**【项目名称】**

**概要设计说明书**

|  |  |  |  |
| --- | --- | --- | --- |
| 文档版本号： | V1.0 | 文档编号： | NG\_TS\_TEMP\_SDS |
| 文档密级： | 内部公开 | 归属部门/项目： | 专业系统部 |
| 编写人： |  | 生效日期： | 2017-09-20 |

**版权信息**

本文件涉及之信息，属南京轨道交通系统工程有限公司所有。

未经南京轨道交通系统工程有限公司允许，文件中的任何部分都不能以任何形式向第三方散发。

网址：http://www.nanjingrail.com/

**文档修订记录**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **版本号** | **修订日期** | **修订人** | **修订说明** | **修订状态** | **审核日期** | **审核人** | **批准人** |
| V1.0 | 2017-12-28 |  | 正式版 | A |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

修订状态：A--增加，M--修改，D--删除

日期格式：YYYY-MM-DD

**目　录**

[1. 引言 1](#_Toc287377107)

[1.1. 编写目的 1](#_Toc287377108)

[1.2. 背景 1](#_Toc287377109)

[1.3. 术语 1](#_Toc287377110)

[1.4. 预期读者与阅读建议 1](#_Toc287377111)

[1.5. 参考资料 1](#_Toc287377112)

[2. 总体设计 2](#_Toc287377113)

[2.1. 设计概述 2](#_Toc287377114)

[2.1.1 设计约束 2](#_Toc287377115)

[2.1.2 设计策略 2](#_Toc287377116)

[2.1.3 设计实现 2](#_Toc287377117)

[2.2. 设计目标 2](#_Toc287377118)

[2.3. 运行环境 2](#_Toc287377119)

[2.4. 平台架构 2](#_Toc287377120)

[2.5. 总体设计思路和处理流程 3](#_Toc287377121)

[2.6. 模块结构设计 3](#_Toc287377122)

[2.7. 功能需求与程序模块的关系（可选） 3](#_Toc287377123)

[2.8. 尚未解决的问题 3](#_Toc287377124)

[3. 接口设计 3](#_Toc287377125)

[3.1. 用户接口 3](#_Toc287377126)

[3.2. 外部接口 4](#_Toc287377127)

[3.3. 内部接口 4](#_Toc287377128)

[4. 界面总体设计 4](#_Toc287377129)

[5. 数据结构设计 4](#_Toc287377130)

[5.1. 设计原则 4](#_Toc287377131)

[5.2. 数据库环境说明 4](#_Toc287377132)

[5.3. 数据库命名规则 4](#_Toc287377133)

[5.4. 逻辑结构 4](#_Toc287377134)

[5.5. 物理存储 4](#_Toc287377135)

[5.6. 数据备份和恢复 4](#_Toc287377136)

[6. 系统出错处理设计[可选] 4](#_Toc287377137)

[6.1. 出错信息 4](#_Toc287377138)

[6.2. 补救措施 5](#_Toc287377139)

[6.3. 系统维护设计 5](#_Toc287377140)

[7. 系统安全设计 5](#_Toc287377141)

[7.1. 数据传输安全性设计 5](#_Toc287377142)

[7.2. 应用系统安全性设计 5](#_Toc287377143)

[7.3. 数据存储安全性设计 5](#_Toc287377144)

[8. 系统部署（可选） 6](#_Toc287377145)

# 引言

* 1. 编写目的

本文档主要描述轨道交通ISCS系统 配置数据库的概要设计，并定义文档的适用范围和人员。

* 1. 背景

轨道交通ISCS系统具有多业务，多模块的特点，ISCS系统各功能模块都需要配置数据，为了统一并且便捷的管理配置数据，系统建立配置数据库管理系统对这些数据进行统一配置和管理。根据业务的特点数据库采用关系型数据库XXX。

* 1. 术语

CFDB Config Database 配置数据库

DQL Data QueryLanguage 数据查询语言标准语法

DB Data Base 数据库

DBMS Data Base Manager System 数据库管理系统

DBS Data Base System 数据库系统

* 1. 预期读者与阅读建议

描述本文档的主要读者，以及这些读者在阅读时的阅读重点与建议。可用列表的方式列出。如表：

|  |  |
| --- | --- |
| 预期读者 | 阅读重点 |
| 项目管理人员 | 仔细阅读文档的系统范围，编写目的，文档约定，系统功能介绍。 |
| 软件设计人员 | 仔细阅读全部内容。 |
| 软件开发人员 | 仔细阅读全部内容。 |
| 测试人员 | 仔细阅读全部内容。 |

