# SCADA模块设计文档

| 日期 | 修改记录 | 修改人 |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

目录

[SCADA模块设计文档 1](#_Toc438043543)

[一、简介 3](#_Toc438043544)

[1．概述 3](#_Toc438043545)

[2．系统功能 3](#_Toc438043546)

[二、总体设计 3](#_Toc438043547)

[1．概述 3](#_Toc438043548)

[1.1测点索引 3](#_Toc438043549)

[1.2 测点逻辑类型 4](#_Toc438043550)

[1.3 测点数据类型 4](#_Toc438043551)

[1.4 设备对象 4](#_Toc438043552)

[2．设计思想 5](#_Toc438043553)

[2.1 模块划分 5](#_Toc438043554)

[2.2 引擎内核模块 6](#_Toc438043555)

[2.3 引擎接口模块 9](#_Toc438043556)

[2.4 数据服务接口 9](#_Toc438043557)

[三、接口说明 10](#_Toc438043558)

[3.1 接口基本规范 10](#_Toc438043559)

[3.2 内核接口 10](#_Toc438043560)

[3.2.1 内存库创建 10](#_Toc438043561)

[3.2.2 系统变量设值 10](#_Toc438043562)

[3.3驱动接口 10](#_Toc438043563)

[3.3.1 内存库关联 10](#_Toc438043564)

[3.3.2 数据模型 11](#_Toc438043565)

[3.3.3 IO设值 11](#_Toc438043566)

[3.3.4 状态接口 11](#_Toc438043567)

[3.4应用接口 12](#_Toc438043568)

[3.5消息接口 12](#_Toc438043569)

[3.5.1 消息定义 12](#_Toc438043570)

[3.5.2 消息接口 13](#_Toc438043571)

[3.5.3 调用的报警接口 13](#_Toc438043572)

# 一、简介

## 1．概述

## 2．系统功能

2.1 测点（变量）的存储管理

2.2 设备对象模型的管理

2.3 报警的处理

2.4 数据服务接口

2.5 调试服务端

注意与五防系统的配合

# 二、总体设计

## 1．概述

系统的测点，也可称之为变量，根据实际外部接线方式，可分为实际测点和虚拟测点，也可以称为定位变量或者非定位变量。非定位变量也称之为内存变量，内存变量分为系统变量与用户变量。系统变量的TagName由系统预定义的，使用者不可修改。具体系统变量可见附表，系统变量的OccNo排行号从1-4096。

### 1.1测点索引

所有的测点支持TagName和OccNo索引。

### 1.2 测点逻辑类型

TYPE\_AIN

TYPE\_DIN

TYPE\_AOUT

TYPE\_DOUT

TYPE\_USER

TYPE\_SYS

### 1.3 测点数据类型

具体如下：

|  |  |  |
| --- | --- | --- |
| **类型名称** | **说明** | **备注** |
| **Boolean** | 位 | 开关量（遥信、遥控） |
| **Char** | 有符号字节（8位） |  |
| **Byte** | 无符号字节（8位） |  |
| **Short** | 有符号字（16位） |  |
| **Word** | 无符号字（16位） |  |
| **Long** | 有符号双字（32位） |  |
| **DWord** | 无符号双字（32位） |  |
| **Float** | 单精度浮点（32位） |  |
| **Double** | 双精度浮点（64位） | 模拟量（遥测、遥调、遥脉） |
| **String** | 字符串（0结束符） | 用于用户变量(针对，系统变量中的字符串型变量，单独建立字符串池存放字符串变量的值和相应的索引，在系统变量中，只存字符串变量的索引) |
| **Array** | 定长数组 |  |

### 1.4 设备对象

设备对象：

支持设备对象的设备操作（支持用户权限）、报警处理、挂牌操作、数据读写。

## 2．设计思想

### 2.1 模块划分

SCADA部分主要包括以下部分：

|  |  |  |
| --- | --- | --- |
| 序号 | 名称 | 描述 |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 | 辅助调试工具 | 调试记录等等 |

### 2.2 内部模块

功能如下：

|  |  |  |
| --- | --- | --- |
| 序号 | 功能 |  |
| 1 | 内存库加载及创建 |  |
| 2 | 数据主从同步 |  |
| 3 | 调试服务端 |  |
| 4 | 脚本引擎 |  |
| 5 |  |  |
| 6 |  |  |
|  |  |  |

#### 2.2.1内存库创建模块

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 说明 | 描述 |
| 1 | 系统节点表 |  |  |
| 2 | 通道表 |  |  |
| 3 | 装置表 |  |  |
| 4 | 模拟量表 |  |  |
| 5 | 开关量表 |  |  |
| 6 | 模出量表 |  |  |
| 7 | 开出量表 |  |  |
| 8 | 用户变量表 | 系统内部维护，通过脚本开放引用 |  |
| 9 | 系统变量表 | 系统自动实时更新系统变量 | 每个节点的系统变量表是**不一样** |
| 10 | 线性转换关系表 |  |  |
| 11 | 非线性转换关系表 |  |  |
| 12 | 计算表 |  |  |
| 13 | 告警表 |  |  |

