**通信接口及协议配置组件项目**

**进度报告**

**凯云联创（北京）科技有限公司**

**2018年11月**

**修订记录**

**类别：**A-增加 M-修改 D-删除

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期** | **版本号** | **类别** | **描述** | **作者** |
| 2018-11-16 |  | A | 创建 |  |
| 2018-12-03 |  | A | 按用户反馈意见增加项目概要及项目工作分析相关内容 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 项目概要

## 项目基本信息

项目名称：通信接口及协议配置组件

用户方：中国工程物理研究院应用电子学研究所

开发方：凯云联创（北京）科技有限公司

## 项目期间

预计项目开始日期：2018年10月10日

预计项目截止日期：2019年01月25日

实际项目开始日期：2018年10月10日

实际项目截止日期：

项目目前状况：

（1）已交付内容:

项目目前C#版本已交付最初版本，期间有一次需求变动，需要继续增加功能，预计在2018年12月5日提交需求变动功能内容。

（2）在研内容

项目包括C++版本，目前处理在研状态。

## 项目成果

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 项目成果内容 | 预计交付日期 | 实际交付期 |
| 1 | 软件需求规格说明书 | 2018-10-29 | 2018-10-29 |
| 2 | 软件设计说明书 | 2018-10-29 | 2018-10-29 |
| 3 | 软件用户使用手册 | 2018-11-25 | 2018-10-29 |
| 4 | C#版本软件 | 2018-11-25 | 2018-10-29第一版  2018-12-05第二版 |
| 5 | C++版本软年 | 2019-01-25 |  |

# 项目工作分析

## 项目投入情况

项目总投入人员为5人，其中包括项目经理1人，研发人员3人、测试人员及文档编写人员1人。

## 项目需求变更

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 变更描述 | 变更时间 | 响应落实情况 |
| 1 | 增加子协议功能 | 2018-11-16 | 调配研发人员  增加研发内容 |

## 项目计划与进度实施

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 计划时间  周期 | 计划内容 | 实际成果 | 偏差说明 |
| 1 | 2018-10-10至  2018-10-12 | 1.需求分析  2.总体设计 | 完成 | 无偏差 |
| 2 | 2018-10-15至  2018-10-19 | 1. 软件界面框架搭建 2. C#通信接口定义编码 3. C#串口、网口功能实现 4. C#协议配置模块实现 5. CRC校验算法实现 | 完成 | 无偏差 |
| 3 | 2018-10-22至2018-10-26 | 1. C#协议生成模块实现 2. C#协议解析模块实现 3. C#CAN、DIO通信模块实现 4. C#源代码生成模块实现 | 完成 | 无偏差 |
| 4 | 2018-10-29至2018-11-02 | 1. 软件功能集成联调 2. 测试及相关文档编写 3. 用户试用 | 1. 功能联调联试 2. 软件测试报告编写 | 1. 功能联调完成。 2. 测试报告部分完成 |
| 5 | 2018-11-05至2018-11-09 | 1. 软件功能单元测试、相关文档编写。 2. 用户反馈问题分析及修改 | 1. 完成软件测试报告。 2. 用户反馈问题确认修改 | 反馈沟通确认 |
| 6 | 2018-11-12至2018-11-16 | 1. 反馈问题修改 2. 软件优化 3. 项目文档完善 | 1. 软件测试报告 2. 相关功能模块修改 | 无偏差 |
| 7 | 2018-11-19至2018-11-23 | 1. 反馈问题修改 2. 软件优化 3. 项目文档完善 | 1. 软件测试报告 2. 相关功能模块修改 | 无偏差 |
| 8 | 2018-11-26至2018-11-30 | 1. 需求变更增加，确定修改方案、模块研发 2. Bug修改 | 1.更改确定 | 无偏差 |
| 9 | 2018-12-03至2018-12-07 | 1. 需求变更模块研发 2. Bug修改 | 1.基本完成修改内容，达到使用状态 |  |
| 10 | 2018-12-10至2018-12-14 | 1. C++版本技术选型 2. C++版本开发环境搭建 3. 试验代码编写 | 1. 确定开发技术 2. 确定开发工具 |  |
| 11 | 2018-12-17至2018-12-21 | 1. C++版本串口、网口开发 2. 通信协议配置模块开发 3. 相关文档编写 |  |  |
| 12 | 2018-12-24至2018-12-29 | 1. C++版本CAN口开发 2. 通信协议配置模块开发 3. 通信协议解析模块开发 4. 相关文档编写 |  |  |
| 13 | 2019-01-02至2018-01-04 | 1. C++DIO接口模块研发 2. 通信协议配置模块研发 3. 通信协议解析模块研发 4. 相关文档编写 |  |  |
| 14 | 2019-01-07至2018-01-11 | 1. C++通信接口测试、联调 2. 通信协议配置研发、联调 3. 通信协议解析研发、联调 4. 代码生成模块研发 5. 相关文档编写 |  |  |
| 15 | 2019-01-14至2018-01-18 | 1. 代码生成模块研发、联调 2. 软件单元测试、集成测试 3. 相关文档编写 |  |  |
| 16 | 2019-01-21至2018-01-25 | 1. 集成测试 2. 相关文档编写 |  |  |

