# Matrix428 LOG模块开发文档

## 需求分析说明书

### 任务概述

Matrix428 LOG模块程序主要提供对于整个Matrix428软件系统的”数据库”的管理。所谓的数据库包括Matrix428各程序组件的配置信息，日志信息，处理支持文件，震源质量控制信息。该模块主要包括两种功能：

1. Matrix428系统内各组件参数的管理和显示
2. 提供对于SPS文件的输入、输出和显示
3. 集成数据支持SEGD文件输出

### 任务目标

该模块主要用于和Matrix428现有的各组件进行集成，从其他组件中获取配置信息和各种设备信息，便于SEGD和SPS文件的输出，同时用于管理和查看系统内各组件的配置。

开发原则：

1. 该模块开发上将本着高内聚，低耦合的原则，尽可能减少对当前系统已存在的各组件修改和影响为原则，
2. 可扩展要求，当前实现的LOG模块，主要参考Sercel e428XL的JLog程序来实现。由于目前Matrix428系统的实现方法和Sercel e428XL存在一定差异，因此需要当前系统存在进一步修改的可能性，由此LOG模块的实现要考虑到未来功能和Sercel e428XL的JLog功能相当的情况，当前已原则1为优先的基础上，对原则2进行实施
3. 移植性 在原则1和2的基础上，开发进度允许情况下，将尽可能将表现层、业务层和数据层分开，便于以后对于不同平台和多主机模式的移植。

#### 运行环境

操作系统： Windows XP-32bit， Windows7和Windows NT Server

硬件环境: Intel, AMD X86系统，1G内存， Netcard 100-1G

#### 条件与限制

条件：

1. 该程序将在统一的Matrix428安全保证条件下运行
2. 依赖Matrix428各部分组件的存在和已经存在配置信息并且在配置信息修改的条件下，同步更新到LOG模块中
3. 需要Line和operation程序中得到测线信息其中包括swath, source point, receive point等

限制：

1. 该程序不提供对于各组件参数的修改后，向其对应程序和相关模块的推送和更新通知，参数的获取和保存为单向过程（LOG只能从其他模块中获取数据）
2. 对于SPS的输出要求版本必须与SEGD版本相对应（SEGD REV1.0对应SPS REV.0；SEGD Rev2.1 对应 SPS Rev2.1）
3. SPS的输入时，需要对炮点进行设置，然后通过swath将SPS输入信息进行配置，作为生成绝对排列的基础
4. 该程序本来提供基本操作和OBServer的安全和操作可打印报告，由于需要大量修改和调整当前已有各模块，因此该功能将被暂时搁置

### 数据描述

#### 静态数据

|  |  |
| --- | --- |
| 界面显示信息 | 其中包括数据分类（参数，结果），各模块名称，菜单信息 |
| 出错信息 | 各种错误信息描述 |

表1

#### 动态数据

输入信息：

|  |  |
| --- | --- |
| 各模块参数信息 | Matrix428中的每个模块的配置信息，其中包括config, line, operation,log,plotter,positioning等 |
| SPS文件 | 针对不同的swath的SPS文件，可以为positioning建立绝对排列 |
| Observer log信息 | 针对每一步操作的安全核查和步骤记录日志 |
| 炮点配置信息 | 在提供的用户界面上配置炮点信息,和输入的SPS信息组合成positioning的绝对排列 |
| 各模块日志 | 根据各模块软件的输出的日志信息 |

表2

输出信息:

|  |  |
| --- | --- |
| SPS文件 | 根据swath导出当前的对应的SPS文件(R,X,S文件) |
| 绝对排列 | 根据LOG中的炮点配置和输入的swath的SPS文件,生成对应的swath的绝对排列 |
| 各模块参数 | 导出Matrix428中各模块的配置信息,可以用来重新配置模块本身 |

表3

#### 数据库描述

由于当前使用和保存的参数较多,使用文件存储方式存在一定问题,使用数据库将大大降低开发和维护强度。由于程序的应用特性需要尽可能的配置和维护简单，因此嵌入式数据库为最佳选择。Sqlite是一种关系式嵌入数据库。使用其将对各模块的配置信息进行输入、输出和保存。

#### 数据字典

|  |  |
| --- | --- |
| Swath | 测区 |
| SPS | 震源位置 |
| Gain | 增益 |
| Observer | 操作员管理服务器 |
| SEGD | 地震地理数据 |

