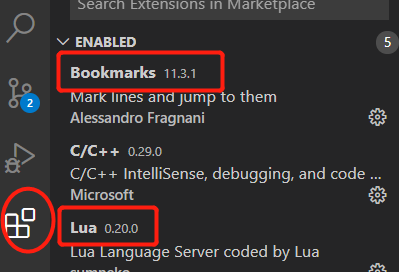
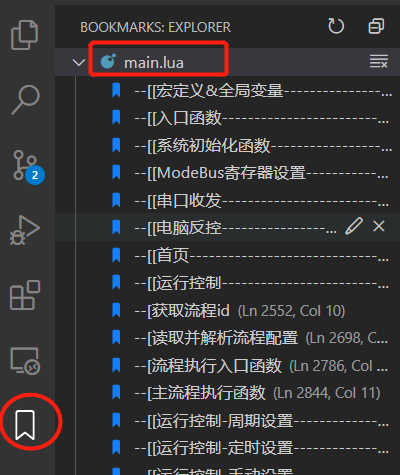
# 一 环境搭建

界面设计以及程序编译采用Visual TFT软件.

Lua脚本编辑采用Visual studio code软件, 下载安装该软件后,需要安装两个插件:一个是Lua插件,用于检测lua语法,智能提示; 一个是Bookmarks插件,添加书签后可以快速定位,因为Visual TFT编译Lua脚本时不支持Lua模块, 所以所有的脚本都写在main.lua文件当中,导致文件比较大, 添加书签后就可以快速跳转, 类似目录的功能.软件安装好后可以使用visual studio code 打开2300project.code-workspace文件。



# 二 宏定义以及全局table说明

在main.lua文件的顶部,定义了一些全局变量与宏,这里做一个简要说明

* 每一个画面都有画面ID, 都定义为了宏, 比如首页定义为MAIN\_SCREEN = 0, 所有的画面都以\_SCREEN结尾.
* **ScreenWithMenu**中保存了带有底部状态栏的画面ID, 在更新状态时,需要将所有的带有底部状态栏的画面都进行更新.
* **TipsTab**用于在画面底部显示一些提示信息,该表主要作为ShowSysTips函数的参数
* **LogTab**里面保存了报警信息以及日志信息.其中信息是根据<<附件4 国家地表水自动监测仪器通信协议技术要求.docx>>中的”表10告警信息表”而来
* **Sys**是一个table,基本所有的全局变量都定义在该部分.
* **WorkStatus** 定义了工作状态,有运行, 停止, 待机(该状态会在定时器中不断查询是否有符合条件的流程需要执行)
* **WorkType** 定义了工作类型,有手动/定时/周期/受控
* **SysUser** 定义了系统用户,有操作员/运维员/管理员
* **OnOffStatus** 定义了阀状态,有打开/关闭.
* **SuddenPowerOff** 定义了运行时断电,再次上电后,可以选择的动作类型,无表示上电后不做任何动作、排空清洗表示做完排空清洗就停止、自动恢复表示恢复为上次断电前的状态。
* **CalcWay** 定义了计算方式，有对数与差值两种方式
* **InjectPara** 定义了注射泵加液参数，“无”表示会按照注射泵加液界面中设置的体积进行加液，“加标体积”表示会按照计算出的加标体积进行加液，只在加标回收流程中使用。
* **CalcType** 定义了计算类型，有"分析", "校准", "核查","加标回收","线性核查"，不同的类型其结果将保存到不同的历史记录界面。
* **ProcessItem** 定义了流程类型选项，其内容应与“流程类型选择”画面中相对应。
* **ActionItem** 定义了动作类型选项。其内容应与“动作类型选择”画面中相对应。
* **PeriodicTab** 保存了周期设置界面中的控件id
* **TimedProcessTab** 保存了定时设置界面中的控件id
* **HandProcessTab** 保存了手动设置界面中的控件id
* **ProcessSetTab** 保存了流程设置界面1/2界面中的控件id
* **ActionTab** 保存了流程编辑界面1/2/3/4界面中的控件id
* **ReagentTab** 保存了手动设置3界面中关于试剂余量的控件id
* **uartSendTab** 保存了串口通讯协议,所有通讯协议都在此处定义.新增加协议时,一定得按照现有的格式进行定义, 否则会出现nil错误.
* **UartArg 主要**用于控制串口重发

# 三 常用函数定义

**系统函数**

**on\_init** : 在系统上电是会调用该函数

**on\_timer**: 定时器回调函数

**on\_systick**:系统定时器回调函数, 每1S调用一次

**on\_control\_notify**: 当控件发生触摸时,会调用该函数,针对每个界面都有一个空间触摸回调 函数例如主页有main\_control\_notify函数,运行控制界面有run\_control\_notify函数等等,