## 项目目前状况

此项描述项目前的所有问题，已解决的问题说明解决情况，时间，人员等，未解决的给出解决措施，时间和人员安排。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 问题描述 | 解决状态 | 备注 |
| 1 | 生成代码中枚举不再作为int使用 | 已解决 |  |
| 2 | ui操作模式下增加注释功能 | 已解决 |  |
| 3 | 修正第一次使用时使用datagrid出现的问题 | 已解决 |  |
| 3 | 生成的代码缩进全部使用4个空格 | 已解决 |  |
| 4 | 包长校验增加对部分字段校验的功能 | 已解决 |  |
| 5 | 设置自动计算包长功能 | 已解决 |  |
| 6 | datagrid插入新行时自动复用上一行数据 | 未解决 | 暂时手工处理，项目升级时再彻底解决 |
| 7 | 增加子系统功能 |  | 正在处理 |
| 8 | 分系统属性增加枚举类型 | 已解决 |  |
| 9 | 分系统属性 set设置为私有 | 已解决 |  |
| 10 | 通道设置全部增加超时时间，有默认值，读取的时候超时抛异常 | 已解决 |  |
| 11 | 修复校验算法bug | 已解决 |  |
| 12 | 修复图形界面下保存会失效bug | 已解决 |  |
| 13 | 运行时，runtime错误提示进行细化 | 已解决 |  |

## 风险管理实施情况

此项描述项目的风险分析情况，包括风险内容，发生概率，严重程度，缓解措施，发生后的应急措施等要素。

|  |  |  |
| --- | --- | --- |
| 序号 | 风险要素 | 描述 |
| 1 | **风险内容** | 项目周期短 |
| **发生概率** | 已存在 |
| **严重程度** | 严重 |
| **缓解措施** | 采用现有的、成熟的软件模块功能，抽取出运用到项目中。 |
| 2 | **风险内容** | 需求变更，增加子协议研发工作量，托后交付工期。增加测试维护工作量。软件由于需求改动引起的不稳定。 |
| **发生概率** | 已存在 |
| **严重程度** | 一般 |
| **缓解措施** | 1. 增加研发人员投入 2. 研发人员增加工作时长 3. 加大注重测试验证过程，提高测试验证覆盖率。 |

# 项目总体进度

|  |  |  |  |
| --- | --- | --- | --- |
| **项目名称** | 通信接口及协议配置组件 | **项目期间** | 开始日：2018年10月1日  结束日：2018年XX月XX日 |
| **项目经理** |  | **填写日期** | 2018年11月16日 |
| **计划项目时间(天)** | 90 | **已经实施时间(天)** | 40 |

|  |  |  |
| --- | --- | --- |
| **起始日期** | **结束日期** | **项目总体计划及阶段（里程碑）** |
| 2018年10月1日 | 2018年10月29日 | C#版本设计、编码、测试、项目文档、交付 |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **起始日期** | **结束日期** | **项目总体计划已完成部分** |
| 2018年10月1日 | 2018年10月29日 | C#版本设计、编码、测试、项目文档、交付 |
|  |  |  |

|  |
| --- |
| **余下主要工作内容** |
| 1.C++版本设计、编码、测试 |

|  |
| --- |
| **其它需要说明的重要问题** |
|  |

# 具体项目阶段及内容

## 项目设计阶段

项目设计阶段的主要工作成果是通信接口及协议配置组件的软件需求文档、项目设计文档。需求文档描述了软件所要完成的功能需求，为设计提供依据。软件设计文档详细设计软件的各个功能模块的具体应用技术及实现方法。

