备注：本说明文档属于我方软件设计文件的第6、7章节，请按已给出的章节框架填写相应功能设计内容。

参考示例：

**第6章：**

6.3保存模块（JRAC.SAVE） //此处不要修改

6.3.1保存函数 saveTable2Excel()//此处写函数名称

6.3.1.1保存函数设计及约束

保存VS调试和精度调试的表格完整信息，只有在调试结束才可以保存。

//此处写函数的功能，在什么过程中使用

1. 调试过程保存。
2. 调试结束后。

6.3.1.2保存函数设计

1. 输入/输出元素 //下表填写形参和返回值

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 元素名称 | 元素标识 | 数据类型 | 数据长度 | 度量单位 | 极限值/值域 | 精确度 | 分辨率 | 用途 |
| 文件名称 | out | QTextStream | \ | 无 | 无 | 无 | 无 | 保存内容 |

1. 局部数据元素

无。//此处根据实际情况填写

1. 逻辑流程图



1. 数据结构

//VS1

QVector<double> original\_VS1;

QVector<double> b\_VS1;

QVector<double> r\_VS1;

QVector<double> differential\_VS1;

QVector<double> a\_VS1;

QVector<double> adjust\_VS1;

QVector<double> final\_VS1;

QVector<QString> status\_VS1;

QVector<QString> date\_VS1;

QVector<QString> worker\_VS1;

//VS2

QVector<double> original\_VS2;

QVector<double> b\_VS2;

QVector<double> r\_VS2;

QVector<double> differential\_VS2;

QVector<double> a\_VS2;

QVector<double> adjust\_VS2;

QVector<double> final\_VS2;

QVector<QString> status\_VS2;

QVector<QString> date\_VS2;

QVector<QString> worker\_VS2;

// 精度调试

//原值

QVector<double> originalVS1\_Jingdu;

QVector<double> originalVS2\_Jingdu;

QVector<double> originalFlow\_Jingdu;

//调整值

QVector<double> adjustVS1\_Jingdu;

QVector<double> adjustVS2\_Jingdu;

QVector<double> adjustFlow\_Jingdu;

//0周期

QVector<double> b0\_Jingdu;

QVector<double> r0\_Jingdu;

QVector<double> differential0\_Jingdu;

QVector<double> a0\_Jingdu;

QVector<double> accuracy0\_Jingdu;

//1周期

QVector<double> b1\_Jingdu;

QVector<double> r1\_Jingdu;

QVector<double> differential1\_Jingdu;

QVector<double> a1\_Jingdu;

QVector<double> accuracy1\_Jingdu;

//备注

QVector<QString> status\_Jingdu;

QVector<QString> date\_Jingdu;

QVector<QString> worker\_Jingdu;

//此处填写结构体

1. 使用的其它元素

QXlsx数据类型：用来存储为Excel表格。//此处根据实际情况填写

**第7章：**

7.1公式数据 vsformulaList //此处写数据名称

7.1.1数据设计说明

结构名称：QVector<QString> //此处写源码中公式数据名称

结构功能：计算公式 //此处写功能

说明： //此处可不填写

7.1.2详细定义

stack<char>Ch;//存储操作符

stack<char>Ch2;//存储ch的逆置，实现ch栈的遍历

stack<double>Num;//存储操作数

stack<double>Num2;//存储Num栈的逆置，实现遍历

class P{

public:

int flag;//记录当前为操作数还是操作符，0--数，1--符

double num;//0

char c;//1

};

P p[MAX\_LENGTH];//存储表达式

=================分割线=====以下为正文====================

6详细设计

6.1初始化模块（JRAC.INIT）

6.1.1主程序初始化 init()

6.1.1.1主程序初始化的设计及约束

1. 界面和参数初始化，绑定ip，配置文件读取；

initUI();//界面

//初始化其他设置窗口

initIpWidget();

initTieGroupWidget();

initUntieGroupWidget();

initWorkerWidget();

initVSFormulaWidget();

initTcpServer();

readConfig();//读取配置

1. 无约束。

6.1.1.2 初始化函数设计

1. 输入输出元素

无。

1. 局部数据元素

无。

1. 逻辑流程图



1. 数据结构

系统参数数据结构。

无共享数据结构。

1. 使用的其它元素

无其它元素。

6.2设备识别模块（JRAC.RCG）

6.2.1设备识别函数

6.2.1.1设备识别函数设计及约束

1. 识别设备ID；
2. 显示在线状态。

6.2.1.2设备识别函数设计

1. 输入/输出元素

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 元素名称 | 元素标识 | 数据类型 | 数据长度 | 度量单位 | 极限值/值域 | 精确度 | 分辨率 | 用途 |
| 执行  结果 | header | QByteArray | 6 | 无 | 无 | 无 | 无 | 识别设备ID |

1. 局部数据元素

无

1. 逻辑流程图

收到消息 -> 解析消息头 -> 识别出设备

1. 数据结构

无

1. 使用的其它元素

无其它元素。

6.3设备绑定模块（JRAC.BD）

6.3.1设备绑定函数 tieTwoMachine()

