

# 项目一

小组成员: 陈宇 516021910399

李萌 516021910153

## [要求]

- 1、熟悉 ELVIS II+, 利用信号发生器功能产生信号, 类型为正弦波和方波, 频率不大于 100Hz。
- 2、通过 USB-4704 的模拟输入端和 DAQ Navi 采集所产生信号, 对比产生波形和采集波形, 并记录;
- 3、利用 DAQ Navi SDK, 选择一门语言进行编程, 编写用户界面供显示和用户交互;
- 4、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;
- 5、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 实验结果与讨论

#### I、使用 ELVIS II+产生信号

#### II、讨论

- 1) 如何针对不同频率的信号设置合适的采样率, 并分析设置采样率时考虑的因素;
- 2) 分析 USB-4704 的模拟输入功能可采集信号的频率范围, 若输入信号在该范围外, 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

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

### I、使用 ELVIS II+产生信号并通过 USB-4704 的模拟输入端和 DAQ Navi 采集所产生信号

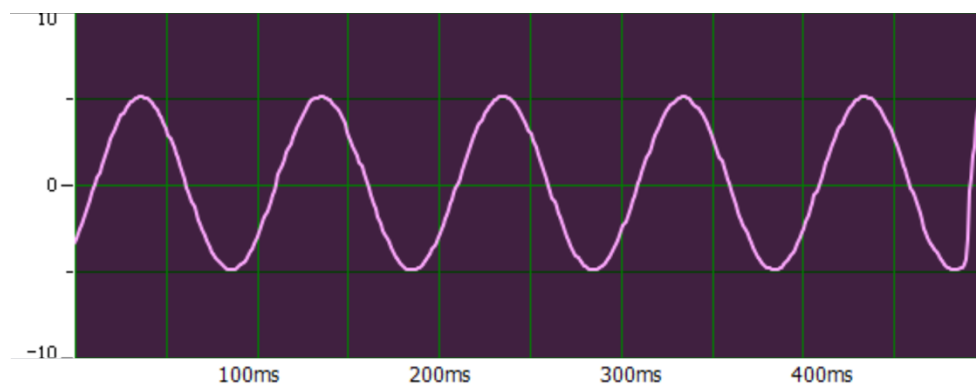


图 1-1 输入信号：正弦波，10Hz；采样频率：1000Hz

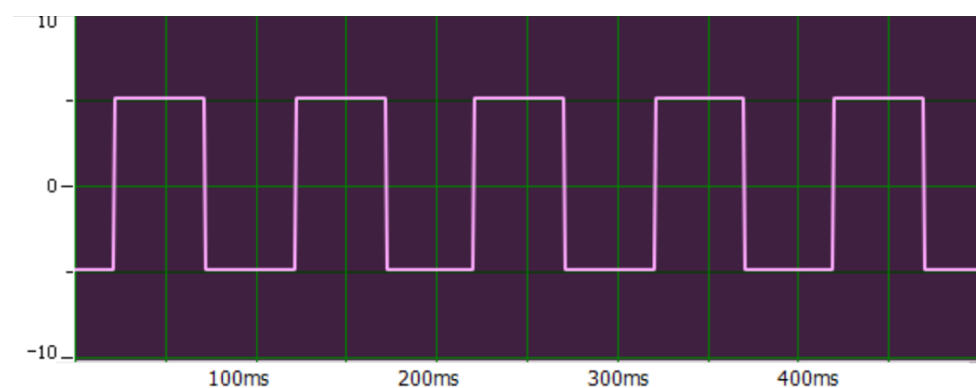


图 1-2 输入信号：方波，10Hz；采样频率：1000Hz

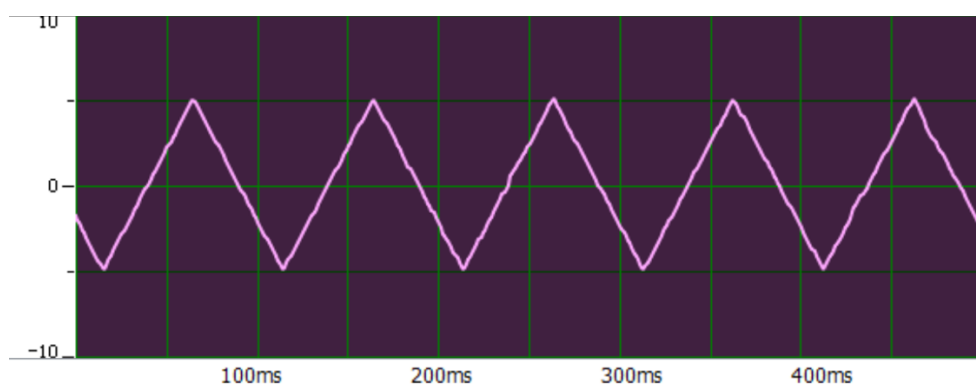


图 1-3 输入信号：三角波，10Hz；采样频率：1000Hz

如图 1-1 ~1-3 所示，当采样率大于输入信号频率的 10 倍时，采样信号几乎不失真。

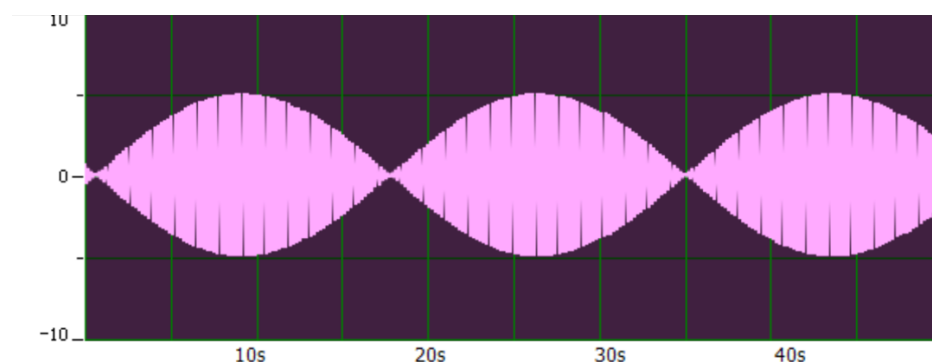


图 1-4 输入信号：正弦波，5Hz； 采样频率：10Hz