**on\_screen\_change**: 当画面发生改变时, 会调用该函数

**on\_usb\_inserted**: 当插入USB时调用该函数

**on\_usb\_removed**: 当拔出USB时调用该函数

**on\_sd\_inserted**:当插入SD时调用该函数

**on\_sd\_removed**:当拔出SD卡是调用该函数

**on\_client\_recv\_data**:作为TCP客户端,当接受到数据时,调用该函数

**on\_uart\_recv\_data**:当接受到串口数据时,调用该函数

**on\_http\_download**:http\_download回调函数,系统自动调用

**各触摸屏控件对应的回调函数**

以下函数均在中调用,是各触摸屏对应的控件回调函数

**main\_control\_notify**(screen, control, value);--首页

**run\_control\_notify**(screen, control, value);--运行控制界面

**run\_control\_period\_notify**(screen, control, value);--运行控制-周期设置界面

**run\_control\_timed\_notify**(screen, control, value);--运行控制-定时设置界面

**run\_control\_hand\_notify**(screen, control, value);--运行控制-手动设置界面

**process\_type\_select\_control\_notify**(screen, control, value);--流程选择界面

**process\_name\_select\_control\_notify**(screen, control, value);--流程名称选择界面

**process\_set12\_control\_notify**(screen, control, value);--流程设置1/2界面

**process\_edit\_control\_notify**(screen, control, value);--流程编辑1/2/3界面

**range\_select\_control\_notify**(screen, control, value);--量程选择界面

**action\_select\_control\_notify**(screen, control, value);--动作选择界面

**process\_init\_control\_notify**(screen, control, value);--流程设置-初始化界面

**process\_inject\_add\_control\_notify**(screen, control, value);--流程设置-注射泵加液

**process\_peristaltic\_control\_notify**(screen, control, value);--流程设置-蠕动泵加液

**process\_dispel\_control\_notify**(screen, control, value);--流程设置-消解

**process\_read\_signal\_control\_notify**(screen, control, value);--流程设置-读取信号

**process\_calculate\_control\_notify**(screen, control, value);--流程设置-计算

**process\_valve\_ctrl\_control\_notify**(screen, control, value);--流程设置-阀操作

**process\_wait\_time\_control\_notify**(screen, control, value);--流程设置-等待时间

**process\_linear\_set\_control\_notify**(screen, control, value);--流程设置-线性核查稀释

**range\_set\_control\_notify**(screen, control, value); --量程设置

**hand\_operate1\_control\_notify**(screen, control, value);--手动操作1

**hand\_operate2\_control\_notify**(screen, control, value);--手动操作2

**hand\_operate3\_control\_notify**(screen, control, value);--手动操作3(目前该界面是试剂余量)

**in\_out\_control\_notify**(screen, control, value);--输入输出界面

**system\_info\_control\_notify**(screen, control, value);--系统信息界面

**login\_system\_control\_notify**(screen, control, value);--登录系统界面

**dialog\_screen\_control\_notify**(screen, control, value);--对话框界面

**password\_set\_control\_notify**(screen, control, value);--密码设置界面

**wifi\_connect\_control\_notify**(screen, control, value);--Wifi设置界面

**remote\_update\_control\_notify**(screen, control, value);--远程更新界面

**history\_control\_notify**(screen, control, value);--历史记录界面

**password\_dialog\_screen\_control\_notify**(screen, control, value);--密码对话框界面

**process\_copy\_control\_notify**(screen, control, value);--流程复制

**server\_set\_control\_notify**(screen, control, value);--服务器设置

**reagent\_select\_control\_notify**(screen, control, value);--试剂选择界面

**初始化时调用的函数:**

**sys\_init:**

在on\_init中调用, 上电获取各驱动板的版本号,且会在该函数末尾处判断是否设置了异常断电检测,是否恢复为上次异常断电时的状态.

**record\_control\_check:**

用于读取历史记录控件中保存的管理员密码/运维员密码/wifi设置/TCP设置参数/首页参数/量程设置参数/流程设置界面参数/试剂余量设置参数等

**setTextToBlankSpace**:

在on\_init中调用,将一些文本控制设置为一个空格,主要是需要选择流程的文本框

**Modebus相关函数**

**modebus\_file\_init:** 在on\_sd\_inserted中调用, 插入SD卡时,检测是否有record/ModeBus文 件,如果有,则从该文件中读书数据,赋值到ModeBus数组; 如果没有该文件,则将 ModeBus数组初始化为0,并将该数据中的数值写入record/ModeBus文件,用逗号分隔.

**SetModebusResultArea**:设置Modebus协议中的结果区域,一般在计算结果后会调用该函数更 新ModeBus数组以及record/ModeBus文件

**SetModebusSysStatus**: 设置Modebus协议中的系统状态区

**SaveModeBusToFile** 将ModeBus数组中的数据保存到record/ModeBus文件

**setModebusRunMode** 设置Modebus中的运行模式

**setModebusUnite** 设置Modebus中的单位

**串口发送相关函数**

**close\_single\_valve:**关闭单个阀

**open\_single\_valve**打开单个阀

**control\_multi\_valve**控制多个阀打开或者关闭

**control\_valco**控制十通阀

**enable\_inject1**使能注射泵

**reset\_inject1**复位注射泵

**set\_inject1\_speed**设置注射泵速度

**move\_inject1\_to**移动注射泵到指定位置

**set\_dispel\_temp**设置消解温度

**set\_dispel\_time**设置消解时间

**on\_uart\_send\_data(packet, reply)** 通过调用该函数发送串口数据,其中packet为uartSendTab中的某个子项,reply : 表示是否是要等待回复,取值NEED\_REPLY 与NO\_NEED\_REPLY.该函数自动计算CRC16,并在末尾处保存一条通讯记录到手动操作4界面, 且当reply==NEED\_REPLY时,会有超时未收到回复进行3次重发功能.

