原棉测试软件使用说明（动态版本）

**整体框架**

无论是静态实验软件系统还是动态实验软件系统，基本的框架和逻辑是类似的，由C++层负责采集由FPGA传上来的数据，再经由一个本地Socket（TCP网络传输协议）传输到C#层。

数据量较大的数据常由C++层处理，较为简单数据量较小的数据处理由C#层处理。

由FPGA采集到的数据将不断的传给C++层，C++层将每次收到的数据储存在一个固定的buffer（缓冲）中。在C#层中有一个固定的时钟，每隔一段时间C#将主动发送数据给C++层，C++层将最新的buffer里的内容返回给C#层。【因此，总共有两套时钟系统，他们之间通过C++层的buffer来刷新和交换数据】

其他说明：可能需要安装**.Net Framework 4.5**及**MS Visual C++ 2015 C++ Redistributable Package**

**动态实验软件使用说明**

本说明建立在静态软件已经正常使用的前提下，若您还不会使用静态软件，请参见“原棉测试软件使用说明\_静态.docx”文件。

本文主要阐述动态软件的一些新增的内容和动态软件的设计目的。

在静态实验中我们验证了传感器对棉花和塑料的反应区别，而在动态实验中我们将尝试让棉花和塑料在滚筒上转起来，通过扫描滚筒来获得棉花、塑料的相关反应曲线。

然而，在现有的传感器电路和软件、硬件限制下，还存在着一些问题：

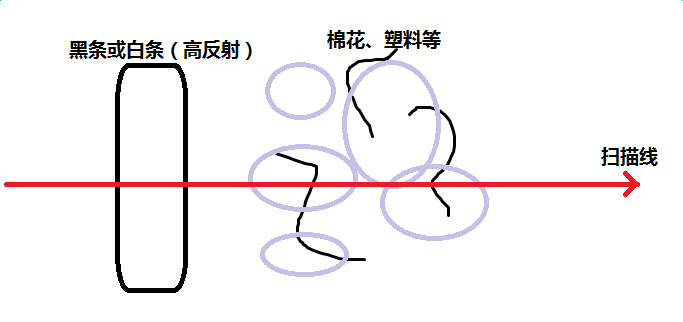
1. 传感器信号本底较高，随机噪声较高，单次采样不稳定因素大
2. 软件无法对上万个点进行绘制图像
3. 滚筒转动较慢，若需要采集一段滚筒上的距离，则需要较多的采样点
4. 两个传感器在纵向上有一些偏移，需要软件或硬件的方式调节相位差

正是因此，我们设计了这一套动态软件系统。

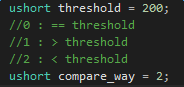
与静态软件系统的主要区别：

【1】

为了应对采样点不宜过多（显示不方便）的问题，设计了一种启动采样机制，具体做法是，在采样滚筒上粘贴一个黑胶带（反射率较低）或锡纸（反射率较高）作为采样的起点，在C++层增加判断条件来判断何时开始采样。如下图所示：



在C++层中增加了两个变量处理这个起始点：



如上图所示，threshold表示阈值，compare\_way表示比较方式，上图表示当传感器收到信号(x<200)时，开始采集。若为高反射白条，可将threshold改为4000（传感器最大输出4096），compare\_way改为1。

这两个变量可通过C#层利用Socket设置，已经在C++的Server和C#的Client中实现。

【2】

同样为了应对采样点不宜过多的问题，在C++层设置了一个采样长度（单位：short），这个长度应该是2的倍数（对应两个传感器），如下图：



这个长度指示了当触发采样的时候，C++层将连续采集多少个数据**点**（每两个数据点是一次双传感器采到的数据**对**）

这个长度和传感器在滚筒上采集的长度有直接关系。

这个变量可通过C#层利用Socket设置，已经在C++的Server和C#的Client中实现。

【3】

为了应对噪声问题，在C++层增加了平均选项。如下图所示：



具体来说，通过多轮平均来降低采集的噪声性。即滚筒滚10圈→做一次平均放到buffer中。但实验中常常遇到起点不一致而导致的采样平均多圈不同步的问题，因此在实验时常常把这个值设置为1，消除噪声通过另一种平均的方式来进行。

到此为止是C++层做的全部操作，其他的数据处理由C#层完成。

【4】

为了应对噪声问题和多圈平均起点不同步的问题，由于转筒转速较慢而采样频率较高，尝试采用横向平均（即连续N个点作为一个数据）的方式来消除一定的噪声。如下图所示：

## 

C#层的平均次数控制、Thr、Cmp和Len均为之前提到的C++层的相关参数，而抽样/平均间隔则为C#层的参数。具体来说，当勾上“平均”时，程序将会把连续的50个（上图中是50个）点做一次平均加到要显示的数据中。若不勾上平均，则会每50个点抽取一个点（噪声更大，更模拟真实情况）加到待显示的数据中。

若为50个点的平均，则每条曲线会有20000（数据总长）/2（传感器2个）/50=200个点。

【需要注意：由于程序绘制曲线的点的数目不能太高，建议这个平均个数不要小于10，即1000个点已经接近绘制曲线的极限】

【5】

为了应对两个传感器有相位差的问题，在C#层加入了传感器1偏移的参数，可将处理好的数据偏移N个点后再进行求差和作图。

经验表明，当显示200个点（平均间隔50）时，这个偏移为5比较合适。



【6】

与之前不同，之前的数据是每0.1秒刷新一次（向C++层请求一次buffer），现在改为了动态可调的值，如下图：



一般经验认为这个值在1秒或2秒比较合适，太快了会反复拿到C++层相同的一次数据，太慢了可能会错过C++层的数据。

程序界面：（实时显示输出的功能已经禁用）

