

StarRiver 数据库设计

卿培 <qingpei@sansitech.com>
研发一部软件组
上海三思电子工程有限公司

2014 年 12 月 6 日

目录

| | | |
|----------|--------------------------------|-----------|
| 1 | 概述 | 3 |
| 1.1 | 通信方式 | 3 |
| 1.2 | 特殊的表 | 3 |
| 1.3 | 表的命名规则 | 3 |
| 1.4 | 表中读写的字段 | 3 |
| 1.5 | 其他 | 3 |
| 2 | 流程 | 4 |
| 2.1 | 前台发送一般的命令 | 4 |
| 2.2 | 前台上载文件 | 4 |
| 3 | 表结构 | 5 |
| 3.1 | system_config | 5 |
| 3.2 | user | 6 |
| 3.3 | controller | 6 |
| 3.4 | controller_status | 7 |
| 3.5 | controller_status_changes | 7 |
| 3.6 | controller_schedule | 8 |
| 3.7 | controller_schedule_edit | 8 |
| 3.8 | device | 8 |
| 3.9 | device_mode | 9 |
| 3.10 | device_status | 9 |
| 3.11 | device_status_edit | 10 |
| 3.12 | device_status_changes | 11 |
| 3.13 | device_status_history* | 11 |
| 3.14 | device_schedule | 12 |
| 3.15 | firmware | 12 |
| 3.16 | frontend_map_bmp | 13 |
| 3.17 | frontend_device_map | 13 |
| 3.18 | frontend_group | 13 |
| 3.19 | frontend_group_devices | 14 |
| 3.20 | frontend_scene | 14 |
| 3.21 | frontend_scene_item | 14 |
| 3.22 | frontend_controller_devices | 14 |
| 3.23 | sensor | 15 |
| 3.24 | auto_policy | 15 |
| 3.25 | auto_policy_item | 16 |
| 3.26 | time_schedule | 16 |
| 3.27 | time_schedule_item | 16 |
| 3.28 | task_todo | 16 |
| 3.29 | task_done | 17 |
| 4 | 附录：表 task_todo 的任务类型及参数 | 18 |

1 概述

1.1 通信方式

StarRiver 服务采用异步通信方式，除非有明确的时序关系，否则全部操作都可以认为是并行的。

1.2 特殊的表

- `task_todo`: 记录待操作的命令。前台程序添加的用户指令，或是 StarRiver 服务添加的定时操作。
- `task_done`: 记录已完成的命令的结果。命令完成后会从 `task_todo` 中删除，并在此插入一条记录。

1.3 表的命名规则

- 控制器相关的表，加 `controller_` 前缀；
- 终端设备相关的表，加 `device_` 前缀；
- 前台程序相关的表，加 `frontend_` 前缀。

1.4 表中读写的字段

除非明确指出，否则前台程序从 `controller*` 和 `device*` 表中读数据，写操作主要在 `frontend*` 和 `task_todo` 中进行。

StarRiver 服务不写 `frontend*` 表。

前台程序和 StarRiver 服务器的时间同步：如果有 Internet 连接，都与 `cn.pool.ntp.org` NTP 服务器对时；否则可以考虑在局域网内由一台 Windows Server 或 Linux 主机作为 NTP 服务器，其他主机与该主机对时。

多个前台程序进程：依赖不同的用户名进行区分。暂不考虑软件上限制同一用户多处同时登录的情况。

1.5 其他

假设表中某个字段定义了 N 个字节，但是某个协议只用 M 个字节 ($M < N$)，默认都是用最低 M 个字节，除非有特殊说明。

目前仅支持系统下的控制器属于同一种协议。

2 流程

2.1 前台发送一般的命令

1. 前台线程往 `task_todo` 插一条记录
2. StarRiver 服务从 `task_todo` 读记录
3. StarRiver 服务执行，对相关的表进行必要的更新，然后把结果写到 `task_done` 。
4. 前台线程从 `task_done` 读结果，去相应表读数据，然后显示。