**uart\_time\_out** 串口接收超时会调用该函数,在该函数中重发上次发送的串口数据,如果一直没有回复,3次后记录一次串口超时记录,并停止重发.

**ComputerControl** 当接受到上位机发送的数据时,会调用该函数.

**底部状态栏显示函数**

**ShowSysTips** 在屏幕的最底部显示一些提示信息, 5S后自动消失

**SetSysWorkStatus** 设置系统工作状态,有运行/停止/待机

**ShowSysCurrentProcess** 显示当前流程名称

**ShowSysCurrentAction** 显示当前动作名称

**ShowSysCurrentMode** 显示当前模式, 有周期/定时/手动/反控等.

**ShowSysAlarm** 显示告警信息

**运行控制相关函数**

**getProcessIdByName** 通过流程名称获取流程id.

**getProcessIdByType** 通过流程类型获取流程id, 例如在开机的时候需要检测是否要执行排空清洗, 这时候就会调用该函数获取排空清洗的流程id,并执行流程.

**getNewTimeByFreq** 通过频率计算出下次运行的时间,一般在周期模式下使用.

**setNextPeriodTime** 设置下次开始时间(周期模式下)

**setPeriodStartTime** 将当前时间设置为周期开始时间, 在周期模式下, 点击开始按钮会调用该函数.

**getTagByProcessType** 通过流程类型初始化流程标识,例如分析为”at”

**get\_period\_process\_id** 在周期模式下, 一次流程运行结束后会调用该函数, 该函数确认下一次运行的是什么流程,其id等于多少, 且如果是标样核查, 还需要将当前时间加上6小时,然后保存到Sys.checkEndTime变量

**get\_auto\_check\_process\_id** 在周期模式下,当自动标样核查不合格时, 调用该函数判断当前需要执行校准流程还是执行标样核查流程. (自动标样核查不合格时需要反复进行校正与标样核查, 知道合格为止,或者超过6小时为止)

**get\_auto\_range\_process\_id** 开启自动量程切换时,当需要切换量程时,通过该函数获取需要执行的流程id, 比如执行量程为1的分析流程时,得出的结果高于量程1的最大值, 此时就需要执行量程为2的分析流程, 就是通过该函数进行判断,并获取流程id.

**get\_controled\_process\_id** 反控模式下获取流程id, 反控模式下要启动某个流程时,需要有流程类型以及量程,流程类型保存在Sys.controledProcessTypeId变量, 流程量程保存在Sys.controledRangeId变量

**get\_current\_process\_id** 在运行控制界面点击启动按钮时,就是调用该函数获取流程id,例如在手动模式下, 直接回返回第一流程的id

**set\_process\_edit\_state** 设置某些空间不可触摸, 之前主要用权限操作, 现在没用这个函数了

**excute\_process** 在on\_timer调用,运行状态下每1ms调用一次

**process\_ready\_run** 在待机模式下,会一直调用该函数, 该函数会一直获取流程id, 当能够获取到流程id后, 会将系统状态设置为运行.

**LoadActionStr** 在process\_ready\_run中调用,该函数首先会从sd卡中的config文件夹读取流程配置文件,文件名与流程id相等, 在这个函数中会将配置文件中的参数读取出来保存到Sys表中的相关变量里, 在执行流程的时候就直接从Sys相关变量中获取了. 此外,在该函数中,还计算了运行该流程需要消耗多少试剂,如果开启了试剂余量告警,且试剂不够用了, 流程不会运行.

**SystemStop** 流程停止函数

**excute\_init\_process** 执行初始化流程

**excute\_inject\_add\_process** 执行注射泵加液流程

**excute\_peristaltic\_process** 执行蠕动泵加液流程

**excute\_dispel\_process** 执行消解流程

**excute\_read\_signal\_process** 执行读取信号流程

**excute\_calculate\_process** 执行计算流程

**excute\_valve\_ctrl\_process** 执行阀操作流程

**excute\_wait\_time\_process** 执行等待时间流程

**excute\_linear\_set\_process** 执行线性核查稀释流程

**hand\_control\_inject1** 手动操作控制注射泵1

**hand\_set\_led\_current** 手动操作设置LED电流

**hand\_get\_voltage** 手动操作获取电压

**hand\_get\_temperature** 手动操作获取温度

**getReagentIdByName** 通过试剂名称获取试剂id

**setPwmOutput** 设置pwm输出

**setPwmAdjust** 设置pwm中4/20mA校正值

**远程控制相关函数**

**scan\_ap\_fill\_list** 扫描wifi并显示

**upload\_screen\_pic** 截取屏幕发送给远程上位机

**upload\_pic\_after\_10ms** 10ms后自动截取参数指定的区域发送给远程上位机

**upload\_middle\_area\_after\_10ms** 10ms后自动截取中间的区域发送给远程上位机

**upload\_bottom\_status\_area** 截取底部状态栏信息发送给远程上位机.

**parse\_xy** 解析远程上位机发送过来的xy坐标

**click\_menu\_button** 在parse\_xy函数中调用,判断为点击了菜单按钮,在该函数中会判断点了哪个菜单按钮,然后进行相应的界面切换, 10ms后再讲画面截图传送给远程上位机, 这里需要延时10ms是因为调用调用界面切换函数screen\_change后,界面不会立马切过去,有一个响应时间.