* 1. 参考资料

列出有关的参考资料，如：

a、本项目经核准的计划任务书或合同、上级机关的批文；

b、不属于本项目的其他已发表的文件；

c、本文件中各处引用的文件、资料、包括所要用到的软件开发标准；

d、列出这些文件资料的标题、文件编号、发表日期和出版单位，说明能够得到这些文件资料的来源。

# 总体设计

* 1. 设计概述

配置数据库系统是综合监控系统的一部分，为综合监控系统提供配置数据和配置服务以及人机界面。配置数据库系统管理维护配置数据，包括数据同步，备份，导库比库等。提供统一的数据写入接口，数据控制接口和数据查询接口。

配置数据库系统在线路中心和车站分别部署配置数据库系统。

在车站内，根据需要部署两至多套配置数据库系统，组成站内的一主多备。车站配置数据库只管理维护本站配置数据，车站不对其它车站提供本站的配置数据服务。

在线路中心，根据需要部署两至多套配置数据库，组成线路中心的一主多备。线路中心的配置数据库只管理维护线路中心的设备、系统等的配置数据。

当车站中主配置数据库的数据发生变化时，配置数据库将变化的数据同步更新至其他车站的配置数据库。

### 2.1.1 设计约束

需求约束：

配置数据库表类型应包含字典表，基础信息表，模型数据表，专业数据表。

封装数据库的底层接口，对外提供通用的统一调用方法，这些调用方法应包含增、删、查、改，并能够根据应用需求，提供更加丰富的定制化接口，如批量操作的接口。

建模工具具备设备生成，由设备模版生成具体设备，按专业建模版，按车站生成设备，设备模版带有点模版信息，对于某一类设备模块，可添加、删除对应的点模版内容，可由设备模版生成具体的设备，生成的设备继承点信息，可批量生成设备，可实现站间设备拷贝，可批量导入导出点数据，可查看所有点的属性状态数据。

配置服务具备数据下装模块，将模型数据下装至实时库等模块。实现主备配置库数据同步，增加修改过数据后应当能主动同步到实时库。

辅助数据工具具备数据备份恢复功能（人工备份、自动备份、定时备份等，并提供数据恢复功能)，导库比库（应有导库功能可按专业、车站等删选条件进行自定义导库。应有比库功能，可比较两份数据的异同），需有可视化的表结构设计工具及可视化的数据查询修改工具。