2.2 前台上载文件

暂定：配置一个文件服务器，协议可以从 FTP、HTTP、Samba、NFS 中选择。

前台程序往服务器上传一个文件后记录该文件的 URI 。

StarRiver 服务通过参数中读到的 URI 提取该文件。

3 表结构

除特殊说明外，每张表的第一个字段为主键 (Primary Key)。

3.1 system_config

| 字段名 | 类型 | 含义 | 备注 |
|------------------------------|-------------------|-----------------------|--|
| id | int unsigned | | AI PK |
| network_mode | tinyint unsigned | StarRiver 服务角色 | 0: 服务器, 1: 客户端。考虑同时作为多种角色与不同控制器通信的需求 |
| listening_port | smallint unsigned | 服务器侦听端口 | |
| last_server_activity | timest_amp | StarRiver 服务最近连接时间 | StarRiver 服务定时更新这个字段。 |
| interval_report_activity | int unsigned | 更新最近连接时间间隔 | |
| interval_status_query | int unsigned | 自动查询控制器状态间隔 | |
| interval_device_status_query | int unsigned | 自动查询设备状态间隔 | |
| interval_retry_write | int unsigned | 连接建立期间，写操作重试间隔 | 单位为毫秒 |
| interval_keepalive | int unsigned | 心跳包发送间隔 | |
| timeout_cmd_ack | smallint unsigned | 指令响应超时 | CSA 6.1 T1 |
| timeout_cmd_result | smallint unsigned | 指令结果超时 | CSA 6.1 T2 |
| timeout_event_ack | smallint unsigned | 事件响应超时 | CSA 6.2 T3 |
| timeout_idle | smallint unsigned | 通信空闲时间 | CSA 6.3 T4 |
| timeout_keepalive_ack | smallint unsigned | 心跳响应超时 | CSA 6.3 T5 |
| cmd_retries | tinyint unsigned | 指令重试次数 | CSA 6.1 N1 |
| event_retries | tinyint unsigned | 事件重试次数 | CSA 6.2 N2 |
| keepalive_retries | tinyint unsigned | 心跳重试次数 | CSA 6.3 N3 |
| map_format | int | 地图类型 | 0: BMP, 1: GIS |
| emergency_mode | int | 紧急调光模式 | 0: default brightness, 1: last brightness, 2: schedule |
| ts_device_mode | timest_amp | deviceMode 表最后更新时间 | |
| ts_auto_policy | timest_amp | AutoPolicy 表最后更新时间 | |
| ts_time_schedule | timest_amp | Time_schedule 表最后更新时间 | |
| ts_frontend_scene | timest_amp | FrontendScene 表最后更新时间 | |

3.2 user

| 字段名 | 类型 | 含义 | 备注 |
|--------------------|--------------|------------------------|------------------------------------|
| id | int unsigned | | |
| username | varchar(64) | | |
| md5 | varchar(40) | MD5(username+password) | 128-bit, 32 digits long hex string |
| permission | tinyint | 用户权限 | 0: StarRiver 服务; 1: 一般用户, 2: 管理员 |
| first_name | varchar(64) | | |
| last_name | varchar(64) | | |
| email | varchar(128) | | |
| registration_key | varchar(128) | | 预留注册用户名时使用 |
| reset_password_key | varchar(128) | | 预留重置密码时使用 |
| registration_id | varchar(128) | | 预留注册用户名时使用 |
| note | varchar(450) | 备注 | UTF8 |

默认分配如下两个用户：

| id | username | note |
|----|----------|--------------|
| 0 | comm | StarRiver 服务 |
| 1 | admin | 管理员 |

3.3 controller

| 字段名 | 类型 | 含义 | 备注 |
|---------------|-------------------|------------------|-------------------------------------|
| id | int unsigned | | |
| name | varchar(128) | | |
| addr | binary(8) | 8 字节控制器地址 | |
| comm_protocol | tinyint unsigned | 通信协议 | 0: Shanghai, 1: Sansi LC300, 2: CSA |
| data_protocol | tinyint unsigned | 通信方式 | 0: 串口, 1: TCP, 2: UDP |
| com_port | varchar(8) | 串口号 | COM0, COM1, ttyS0, ttyS1 等 |
| com_baud | int unsigned | 串口波特率 | 9600, 115200 等 |
| com_addr | tinyint unsigned | 使用 485 串口通信时指明地址 | |
| ip_addr | varchar(16) | | xxx.xxx.xxx.xxx |
| ip_port | smallint unsigned | | |
| note | varchar(450) | 备注 | 比如位置信息 |
| display_order | int unsigned | 前台程序界面上显示的顺序 | 不宜重复 |