**配置文件相关函数**

**changeCfgFileLanguage** 将配置文件进行中英文相互转换

**getFileNameByProcessName** 通过流程名称查找该流程对应的文件名称(流程id)

**ReadActionToTabAndScreen** 将文件中的每一个Action加载到ActionStrTab表并显示到界面 当中,配置文件中,每一行代表一个Action, 每一行都有一对<type>标签与<content>标 签,type标签保存的是动作类型.content保存的是按钮值与文本内容.

**WriteActionStrTabToFile** 将流程编辑界面1/2/3/4中的内容保存到文件当中, 当在流程编辑界面点击保存按钮时会调用该函数.

**WriteTypeAndContentToActionStrTab** 将流程编辑界面1/2/3/4中的内容写入ActionStrTab表中,在流程编辑子界面当中点击确认按钮时,会调用该函数,例如在注射泵加液界面点击确认按钮.

**MoveActionStrTabElement** 移动ActionStrTab表中的元素,当在流程编辑界面点击插入或者删除按钮时会调用该函数.

**SetActionToScreen** 将ActionStrTab中的内容设置到界面当中的内容加载到界面当中,在流程编辑界面点击”编辑”按钮时会调用该函数.

**readFileToTab** 将文件中的内容读取到tab当中,每一行对应了一个table元素.

**writeTabToFile** 将tab中的元素按写入文件当中,每一个元素占用一行.

**ConfigFileCopy** 复制配置文件,需要传入被复制的文件名与复制后的文件名.

**数据记录控件相关函数**

**saveSysInfo** 保存系统信息到记录空间,包括仪器型号/仪器序列号/系统语言

**saveNetInfo** 保存网络设置,包括wifi名称/密码/tcp服务器地址/端口号/远程升级地址.

**saveMainScreenInfo** 保存首页信息,包括分析时间/分析结果/E1/E2等.

**saveRunCtrlInfo** 保存运行控制界面参数

**saveRunCtrlPeriodInfo** 保存运行控制-周期设置界面参数

**saveRunCtrlHandInfo** 保存运行控制-手动设置参数

**saveRunCtrlTimedInfo** 保存运行控制-定时设置参数

**saveProcessSetInfo** 保存流程设置1/2参数

**saveRangeSetInfo** 保存量程设置参数

**saveHandOperation2Info** 保存手动操作2界面参数

**saveHandOperation3Info** 保存手动操作3界面参数

**saveInOutInfo** 保存输入输出界面参数

**saveStatusInfo** 保存系统状态

**add\_history\_record** 增加一条历史记录,参数为历史记录类型,例如分析/核查/日志等.

**getMaxNumByScreen** 通过该函数获取历史记录的最大能记录的条数,这个是自己配置的.

**getFilePathByScreen** 针对不同的历史记录界面,返回其对应的文件路径.

**checkHistoryFile** 用于检测历史文件是否存在,如果不存在,则创建一个空的文件.