模块化约束：配置数据库需要为多种不同的服务提供数据，为减少数据耦合，配置数据库根据不同服务所需的数据进行设计，保持数据的独立性。

易操作性约束：配置数据库含有大量的基础数据，在后期录入数据时就要求配置数据库提供简便可靠的操作方式。

### 2.1.2 设计策略

1.数据库统一接口

为加强配置数据库管理，为使用者提供统一的配置库API，使用者通过统一接口对数据库进行操作。

2.数据库备份恢复

为保证数据库数据安全，数据库支持人工备份、自动备份、定时备份等。

3.表结构设计

为了提高性能，减少表关联，允许数据适当冗余。

为了提高数据的可复用性，减少不必要的数据重复，对数据进行抽象再设计。

物理独立性: 存储结构变，逻辑结构可以不变，从而应用程序也不必改变。

逻辑独立性: 总体逻辑结构变，局部逻辑结构可以不变，从而应用程序也不必改变。

### 2.1.3 设计实现与限制

系统包含三大部分：

1.可视化工具集界面。

2.函数API（声明各个外部接口的头文件）、静态库（.a文件）和运行时动态链接库（.so文件）。

3.独立的配置库数据库管理软件，包含基于控制台的数据库管理软件和基于图形界面的数据库管理软件。可实现对历史数据库软件的基本管理，包括数据库的管理、数据表的管理、数据查询、数据更新、数据删除等操作。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **功能点** | **子功能** | **需求编号** | **优先级** | **内容描述** |
| 数  据  库 | 表结构设计 | CFDB-DB-001 | P2 | 数据库的表结构 |
| 触发器及存储过程 | CFDB-DB-002 | P2 | 定义数据库触发器及存储过程 |
| 数据备份与恢复 | CFDB-DB-003 | P2 | 定义数据库备份与恢复功能 |
| 数  据  库  接  口 | 数据查询 | CFDB-IDAT-001 | P2 | 数据库统一的查询接口 |
| 数据存储 | CFDB-IDAT-002 | P2 | 数据库统一的存储接口 |
| 数据控制 | CFDB-IDAT-003 | P2 | 数据库统一的控制接口 |
| 配  置  服  务 | 数据同步 | CFDB-SER-001 | P2 | 数据库数据变更后，将变更后的数据同步到所有配置数据库 |
| 数据下装 | CFDB-SER-002 | P2 | 将数据下装至实时库等模块 |
| 在线同步实时库 | CFDB-SER-003 | P2 | 数据库变更数据后，将变更后的数据同步到实时库 |
| 导库 | CFDB-SER-004 | P2 | 可自定义导库 |
| 比库 | CFDB-SER-005 | P2 | 比较两份数据异同 |
| 建  模  工  具 | 设备建模 | CFDB-MOD-001 | P2 | 建立设备模板 |
| ATS建模 | CFDB-MOD-002 | P2 | 建立ATS模板 |
| 公式定义 | CFDB-MOD-003 | P2 | 定义公式 |
| 报警定义 | CFDB-MOD-004 | P2 | 定义报警权限 |
| 用户定义 | CFDB-MOD-005 | P2 | 定义用户权限 |
| 通用配置定义 | CFDB-MOD-006 | P2 | 定义通用配置 |
| 其  他  工  具 | 日志查询工具 | CFDB-TOOL-001 | P2 | 查询系统日志的工具 |
| 点模拟工具 | CFDB-TOOL-002 | P2 | 模拟信号点的工具 |
| 检索器 | CFDB-TOOL-003 | P2 | 根据条件搜索数据的工具 |
| 辅助  工具 | 客户端接口 | CFDB-ASST-001 | P1 | 客户端接口 |
| 报警 | CFDB-ASST-002 | P2 | 异常报警推送至RTDB |
| 系  统  功  能 | 系统性能 | CFDB-PERF-001 | P4 | 系统运行性能指标 |
| 界面要求 | CFDB-MMI-002 | P2 | 人机界面 |
| 系统安全 | CFDB-SEC-003 | P1 | 用户登录安全控制 |
| 系统日志 | CFDB-LOG-004 | P1 | 本地系统日志管理 |