表4

### 功能要求

#### 功能划分和描述

|  |  |  |
| --- | --- | --- |
| 1 | SPS文件导入 | 在指定swath的情况下,将SPS文件导入到swath中 |
| 2 | SPS文件导出 | 指定swath,导出SPS文件(R,S,X) |
| 3 | 模块  Parameters  导入 | 读取各模块的parameters文件信息,并将信息放入程序中内存中,便于显示 |
| 4 | 显示parameters | 将各模块的参数信息在界面上,并且分类显示到界面 |
| 5 | 显示SPS文件 | 根据swath，显示对应的SPS文件在界面上（其中包括S,X,R文件） |
| 6 | 存取数据 | 读取和保存各模块的参数数据，相关的swath信息（包括SPS信息，炮点信息等等） |
| 7 | 炮点配置 | 在特定的swath上，配置炮点信息 |

表5

### 性能需求

#### 数据精确度

|  |  |
| --- | --- |
| SPS导出导入文件 | 数据精度参看e428XL Vol.2中的SPS Rev2.1标准 |
| 模块参数 | 各模块参数根据模块情况存储响应的格式,注意float, double,datetime和string大小 |
| Swath参数 | 根据SEGD Rev2.1和SPS Rev2.1的要求保存 |

表6

#### 时间特性

##### 响应时间

LOG程序为桌面应用程序,因此该程序的响应效率必须符合桌面程序特性,该程序不存在网路应用和大规模的I/O读写操作,因此响应时间问题基本不用考虑,以windows的基本要求为准.

##### 更新处理时间

该时间主要是针对嵌入式数据库对于数据表的UPDATE效率,这可能和数据表大小存在一定关联,需要针对较大数据表进行测试,该时间必须在毫秒级别来完成.

##### 数据转换和传输时间

无需考虑

##### 运行时间

该程序为GUI程序,要求能保证24\*2连续不停歇运行,不能出现程序崩溃或效率问题.

#### 适应性

##### 操作方式

该程序使用标准的Windows操作界面,遵循Windows标准操作方式(其中包括鼠标的使用,键盘的快捷方式等)

##### 运行环境

硬件: INTEL兼容CPU, 512M内存, 200M以上硬盘空间

软件: WinXP 32bit, Win Vista, Win7 32bit, WinServer 2003-2008

##### 与其他软件接口

|  |  |
| --- | --- |
| 软件名称 | 接口关系 |
| Sqlite | 数据库接口 |
| 测线程序 | 导入震源,接受点等数据;配置信息 |
| 施工程序 | 导入炮点信息,震源和接受点关系信息;配置信息 |
| Config程序 | 配置信息 |
| AdminServer程序 | 配置信息 |

表7

## 运行需求

#### 用户界面

屏幕格式:

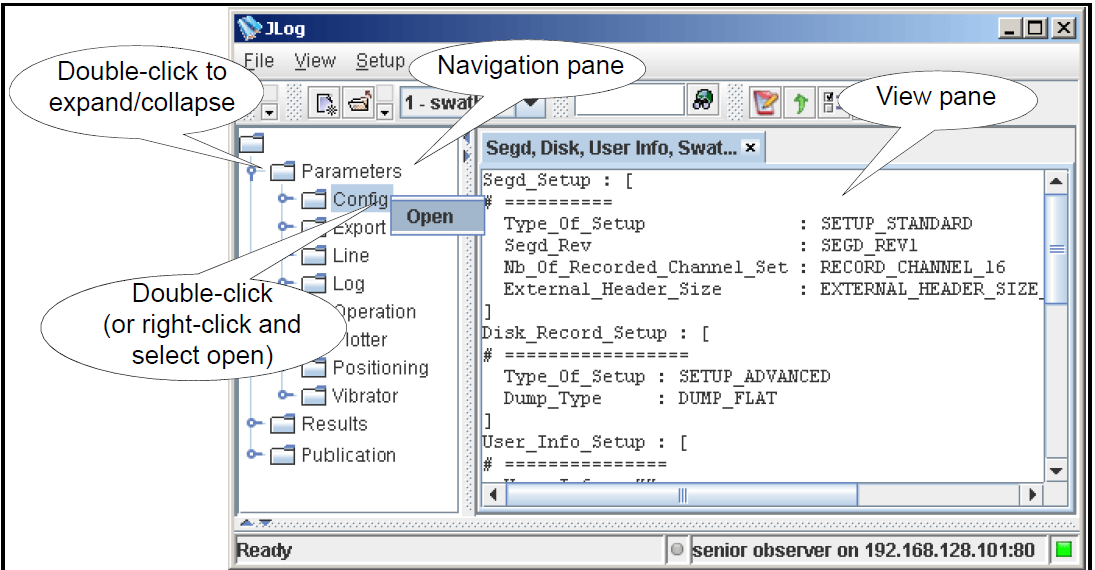


图1

工具条:

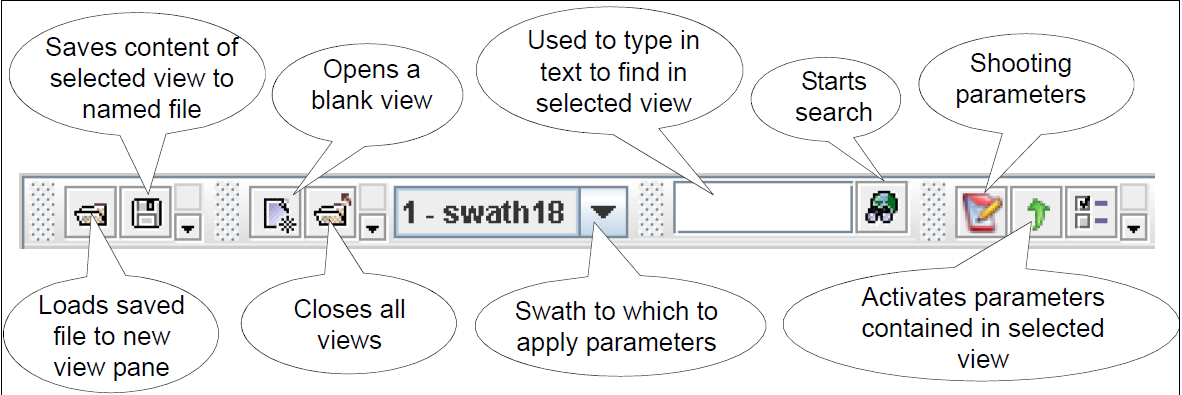


图2

视图界面

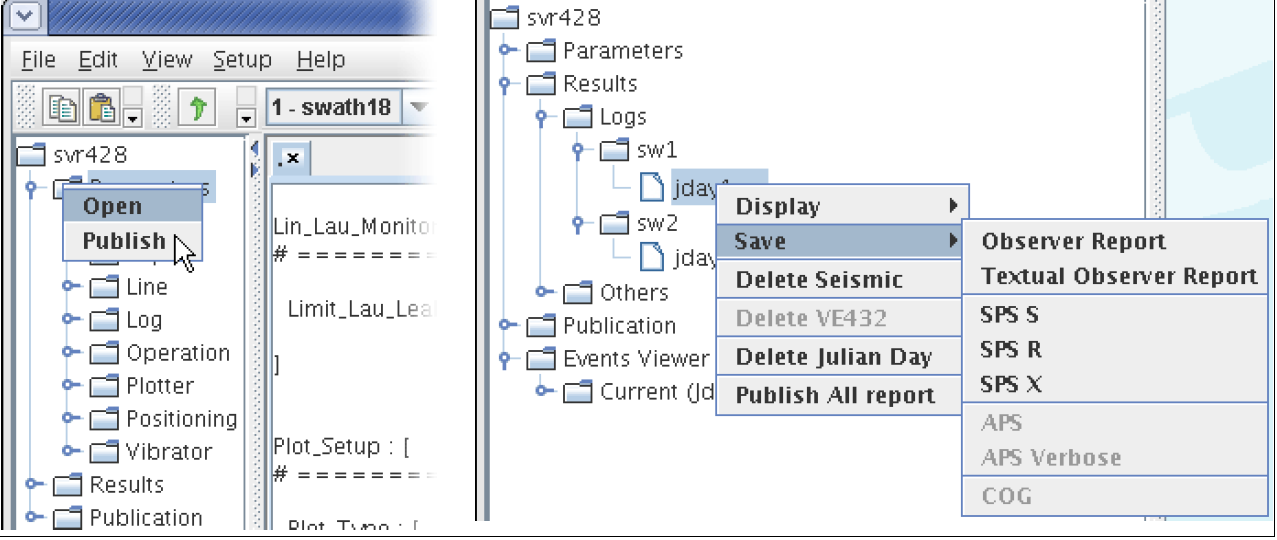


图3

炮点设置

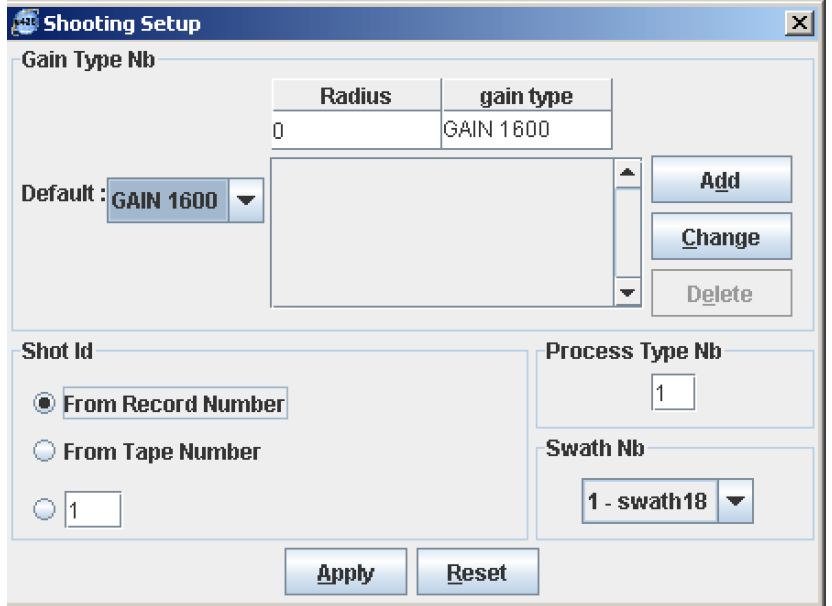


图4

输入SPS界面:

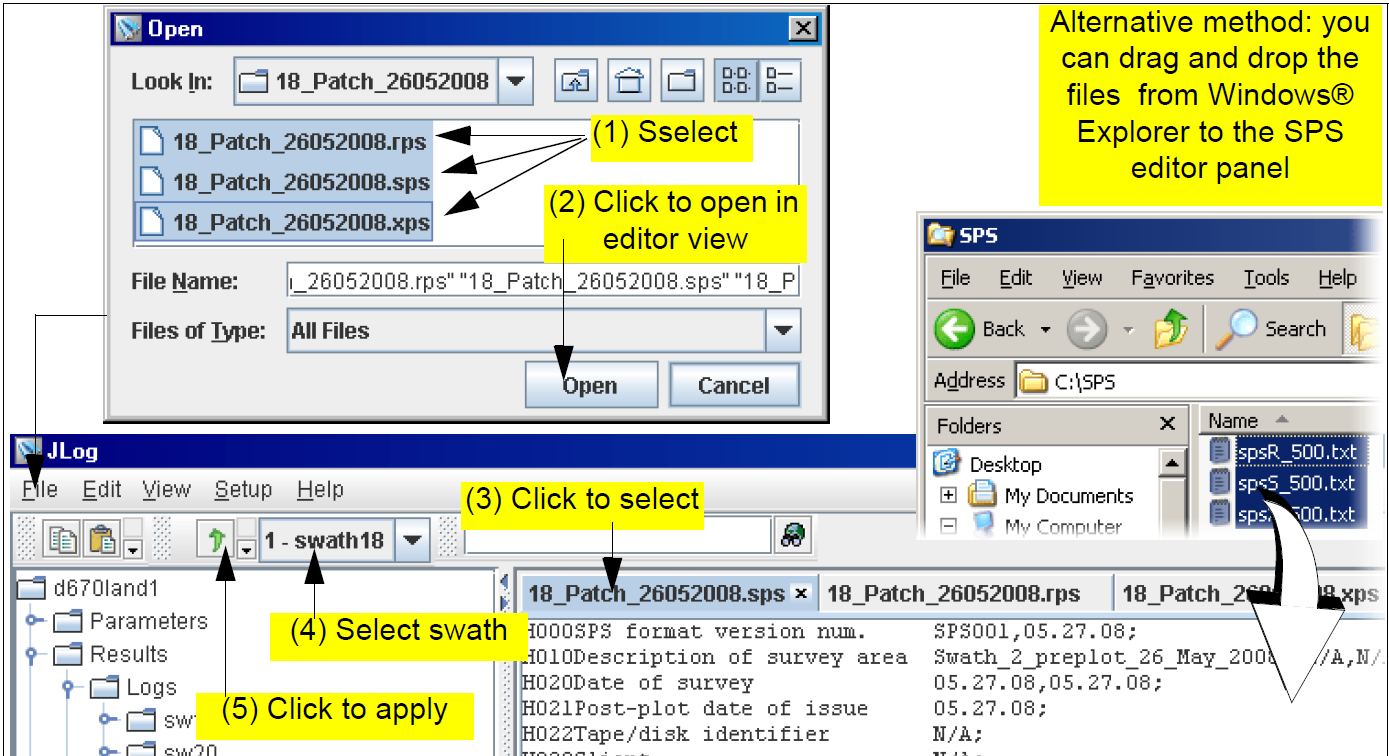


图5

输出SPS界面:

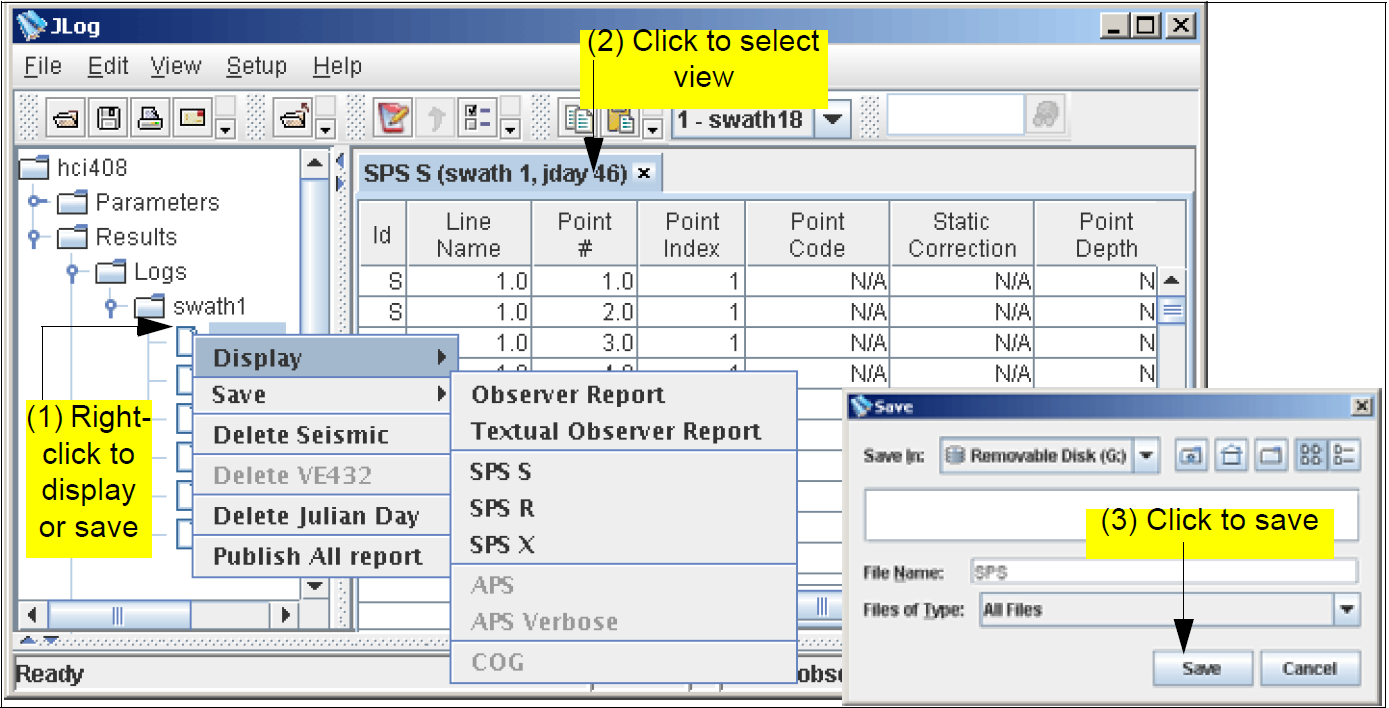


图6

#### 硬件接口

|  |  |
| --- | --- |
| 硬件 | 描述 |
| 打印机 | 可以用于输入日志 |

表8

#### 软件接口

|  |  |
| --- | --- |
| 软件 | 描述 |
| Sqlite | 嵌入式数据库 |
| SPS I/O | 导入导出SPS文件组 |
| Param Input | 模块参数参数输入 |