3.4 controller_status

状态量不支持查询的话对应字段写 NULL。

| 字段名 | 类型 | 含义 | 备注 |
|-------------------|-------------------|-------------------------|--|
| controller_id | int unsigned | controller.id | |
| comm_state | tinyint unsigned | 通信状态 | 0: 正常, 1: 故障 |
| login_state | tinyint unsigned | 是否已注册到 Star-River 服务服务端 | 0: 未注册, 1: 已注册 |
| version_software | binary(4) | 软件版本 | |
| version_system | binary(4) | 系统版本 | |
| version_kernel | binary(4) | 内核版本 | |
| version_hardware | binary(4) | 硬件版本 | |
| manufacturer_info | varchar(450) | 厂商信息文本内容 | UTF8 |
| type | binary(2) | 控制器类型 | 上海: 0x0001 => 485 控制器, 0x0002 => PLC 控制器, 0x0003 => 无线控制器。LC300 同上 |
| work_mode | binary(1) | 工作模式 | 上海: 0x01 => 远程控制模式, 0x02 => 时控工作模式。LC300 同上。Sr-Config 程序初始化数据库时写 0xFF, 代表控制器尚未查询过。 |
| status_code | binary(1) | 控制器状态代码 | 上海: 0x00 => 正常工作, 0x01 => 下行通道异常, 0x02 => 升级模式, 0xFF => 未知错误。LC300 未描述。 |
| last_time | datetime | 最后一次从控制器读取的时间 | 用于检查控制器时间是否正常 |
| sms_count | smallint unsigned | 短信数量 | 从控制器读回来的 |

3.5 controller_status_changes

记录控制器状态变化。

| 字段名 | 类型 | 含义 | 备注 |
|---------------|--------------|---------------|-----|
| controller_id | int unsigned | controller.id | PK1 |
| field | varchar(32) | 被修改的字段名 | PK2 |
| field_value | varchar(255) | 变化后的字段值 | |
| time | timest_amp | 修改时间 | PK3 |

目前记录如下两个状态的变化, 由 controller_status 表中名为 log_controller_changes

的 trigger 执行:

1. comm_state: field_value 存储 CAST(comm_state AS char)
2. status_code: field_value 存储 HEX(status_code)

每月执行的事件 Delete outdated status history 会将超过 365 天前的记录删除。

3.6 controller_schedule

查询时控计划时把结果放到这张表里, 将原来该控制器的删掉, 再重新插入。

| 字段名 | 类型 | 含义 | 备注 |
|---------------|-----------------|---------------|--|
| id | bigint unsigned | 自增 ID | |
| controller_id | int unsigned | controller.id | |
| item | varchar(128) | 时间亮度表的一项 | 上 海: MM, DD, MM, DD, HH, mm, ss, Addr, mode, value (逗号后不加空格) |

3.7 controller_schedule_edit

用户编辑时控计划时把结果放到这张表里。

| 字段名 | 类型 | 含义 | 备注 |
|---------------|-----------------|---------------|--|
| id | bigint unsigned | 自增 ID | |
| controller_id | int unsigned | controller.id | |
| item | varchar(128) | 时间亮度表的一项 | 上 海: MM, DD, MM, DD, HH, mm, ss, Addr, mode, value (逗号后不加空格) |

3.8 device

| 字段名 | 类型 | 含义 | 备注 |
|---------------|--------------|--------------|----|
| id | int unsigned | | |
| name | varchar(128) | | |
| addr | binary(8) | | |
| display_order | int unsigned | 前台程序界面上显示的顺序 | |

| 字段名 | 类型 | 含义 | 备注 |
|------|--------------|-----------|------|
| note | varchar(450) | 备注，比如位置信息 | UTF8 |

3.9 device_mode

各控制器下各个设备的调光模式

| 字段名 | 类型 | 含义 | 备注 |
|-----------------|--------------|----|---------------------------------|
| controller_addr | binary(8) | | PK1 |
| device_addr | binary(8) | | PK2 |
| mode | int unsigned | | 0: manual, 1: auto, 2: schedule |