* 1. 设计目标

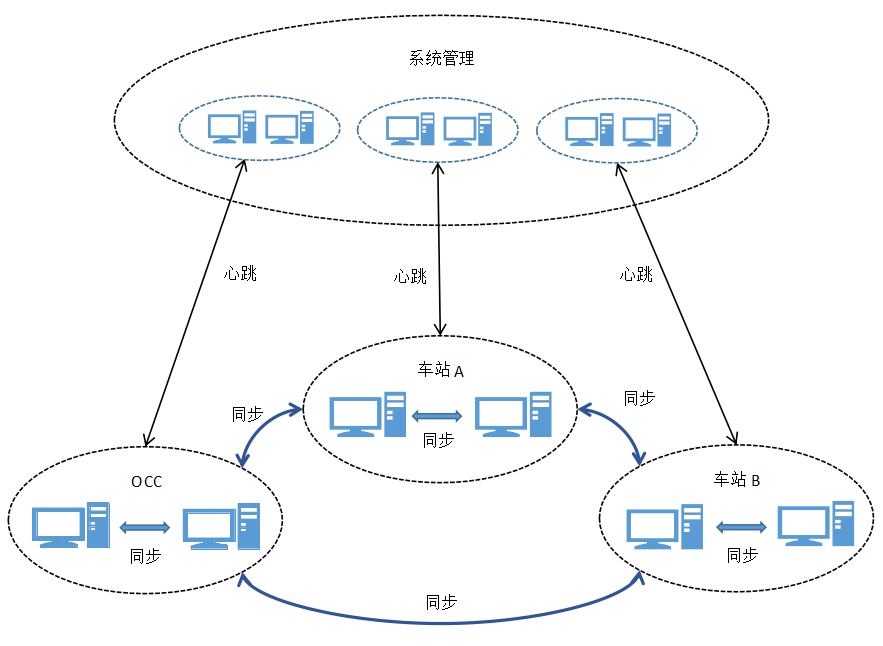
统一接口：配置数据库提供统一的操作接口。

查询速度：在同等内存和CPU资源下，配置数据库软件的查询速度不应随着数据规模的扩大而剧烈降低。

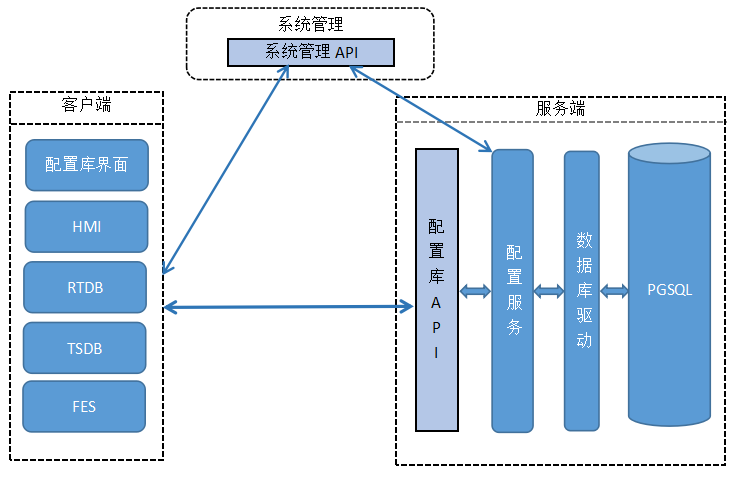
* 1. 运行环境

|  |  |  |
| --- | --- | --- |
| **类别** | **标准配置** | **最低配置** |
| 计算机硬件 | Intel Xeon 12核CPU（64  位指令集兼容）64GB内存/1000GB硬盘,两块相同的以太千兆网卡 | Intel Xeon 双核CPU（64  位指令集兼容）2GB内存/100GB硬盘,两块相同的以太百兆网卡 |
| 软件 | CentOS 7 x64 | CentOS 7 x64 |
| 网络通信 | TCP/IP协议栈支持 | TCP/IP协议栈支持 |

* 1. 平台架构

轨道交通综合监控系统中配置数据库以主，从冗余互备结构出现，配置库系统定时向系统管理发送心跳报文，由系统管理根据配置数据库系统的心跳报文以及权值分配主从角色，配置数据库根据主服选举的情况进行数据同步。

* 1. 数据处理流
     1. 数据流概图



* + 1. 数据写入流程

配置数据库数据写入流程如下：



流程描述如下：

1. 客户端通过SOCKET将HTTP协议将DATA发送至服务端；
2. 服务端跟据解析的数据调用API；
3. API处理数据，调用DB Driver；
4. DB Driver根据参数对PGSQL进行操作；
5. DB Driver返回执行结果；
   * 1. 数据读取流程

配置数据库数据读取流程如下：



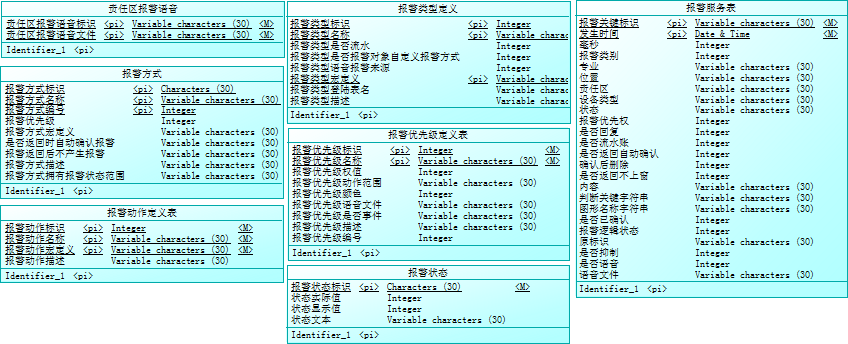
流程描述如下：

1. 客户端通过SOCKET将HTTP协议将数据发送至服务端；
2. 服务端跟据解析的数据调用API；
3. API处理数据，调用DB Driver；
4. DB Driver根据参数对PGSQL进行操作；
5. DB Driver返回数据。；
6. API调用SOCKET接口写入数据；
7. 客户端从SOCKET读取数据；
   1. 表结构设计