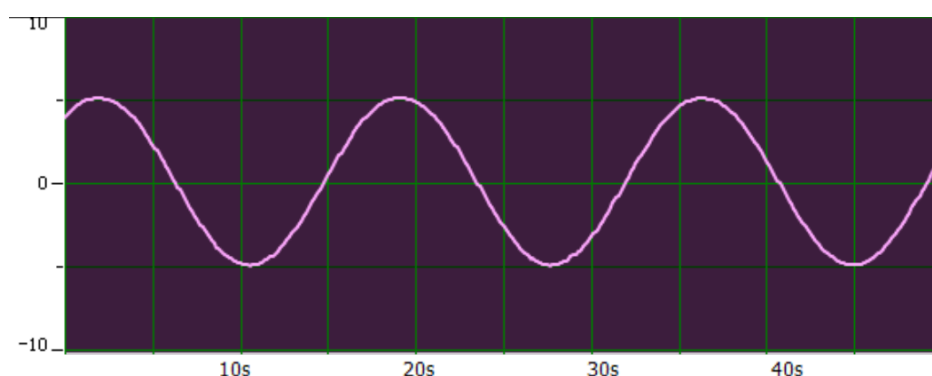


图 1-5 输入信号：正弦波，10Hz； 采样频率：10Hz

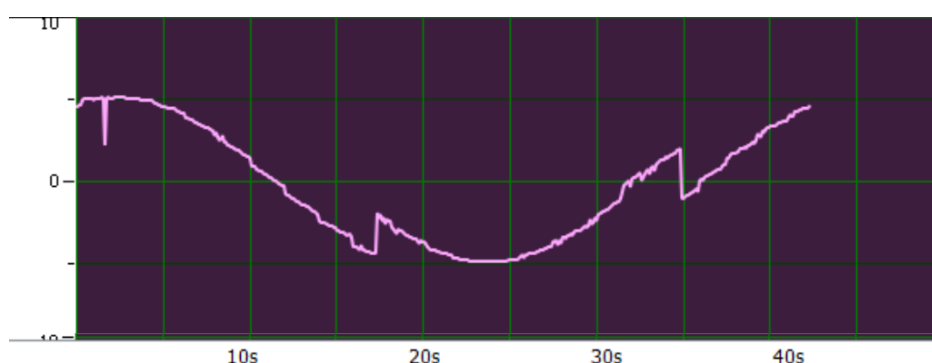


图 1-6 输入信号：正弦波，100Hz； 采样频率：10Hz

如图 1-4~1-6 所示，当采样频率为两倍输入信号频率、等于信号频率以及为信号频率的 1/10 时，得到的失真波形有各自的特征。

当采样频率为两倍输入信号频率，且输入信号为正弦波时，理论上推导采样值应该正负交替，但绝对值相等，失真信号呈矩形。数学描述如下：

输入信号：  $x(t) = \sin \omega t, \omega = 10\pi, T = \frac{1}{f} = 0.2s$  ；

采样率  $f_s = 10Hz$ ，采样周期  $T_s = 0.1s$ ；

理论上的采样信号应为  $x_{st}[n] = \begin{cases} x_{st}[0], & n = 2kT_s \\ -x_{st}[0], & n = (2k+1)T_s \end{cases}, k = 0, 1, \dots$

其中， $x_{st}(0) = \sin \omega t_0$  为第一个采样值。由于采样周期  $T_s$  为正弦输入信号周期  $T$  的一半，则每次采样时输入信号都后移了  $T_s = \frac{1}{2}T$ 。

设第  $n$  个采样值为：  $x_{st}[n] = \sin \omega t_0$

则第  $n+1$  个采样值为:  $x_{st}[n+1] = \sin \omega \left( t_0 + \frac{1}{2}T \right) = -\sin \omega t_0$

从以上推导可以看出, 第  $n+1$  个采样值总是第  $n$  个采样值的负值。

但实际上我们观察到, 采样信号为一个周期  $T_r \approx 34s, f \approx 0.05Hz, \omega_r \approx 0.1\pi$  的有正负正弦包络线的信号, 表达式约为  $x_{sr}[n] = \begin{cases} \sin \omega_r(t + t_0), & n = 2kT_r \\ \sin \omega_r(t + t_0 + \pi), & n = (2k+1)T_r \end{cases}, k = 0, 1, \dots$  这种现象产生的原因是: 理论上认为采样是瞬时的, 但实际中采样本身需要一定的时间  $t_1$ , 对于该输入信号, 以采样周期  $T_s = \frac{1}{2}T$  进行采样, 每次采样时两个采样值并非如上述理论推导, 是对输入信号间隔  $T_s = \frac{1}{2}T$  的采样, 而是间隔为  $\left(\frac{1}{2}T + t_1\right)$  的采样。于是, 两次采样值之间的关系并不是简单取反的关系。推导如下:

假设第  $n$  个采样值:  $x_{sr}[n] = \sin \omega t$

则第  $n+1$  个采样值:  $x_{sr}[n+1] = \sin \omega (t + T_s + t_1) = -\sin (\omega t + \omega t_1)$

第  $n+2$  个采样值:  $x_{sr}[n+2] = \sin \omega (t + 2T_s + 2t_1) = \sin (\omega t + 2\omega t_1)$

第  $n+3$  个采样值:  $x_{sr}[n+3] = \sin \omega (t + 3T_s + 3t_1) = -\sin (\omega t + 3\omega t_1)$

可以看出: 对奇数次采样信号和偶数次采样信号的幅值构成频率相同, 相位相差  $\pi$  的正弦波, 其周期约为  $T_r \approx \frac{2\pi}{\omega t_1}$ , 则执行一次采样需要花费  $t_1 \approx 0.00588s$ 。

同理, 当信号频率为 10Hz, 采样频率为 10Hz 时,  $T_s = T = 0.1s, \omega = \frac{2\pi}{T} = 20\pi$ 。

假设第  $n$  个采样值:  $x_{sr}[n] = \sin \omega t$

则第  $n+1$  个采样值:  $x_{sr}[n+1] = \sin \omega (t + T_s + t_1) = \sin (\omega t + \omega t_1)$