**setHistoryScreen** 将最近的10条历史记录显示到对应的界面上

**clearHistoryFile** 清楚历史记录文件

**showHistoryByScreenAndPage** 跳转到指定页时调用该函数

**exportHistory** 导出历史记录时调用该函数

**其他函数**

**operate\_permission\_detect** 操作权限检测

**UpdataDriverBoard** 通过wifi升级驱动板

**set\_unit** 设置单位

**set\_equipment\_type** 设置一起型号

**SetSysUser** 设置系统用户

**Decimal\_BCD** 十进制数转BCD码, 例如11转为0x11

**BCD\_Decimal BCD**码转为十进制数,例如0x11转为11

**Split** 字符串分隔函数,将字符串按照制定的字符(逗号)进行分割, 并将分割后的字符存入table当中

**GetSubString2** 遍历历字符串，截取字符串1与字符串2之间的字符串

**GetSubString** 遍历历字符串，截取字符串1与字符串2之间的字符串, 与**GetSubString2**的实现方式稍有区别

**bcd\_to\_string** 将BCD码转换为字符串格式,例如将0x11转换为”11”

**hexToFloat** 将16进制数转换为浮点数

**FloatToHex** 将浮点数转换为16进制数

**encryptChar** 单个字符加密函数

**decodeChar** 单个字符解密函数

**encryptStr** 字符串加密函数

**decodeStr** 字符串解密函数

**GetPreciseDecimal** 保留小数点后n位

**right\_shift\_one** 右移一位

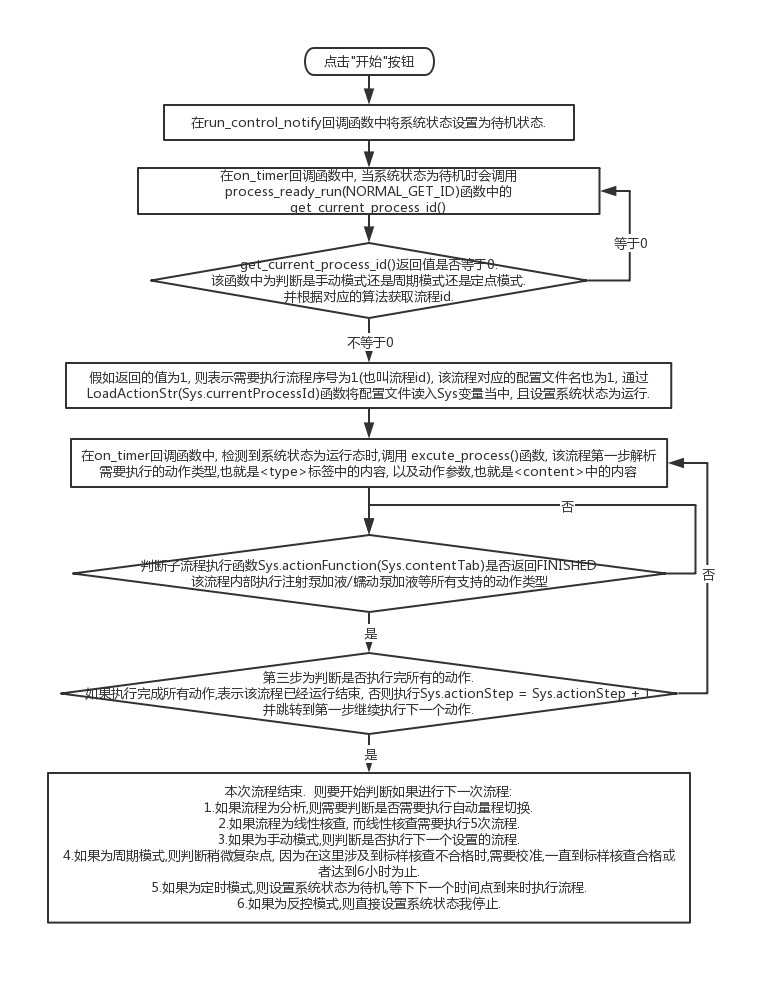
**right\_shift** 右移n位

**CalculateCRC16** 计算校验码

# 四 部分功能设计

**流程运行**

* 流程配置文件保存在sd卡中的config目录, 文件名为1,其对应序号为1的流程.依次类推. 文件名为0的配置文件中保存的是流程设置1/2界面中的参数. 程序依赖SD才能运行,所以建议出货时sd卡上热熔胶固定.



* 手动操作界面中的流程控制,列如手动操作一种的移动注射泵,当点击发送按钮时,其函数调用流程为:on\_control\_notify -> hand\_operate1\_control\_notify -> Sys.hand\_control\_func = hand\_control\_inject1, 而在on\_timer 中会一直检测Sys.hand\_control\_func是否为nil,当其不会nil时,会执行该函数Sys.hand\_control\_func()

**历史记录**

* 所有的历史记录都通过调用add\_history\_record接口实现,添加一条历史记录则表示在文件末尾处添加一行.
* 历史记录保存在sd卡的record目录下不同的文件,可以通过查看getFilePathByScreen接口得知不同的历史记录对应的文件名.
* 历史记录在文本中以密文的方式保存,直接查看到的是乱码,只有从触摸屏的历史界面查看或者导出的历史记录才是明文.
* 导出历史记录就是将sd卡中的密文解密后再保存到u盘当中. 历史记录文件保存在record文件夹中, 配置文件保存在config文件夹中,由于未实现在sd卡或者在u盘实现创建文件的工具,所有需要在电脑端手动在sd卡或者u盘创建record与config文件夹.
* 所有的告警记录都同步保存到日志当中.
* 历史记录控件里的内容是不进行存储的,其显示的内容是从历史记录文件中读取的.

**保存界面参数的历史记录控件**

共有两个历史记录控件保存界面, 这两个控件都隐藏在系统信息界面,id号分别是25与26,

id为25的其保存的内容如下:

|  |  |
| --- | --- |
| 序号 | 内容 |
| 0 | 管理员密码,运维员密码,仪器型号,仪器序列号,系统语言 |
| 1 | Wifi名称,wifi密码,tcp服务器ip地址,tcp服务器端口,远程升级服务器地址 |