3.10 device_status

| 字段名 | 类型 | 含义 | 备注 |
|-------------------------|------------------|--------------|---|
| device_id | int unsigned | device.id | |
| version_software | binary(4) | 软件版本号 | LC300: 只用 2 字节 |
| version_hardware | binary(4) | 硬件版本号 | LC300: 只用 2 字节 |
| manufacturer_info | varchar(450) | 厂商信息文本 | 上海: 厂商描述。 LC300: 终端设备产品信息, <=32 字节, ASCII 码。 |
| manufacture_time | datetime | 出厂时间 | LC300: 无 |
| type | binary(4) | 设备类型 | 上海 4.4.1 LC300 4.6.1 0x0E |
| sn | varchar(64) | 产品序列号 | |
| input_volt | double | 输入电压采样值 | |
| input_amp | double | 输入电流采样值 | |
| output_volt | double | 输出电压采样值 | |
| output_amp | double | 输出电流采样值 | |
| active_power | double | 有功功率采样值 | |
| temperature | smallint | 温度采样值 | LC300: 1 字节 |
| temperatureGuard | binary(4) | 过温保护状态过温保护参数 | LC300 4.6 0x0A |
| uptime | int unsigned | 上电工作时间 | |
| total_uptime | int unsigned | 总工作时间 | |
| electricity_consumption | int unsigned | 消耗电量值 | |
| transition_duration_ | tinyint unsigned | 调光渐变时间 | |
| brightness | tinyint unsigned | 当前亮度 | |
| brightness_min | tinyint unsigned | 最小亮度值 | |
| brightness_max | tinyint unsigned | 最大亮度值 | |

| 字段名 | 类型 | 含义 | 备注 |
|-------------------------|------------------|-----------|--|
| brightness_min_physical | tinyint unsigned | 物理最小亮度值 | LC300 未定义 |
| brightness_max_physical | tinyint unsigned | 物理最大亮度值 | |
| brightness_power_on | tinyint unsigned | 上电亮度值 | |
| brightness_default | tinyint unsigned | 默认故障亮度值 | |
| brightness_coefficiency | tinyint unsigned | 调光系数 | |
| status | binary(1) | 设备状态 | 实际亮度 = 设置亮度 * 调光系数 / 100 上海: 4.5.1 LC300: 4.6 0x06 |
| comm_status | binary(1) | 通讯状态 | 上海: 4.5.1 |
| lamp_status | binary(1) | 灯具状态 | 上海: 4.5.1 |
| sensor_i | binary(4) | 光强传感器采样值 | 上海: 光感亮度采样值 |
| LC300: 终端传感器设备光强值 | | | |
| sensor_l | binary(4) | 光照传感器采样值 | LC300: 终端传感器设备光照值 |
| sensor_h | binary(4) | 湿度传感器采样值 | |
| sensor_t | binary(4) | 车流量传感器采样值 | |
| group_mask | binary(32) | 分组掩码 | 一共 256 bit, 若灯属于第 N 个组, 则 bit(N)=1。上海: 组地址是 1-254。LC300: 组地址是 0-63 |

3.11 device_status_edit

StarRiver Config 在此记录用户录入的设备初始信息。

| 字段名 | 类型 | 含义 | 备注 |
|-------------------------|------------------|-----------|---|
| device_id | int unsigned | device.id | |
| transition_duration | tinyint unsigned | 调光渐变时间 | |
| brightness_min | tinyint unsigned | 最小亮度值 | |
| brightness_max | tinyint unsigned | 最大亮度值 | |
| brightness_min_physical | tinyint unsigned | 物理最小亮度值 | LC300 未定义 |
| brightness_max_physical | tinyint unsigned | 物理最大亮度值 | |
| brightness_power_on | tinyint unsigned | 上电亮度值 | |
| brightness_default | tinyint unsigned | 默认故障亮度值 | |
| brightness_coefficiency | tinyint unsigned | 调光系数 | |
| group_mask | binary(32) | 分组掩码 | 实际亮度 = 设置亮度 * 调光系数 / 100 一共 256 bit, 若灯属于第 N 个组, 则 bit(N)=1。上海: 组地址是 1-254。LC300: 组地址是 0-63。 |