第  $n+2$  个采样值:  $x_{sr}[n+2] = \sin \omega (t + 2T_s + 2t_1) = \sin (\omega t + 2\omega t_1)$

采样信号呈正弦波, 周期约为  $T_r \approx \frac{2\pi}{\omega t_1} \approx 17s$ 。

当采样频率低于两倍信号截止频率时, 采样信号会出现失真。但当采样频率与信号频率有倍数关系时, 当输入信号为正弦波, 由于  $\omega T_s = n\pi, (n = 1, 2, \dots)$ , 采样信号会呈和原信号幅值相同, 频率改变的正弦波。由于程序实际执行一次采样约  $0.00588s$ , 则实际采样频率不能大于  $f = \frac{1}{T} \approx 170Hz$ , 则频率在约 20Hz 以下的信号能较好地被 DAQ Navi 采集, 超过约 85Hz 的信号会出现严重失真。

如图 1-7~1-9 所示, 采集信号严重失真。一方面是因为当采样频率低于两倍信号截止频率 (抽样定理), 且采样频率与信号频率无倍数关系时, 从理论上推导, 采样信号波形会出现明显失真。另一方面, DAQ Navi 的实际采样率有限, 这也限制了采样的准确性。

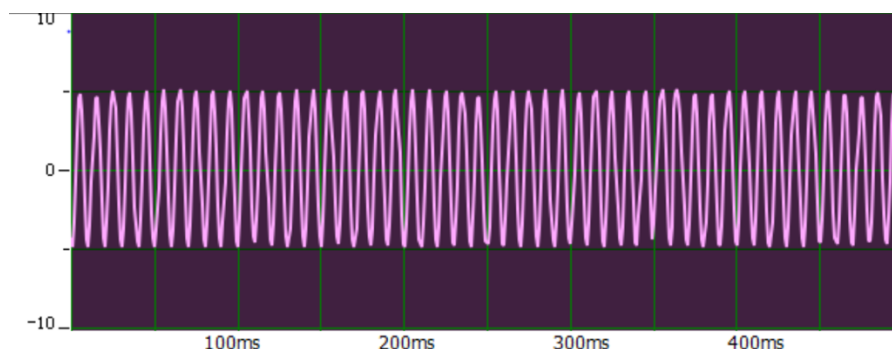


图 1-7 输入信号: 正弦波, 100Hz; 采样频率: 1000Hz

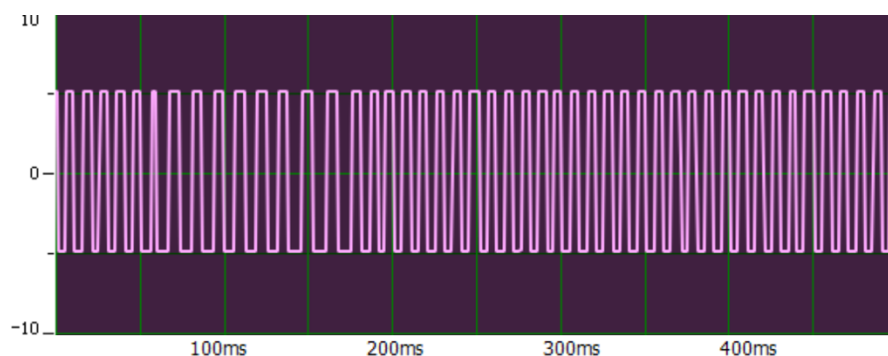


图 1-8 输入信号：方波，100Hz； 采样频率：1000Hz

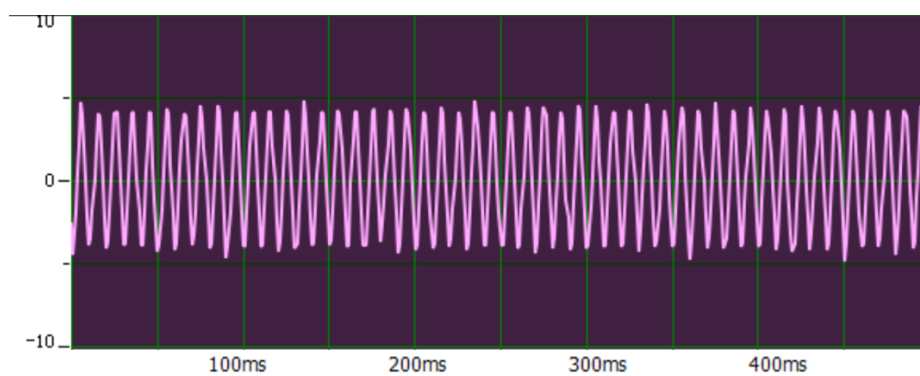


图 1-9 输入信号：三角波，100Hz； 采样频率：1000Hz

## II、讨论

### 1) 如何针对不同频率的信号设置合适的采样率，并分析设置采样率时考虑的因素

一般采样率需要大于或等于信号频率的 10 倍，当对方波这种频域有多个波峰的信号进行采样，需要更大的采样频率才能够完美再现信号波形。另外，在实际设置采样率时还需考虑程序本身能实现的采样率极限。

### 2) 分析 USB-4704 的模拟输入功能可采集信号的频率范围，若输入信号在该范围外，会出现哪些问题，并探讨可能的解决方案

USB-4707 的 Instant AI 最大只能设置 1000Hz 的采样率，但通过 I 的讨论发现，其实际采样率只能达到约 170Hz，所以实际可采集信号的频率最高约为 80Hz。当输入信号在该范围外，可能出现 1) 信号波形失真；2) 采样信号与输入信号波形相似，但频率不一致。

对于 1) 信号波形失真的情况，其原因是采样率未大于信号频率的十倍（理论上两倍即可，在实际中大于十倍的采样率才能比较好地保证信号可恢复），且采样率与信号频率无倍数关系。解决方案为：优化程序，降低一次采样需要的时间，提高实际采样率的上限。

对于 2) 采样信号与输入信号波形相似，但频率不一致的情况，其原因是采样率未大于两倍信号频率且与信号频率存在倍数关系。在这种情况下，只要一次实际采样需要一定的时间，采样信号都会出现失真。仅能通过改变采样率，避免与信号频率倍数关系来解决。