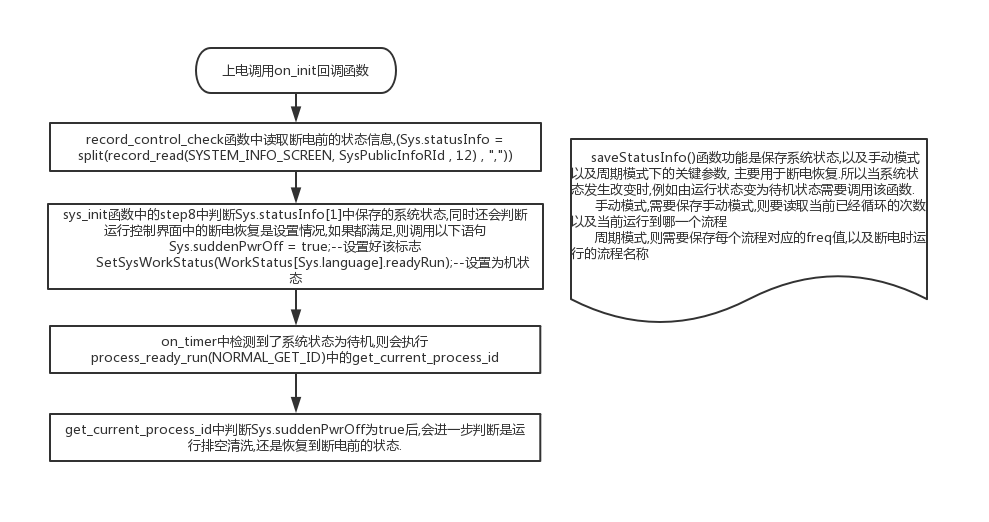
Id为26的控件其保存的内容如下:

|  |  |
| --- | --- |
| 序号 | 内容 |
| 0 | 首页信息 |
| 1 | 运行控制界面参数 |
| 2 | 周期设置界面参数 |
| 3 | 手动设置界面参数 |
| 4 | 定时设置界面参数 |
| 5 | 流程设置1/2界面的流程类型参数 |
| 6 | 流程设置1/2界面的流程名称参数 |
| 7 | 流程设置1/2界面的流程量程参数 |
| 8 | 量程设置界面参数 |
| 9 | 手工操作2界面参数 |
| 10 | 手工操作3界面参数 |
| 11 | 输入输出界面参数 |
| 12 | 系统状态,周期运行以及手动运行时的关键参数,主要用于断电恢复功能. |

**试剂余量检测**

* 试剂余量设置在”手工操作->试剂余量”界面中进行设置, 气泡检测功能暂未实现.
* 试剂余量检测功能可以通过”运行控制->试剂余量”进行关闭与打开
* 在注射泵加液界面或者蠕动泵加液界面中,有个试剂选项,选择试剂名称与体积,程序通过这个设置来判断执行该步骤需要消耗多少试剂.
* 在LoadActionStr函数中,会读取流程配置文件,解析里面的每一个动作, 当动作类型为注射泵加液或者蠕动泵加液时,则表示会消耗多少试剂,从而可以对试剂进行预算,看剩下的试剂是否够跑完一次流程
* 在程序试剂运行到注射泵加液excute\_inject\_add\_process()或者蠕动泵加液(excute\_peristaltic\_process)时,会实时减少试剂余量.

**断电恢复功能**

****

**反控模式功能**

* 反控模式主要是在该模式下解析上位机发送的1200指令,可参考<2300 水质在线自动监测仪动态管控通讯地址.xlsx>与<附件4 国家地表水自动监测仪器通信协议技术要求.docx>
* 在反控模式下隐藏了运行控制界面的开始按钮
* 是否为上位机发送的指令,是通过判断接受的首个数据是否与输入输出界面的设备地址来判断的,当接受到串口数据时会调用on\_uart\_recv\_data函数,判断为上位机发送的数后,会调用ComputerControl函数进行解析. 如果控制编码为1200,则表示为控制命令,如果命令为启动流程,则会调用process\_ready\_run(CONTROLLED\_GET\_ID)函数来解析需要启动什么类型的流程.

**周期模式下,自动标样核查不合格**

完整的需求如下:周期（自动）模式下，自动标样核查的结果不合格（超过“允许误差”）时，仪器执行自动校正并更新斜率（不管斜率更新选项为是或否），然后再次执行自动标样核查，如仍不合格，再次进行自动校正并更新斜率，如此反复连续进行。自首次标样核查开始起，如6 h内标样核查结果合格，则仪器继续按原频率设置执行后续任务，如超过6 h标样核查结果仍不合格，则终止周期（自动）模式，使仪器进入离线模式。