3.12 device_status_changes

记录设备状态变化。

| 字段名 | 类型 | 含义 | 备注 |
|-------------|--------------|-----------|-----|
| device_id | int unsigned | device.id | PK1 |
| field | varchar(32) | 被修改的字段名 | PK2 |
| field_value | varchar(255) | 变化后的字段值 | |
| time | timest_amp | 修改时间 | PK3 |

目前记录如下三个状态的变化，由 device_status 表中名为 log_changes 的 trigger 执行：

1. status: field_value 存储 HEX(status)
2. comm_status: field_value 存储 HEX(comm_status)
3. lamp_status: field_value 存储 HEX(lamp_status)

3.13 device_status_history*

以下三张表的字段定义大部分相同，分别记录不同粒度的状态量历史。

1. device_status_history
2. device_status_history_hourly
3. device_status_history_daily

三张表共同部分的定义如下：

| 字段名 | 类型 | 含义 | 备注 |
|-------------|--------------|-----------|-----|
| device_id | int unsigned | device.id | PK1 |
| field | varchar(32) | 被修改的字段名 | PK2 |
| field_value | double | 字段采样值 | |
| time | timest_amp | 修改时间 | PK3 |

两张汇总表有额外记录汇总时段内峰值、谷值的字段。

1. device_status_history_hourly
2. device_status_history_daily

峰谷值定义如下：

| 字段名 | 类型 | 含义 | 备注 |
|-----------|--------|------------|-----------------|
| field_max | double | 汇总时段内最大采样值 | 浮点数可能无法精确表示整数采样 |
| field_min | double | 汇总时段内最小采样值 | 浮点数可能无法精确表示整数采样 |

有三个 event 更新这些表:

1. 每 5 分钟, 对 device_status 进行采样, 记录设备的状态量。
2. 每天, 将前一天的采样归纳成每小时均值。删除 3 天前的采样。
3. 每周, 将前一周的采样归纳成每日均值。删除一周前的小时均值。

每月执行的事件 Delete outdated status history 会将 device_status_history_daily 中超过 365 天前的记录删除。

3.14 device_schedule

LCP-SH-D: 该功能是在单灯控制器与集中控制器失去联系后采用的异常模式, 如果在某一时间段内没有设置该亮度值的话, 将采用默认故障亮度值显示。

注意调光计划的总数不能超过 7。

LC300 没有定义。

| 字段名 | 类型 | 含义 | 备注 |
|-----------|-----------------|-----------|-------------|
| id | bigint unsigned | 自增 ID | |
| device_id | int unsigned | device.id | |
| item | varchar(32) | 自动亮度表的一项 | HH,mm,value |

3.15 firmware

| 字段名 | 类型 | 含义 | 备注 |
|---------|--------------|--------------|------|
| id | int unsigned | | |
| content | mediumblob | 二进制内容 | |
| hash | char(32) | md5(content) | |
| name | varchar(64) | 固件包名称 | 终端用 |
| version | binary(4) | 固件版本号 | 终端用 |
| note | varchar(300) | | UTF8 |

MediumBlob 可以存储 16MB 大小的固件，MySQL 服务器也已经设置：

max_allowed_packet=16M

3.16 frontend_map_bmp

| 字段名 | 类型 | 含义 | 备注 |
|---------------|------------------|----------------|------|
| id | tinyint unsigned | | |
| name | varchar(64) | | |
| display_order | tinyint unsigned | 前台程序界面上显示地图的顺序 | 不宜重复 |

3.17 frontend_device_map

描述各种设备在 BMP 地图上的信息。

| 字段名 | 类型 | 含义 | 备注 |
|---------------------|------------------|----------------------------|-------------|
| frontend_map_bmp_id | int unsigned | frontend_map_bmp.id | PK1 |
| device_type | tinyint unsigned | 表明设备类型，如控制器、灯等 | PK2 |
| device_id | int unsigned | device.id or controller.id | PK3 |
| pos_x | int | 设备在地图上的横坐标 | |
| pos_y | int | 设备在地图上的纵坐标 | |
| icon_id | tinyint unsigned | 标 | 对应的图标文件必须存在 |

