Project1 项目报告

1. 程序开发逻辑
2. 初始化

建立GUI，并将显示缓冲区XData，YData，时间点TimePoint，保存文件编号FileNum，保存采到信号的缓冲区FileSave等初始化。

1. 采样

用户输入的采样率，确认并点击开始采样后，首先设打开和初始化硬件，然后利用采样率定义并启动一个采样周期为采样率倒数的timer，在其回调函数中，利用Get方法将采样值取出，并更新时间TimePoint。接着，将采到的值和对应的时间点添加入缓冲区FileSave即可。

1. 显示

最初版本的程序中为每次得到采样点和对应时间点后，通过plot显示该点，并通过hold on 和drawnow命令保留过去作图痕迹和实时更新图像，完成动态作图效果。但在测试过程中，发现drawnow执行过慢，对采样效果影响较大，因此进行了改进，详见2.2。

1. 存储

每次新采到的值加入FileSave后，检查FileSave的长度，当达到一定长度后，将其以.mat格式写入磁盘，并清空FileSave。

1. 放大缩小

放大缩小通过改变缩放系数z实现。Z通过改变显示axes的Xlim，XTick属性改变x坐标轴的范围和分度，从而实现放缩，如下：

set(PlotArea,'Xlim',[Y(1)-5\*z,Y(1)+0.2\*z],'Ylim',[-1.5,1.5],'XTick',[round(Y(1)-5\*z):z:round(Y(1)+0.2\*z)]);

显然，需要放大时，z应小于1，需要缩小时，z应大于1。

每次开始采样时，将z设为0，每次点下放大button，就将他乘以一个小于1的数a；殿下缩小button时，则除以一个大于1的数b。为符合习惯，ab=1。

1. 测试中出现的问题及其解决方案
2. GUI按键的逻辑顺序

在最初版本中，未对用户在GUI上按键先后进行设置。比如用户可以在确认采样率前点下开始采集按钮，造成报错。为了防止这种现象，我在GUI初始化时将除采样率输入，确认控件以外的所有控件的enable属性设为off，并在每个button的回调函数中重新设置了此button按下后应禁止或允许的按钮。

另外，为防止用户强行使用窗口关闭按钮关闭GUI导致timer未停止或有关控件没有删除，在关闭按钮的回调函数中也加入了检查，如有必要则停止timer，删除控件的代码，保证程序能安全退出。

1. 绘图的时间成本

当采样率设置较高时，从实时显示中可以看出，采到信号的频率会逐渐和输入信号产生偏差，由于实测drawnow指令的耗时很长（~0.015s），导致timer回调函数的执行时间大大长于其本身被timer放入执行队列的周期（即采样率的倒数）。

为了节约执行时间，我不再使用drawnow命令，而是实现用line命令将待显示的图像对象建立，并将其XData,YData设空。每次采样后将新采到信号的时间点和值分别加在XData和YData的末尾。为了防止XData，YData太大，当他们到达一定长度时，将其中指用get函数取出，并把其中最后若干个点重新赋回XData/YData。这样，这两个向量的长度就不会无限增长导致内存溢出。

1. x坐标轴Label的可读性

在实时显示过程中，由于x坐标轴会随着在不断移动。为了防止移动过快导致无法阅读，我将xTick设置为',[round(Y(1)-5\*z):z:round(Y(1)+0.2\*z)]，其中z为放大缩小系数，利用round，使x轴label事实上每秒才更新一次，增强了可读性。

1. 采样率设置

理想状态下，采样率应不低于输入信号最高频谱分量的两倍。实际上，由于在本方案中使用Timer定时采样，而matlab中Timer的最小周期为0.001s，所以支持最高采样率为1000S/s。另外，由于Timer的回调函数本身的执行时间为0.01~0.02s（主要耗时在读硬件端口），可以支持的最高采样率约为100S/s以内。

设置采样率时，考虑到程序本身运行的速度，在设置采样率时在满足需求的前提下可尽量低一些，以利于采样时间点的准确。另外，Timer周期最大的精度也为0.001s，这也会导致实际采样率和设置采样率的微小偏差。

如果采用buffered AI模式，可以大大提高最高采样率，但由于显示部分有些未解决的问题，在本项目中暂未实现。

1. 分析 USB-4704 模块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-4704的采样率最高为48kS/s，由采样定理可知，对于单通道输入信号的频率应小于24kHz。若信号有超过此频率的成分，则在频域会出现频谱混叠现象，导致频谱失真。

为了解决这个问题，可以在采集前使原始信号通过一个截止频率适当的低通滤波器，使其最高频率成分低于24kHz。由于甚高频信号往往是噪声，这种处理一般不会造成有效信息的损失。

如果高频部分含有用信息，又无法更换支持更高采样率的硬件，可以考虑使用多个采集卡，利用单片机或微机控制轮替采样，最后把采到的结果合并。