### 2.6.1.用户权限表结构



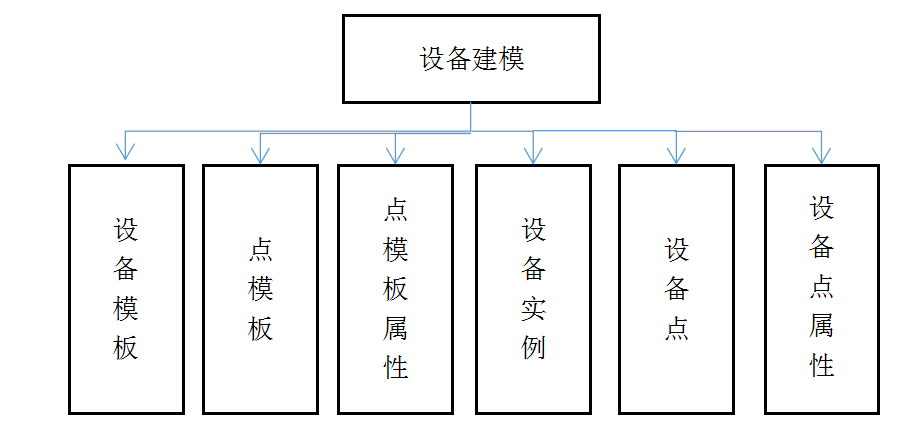
### 2.6.2.报警表结构



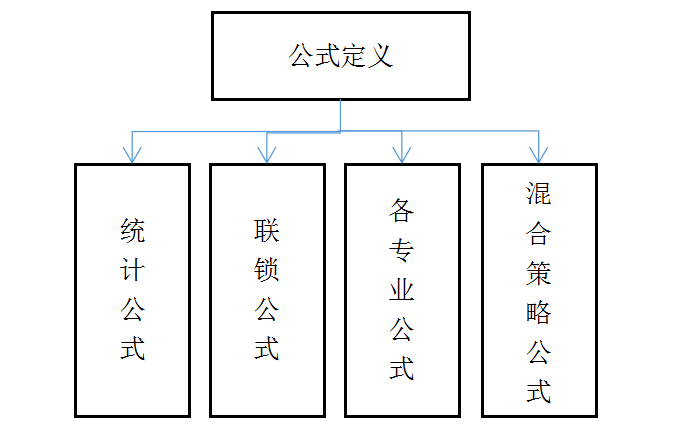
* 1. 模块结构设计

配置数据库工具包含设备建模，公式定义，报警定义，用户权限定义，通用模块，筛选器。

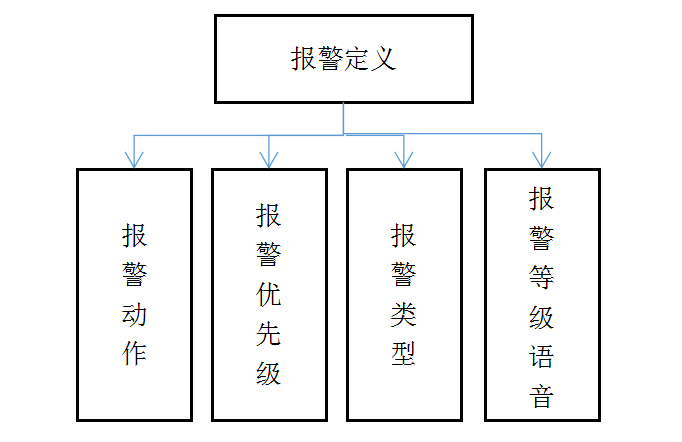
设备建模：



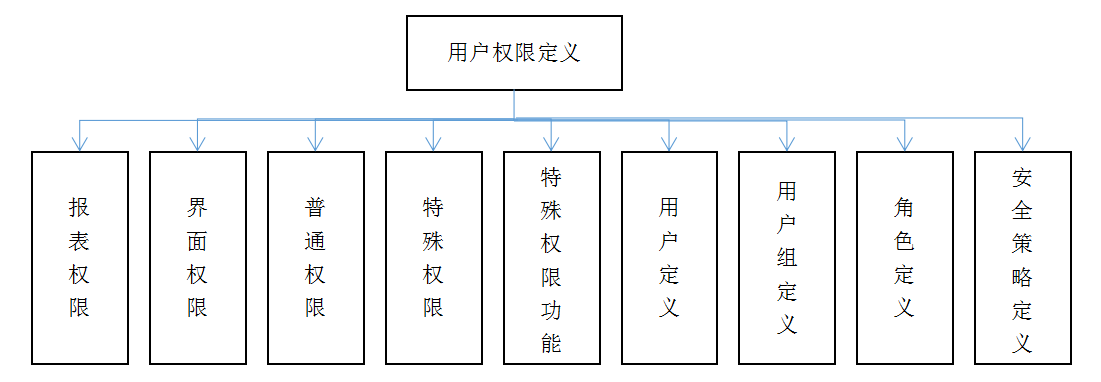
公式定义：



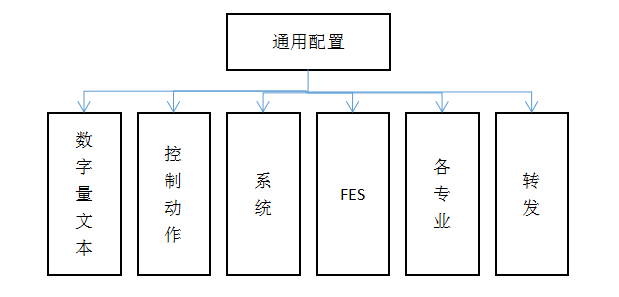
报警定义：



用户权限定义：



通用模块：



# 接口设计

* 1. 用户接口

说明与用户接口的输入输出关系，以及将向用户接口提供的命令和它们的语法结构，以及软件的回答信息。

* 1. 外部接口

说明本系统同外界的所有接口的安排包括软件与硬件之间的接口、本系统与各支持软件之间的接口关系、协议要求等。

* 1. 内部接口

说明本系统之内的各个系统元素（各层模块、子程序、公用程序等）之间的接口的安排，包括设计用关系、输入输出要求、语法结构等。

# 界面总体设计

说明界面总体布局和风格设计。

# 数据结构设计

可以在本文说明也可以单独使用数据库设计说明书描述

* 1. 设计原则

给出系统数据库的设计原则。

* 1. 数据库环境说明

简单介绍一些数据库直接有关的支持软件，如数据库管理系统、存储定位程序和用于装入、生成、修改、更新数据库的程序等。说明这些软件的名称、版本号和主要功能特性。

* 1. 数据库命名规则

联系用途，详细说明用于唯一地标识该数据库的代码、名称或标识符，附加的描述性信息亦要给出。

* 1. 逻辑结构

数据库设计人员根据需求文档，利用数据建模技术来描述逻辑数据库结构。要求使用PowerDesigner或Visio创建数据库PDM模型。此处只需列出PDM模型名称。

* 1. 物理存储

描述整个逻辑数据模型是如何被转换为数据文件（物理模式）。文件结构类型在这里应清楚的体现。

* 1. 数据备份和恢复

描述数据库的备份和恢复策略。