3.18 frontend_group

对于 LCP-SH-D 和 LC300 协议：现在用户的组和灯的组是统一的，即用户的第一个组就是所有灯的第一个组；对于 CSA 来说，一个用户的组可能对应控制器 1 的网关 ID1 和控制器 2 的网关 ID2，所以还要建一张表来指定用户组和控制器的网关 ID 的关系（有点像以前 LC200）。

| 字段名 | 类型 | 含义 | 备注 |
|-------|------------------|----|----------------|
| id | tinyint unsigned | | |
| name | varchar(64) | | |
| color | tinyint unsigned | | Config 程序里作为选项 |

3.19 frontend_group_devices

描述灯属于哪个组。

| 字段名 | 类型 | 含义 | 备注 |
|----------------------|------------------|-------------------|------|
| frontend_group_id | tinyint unsigned | frontend_group.id | PK1 |
| device_id | int unsigned | device.id | PK2 |
| device_display_order | int unsigned | 界面显示该组内的灯时的序号 | 不宜重复 |

3.20 frontend_scene

这是用户定义的场景。

| 字段名 | 类型 | 含义 | 备注 |
|---------------|--------------|----|------|
| id | int unsigned | | |
| name | varchar(64) | | |
| display_order | int unsigned | | 不宜重复 |

3.21 frontend_scene_item

这是用户定义的场景。

| 字段名 | 类型 | 含义 | 备注 |
|-------------|-----------------|----|---|
| id | bigint unsigned | | |
| scene_id | int unsigned | | frontend_scene.id |
| device_addr | binary(8) | | 可以是单灯地址，也可以是组地址 |
| mode | int unsigned | | 0: manual, 1: auto, 2: schedule |
| value | int unsigned | | brightness OR id auto policy OR id schedule |

3.22 frontend_controller_devices

描述灯属于哪个控制器。

对于 CSA 的协议，灯是属于控制器下的某个网关 ID，所以可能会增加一个字段表示灯属于控制器的哪个网关。

| 字段名 | 类型 | 含义 | 备注 |
|----------------------|-------------------|-----------------|-------|
| controller_id | int unsigned | controller.id | PK1 |
| device_id | int unsigned | device.id | PK2 |
| device_display_order | int unsigned | 界面显示该控制器下的灯时的序号 | |
| port_on_controller | smallint unsigned | 灯在控制器的哪个口上 | 可能不需要 |

3.23 sensor

传感器信息。

| 字段名 | 类型 | 含义 | 备注 |
|------------------|--------------|----|---|
| id | int unsigned | | |
| name | varchar(255) | | |
| type | int unsigned | | 0: virtual, 1: brightness, 2: traffic |
| source | varchar(255) | | “controller_addr,port(1-4)” for those attached on a controller; “ip,port,user,pass,sql” for those available in a database |
| raw_value | double | | |
| normalized_value | int | | mapped to range [0,100] |
| normalize_method | int | | predefined method id here. Method lies in source code. |

3.24 auto_policy

自动调光策略。

| 字段名 | 类型 | 含义 | 备注 |
|---------------|--------------|-----------|----|
| id | int unsigned | | |
| name | varchar(255) | | |
| display_order | int unsigned | | |
| sensor_id | int unsigned | Sensor.id | |

3.25 auto_policy_item

自动调光策略单项。

| 字段名 | 类型 | 含义 | 备注 |
|----------------|-----------------|----------------|--------------------------------|
| id | bigint unsigned | auto_policy.id | |
| auto_policy_id | int unsigned | | |
| input | bigint unsigned | | [0,100] |
| output | bigint unsigned | | [0,100] |
| type | bigint unsigned | | 0: calculated, 1: user defined |

对于每个 auto_policy，都应生成 101 个策略项，分别对应 input 为 0 到 100 的情形。其中若干项是用户设定的，其余是插值计算而得。

3.26 time_schedule

调光时间表。

| 字段名 | 类型 | 含义 | 备注 |
|---------------|--------------|----|----|
| id | int unsigned | | |
| name | varchar(255) | | |
| display_order | int unsigned | | |