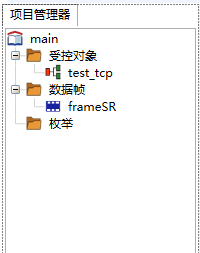
软件设计的功能模块包括通信模块设计、配置项内容设计、配置项编辑方式设计、配置项处理设计、代码生成设计、异常处理设计及软件界面设计。

### 通信模块设计

通信模块提供独立的底层通信接口，实现对硬件通道进行输入输出操作的调用。通信接口类型包括TCP（服务器和客户端）、UDP、串口、CAN和DI/DO。其中CAN通信接口实现对广州致远和研华两个厂家设备驱动的封装，DI/DO接口实现对研华厂家设备驱动的封装。

### 配置项内容设计

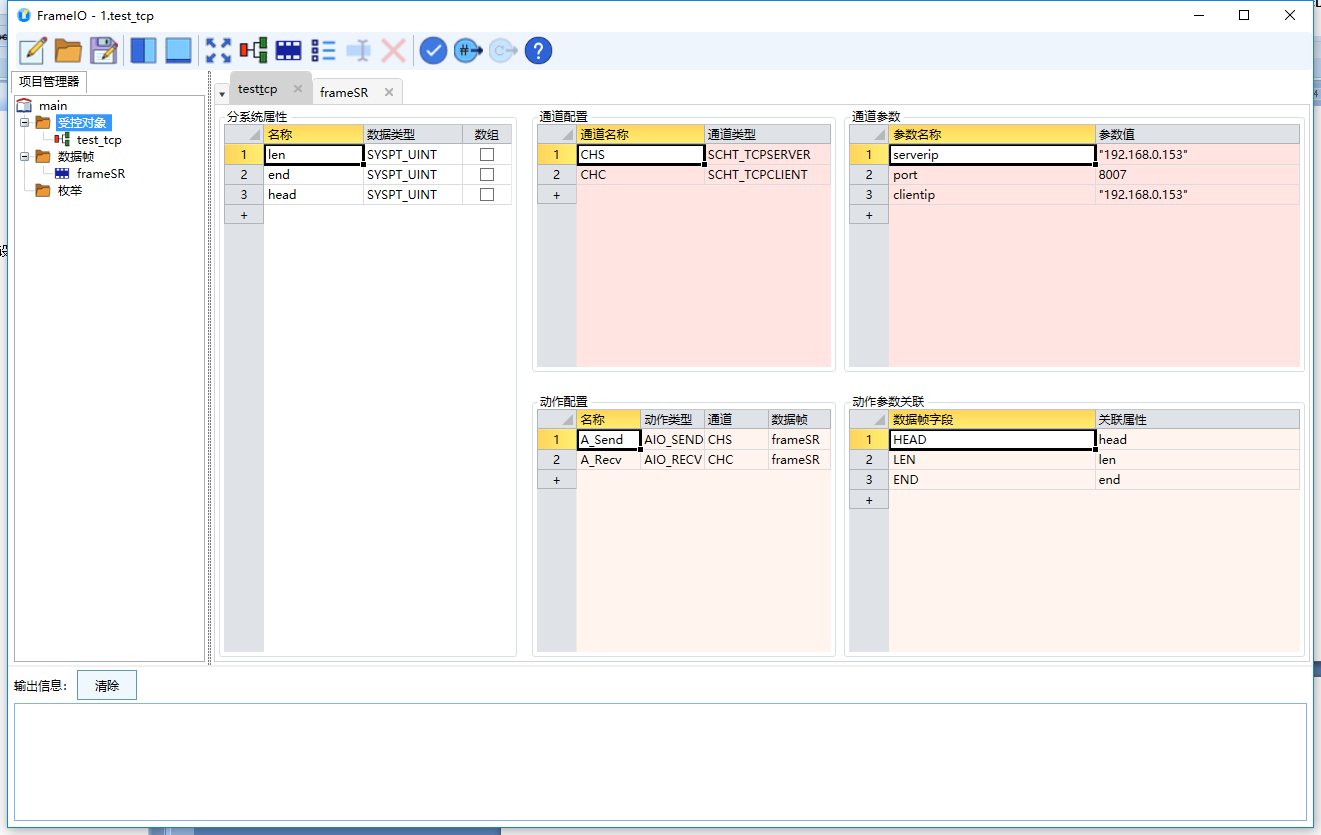
配置内容分为受控对象配置项、数据帧配置项和枚举配置项三类，其中核心的配置项是受控对象配置项，数据帧与枚举配置项用以辅助对受控对象进行配置，配置项的组成关系如下图：