# 系统出错处理设计[可选]

* 1. 出错信息

用一览表的方式说明出错的类型，以及每种可能的出错或故障情况出现时，系统输出信息的形式、含意及处理方法。例如：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **错误分类** | **子项及其编码** | **错误名称** | **错误代码** | **备注** |
| 数据库错误 | 连接 | 连接超时 | 100001001 |  |
| 连接断开 | 100001002 |  |
| 数据库本身错误代码 | 数据库本身错误代码 | 100002+数据库错误代码 |  |
| TCP连接错误 | 连接 | 连接超时 | 101001001 |  |
| 连接断开 | 101001002 |  |
| 其它TCP连接错误(socket自身错误代码) |  | 101002+ socket错误代码 |  |
| 配置信息错误 | 未配置输入参数 |  | 102001 |  |
| 未配置输出参数 |  | 102002 |  |

* 1. 补救措施

说明故障出现后可能采取的变通措施，包括：

a．后备技术 说明准备采用的后图示技术，当原始系统数据万一丢失时启用的副本的建立和启动的技术，例如周期性地把磁盘信息记录到磁带上去就是对于磁盘媒体的一种后备技术；

b．降效技术 说明准备采用的后备技术，使用另一个效率稍低的系统或方法来求得所需结果的某些部分，例如一个自动系统的降效技术可以是手工操作和数据的人工记录；

c.恢复及再启动技术 说明将使用的恢复再启动技术，使软件从故障点恢复执行或使软件从头开始重新运行的方法。

* 1. 系统维护设计

说明为了系统维护的方便而在程序内部设计中作出安排，包括在程序中专门安排用于系统的检查与维护的检测点和专用模块。

# 系统安全设计

* 1. 数据传输安全性设计

说明在数据通信和传输过程中安全性设计。

* 1. 应用系统安全性设计

说明在访问应用系统过程中用户以及访问权限、操作等安全性设计。

* 1. 数据存储安全性设计

说明在数据和文件在存储过程中的安全性设计。

# 系统部署（可选）

给出系统部署方案，尽量使用图表的形式，并辅以必要的文字说明。