3.27 time_schedule_item

调光时间表的逐条记录。

| 字段名 | 类型 | 含义 | 备注 |
|------------------|-----------------|----|--------------------------------------|
| id | bigint unsigned | | |
| time_schedule_id | int unsigned | | time_schedule.id |
| name | varchar(255) | | MM, DD, MM, DD, HH:mm:ss, brightness |

3.28 task_todo

待处理的用户操作。

服务器进程每隔一段时间在这里取任务，分别处理。

| 字段名 | 类型 | 含义 | 备注 |
|---------|-------------------|-------------|--|
| id | bigint unsigned | 自增 ID | 当流水号 |
| time | datetime | | |
| user_id | int unsigned | user.id | |
| cmd | smallint unsigned | 操作类型 | |
| param | varchar(1024) | 操作的参数 | |
| hash | varchar(40) | 命令的 MD5 散列值 | md5(time + user_id + cmd + param) 作为查询条件供前段程序在 task_done 中查询任务完成状态 |

注意：这个表使用 MyISAM 引擎存储。自增的 id 字段如果使用 InnoDB 存储，计数器在内存里，数据库服务器一旦重启，又将从 1 开始计数，完成任务后向 task_done 中插入记录会引发冲突导致失败。

3.29 task_done

记录那些已完成的用户操作。

task_todo 中的任务，处理完毕后，从 task_todo 删除，将结果插入到表 task_done，保持 task_todo 中字段信息不变。

除了 task_todo 的字段，另有以下字段：

| 字段名 | 类型 | 含义 | 备注 |
|-------------|--------------|--------|--|
| finished | timest_amp | 完成时间 | 插入记录时由数据库自动生成 |
| return_code | smallint | 返回代码 | 成功: 0; 失败: 控制器返回的确认码 bitwise-OR; 控制器未能返回结果（如超时）: -2。 |
| message | varchar(255) | 错误描述信息 | UTF8 |

有一个 event (*Remove outdated tasks from TaskDone*) 定期清理创建早前的任务，以保证这张表不会无限膨胀下去。

4 附录：表 task_todo 的任务类型及参数

参数之间用英文的逗号，隔开；项与项之间用 \n 分开。

| Id | Command | Parameters | 备注 |
|----|-------------|---|---|
| 0 | 密钥更新 | | |
| 1 | 重启控制器 | 控制器地址（8 字节 MAC） | |
| 2 | 控制器时间同步 | 控制器地址 | 如果控制器地址（8 字节 MAC）为广播地址的话则所有控制器都要时间同步 |
| 3 | 设置控制器调光模式 | 控制器地址, 模式 | 模式 =0 远程调光，调光指令由中央控制器发送；模式 =1 自动调光，集中控制器根据预先设置的时控计划进入时控模式 |
| 4 | 设置控制器时控调光计划 | 控制器地址, 终端地址（8 字节 MAC），开始月, 开始日, 结束月, 结束日, 时, 分, 秒, 调光模式, 调光值 \n | 调光模式 =0 表示后面是直接亮度值，=1 表示后面是场景值 |
| 5 | 清空控制器时控调光计划 | 控制器地址 | |
| 6 | 设置控制器自动调光计划 | | |
| 7 | 清空控制器自动调光计划 | 控制器地址 | |
| 8 | 下载手机号码到控制器 | | |
| 9 | 查询控制器手机号 | 控制器地址 | |
| 10 | 删除控制器中所有手机号 | 控制器地址 | |
| 11 | 查询控制器内短信数量 | 控制器地址 | |
| 12 | 查询控制器某条短信内容 | | |
| 13 | 删除控制器某条短信内容 | | |
| 14 | 查询控制器状态 | 控制器地址 | 上海：命令 0x0080；LC300：是多条命令的组合。 |
| 15 | 查询控制器时控调光计划 | 控制器地址 | |
| 16 | 查询控制器自动调光计划 | 控制器地址 | |
| 17 | 查询控制器日志 | 控制器地址 | |
| 18 | 升级控制器固件 | 控制器地址,FirmwareMD5 | 控制器进入固件升级模式，固件升级，退出固件升级模式 |
| 19 | 设置控制器所有属性 | | |
| 20 | 初始化终端信息 | 控制器地址 | |
| 21 | 查询终端初始化信息 | | 上海没有 |
| 22 | 删除终端信息 | 控制器地址 | 删除控制器下所有的终端 |

