**朗石水质监测仪业务逻辑源码修改V1.0.0**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **时间** | **责任人** | **备注** |
| **V1.0.0** | **2018.05.07** | **万宇** | **制定初始版本** |

在修改业务逻辑之前，首先要对整体的业务逻辑架构有个基本的认知，水质检测仪需要在设定好参数情况下，在已定的排期下执行测量任务，同时一方面要响应用户UI的输入，另一方面要响应上位机MODBUS下发的命令，还有水质云的通信等，这些都是由程序启动不同的线程去执行，这些当中最核心的部分就是测量流程相关的内容。

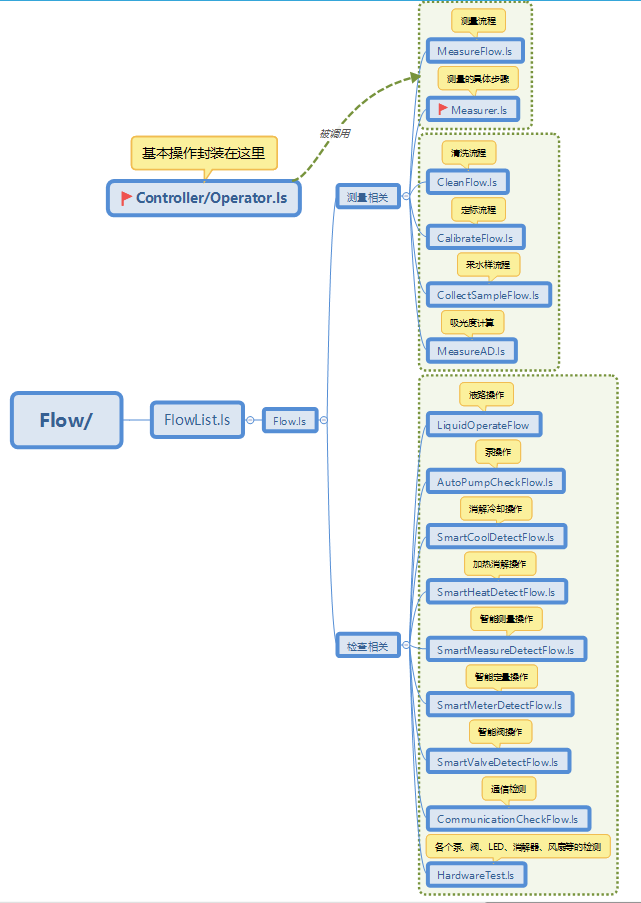
因此，通常业务逻辑的修改，主要是流程的修改，而修改流程，一方面会涉及到相应的参数，另一方面可能会涉及到UI界面的变化。流程所用到脚本主要集中在Flow/文件夹中，参数配置相关脚本位于Config/文件夹与Setting.ls文件中，而UI界面相关的配置在Setting/UI/文件夹下。

以下对若干文件进行简单介绍（请仔细查看所有提及文件的详细参数）：

1. 流程相关
2. Flow/ ：

水质检测仪的流程主要是两种类型，一种是测量相关的流程，一种是检查相关的流程。测量相关的流程一方面是测量、定标流程，另一方面是测量前采水样，以及测量前后的清洗流程；

检查相关的流程一方面包括定量、液路等操作的检查，另一方面是阀、泵等的基本操作的检查；



1. 参数相关
2. Setting.ls ：

此文件中主要是硬件的硬性配置，一旦确定在程序运行期间是不可以更改的，且在大多数参数都是流程中使用的，十分重要。基本所有的参数在定义后面都有相应的注释，基本可以做到见文知意。

setting.unitVolume表示每一单位的体积，单位为mL。

setting.measure.range此项为量程相关的参数，水质监测仪除了基础量程（量程1）外，还允许有若干扩展量程（2，3，… ）；

setting.calibrate此项为定标相关的参数，由于定标所使用的就是测量流程的量程1，只不过加水样的过程换成了加空白水（零点定标）或标准液（标点定标），因此很多项是不需要改的，除了定点0的blankVolume、inseBlankVolume以及定点1的standardVolume、rinseStandardVolume要改成和量程1测量加入的和润洗用到的水样的体积一样；

1. Config/ ：

此文件夹下主要是一些可以在运行期进行更改的参数；其真正有效的文件在Default/文件夹下，外部的文件除了ConfigList.ls、ConfigModifyRecord.ls、ProducerAuthorization.ls外，全部是由Default/中复制而来，因此修改时需要修改Default/文件夹中的文件，然后把外边的同名文件删除。主要修改的文件如下：

