

**RSUC节点管理**

设计文档

|  |  |  |
| --- | --- | --- |
|  | 当前版本： | V4.0.0 |
| 文件状态： | 创 建 人： | 翟海青 |
| [√]草稿 | 创建日期： | \_\_\_2020\_年 4 月 1 日 |
| [ ]正在修改 | 审 核 人： | 翟海青 |
| [ ]正式发布 | 审核日期： | \_\_\_2020\_年 4 月 1 日 |
|  | 密 级： | ■内部资料 □公开资料 |

版权声明和保密须知：

本文件出现的任何文字叙述、文档格式、照片、插图、方法、过程等内容，除特别声明外，版权均归属北京源清慧虹信息科技有限公司所有，受到有关产权及版权法保护。

Copyright© 2014.All rights reserved 北京源清慧虹信息科技有限公司

**本文详细说明了节点表组织、管理、存储方式等，嵌入式程序将按照此文进行设计。**

**存储：**

节点表存储在根目录下（”/”），文件为“EQMA.CFG”。

**组织：**

节点表包含了所有节点的信息以及总线的信息，其结构如下：

#define RSUC\_BUS\_MAX\_DEVICE 247 //总线上支持的从机设备数量

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 设备序号 | 节点地址 | 节点类型 | 节点属性 | 节点分组 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 2 | 1 | 0 | 0 |
| 2 | 3 | 1 | 0 | 0 |
| 3 | 4 | 2 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 |
| 5 | 6 | 3 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 |
| … | 0 | 0 | 0 | 0 |

设备序号：序号从0开始，连续递增至RSUC\_BUS\_MAX\_DEVICE，每一个序号（数组下标）对应一个具体的地址号，从上述组织表中可以看出，设备在存放时，将会根据地址号的值放入对应的序号位置，若不存在某一个设备，则该序号下的地址号为0，例如：新增一个地址6的设备，则该设备会被放入序号为5的表中；

节点地址：数据类型uint8，记录总线上的设备地址，地址为增序排列；

节点类型：数据类型uint8，记录该节点的设备类型，设备表需要单独维护；

节点属性：数据类型uint8，记录该节点的属性，可由上层自定义，eg：HCF700设备中，0可以代表节点，1可以代表基准点；

节点分组：对总线上的节点进行分组，用于记录组号，默认均为0；

/\* 定义单个节点属性列表，单台设备共有4种属性 \*/

typedef struct \_\_attribute\_\_ ((\_\_packed\_\_))

{

unsigned char eq\_addr; //节点地址

unsigned char eq\_type; //节点类型

unsigned char eq\_par; //节点属性

unsigned char eq\_group; //节点分组

}signel\_eq\_manag\_type;

/\* 定义总线上所有的从设备 \*/

typedef struct \_\_attribute\_\_ ((\_\_packed\_\_))

{

unsigned short version; //组件版本号

unsigned int flag; //标志位，用于判断配置表是否初始化过

unsigned char eq\_num; //总线从机数量

signel\_eq\_manag\_type eq[RSUC\_BUS\_MAX\_DEVICE]; //为每个从设备分配一个属性

}eq\_manag\_type;

从具体设计中可以看出，设备表包含以下信息：标志位、设备数量、设备表：



**管理：**

节点管理分为添加、删除、修改节点信息，管理设备时，业务组件需发送包含操作类型、设备地址、设备类型、附加属性、设备分组、校验类型的消息，RSUC组将将会根据此消息对设备进行增删改。