### 配置项编辑方式设计

1. **可视化编辑模式**

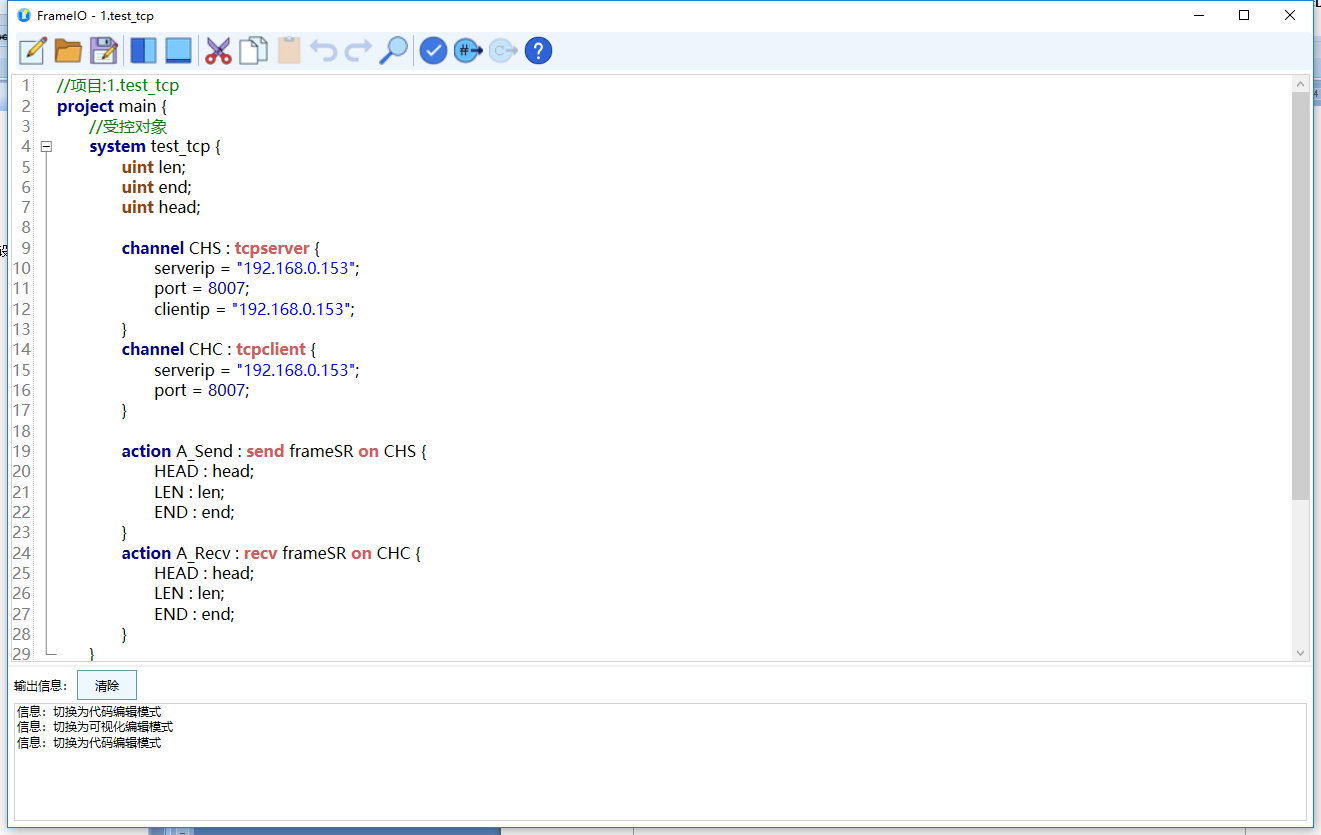
可视化编辑模式中，对所有配置项的编辑均通过windows系统中的标准控件进行，通过直观易操作的界面实现全部配置项的编辑功能。用户通过选择图形界面上的菜单、工具栏按钮控制配置项的添加、删除、重命名，配置项的具体内容通过UI表单中的输入框、复选框、下拉框等控件组合进行控制。