## 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:

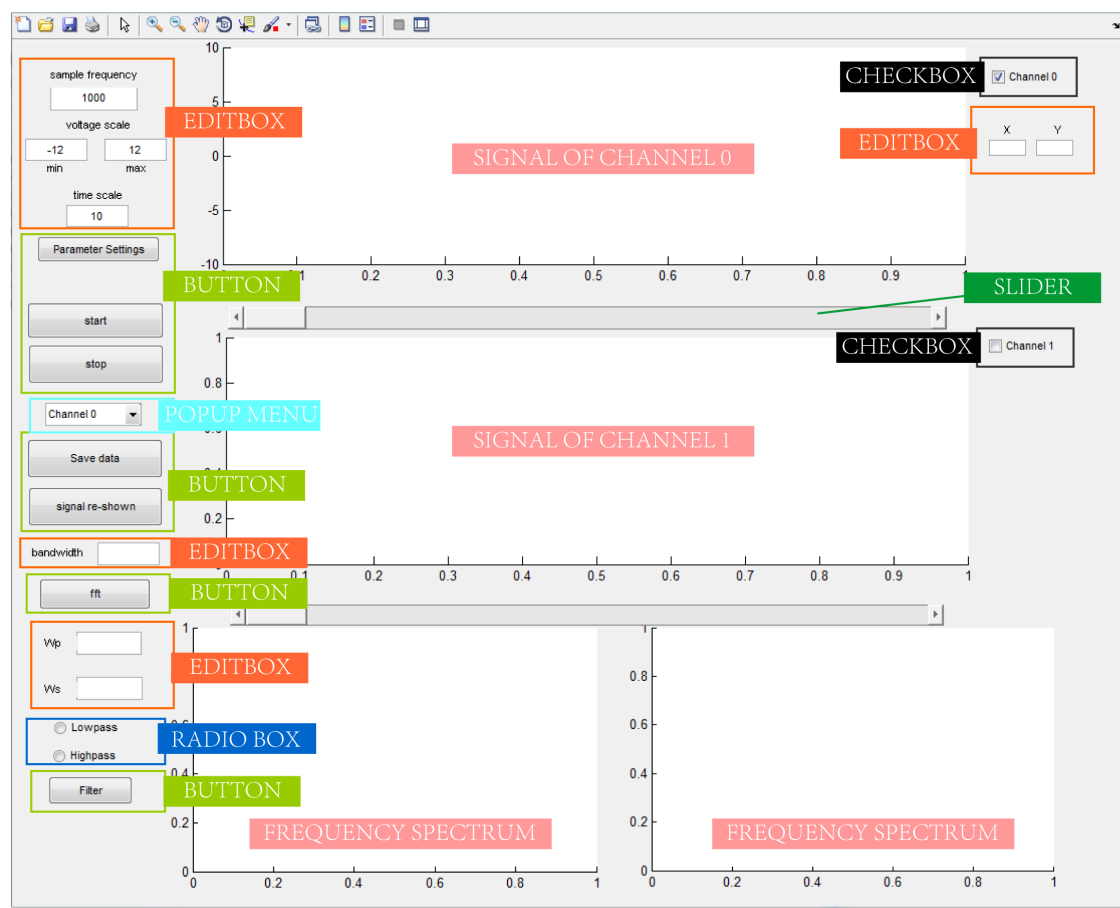


图 2-1 界面

EDIT BOX:

- 1) Sample frequency 采样率 (Hz)
- 2) Voltage\_min, Voltage\_max 显示电压上下限 (V)
- 3) Time\_scale 一个屏幕长度所代表的时间 (s)
- 4) Bandwidth FFT 完成后显示的频率窗宽 (Hz)
- 5) Wp, Ws 滤波器参数, 表示通带和阻带频率 (Hz)
- 6) x, y, 显示最接近鼠标点击处的信号坐标值 (t-V)

RADIO BOX:

- 1) HighPass 高通滤波器 (butterd, butterworth)