6.3.1.1设备绑定函数设计及约束

1. 绑定两个未绑定的设备
2. 无约束

6.2.1.2设备绑定函数设计

1. 输入/输出元素

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 元素名称 | 元素标识 | 数据类型 | 数据长度 | 度量单位 | 极限值/值域 | 精确度 | 分辨率 | 用途 |
| 执行  结果 | MachineID | QString | 不限 | 无 | 无 | 无 | 无 | 标识设备ID |

1. 局部数据元素

无

1. 逻辑流程图

选择两个设备 -> 点击绑定

1. 数据结构

无

1. 使用的其它元素

无其它元素。

6.4保存模块（JRAC.SAVE）

6.4.1保存函数 saveTable2Excel()//此处写函数名称

6.4.1.1保存函数设计及约束

保存VS调试和精度调试的表格完整信息，只有在调试结束才可以保存。

//此处写函数的功能，在什么过程中使用

1. 调试过程保存。
2. 调试结束后。

6.4.1.2保存函数设计

1. 输入/输出元素 //下表填写形参和返回值

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 元素名称 | 元素标识 | 数据类型 | 数据长度 | 度量单位 | 极限值/值域 | 精确度 | 分辨率 | 用途 |
| 文件名称 | out | QTextStream | \ | 无 | 无 | 无 | 无 | 保存内容 |

1. 局部数据元素

无。//此处根据实际情况填写

1. 逻辑流程图



1. 数据结构

//VS1

QVector<double> original\_VS1;

QVector<double> b\_VS1;

QVector<double> r\_VS1;

QVector<double> differential\_VS1;

QVector<double> a\_VS1;

QVector<double> adjust\_VS1;

QVector<double> final\_VS1;

QVector<QString> status\_VS1;

QVector<QString> date\_VS1;

QVector<QString> worker\_VS1;

//VS2

QVector<double> original\_VS2;

QVector<double> b\_VS2;

QVector<double> r\_VS2;

QVector<double> differential\_VS2;

QVector<double> a\_VS2;

QVector<double> adjust\_VS2;

QVector<double> final\_VS2;

QVector<QString> status\_VS2;

QVector<QString> date\_VS2;

QVector<QString> worker\_VS2;

// 精度调试

//原值

QVector<double> originalVS1\_Jingdu;

QVector<double> originalVS2\_Jingdu;

QVector<double> originalFlow\_Jingdu;

//调整值

QVector<double> adjustVS1\_Jingdu;

QVector<double> adjustVS2\_Jingdu;

QVector<double> adjustFlow\_Jingdu;

//0周期

QVector<double> b0\_Jingdu;

QVector<double> r0\_Jingdu;

QVector<double> differential0\_Jingdu;

QVector<double> a0\_Jingdu;

QVector<double> accuracy0\_Jingdu;

//1周期

QVector<double> b1\_Jingdu;

QVector<double> r1\_Jingdu;

QVector<double> differential1\_Jingdu;

QVector<double> a1\_Jingdu;

QVector<double> accuracy1\_Jingdu;

//备注

QVector<QString> status\_Jingdu;

QVector<QString> date\_Jingdu;

QVector<QString> worker\_Jingdu;

//此处填写结构体

1. 使用的其它元素

QXlsx数据类型：用来存储为Excel表格。//此处根据实际情况填写

6.5公式编辑模块（JRAC.FORMULA）

6.5.1公式编辑函数 无

6.5.1.1公式编辑函数设计及约束

1. 在编辑框中直接编辑
2. 公式合法

6.5.1.2公式编辑函数设计

1. 输入/输出元素

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 元素名称 | 元素标识 | 数据类型 | 数据长度 | 度量单位 | 极限值/值域 | 精确度 | 分辨率 | 用途 |
| 执行  结果 | 字符 | QString | 不限 | 无 | 无 | 无 | 无 | 编辑公式 |

1. 局部数据元素

无

1. 逻辑流程图

文本框中直接编辑 -> 点击存储即存储

1. 数据结构

无

1. 使用的其它元素

无其它元素。

6.6设置模块（JRAC.SET）

6.6.1设置函数 setting()

6.6.1.1设置函数设计及约束

1. 设置相关参数
2. 参数大小在有效范围内

6.6.1.2设置函数设计

1. 输入/输出元素

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 元素名称 | 元素标识 | 数据类型 | 数据长度 | 度量单位 | 极限值/值域 | 精确度 | 分辨率 | 用途 |
| 执行  结果 | debugValue | Float | 8 | 无 | 无 | 无 | 无 | 存储每个参数 |

1. 局部数据元素

无

1. 逻辑流程图

填写修改后的参数 -> 点击确定即存储

1. 数据结构

无

1. 使用的其它元素

无其它元素。

6.7 VS调试模块(JRAC.VS）

6.7.1 VS调试函数 startVS1() startVS2()

6.7.1.1 VS调试函数设计及约束

1. 通过交换报文完成调试过程
2. 设备在线

6.7.1.2 VS调试函数设计

1. 输入/输出元素

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 元素名称 | 元素标识 | 数据类型 | 数据长度 | 度量单位 | 极限值/值域 | 精确度 | 分辨率 | 用途 |
| 执行  结果 | MachineSocket | QTcpSocket | 无 | 无 | 无 | 无 | 无 | 存储socket套接字 |