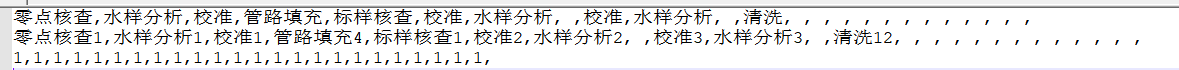


**智能互联功能**

* 智能互联功能已经能够解析上位机发送的x,y坐标, 并根据坐标做出相应的动作,目前只支持按钮类的单击动作.
* 为减少数据量将触摸屏分为三个区域.一为时钟区,该区域的截图每秒都会上传; 二为主显示区,该区域当发生按钮事件,且该事件会导致界面改变时,会将该区域的截图上传上位机显示;三为状态区,该区域在系统状态发生改变时,会将该区域的截图发给上位机显示.
* 上电后需要首先在系统信息界面的”无线-连接”连接公司wifi,连接成功后,再在”连接-服务器”界面连接TCP服务器,此时触摸屏会每隔10秒在on\_systick给上位机发送心跳包
* 当接受到上位机发送的远程控制命令后Sys.remoteControled = true, 此时不再上传心跳包,而是在on\_systick中每1秒上传右上角的时钟给上位机.
* 当底部状态栏发生改变时调用upload\_bottom\_status\_area上传底部的状态显示区域.例如当工作状态发生改变时会调用SetSysWorkStatus函数,而在该函数内部又会调用upload\_bottom\_status\_area函数,将当前状态截图发送给上位机同步显示
* 目前触摸屏cpu与wifi模块的通信方式是usb2.0
* 当上位机发送的x,y坐标对应的是切换界面的功能时,触摸屏会切换界面,并将新的界面截图发送给上位机实时显示. 假如当前触摸屏处于远程控制模式,且当前页面为首页. 当上位机发送的x,y坐标对应了首页的菜单按钮时,触摸屏处理数据的路径为:on\_client\_recv\_data->parse\_xy->click\_menu\_button,在click\_menu\_button中调用change\_screen函数切换界面,调用upload\_pic\_after\_10ms函数在10s后截图上传给上位机. 这里必须要等待一段时间后才能截图,因为调用change\_screen函数后界面不会立马变化.

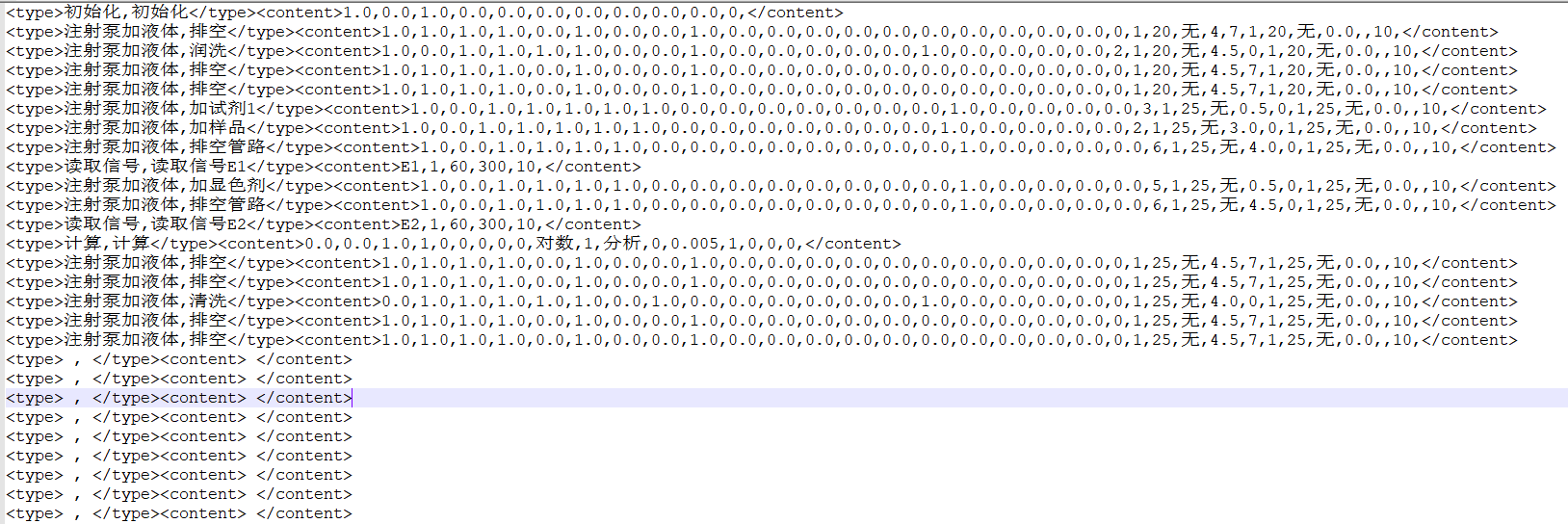
**配置文件说明**

配置文件0, 该文件在导入导出的时候使用,其保存的内容是流程设置1/2界面中的参数,保存在sd卡中的config文件夹中,其内容中的第一行为”流程类型”,第二行为”流程名称”,第三行为”流程量程”,如下图所示



在电脑端调试的时候,为了方便,在on\_init函数中读取将配置文件0的内容加载到流程设置1/2界面当中,当前已经注释掉.

配置文件1-24就对应的流程序号为1-24的流程的配置文件了, 下图为一个配置文件打开后的截图, 每个文件固定的有48行,表示每个流程最多可以设置48个动作,基本够用了.<type>标签里保存的是动作类型与动作名称,<content>标签里保存的是对应页面的按钮与文本的值,这里为了方便,每个页面的按钮与文本框的编号都是连续的,且从1开始编号的.当动作未设置时,其内容为<type> , </type><content> </content>,这里注意到共有三个空格,这些空格是一定不能省略的,因为其在读取的时候是作为判断条件的,main.lua文件使用BLANK\_SPACE宏表示一个空格.