****

1. **代码编辑模式**

代码编辑模式中，主工作界面是一个所见即所得的代码编辑器，在代码编辑器中加载完整的配置代码内容，通过代码编辑器可直接修改配置文件的原始文本。

代码编辑器除支持中英文输入，支持文本内容复制、粘贴、剪切功能，支持按修改步骤执行撤销、恢复功能。代码编辑器最左侧一列显示配置代码行号；代码行号右侧为配置代码内容，根据自定义语法规则，配置代码的不同关键词显示不同的高亮颜色，方便使用者识别，以增强配置代码的可读性。

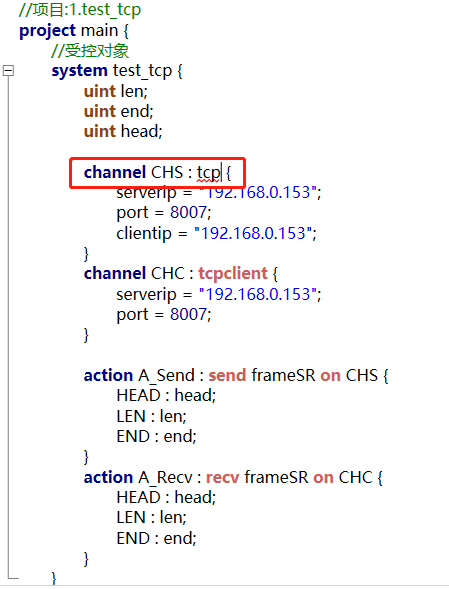


### 配置项处理设计

配置项处理由两个主要环节：语法检查与配置项解析。

1. **语法检查**

语法语义判别通过BNF语法分析，对配置代码中的完整配置项进行逐项识别，识别完成后按照识别的语义进行合规性判别，对出现的不合规语法或语义冲突均记录为一项语法错误。以红色波浪线提示。