## 2) LowPass 低通滤波器 (butterd, butterworth)

### BUTTON:

#### 1) Parameter Settings

- ① 设置采样率 sample frequency, 设置 timer 参数
- ② 设置时间轴和电压范围

#### 2) Start

启动 Timer, 开始采集数据并滚动显示, 并将数据一块一块地写入.txt 文件

#### 3) Stop

关闭 Timer 并删除, 停止采集数据, 将剩余的未存入的数据写入 .txt 文件

#### 4) Save data

将 .txt 文件中的数据保存在 .csv 文件中

#### 5) Data re-shown

将保存在 .csv 文件中的数据读出并显示在相应的屏幕上

#### 6) Fft

将保存在 txt 文件中的数据读出并进行 fft, 根据设定的 bandwidth 参数显示在图表中

#### 7) Filter

将数据读出并根据所选择的滤波器种类和参数构建滤波器, 对时域信号进行滤波, 将滤波后的时域信号显示在图表中, 并做 fft 显示在下方图表中。

### CHECKBOX:

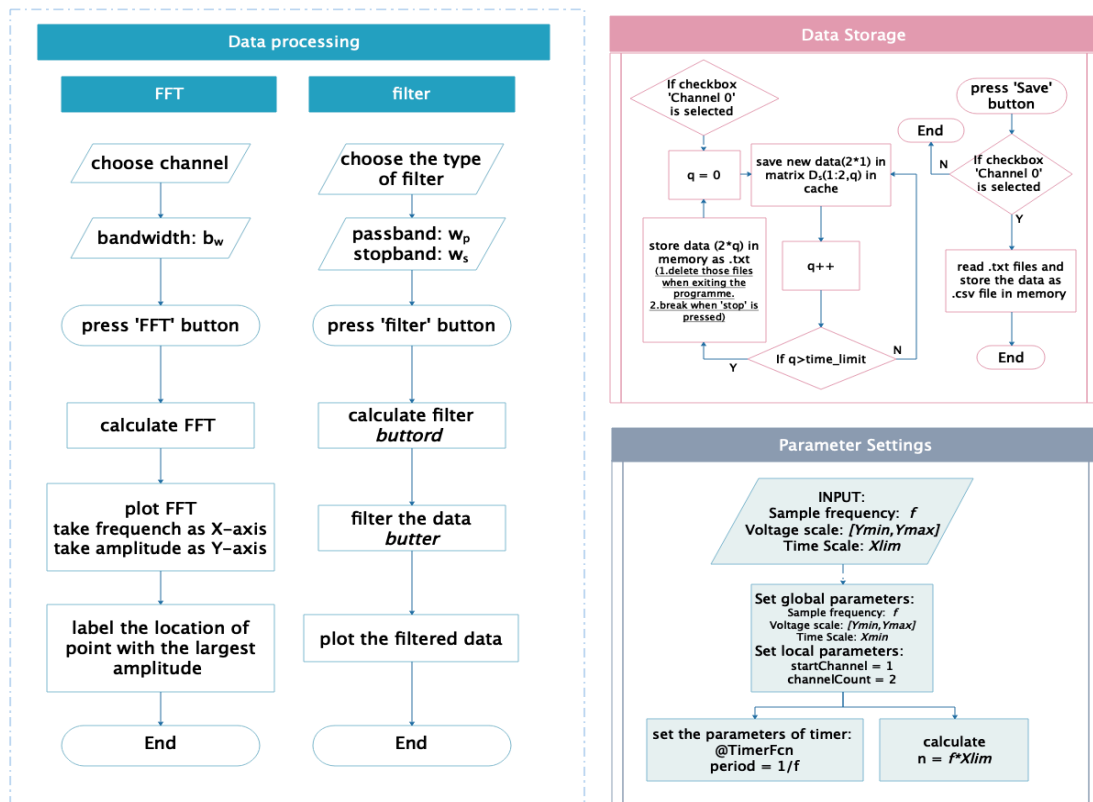
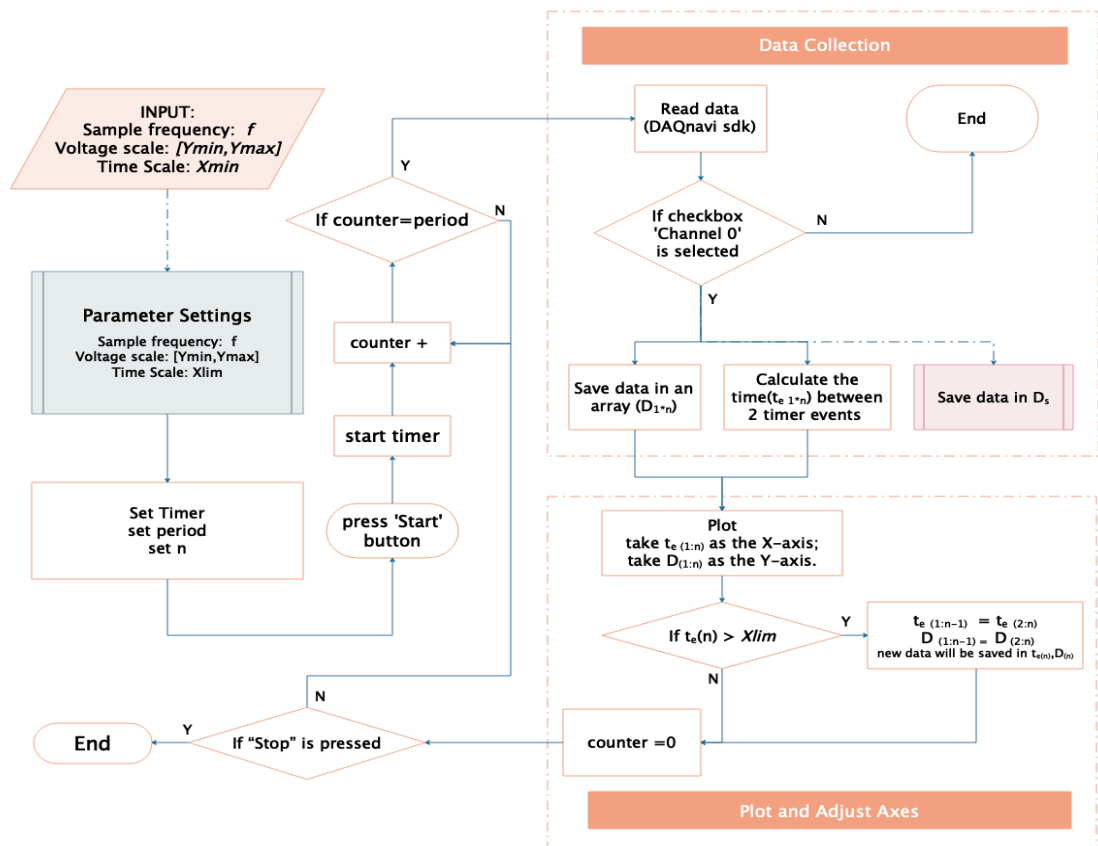
点击小方块表示通道打开, 否则通道关闭