1.部分变量若配置初始值，或者自动记录（自动保存），在变量初始化时应当自动赋值。

2.内存变量（系统变量及内存变量）默认品质均为好。

**2.2.1.1内存变量管理**

1. 各表中的OccNo从1开始，每个表的OccNo独立,0表示非法OccNO。
2. TagName在一个前置节点内全局唯一，模入量、模出量、开入量、开出量、用户变量、系统变量、通道、装置采用TagName描述。

**2.2.1.2 内存表关系**

通道：一个或多个使用相同通信规约的装置形成一个通道。

装置：每个通道下辖多个装置

驱动：每个通道对应一个驱动

**2.2.1.3 基于内存库的关系映射**

点映射：通道号，装置号，点号到OccNo的映射

名映射：TagName到OccNo的映射

#### 2.2.2 IO驱动管理

**2.2.2.1、驱动加载**

前置扫描通道表，根据规约类型和驱动运行方式，加载驱动，并将通道号作为参数传递至驱动

**2.2.2.2 驱动监测**

驱动运行状态监测，驱动将运行状态刷新至前置通道表

**2.2.2.3、驱动卸载**

1. 前置退出时，设置所有通道内核运行状态为退出
2. 驱动循环检测通道内核运行状态，如果为退出状态，则退出驱动

或：前置发送退出命令至驱动，驱动处理命令退出

#### 2.2.3 数据传送

更详细内容见 通信协议

1、定时扫描内存库全数据，上送至SCADA（批量传送测值）

2、上送方式：本地共享内存或SOCKET

3、主从之间的数据同步，同步的内容见下表：

|  |  |  |
| --- | --- | --- |
|  | 内容 | 备注 |
| DIN |  |  |
| AIN |  |  |
| DQ |  |  |
| AQ |  |  |
|  |  |  |

注：主从同步时，系统变量的同步方式**待定**。

#### 2.2.4调试服务端模块

1、根据调试信息分类，建立一个或多个socket服务端

2、调试信息经socket发送至调试客户端

#### 2.2.5控制与调节

实现SCADA、装置以及驱动之间的控制命令交互

1. 实现方式：控制命令交互基于共享内存队列实现；
2. 队列标识：队列以通道ID做为唯一标识，前置的命令队列的ID为0
3. 队列需求：每个通道维护一个命令队列，前置写，驱动读；前置维护一个命令队列，驱动写，前置读；SCADA写，前置读

控制、调节、设值存在以下几种情况：

1． 本节点上的模块（脚本、顺控、面板、调试）修改本地数据；

2． 本节点上的模块（脚本、顺控、面板、调试）修改远程（其他节点）数据；

3． 远程节点修改本地数据；

SetSomeValue(Fi,PointOccNo,value)

如果是本机Fi,则直接下发，否则走总线转

#### 2.2.6脚本与逻辑

1、内核支持脚本

2、脚本中关于数据的读写操作是基于前置开放接口实现

#### 2.2.7 主从冗余同步

### 2.4 数据服务接口

SCADA为高级应用模块提供服务，高级应用模块有：远动通信模块、历史数据服务模块、顺控流程模块、实时画面模块、其他高级应用（AGC\AVC）等。

#### 2.4.1 用户权限管理

运行时，用户登录由SCADA进行管理。SCADA模块提供 用户登录状态接口。运行时画面操作，如果该图元操作关联了设备对象后，则需由验证是否具备操作权限。

#### 2.4.2基于设备对象的数据读写

SCADA提供基于设备对象的数据读写操作接口，

# 三、接口说明

## 3.1 接口基本规范

1. 统一返回值KRESULT，实际返回值以指针参数形式返回
2. 所有接口基于OCCNO，提供TagName到OCCNO的转换接口

## 3.2 内核接口

### 3.2.1 内存库创建

1、内存库创建

KRESULT CreateIO();

2、内存库删除

KRESULT DestroyIO();

### 3.2.2 系统变量设值

KRESULT SetSYSValue(INT32U occNO,VARIANT vValue,INT32U nQuality);

## 3.3驱动接口

### 3.3.1 内存库关联

1、内存库打开,并通过消息向前置注册该通道

KRESULT OpenIO(INT32U nChannelOccNo);

2、内存库关闭，并通过消息向前置注销该通道

KRESULT CloseIO(INT32U nChannelOccNo);

### 3.3.2 数据模型

1.获取通道下所有的装置号

KRESULT GetDevicesInChannel(INT32U nChanelOccNO，vecotr<INT32U>& vecDevices);