| Id | Command | Parameters | 备注 |
|----|---------------------|--|---|
| 23 | 终端调光 | 控制器地址, 终端地址, 亮度值 \n | 这个可能主要用于单灯调光。控制器地址（8 字节 MAC）为广播地址的话是向所有控制器发；终端地址（8 字节 MAC）是广播地址的话是向控制器下的所有终端发、终端地址（8 字节 MAC）是组地址的话是向控制器下这个组的终端发、终端地址（8 字节 MAC）也可以是单个终端的地址 |
| 24 | 初始化终端亮度 | 终端地址, 调光渐变时间, 上电亮度值, 最大亮度值, 最小亮度值, 故障亮度值, 调光系数 | 25-30 的批处理 |
| 25 | 设置终端调光渐变时间 | 终端地址, 调光渐变时间 | |
| 26 | 设置终端上电亮度值 | 终端地址, 上电亮度值 | |
| 27 | 设置终端最大亮度值 | 终端地址, 最大亮度值 | |
| 28 | 设置终端最小亮度值 | 终端地址, 最小亮度值 | |
| 29 | 设置终端故障亮度值 | 终端地址, 故障亮度值 | |
| 30 | 设置终端调光系数 | 终端地址, 调光系数 | |
| 31 | 终端恢复出厂设置 | 终端地址 | |
| 32 | 终端组设置 | 终端地址, 组号 \n | LC300: 组 0 不用, 就用组 1-63 |
| 33 | 终端组删除 | 终端地址 | 删除终端下的所有组 |
| 34 | 终端场景设置 | 终端地址, 场景号, 亮度值 \n | |
| 35 | 终端场景删除 | 终端地址 | 删除终端下的所有场景 |
| 36 | 设置终端自动调光计划 | | |
| 37 | 删除终端自动调光计划 | 终端地址 | |
| 38 | 设置终端信息上报条件 | | |
| 39 | 设置终端报警阈值条件 | | |
| 40 | 查询终端状态量终端地址 | 终端地址 \n | |
| 41 | 查询终端自动调光计划 | 终端地址 | |
| 42 | 查询终端组号 | 终端地址 | |
| 43 | 查询终端场景 | 终端地址 | |
| 44 | 升级终端固件 | 终端地址,FirmwareMD5 | StarRiver 服务按流程走 |
| 45 | 清零终端总工作时间 | 终端地址 | |
| 46 | 清零终端总用电量 | 终端地址 | |
| 47 | 设置终端过温保护参数 | | |
| 48 | 开启关闭终端过温保护 | | LCP-SH-D 没有 |
| 49 | 重启 StarRiver Server | | 任务返回值只有 0，重启失败时从原理上就无法再将这个任务加到 TaskDone 了，超时还未从 TaskDone 读到重启结果可以认为是失败。 |
| 58 | 终端自动调光 | 控制器地址, 终端地址, 自动表 ID\n | |
| 59 | 终端时控调光 | 控制器地址, 终端地址, 时控表 ID\n | |

| Id | Command | Parameters | 备注 |
|----|----------|------------|--|
| 60 | 设置紧急调光模式 | 控制器地址, 模式 | 0: 终端故障亮度, 1: 最近亮度, 2: 预存时控计划。通信时复用 0x000D 指令, 用 0xA0, 0xA1 和 0xA2 表示上述模式。注意全局模式应统一, 这里参数加控制器地址仅为了方便客户端识别操作失败的控制器。 |
| 61 | 传感器状态查询 | 传感器 ID\n | |

以下任务类型不由 StarRiver 提交, 尽在 StarRiver Server 中用于数据帧的识别。

| Id | Command | 备注 |
|----|----------------------------------|----|
| 50 | DownloadControllerFirmwareBin | |
| 51 | ExitControllerFirmwareUpdateMode | |
| 52 | DownloadDeviceFirmwareMetadata | |
| 53 | DownloadDeviceFirmwareBin | |
| 54 | DownloadFirmwareToDevice | |
| 55 | ExitDeviceFirmwareUpdateMode | |
| 56 | KeepAlive | |
| 57 | Register | |