先设定好的特定波形

可以实现两个通道的模拟输入，选择通道的 checkbox 即可开启相应通道。

1、连续输出波形

- 1) 设定每周期输出点数 (SampleCounts)，频率 (Hz)，信号幅值 (V)，偏置 (V) 和直流电压 (V，仅直流信号)
- 2) 点击 sine (正弦波)/tri(三角波)/square (方波) /level (直流信号),一周期的信号将在对应通道的绘图区画出
- 3) 点击 start，开始输出波形
- 4) 点击 stop，停止输出波形

2、输出固定点数的波形

- 1) 设定每周期输出点数 (SampleCounts)，频率 (Hz)，信号幅值 (V)，偏置 (V) 和直流电压 (V，仅直流信号)
- 2) 点击 checkbox"是否输出固定点数"，在固定输出点数栏输入需要输出的点数
- 3) 点击 start，开始输出波形，输出固定点数后停止输出

3、读取文件并输出

- 1) 点击 Read File, 在弹出窗口选择.csv 信号文件
- 2) 设定频率 (Hz)
- 3) 点击 start, 开始输出波形
- 4) 点击 stop, 停止输出波形

* 文件数据格式:

1. 只有两行的 csv 文件, 包含 1s 内信号的电压和对应的时间点 (时间点建议 ≤ 100 个, 点电压幅值为 0-5V)

2. 第一行为点电压, 第二行为对应的时间点

* AO 输出限制

信号频率 $\leq 8\text{Hz}$

信号频率 \times 每周期点数 < 1000 (USB-4704 的每秒输出点数上限)

4. DI_DO

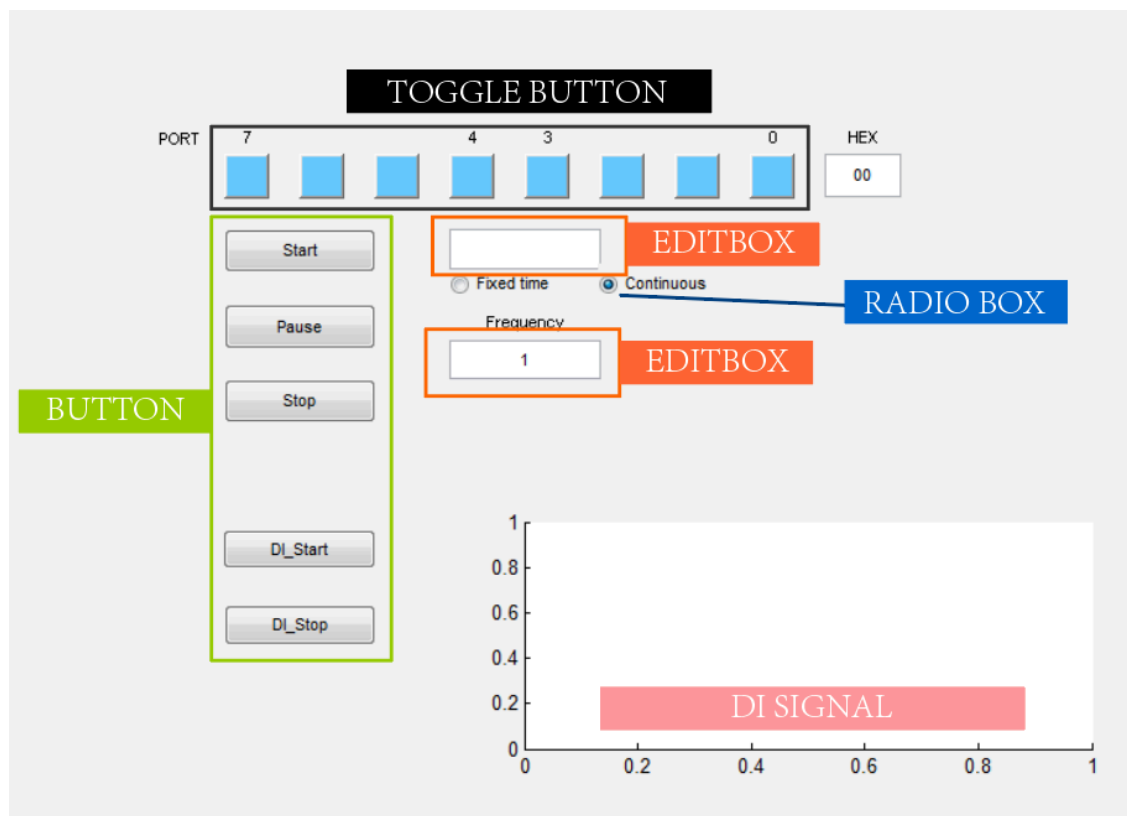


图 2-5 DI_DO 界面

可以实现 DO 端口的数字输出, 每个端口可输出特定频率的方波
可利用 DI 采集到的八个端口电平绘制理论的方波波形

1、数字输出

1) 点击界面上方的 TOGGLE BUTTON, 控制对应端口的电平。浅蓝色为低电平, 深蓝色为高电平

2、数字端口连续输出方波

1) 设定频率 (Hz)

2) 选择 "Continuous"

3) 点击 start, 输出方波

4) 点击 Pause, 暂停输出方波

5) 点击 stop, 停止输出方波

* 方波输出频率建议不大于 55Hz; 可在输出过程中实时改变频率

3、数字端口输出固定时间方波

1) 设定频率 (Hz)

2) 选择 "Fixed time"

3) 输入需要输出的时间 (s)

4) 点击 Start, 开始输出对应时间方波, 输完后停止

4、数字输入

为了测试数字输入功能, 将 USB-4704 的 DO 口与 DI 口相连, 在界面上方设置了八个 TOGGLE BUTTON 修改数字输出的值, DI 接收到 DO 的输出值后进行解码; 其中 DI 的 port0~4 控制频率 (Hz, 00000~11111, 0-31Hz), port5~6 控制幅值 (V, 00~11, 0-3V), port7 表示开/关 (0~1)

1) 点击 DI_Start, 开始接收数字输入

2) 改变 port0~7 的值(00000000~11111111), 软件检测到对应八个端口的数字输入信号, 在下方绘图区画出对应频率、幅值的方波波形

3) 点击 DI_Stop, 停止接收数字输入

III、测试中出现的问题及解决方案

1. 如何实现多 GUI 融合

1) 比较复杂的方法, 先将子 GUI 的 handles 放在主 GUI 的工作区内, 通过设置按键的 callback 函数调整子 GUI 的 "visible" 参数, 实现 GUI 之间的切换。但是这个有个坏处, 就是子 GUI 右上角的关闭键的 delete_func 会自动把他自己的 handles 从主 GUI 的工作区内删除, 所以第二次运行该子 GUI 时就会报错, 除非重新添加子 GUI 的 handles, 但是这样就有点没意义了。

2) 比较简单的方法, 在主 GUI 中设置按键, 按下后, 用 gcf 保存主 GUI figure 的 handles, 运行子 GUI, 通过保存的 handles 关闭主 GUI 的 figure, 实现界面切换; 然后在子 GUI 的 delete_func 里面重新运行主 GUI, 重现回到主界面。