表9

#### 故障处理

建立出错机制,将错误逐级上传到界面系统,然后输出数据到姐

### 其他需求

|  |  |
| --- | --- |
| 安全保密 | 在AdminServer统一的安全机制下运行 |
| 可维护性 | 要有一定的可扩展能力,以准备对程序功能的扩充 |
| 可移植性 | 在Windows基础上,使用可跨平台UI开发 |

表10

## 概要设计说明书

### 总体设计

#### 处理流程

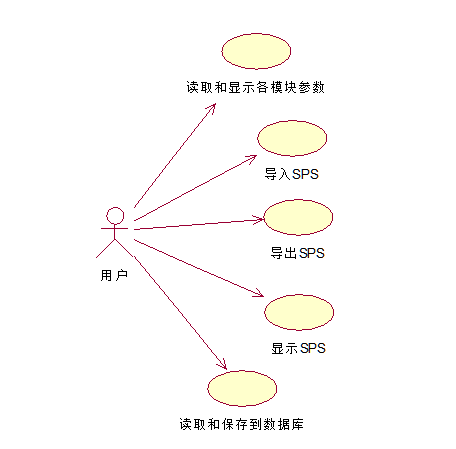


图7

#### 总体结构和模块外部设计

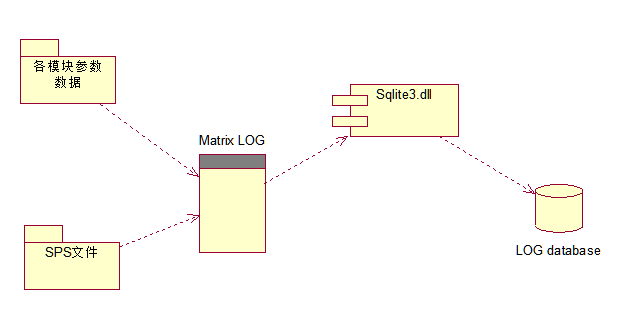


图8

#### 功能分配

|  |  |
| --- | --- |
| 读取模块配置信息 | 读取各模块配置信息从各模块配置文件中 |
| 读取保存数据 | 读取和保存到数据库 |
| 导出导入SPS | 根据swath导出导入sps文件 |
| 显示界面 | 显示参数列表,显示sps文件 |
| 配置炮点 | 设置swath炮点信息 |

表11

### 接口设计

#### 外部接口

外部界面

采用多文档多视图结构,将针对不同的文档内容显示数据

|  |  |
| --- | --- |
| LOG整体结构 | 当前使用TreeView视图结构,将显示各模块参数,结果 |
| SPS文件内容 | 使用GridView视图结构,描述各部分信息 |
| 模块参数 | TextView视图显示个参数信息 |
| 操作输出 | ListView输出操作和错误 |

表12

软件结构

Matrix LOG

SPS I/0 module

Sqlite

Param Input

图9

当前Matrix LOG将分解为4个子部分:

|  |  |
| --- | --- |
| Matrix LOG主程序 | 该模块实现界面和显示功能 |
| SPS I/O模块 | 读入和写出SPS文件 |
| Sqlite | 实现存取数据信息功能 |
| Param Input | 读取和导入各模块的配置信息 |

表13

硬件接口 忽略

#### 内部接口(模块间接口)

##### Matrix LOG 与 SPS I/O模块

|  |
| --- |
| Function: int init\_sps\_io()  Brief: SPS import/extport 初始化函数  Return: 如果成功返回SPS\_SUCCESS,否则返回SPS\_FAULT, 通过调用get\_last\_error()返回错误信息 |
| Function: int import\_sps(char const\* sps\_path, SPS\_DATA\_PTR sps\_data)  Brief: 输入sps文件,得到SPS数据  Param: [in] sps\_path 输入路径,将打开后缀s, x, r 文件  Param: [in/out] sps\_data 数据信息  Return: 如果成功返回SPS\_SUCCESS,否则返回SPS\_FAULT, 通过调用get\_last\_error()返回错误信息 |
| Function: int export\_sps(char const\* sps\_path, const SPS\_DATA\_PTR sps\_data)  Brief: 输出SPS文件  Param: [in] sps\_path 输入路径,将文件后缀加入s, x, r  Param: [in] sps\_data 数据信息  Return: 如果成功返回SPS\_SUCCESS,否则返回SPS\_FAULT, 通过调用get\_last\_error()返回错误信息 |
| Function: int release\_sps\_io()  Brief: 释放模块  Return: 如果成功返回SPS\_SUCCESS,否则返回SPS\_FAULT, 通过调用get\_last\_error()返回错误信息 |
| Function: int get\_last\_error(char\* error\_info, int\* error\_info\_size)  Brief: 得到当前程序模块错误信息  Param: [in/out] error\_info 传入有效缓冲区,返回当前错误缓冲  Param: [in/out] error\_info\_size 传入error\_info大小,返回缓冲实际填充大小  Return: 返回当前的错误编号,如果返回SPS\_SUCCESS说明没有错误 |