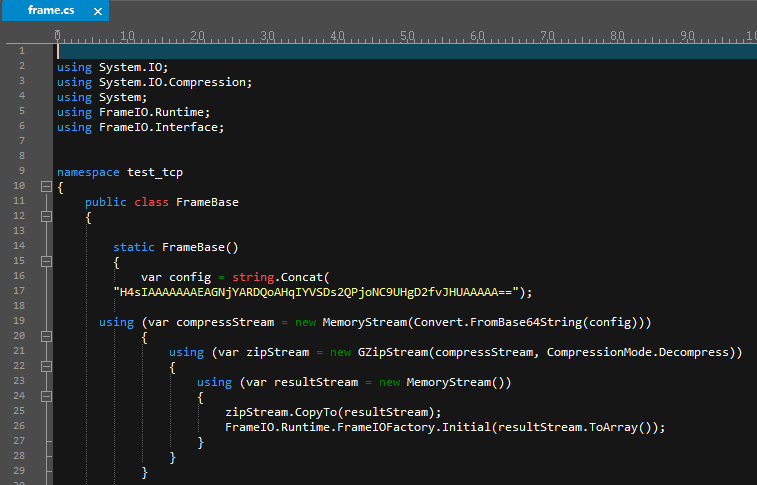
1. **配置项分析**

配置项解析是通信配置组件产生生成代码的基础，配置项解析应确保准确。该过程由通信配置组件后台自动调用，对使用者是透明的。

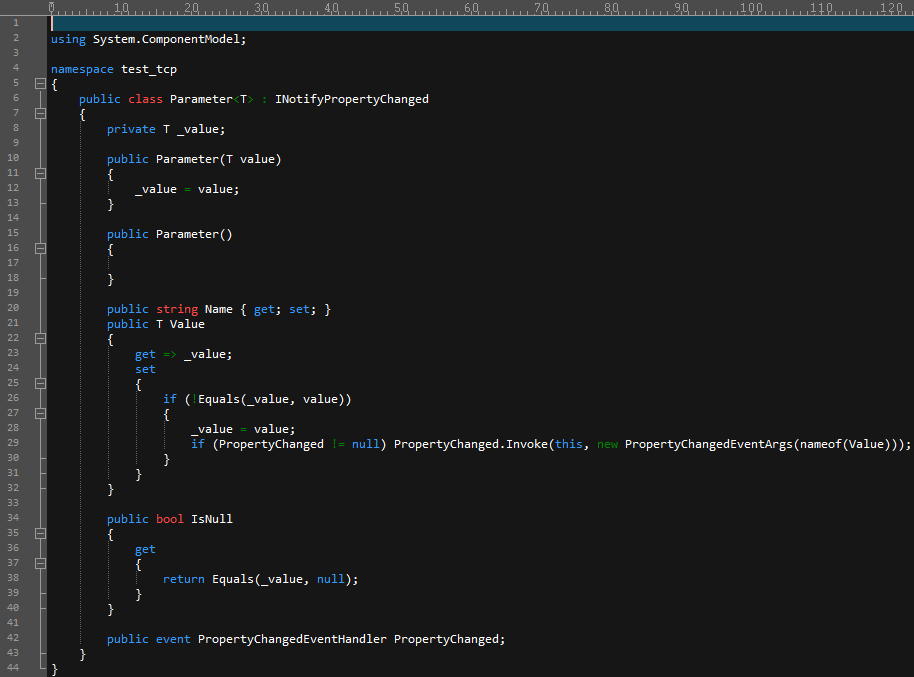
### 代码生成设计

由通信配置组件自动生成的程序代码中包含分系统类代码、数据帧类代码和枚举定义代码，分别由配置项中的受控对象配置项、数据帧配置项和枚举配置项对应产生。

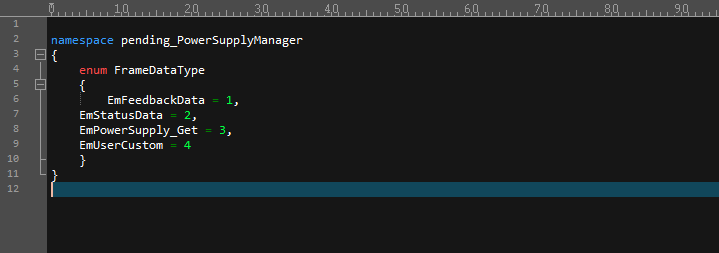
1. **分系统类代码（摘录部分代码）**



1. **数据帧类代码（摘录部分代码）**

****

1. **枚举定义代码（摘录部分代码）**

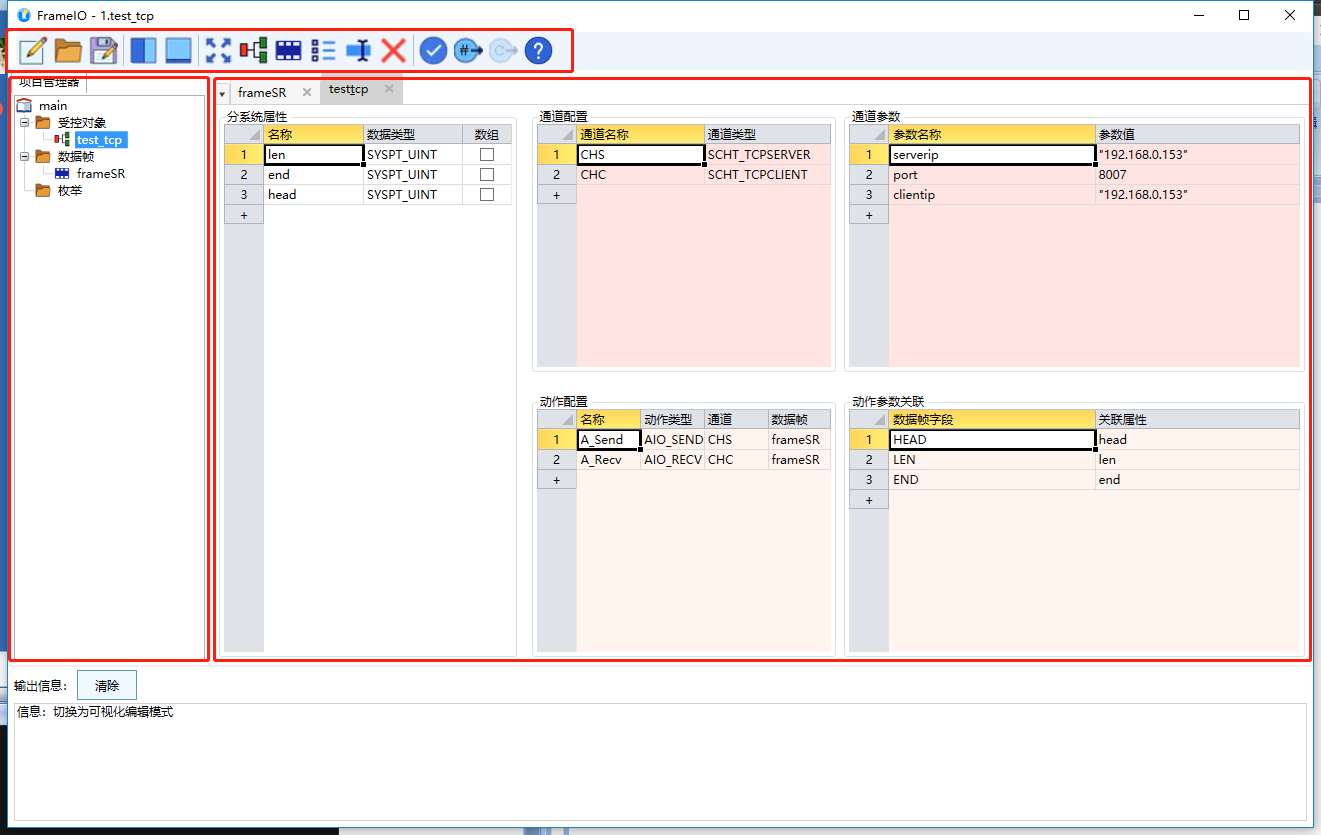