# 五 如何增加一个动作类型(滴定)

第一步:

增加一个界面,命名为”流程设置-滴定”, 进行一次编译后,在右侧的”资源窗口”可以查看到该界面的编号为20, 所以在main.lua的顶部增加一个宏定义PROCESS\_TITRATION\_SCREEN = 0,并添加宏定义TITRATION\_TextStartId = 1, TITRATION\_TextEndId = 6; 其中1表示需要保存的文本框的开始编码,6表示需要保存的文本框的结束编码.

第二步:

定义子回调函数process\_titration\_control\_notify, 如下所示,主要定义了确认按钮与取消按钮时需要执行的代码

function process\_titration\_control\_notify(screen, control, value)

    if control == SureButtonId and value == ENABLE then --确认按钮

        if operate\_permission\_detect(CHK\_RUN\_USER) == ENABLE then--检测权限

            WriteTypeAndContentToActionStrTab(DestActionNum);

            change\_screen(DestScreen);

        end

    elseif control == CancelButtonId then --取消按钮

        change\_screen(DestScreen);

    end

end

在on\_control\_notify回调函数中增加一个段代码,如下所示,这样在当点击”流程设置-滴定”中的按钮时就会调用process\_titration\_control\_notify函数.

elseif screen == PROCESS\_TITRATION\_SCREEN then --流程设置-滴定

        process\_titration\_control\_notify(screen, control, value);

第三步:

在”动作选择”界面增加一个”滴定按钮”,将其编号设置为10. 同时在main.lua文件中找到名称为ActionItem的table,增加下标为10的元素,中文命名为”滴定”,英文命名为”Titration”

ActionItem = {

    [CHN] = {

        [1] = "初始化",

        [2] = "注射泵加液体",

        [3] = "读取信号",

        [4] = "蠕动泵加液",

        [5] = "计算",

        [6] = "等待时间",

        [7] = "消解",

        [8] = "阀操作",

        [9] = "线性核查稀释",

        [10]= BLANK\_SPACE,

     },

    [ENG] = {

        [1] = "Initialize",

        [2] = "Injector Add",

        [3] = "Read Signal",

        [4] = "Pump Add",

        [5] = "Calculation",

        [6] = "Wait Time",

        [7] = "Dispel",

        [8] = "Valve",

        [9] = "Linear Dilution",

        [10] = BLANK\_SPACE ,

    },

};

第四步:

在set\_edit\_screen中增加对滴定的判断, 这样在流程编辑1/2/3/4中点击”编辑”按钮时,当判断出该动作为”滴定”时,才会跳转到滴定界面.

elseif para == ActionItem[Sys.language][10] then --滴定

        change\_screen(PROCESS\_TITRATION\_SCREEN);

    end

第五步:

在parse\_xy函数中增加如下判断,这里是当远程控制时, 在”滴定”界面点击了确认或者返回按钮,会退出”滴定”界面,并将最新的界面10ms后发送给上位机显示.

elseif screen == PROCESS\_LINEAR\_CHK\_SET\_SCREEN then--流程设置-滴定

        if (x>=96 and x<=(96+109) and y>=657 and y<=(657+35)) or (x>=418 and x<=(418+109) and y>=655 and y<=(655+35)) then--确认/取消按钮

            change\_screen(DestScreen);

            upload\_middle\_area\_after\_10ms()

        end

第六步:

在WriteTypeAndContentToActionStrTab函数中增加如下if判断语句,其主要作用是,当在”滴定”界面点击确认按钮时,将该界面中的文本参数写入ActionStrTab变量中.

 --------------------------------写-滴定参数-------------------------------------

    elseif actionType == ActionItem[Sys.language][10] then

        for i = TITRATION\_TextStartId, TITRATION\_TextEndId, 1 do

            ActionStrTab[actionNumber] = ActionStrTab[actionNumber] .. get\_text(PROCESS\_TITRATION\_SCREEN, i) .. "," --写入文本值

        end

第七步:

在SetActionToScreen函数中增加如下if语句,其主要作用是点击”编辑”按钮,跳转到”滴定”界面时,将之前保存下来的文本值在界面上进行显示.

--------------------------------读-滴定参数-----------------------------

    elseif actionType[1] == ActionItem[Sys.language][10] then

        for i = LINEAR\_TextStartId, LINEAR\_TextEndId, 1 do

            if tab[i] ~= nil then

                set\_text(PROCESS\_TITRATION\_SCREEN, i, tab[i]);--写入文本值

            end

        end

第八步:

在excute\_process函数中添加如下代码,

elseif Sys.actionType == ActionItem[Sys.language][10] then--执行-滴定

            Sys.actionFunction = excute\_titration\_process;

并定义excute\_read\_signal\_process函数,这个函数目前还是一个空函数,需要完善,可以参考其他动作执行函数.

--执行滴定流程

function excute\_read\_signal\_process(paraTab)

    return Sys.actionSubStep

end