##### Matrix LOG 与 模块Param Input模块

|  |
| --- |
| Function: int init\_params\_io();  Brief: params import/extport 初始化函数  Return: 如果成功返回PARAMS\_IO\_SUCCESS,否则返回PARAMS\_IO\_FAULT, 通过调用get\_last\_error()返回错误信息 |
| Function: int import\_params(char const\* params\_path, enum \_\_enuPARAMS\_IO\_SET\_TYPE type, void\* params\_list)  Brief: 导入指定的类型参数文件的参数列表  Param: [in] params\_path 输入路径  Param: [in] type 参数集合类型,参见enum \_\_enuPARAMS\_IO\_SET\_TYPE  Param: [in/out] params 数据集合信息  Return: 如果成功返回PARAMS\_IO\_SUCCESS,否则返回PARAMS\_IO\_FAULT, 通过调用get\_last\_error()返回错误信息 |
| Function: int export\_params(char const\* params\_path, enum \_\_enuPARAMS\_IO\_SET\_TYPE type, void const\* params\_list)  Brief: 导出指定的类型参数文件的参数列表  Param: [in] params\_path 输入路径  Param: [in] type 参数集合类型,参见enum \_\_enuPARAMS\_IO\_SET\_TYPE  Param:[in] params 数据集合信息  Return: 如果成功返回PARAMS\_IO\_SUCCESS,否则返回PARAMS\_IO\_FAULT, 通过调用get\_last\_error()返回错误信息 |
| Function: int release\_params\_io()  Brief: 释放模块  Return: 如果成功返回PARAMS\_IO\_SUCCESS,否则返回PARAMS\_IO\_FAULT, 通过调用get\_last\_error()返回错误信息 |
| Function: int get\_last\_error(char\* error\_info, int\* error\_info\_size)  Brief: 得到当前程序模块错误信息  Param: [in/out] error\_info 传入有效缓冲区,返回当前错误缓冲  Param: [in/out] error\_info\_size 传入error\_info大小,返回缓冲实际填充大小  Return: 返回当前的错误编号,如果返回PARAMS\_IO\_SUCCESS说明没有错误 |

##### Matrix LOG 与 Sqlite

### 数据结构设计

#### 逻辑结构设计

#### 物理结构设计

#### 数据结构和程序关系

### 运行设计

#### 运行模块的组合

#### 运行控制

#### 运行时间

### 出错处理设计

#### 出错输出信息

#### 出错处理对策(设置后备,性能降级,恢复及再启动)

### 安全保密设计

### 维护设计

## 详细设计说明书

|  |  |
| --- | --- |
| 功能 |  |
| 性能 |  |
| 输入项目 |  |
| 输出项目 |  |
| 算法(本模块选用的算法) |  |
| 程序逻辑(详细描述本模块实现算法,可采用:各种图) |  |
| 接口 |  |
| 存储分配 |  |
| 限制条件 |  |
| 测试要点 |  |

## 测试说明书

测试项目(每一个项目名称, 内容)

测试用例

输入(输入数据和命令)

输出(预计的输出结果)

步骤和操作

允许的偏差(给出实测结果和预期结果之间允许的误差范围)

测试条件

测试资料

评测: 范围 – 测试说明问题的范围和局限性; 准则-评测结果准则

评价

测试结果(实测数据结果,与预期的偏差,该项表明的事实,该测试发现的问题)

软件能力(经过测试表明软件的能力)

缺陷和限制(测试的揭示的软件缺陷和不足,以及给软件运行带来的不足,以及给软件运行带来的影响)

建议(弥补以上缺陷的建议)

测试结论(是否通过测试)