ConsumableConfig.ls

此项为各种耗材的更新配置，其中每一子项的cycle条目即为相应的更新周期；

HardwareConfig.ls

此项为定量管、消解器、泵、定量光感等一些参数设置，通常只要了解第一项定量管的相关参数即可，也就是定量管的数量，以及定量点（第一个表示定量点1，依次为定量点2，… ）的设定与实际定量值（通常是一样的）。

InterconnectionConfig.ls

此项主要是一些测量、通信相关的限制；重要的有水样与核查样的4-20浓度上下限等；

MeasureParamConfig.ls

此项主要为测量相关参数的配置，是修改的重点，消解温度、消解时间，反应温度，反应时间，标定0（空白水）浓度、标定1（标准液）浓度、高标校正液浓度，测量前后清洗用的空白水体积，读初始值前静置时间等，都在这里设置；

RemainConfig.ls

此项主要是试剂余量管理的相关参数，每一项表示一种试剂，见名知意；其中cycle表示更换周期/月；total表示试剂总量/mL。

SchedulerConfig.ls

此项是自动测量等排期的设置，每一子项表示一种排期，每一子项的interval条目表示它的排期间隔。

SystemConfig.ls

此项是系统相关的参数，包括屏保的开关和时间，CCEP认证的打开和关闭，水质云的域名与端口等；

1. UI界面相关
2. Setting/UI ：

此文件夹主要是UI界面的修改，主要是添加、删除或者改变系统中的某些元素时需要对照修改。

除了参数的改变之外，另一个重要的点就是流程的变化，尤其是测量的流程。流程相关的业务逻辑主要在Flow/文件夹下。

这里的逻辑大致可以这样描述：首先创建一个流程，也就是一个Flow“类”（LUA并没有类的概念，这里只是利用LUA的表构造及闭包来模拟），然后将这个Flow加入到FlowList队列中，然后执行之（即调用其自身的run函数）。run函数又分为若干步骤，大致有OnStart（开始）、OnProcess（执行中）、Onstop（结束）等几步。OnStart函数通常是做一些初始化动作，比如一些参数的赋值等；OnProcess函数是执行的主干；OnStop函数是结束后的一些收尾工作，比如仪器状态的复位等。OnProcess函数作为执行流的主要部分，也是由一个个步骤组成的；每一个部分都会调用一些基本操作的函数，这些函数主要在Operator.ls中定义。

以下以测量流程为例介绍一下一个流程是怎样从开始到结束的。

测量流程开始时会调用Flow:run()函数，run函数又会调用自身的OnStart()、OnProcess()、异常处理的ExceptionHandle()函数以及OnStop()函数；对于测量流程，这些函数都在MeasureFlow.ls中定义；

在OnStart()函数中，主要是一些状态的设置以及仪器状态的检查等；在OnProcess()函数中，首先设置了重测循环的次数，然后进入这个循环，执行测量步骤；如果测量成功，则一遍过；测量失败，会重新测量，三次失败则抛出异常。在测量步骤中，主要的执行顺次可以这样描述：首先复位了流程表和参数（流程表是流程执行了哪些操作以及执行的顺序；参数配置就是流程需要用到的参数，比如试剂一加多少，使用哪个定量点计量等）；然后根据测量的类型设定加液的参数（加样加多少，润洗加多少，等等）；然后执行了Measurer:Measure()函数，这就是核心的执行流程了。最后进行异常处理。OnStop()函数中，主要进行了仪器的复位，并保存了试剂余量表（根据使用了多少试剂算出还剩下多少）。

Measure()函数在Measurer.ls中定义，在Measure()函数中引用了procedure列表和steps表构造；当procedure表构造为空时，会按照steps中从上到下的顺序执行其中的子项（step），而当procedure不为空时，则会按照procedure里的数字排序的顺序去执行steps里的step。

以下是两个案例，一个是COD的大体系转小体系的修改；另一个是将原来控制三个板的脚本改为单板的脚本，也就是三板合一。首先要说明一下基本的修改方法。修改主要是三种形式增、删、改。一般是分四个步骤进行，定位、分析、更新、调试（如果对代码熟悉了，完全可以一步到位）。增删主要是增加/删除某个步骤或内容；改主要是针对某项流程或操作进行修正。

定位，就是要找到此修改项用在什么地方（如果对整个代码非常熟悉，通常不用费多大力气）。首先进行中文的查找；然后进行对应代码中的名称查找；最后还要查找被赋值的变量名或者该变量的索引进行查找。分析，就是分析所修改的这项内容在定位的位置做什么用途（如果对整个代码非常熟悉，这一步也是不必要的），看使用这项内容的函数是干什么用的；更新即修改；调试则是将修改好的代码在机器上测试若干次，看看有没有什么问题。