****

### 异常处理设计

在协议配置组件运行过程中需要处置各种可能出现的错误，比如通信模块调用硬件驱动时可能会收到系统内核返回的错误码，数据帧解析过程中可能会出现解析失败或校验错误，生成的程序代码运行时也会有很多运行时错误；对于可能出现的错误协议配置组件以统一的方式进行处理。

### 软件界面设计

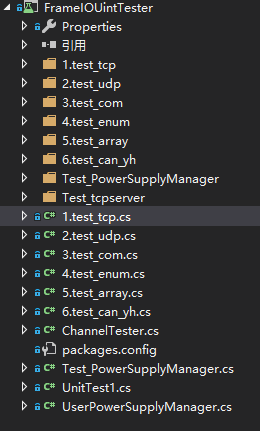
软件界面主要包括功能主菜单、项目管理器及分项子界面组成。



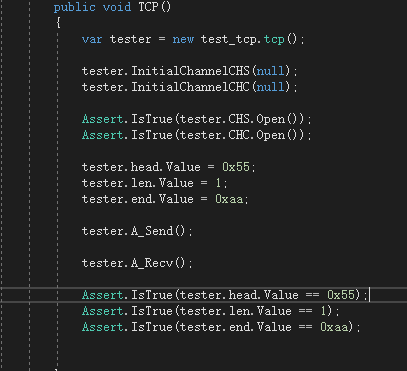
## 项目编写、测试阶段

项目编写主要完成了C#代码的编写。测试采用单元测试用例方式对编写的模块代进行测试。

1. **单元测试项目（结构）**



1. **单元测试内容（摘录）**



## 项目文档制作阶段

本项文档包括软件需求规格说明书、软件设计说明书、软件用户手册