2.获取装置下所有的AIN OCCNO

KRESULT GetAinsInDevice(INT32U nDeviceOccNO，vecotr<INT32U>& vecAins);

3.获取装置下所有的DIN OCCNO

KRESULT GetDinsInDevice(INT32U nDeviceOccNO，vecotr<INT32U>& vecDins);

4.获取装置地址等参数

KRESULT GetDevice(INT32U nDevOccNO,CDevice\*\* ppDevice);

5.获取通道参数

KRESULT GetChannel(INT32U nChannelOccNO,CChannel\*\* ppChannel);

6.获取AIN

KRESULT GetAIN(INT32U occNO，AIN\*\* ppAIN);

7.获取DIN

KRESULT GetDIN(INT32U occNO，DIN\*\* ppDIN);

8.获取系统变量

KRESULT GetSYS(INT32U occNO，SYS\*\* ppSYS);

### 3.3.3 IO设值

1.AIN设值

KRESULT SetAINValue(INT32U occNO,FP32 fValue,INT32U nQuality);

2.DIN设值

KRESULT SetDINValue(INT32U occNO,INT8U nValue,INT32U nQuality);

3.DOUT设值

KRESULT SetDOUTValue(INT32U nOccNo, INT8U Value, INT8U Quality);

4.AOUT设值

KRESULT SetAOUTValue(INT32U nOccNo,FP32 Value,INT8U Quality);

### 3.3.4 状态接口

1 驱动是否退出

KRESULT IsDriverQuit(INT32U nChannelOccNo);

2 前置退出

KRESULT IsFesQuit();

## 3.4应用接口

应用类接口主要为顺控、脚本等基于前置的**本地**高级应用模块服务。

1.TagName转为OCCNO

KRESULT GetOccNoByTagName(INT8S\* strTagName,INT32U\* pOccNo);

KRESULT GetOccNoByTagName(INT32U nDataType,INT8S\* strTagName,INT32U\* pOccNo);

2.AIN取值

KRESULT AppGetAinValue(INT32U nOccNo,FP64\* pValue);

3.DIN取值

KRESULT AppGetDinValue(INT32U nOccNo,INT8U\* pValue);

4.系统变量取值

KRESULT AppGetSysValue(INT32U nOccNo,VARIANT\* pValue);

## 3.5消息接口

前置引擎与IO驱动程序之间采用邮箱通信，通道号作为邮箱的标识符。可按邮箱ID作为接收地址进行邮箱通信。

消息有以下类型：

|  |  |  |
| --- | --- | --- |
| 序号 | 简介 |  |
| 1 | 调试类消息 | 强制设值、修改定值、驱动退出等 |
| 2 | IO类消息 | 遥控、遥调发给内核 |
| 3 | 告警类消息 | 驱动告警发往内核 |
|  |  |  |
|  |  |  |
|  |  |  |

### 3.5.1 消息定义

|  |  |  |  |
| --- | --- | --- | --- |
| 属性序号 | 名称 | 说明 | 备注 |
| 1 | Type | 命令类型 |  |
| 2 | Length | 命令长度 |  |
| 3 | ChannelOccNo | 目标通道OCCNO | 发往前置则该域为0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 命令序号 | 名称 | 说明 | 备注 |
| 1 | CT\_CTRL | 控制命令 |  |
| 2 | CT\_QUIT | 退出驱动 |  |
| 3 | CT\_MSG | 自定义消息 | 告警，日志等 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### 3.5.2 消息接口

1前置下发命令至驱动

KRESULT SendCommand(INT32U nChannelOccNo,INT8u\* pCommand,INT32U nLength);

2驱动接收命令

KRESULT RecvCommand(INT32U nChannelOccNo,INT8u\* pCommand,INT32U nLength);

3驱动上送消息至前置

KRESULT SendCommand(INT8u\* pCommand,INT32U nLength);

4前置接收命令

KRESULT RecvCommand(INT8u\* pCommand,INT32U nLength);

### 3.5.3 调用的报警接口

5 自诊断

FESAPI void IoDiagAlarm(INT32U nChannleNo, INT32U nDeviceNo, INT8S tmpstr[]);

由驱动调用，驱动程序发现装置通信故障，或者通道异常时，通过调用该接口通知系统。

6 操作报警

FESAPI void IoOperAlarm(INT32U nChannleNo, INT32U nDeviceNo, INT8S tmpstr[]);

7 保护的自诊断

FESAPI void IoRelayDiagAlarm(INT32U nChannleNo, INT32U nDeviceNo, INT8S tmpstr[], TIMEPAK \* pTm);

由驱动调用，装置内部自诊断发现异常时，通过驱动程序调用该接口通知系统。

8 通用的告警

FESAPI void IoAlarmMsg(INT32U nChannleNo, INT32U nLogType, INT8S tmpstr[], TIMEPAK \* pTm);