1. 局部数据元素

无

1. 逻辑流程图

发送调试请求 -> 收到返回的报文 -> 根据逻辑判断是否继续请求

1. 数据结构

class Group : public QObject

{

Q\_OBJECT

signals:

void SendLog(QString, QString);//group + msg

public:

Group();

GroupInfo groupInfo;

AllData allData;

void login(MyThread \*machine); //绑定 设备a/b的线程

void logout(MyThread \*machine); //解除绑定 设备a/b的线程

int getOnlineStatus(); //获得在线的状态

void setMachineA(MyThread\* a);

void setMachineB(MyThread\* b);

QString getMachineA\_id();

QString getMachineB\_id();

void tie\_byID(QString gname,QString a, QString b); //通过id绑定设备组

void tie(QString gname, MyThread\* a, MyThread\* b); //通过线程绑定设备组

void untie(); //解绑该设备组

void setWorker(QString w); //设置工作人员

bool stop(); //中止当前调试进程

//data processing

void request\_b(); //寻求初值

void request\_a(); //寻求实际值

void request\_r(); //寻求末值

void analyzeData\_a(QByteArray data); //存储处理实际值

void analyzeData\_b(QByteArray data); //存储处理初值

void analyzeData\_r(QByteArray data); //存储处理末值

void returnFinalResult(int mode); //vs调试返回最终值给设备

void returnThreeResult(int mode); //精度调试返回调整后的三个值

void analyzeData\_answer();

void request\_buchong();

void receive\_buchong();

void tellToJD(); //通知设备A转化为精度调试

void saveData(FILE \*fp);

void readData(FILE \*fp);

bool meishuile;

private:

MyThread \*machineA;

MyThread \*machineB;

};

1. 使用的其它元素

无其它元素。

6.8精度调试模块（JRAC.ACCURACY）

6.8.1精度调试函数 startJingdu()

6.8.1.1精度调试函数设计及约束

1. 通过交换报文完成调试过程
2. 设备在线

6.8.1.2精度调试函数设计

1. 输入/输出元素

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 元素名称 | 元素标识 | 数据类型 | 数据长度 | 度量单位 | 极限值/值域 | 精确度 | 分辨率 | 用途 |
| 执行  结果 | MachineSocket | QTcpSocket | 无 | 无 | 无 | 无 | 无 | 存储设备的socket套接字 |

1. 局部数据元素

无

1. 逻辑流程图

发送调试请求 -> 收到返回的报文 -> 根据逻辑判断是否继续请求

1. 数据结构

class Group : public QObject

{

Q\_OBJECT

signals:

void SendLog(QString, QString);//group + msg

public:

Group();

GroupInfo groupInfo;

AllData allData;

void login(MyThread \*machine); //绑定 设备a/b的线程

void logout(MyThread \*machine); //解除绑定 设备a/b的线程

int getOnlineStatus(); //获得在线的状态

void setMachineA(MyThread\* a);

void setMachineB(MyThread\* b);

QString getMachineA\_id();

QString getMachineB\_id();

void tie\_byID(QString gname,QString a, QString b); //通过id绑定设备组

void tie(QString gname, MyThread\* a, MyThread\* b); //通过线程绑定设备组

void untie(); //解绑该设备组

void setWorker(QString w); //设置工作人员

bool stop(); //中止当前调试进程

//data processing

void request\_b(); //寻求初值

void request\_a(); //寻求实际值

void request\_r(); //寻求末值

void analyzeData\_a(QByteArray data); //存储处理实际值

void analyzeData\_b(QByteArray data); //存储处理初值

void analyzeData\_r(QByteArray data); //存储处理末值

void returnFinalResult(int mode); //vs调试返回最终值给设备

void returnThreeResult(int mode); //精度调试返回调整后的三个值

void analyzeData\_answer();

void request\_buchong();

void receive\_buchong();

void tellToJD(); //通知设备A转化为精度调试

void saveData(FILE \*fp);

void readData(FILE \*fp);

bool meishuile;

private:

MyThread \*machineA;

MyThread \*machineB;

};

1. 使用的其它元素

无其它元素。

7 CSCI 数据(以下内容请按本文开头示例编写，源码中所有与数据相关的结构都应列出，如公式参数数据、IP数据、VS参数数据、精度参数数据等)

7.1公式数据 //此处写数据名称

7.1.1数据设计说明

结构名称：Status\_Vol //此处写源码中公式数据名称

结构功能：计算公式 //此处写功能

说明： //此处可不填写

7.1.2详细定义

struct Status\_Vol

{

float cur\_vol; //当前输注量

float cur\_press; //当前压力值

};

7.2系统参数数据

7.2.1数据设计说明

结构名称：Systemparam

结构功能：系统参数

说明：

7.2.2详细定义

struct Systemparm

{

short sound\_vol; //提示音音量

bool bauto\_resorption; //自动回吸

bool bauto\_prime; //自动排气

};