# 需求说明

该工具用于对串口、网口的时域数据进行数据收集、波形图绘制、数据导出等功能。如下将对该工具的方案需求分为数据采集、数据监视、数据导出等几个方面进行简要说明。

## 数据采集

数据采集指工具接收并解析来自串口、网口、USB接口等端口的时域数据。这些时域数据默认格式为CSV格式，以一行数据表示一个数据帧，每行数据通过Dos（\r\n）格式的换行符间隔开；数据帧的数据由英文逗号隔开，每个数据的数据类型为实数，其精度可配置。采集开始时，以第一行的数据个数作为数据采集的通道数目，往后的数据帧如果数据个数少于采集通道数，则默认填充为非法浮点数。数据采集时支持数据帧时间记录，且该功能可配置。此外，工具支持数据量超过4GB的数据采集。

考虑到数据格式的复杂性以及数据来源的多样性，数据采集模块以插件的方式安装到工具中，工具提供名为“通用网口”和“通用串口”的两个默认插件用于对默认数据格式进行采集。插件以DLL形式存在，工具启动时遍历存放插件的目录并对目录内的插件进行加载。插件加载完成后以子菜单项的方式存在于菜单栏的【打开】菜单项下，点击对应的子菜单项时由对应插件弹出UI控制界面，引导用户完成数据端口的打开。

## 数据监视

数据监视是指对采集到的数据每个通道进行绘制，绘制方式包括散点、折线图、曲线图。绘制过程中支持显示通道的选择，每个通道支持独立的XY轴的平移与缩放控制。此外还支持自定义曲线的绘制，即将数据通道的某几个通道的数据进行计算后作为新的虚拟通道。例如数据通道只包含电流数据通道与电压数据通道，数据显示时则可以直接新增一个虚拟的功率通道，将电压通道与电流通道的数据的乘积作为新通道的数据。这个功能是在界面上输入简单的C#代码实现的，也可以输入其它代码完成特殊函数图像的显示。

## 数据导出

为节省磁盘空间，工具最终的存储格式是二进制的方式存储的，但是工具支持将数据导出到CSV格式，方便其它程序加载分析。同时工具也支持导出为Python代码，用于自定义的数据处理。

# 框架说明

框架分为插件模块、数据泵模块、存储模块、绘制模块、UI交互五个部分，分别负责数据采集、数据分发、数据存储、数据展示、交互。

## 插件模块

插件模块分为两部分，一部分是通用的插件接口与插件实现。该部分负责对插件进行抽象，定义插件所需要的公共接口。插件的实现则是网口与串口数据采集的通用插件的具体实现，并在此基础上提供插件开发示例。插件模块的另一部分则是插件的管理，它负责程序启动阶段加载并管理插件，以向外提供插件信息以及用于数据读取的插件实例。

## 数据泵模块

数据泵模块作为整个工具的数据交汇点，它的主要任务就是通过插件读取数据，并将其分发到存储模块和绘制模块。除此之外，它还提供虚拟通道的数据计算任务，新的数据点就是在此阶段产生的；因此，该模块还提供新数据点计算公式对应代码的编译及加载服务。

## 存储模块

数据存储只负责将采集到的数据存储到文件中去。其中，每个端口的数据对应一个数据队列，数据泵模块交付的数据都会投递到对应端口的数据队列中。整个模块只有一个工作线程，该工作线程会遍历所有的数据队列，一旦数据队列里面有数据，则将数据写入对应的文件中。

## 绘制模块

绘制模块同样采用队列的方式存储数据，只是该队列长度是有限制的。因为队列中的数据就对应着界面上的曲线，队列的长度对应着绘制曲线的长度。当队列中数据个数达到一定限制时，每加入一个新的数据点就会去除一个旧的数据点，相当于曲线的平移显示。

每条曲线对应着一个对象，该对象内存储了该曲线的颜色、平移参数、缩放参数、绘制偏移等诸多绘制时要用到的参数。所有曲线的绘制都是由一个线程完成的，该线程会将所有曲线绘制完成后将绘制好的图像交付给UI线程，实现界面更新。

另外，曲线绘制时采取的策略是：若只有新增数据点的更新，则只绘制新的数据点，并添加到上次绘制的结果上，从而实现绘制加速；若线条颜色、平移参数、缩放参数发生了更新，则意味着上次绘制结果无法被复用，此时将进行全部重绘。

## 交互模块

交互模块负责与用户的交互，除了管理用户交互的相关数据外，它还管理着上述模块的所有实例。每新增一个端口的数据采集任务，交互模块就会创建一个会话实例，该实例内保存了此次会话所用到到插件实例信息、数据泵模块实例、存储模块实例、绘制模块实例。UI上对数据采集、存储、绘制的控制都是通过会话实例找到对应的模块来完成的。