### SLIDER:

data re-shown 按钮按下之后, 左右移动能够滚动显示对应的保存的信号数据

### MENU:

选择 FFT, 滤波, Save data, Data re-shown 的通道



Notes:

- 1、There are 2 checkboxes in the programme. All the following process for checkbox 'Channel 1' are the same as 'Channel 0'.
- 2、'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. 使用 GUI 外部函数画图，Figure 会另外产生

我们将 GUI 的 handles 通过函数传递到外部函数中，但是发现在使用 handles.axes 时会重新产生一个 figure 来画坐标轴，这个是因为 GUI 内部本身有保护句柄的设置，在句柄通过参数进行传递的时候会被隐藏，可以进行数据的传递，但是无法从外部访问句柄内部的模块；设置 HandlesVisibility 为 On 即可解决这个问题。

### 2. 屏闪发生的原因

采集数据和画图是在 timer 的 TimerFun 函数中一起进行的，当采样率很高时，频率刷新的速率也很快，所以出现闪屏的现象。理论上，采集数据和刷新屏幕不应该在同一个模块中进行，有尝试过用两个 timer 分别执行不同的功能但是由于 matlab 的线程不够，导致很多模块无法响应，所以放弃了两个 timer 的做法。

### 3. matlab 中实际采样率的计算

在采样的时候，由于采样和画图的过程需要时间，所以实际的采样间隔时间并不是理想的设定值，这就导致了采样率大大降低的问题。正如前文所说，我们没做到在 matlab 上把数据采样和画图分开，导致采样时间与设定的时间间隔相差甚远；为了保证波形的不失真，我们需要知道每次采样间隔的实际时间，为此我们使用了 get(instantperiod) 函数，得到了最近两次 TimerFun 的间隔时间，并以此计算实际的横坐标，在采样率足够的前提下，保证了信号波形的不失真。

我们计算过输入 1000Hz 采样率时，程序的实际采样率：

- 1) 在两个通道同时打开并且开始滚屏，最高采样率为 33Hz 左右
- 2) 双通道打开但不滚屏为 40Hz
- 3) 单通道打开滚屏为 60Hz
- 4) 单通道打开不滚屏为 150Hz 左右

由此可以看出，画图和滚屏花的时间还是非常多的；

我们也测试过没有画图时采一个点所需要的时间是 2ms，如果 matlab 能厉害一点，确实能够达到更高的采样率。

### 4. fft 以及滤波等距采样的构建

由于我们的横坐标是通过 get(instantperiod) 得到的，所以相当于是非等间距的采样，在这种条件下做 fft 会出很大的问题。所以我们选择利用现有的数据，通过 spline 函数，进行重新插值，得到等间距采样的数据，利用这样的数据做 fft 以及滤波，效果还可以。

### 5. 数据存储与缓存占用问题

一开始我们写程序，构建了一个无限大扩充的缓存数组（没有规定范围），但是测试的时候发现，内存存在逐渐增加，到后面因为内存被占满了，采样的速率变得越来越慢，完全采不到想要的信号；所以我们选择在缓存中只保存屏幕上的点的数据（横轴长度\*采样率），完整的数据在一段时间之后会通过 fprintf 函数写到 txt 文件中，然后点击 save data 能够将完整数据保存在 csv 文件中，通过点击 data re-shown 按钮能够将保存的数据重现在想要的坐标轴上。

### 6. 如何用 Matlab 程序生成 .exe 文件

在 Matlab 中使用命令行 'deploytool'.