车牌识别一体机

TCP通讯协议说明手册

(版权所有,翻版必究)

|  |
| --- |
| 非常感谢您使用我们公司的产品，我们将竭诚为您提供最好的服务。  本手册可能包含技术上不准确的地方或文字错误，欢迎您的纠正。  本手册内容将做定期的更新，更新内容将在本手册的新版本中加入。  我们随时会改进或更新本手册中描述的产品或程序。 |

版本变更记录

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **拟制日期** | **版本描述** | **存档编号** |
| 1.0.12 | 2021.11.29 | 1、增加交通灯控制协议 |  |
| 1.0.11 | 2021.6.21 | 1. 增加设置获取客户定制sn 2. 增加设置获取gpio加锁状态 |  |
| 1.0.10 | 2018.10.23 | 1. 增加NTP配置获取；  2. 增加NTP配置设置； |  |
| 1.0.9 | 2017.03.29 | 1. 增加获取/设置用户自定义数据  2. 增加获取/设置串口参数  3. 增加获取IO输出状态  4. 获取设备存储信息 |  |
| 1.0.8 | 2017.12.12 | 1. 增加算法识别参数的获取/设置  2. 增加算法识别属性的获取  3. 增加LED控制参数的设置/获取  4. 增加LED控制属性的获取  5. 增加恢复设备默认配置 |  |
| 1.0.7 | 2017.11.24 | 1. 增加设置/获取虚拟线圈参数  2. 增加设置/获取识别区域参数 |  |
| 1.0.6 | 2017.10.16 | 1 设备维护管理相关  2 车牌识别中获取视频播放uri  3 注册脱机事件  4 设备组网相关  5 用户登录相关 |  |
| 1.0.5 | 2017.6.16 | 1 清除组网数据  2 得到当前组网内部所有设备配置  3 获取组网图片 |  |
| 1.0.4 | 2016.6.13 | 1 增加设置、获取网络参数  2 增加设置中心服务器配置 |  |
| 1.0.3 | 2016.3.25 | Tcp 协议纠错 |  |
| 1.0.2 | 2016.2.16 | 1 新增Linux环境下java的Demo  2 从此版本开始，版本号以三位整数进行发布 |  |
| 1.0.0.1 | 2016.1.12 | 1 开发手册对接口说明，进行分类，方便用户查看 |  |
|  |  |  |  |

目录

[1 基本说明 7](#_Toc23756)

[1.1 收发命令包与数据包格式 7](#_Toc21242)

[1.2 具体命令 7](#_Toc9600)

[2 Tcp协议命令 8](#_Toc2795)

[2.1 设备维护管理 8](#_Toc3876)

[2.1.1 恢复设备默认配置：set\_factorydefault 8](#_Toc17079)

[2.1.2 获取设备的序列号: getsn 8](#_Toc3639)

[2.1.3 获取设备的硬件版本信息: get\_hw\_board\_version 9](#_Toc17470)

[2.1.4 获取设备当前时间戳: get\_device\_timestamp 9](#_Toc14090)

[2.1.5 设置系统时间：set\_time 10](#_Toc17585)

[2.1.6 设置网络参数：set\_networkparam 10](#_Toc15241)

[2.1.7 获取网络参数：get\_networkparam 11](#_Toc16843)

[2.1.8 设置中心服务器网络参数：set\_centerserver\_net 11](#_Toc20502)

[2.1.9 设置当前配置为用户默认配置：set\_user\_default\_cfg 12](#_Toc23981)

[2.1.10 修改设备admin密码：set\_adminpass 12](#_Toc30835)

[2.1.11 重启设备:reboot\_dev 12](#_Toc22188)

[2.1.12 建安顺APP过期设置：jasauthtime，jasauthtime\_rsp 13](#_Toc1236)

[2.1.13 对DM8127的单板：获取降噪模式和降噪强度：get\_denoise 13](#_Toc25241)

[2.1.14 对DM8127的单板：设置降噪模式和降噪强度：set\_denoise 14](#_Toc2178)

[2.1.15 开始自动聚焦：startfocusandzoom 15](#_Toc12621)

[2.1.16 停止自动聚焦：stopfocusandzoom 15](#_Toc7879)

[2.1.17 获取设备版本信息: get\_product\_info 15](#_Toc14458)

[2.1.18 获取设备4G参数信息: get\_4g\_param 16](#_Toc14321)

[2.1.19 设置设备4G参数信息: set\_4g\_param 17](#_Toc15115)

[2.2 车牌识别 18](#_Toc14253)

[2.2.1 配置推送数据方式: ivsresult 18](#_Toc20645)

[2.2.2 获取最近一次识别结果: getivsresult 20](#_Toc14097)

[2.2.3 手动触发车牌识别：trigger 21](#_Toc5633)

[2.2.4 获取记录最大id: get\_max\_rec\_id 21](#_Toc4101)

[2.2.5 获取历史记录: get\_record 21](#_Toc13618)

[2.2.6 获取记录图片: get\_image、get\_offline\_image 22](#_Toc17646)

[2.2.7 抓取当前图片get\_snapshot 22](#_Toc11550)

[2.2.8 获取视频播放的uri: get\_rtsp\_uri 22](#_Toc11983)

[2.2.9 获取虚拟线圈参数：get\_virloop\_para 23](#_Toc14766)

[2.2.10 设置虚拟线圈参数：set\_virloop\_para 24](#_Toc27952)

[2.2.11 获取虚拟线圈属性：get\_virloop\_prop 24](#_Toc2683)

[2.2.12 获取识别区域参数：get\_reco\_para 25](#_Toc11923)

[2.2.13 设置识别区域参数：set\_reco\_para 26](#_Toc9232)

[~~2.2.14 获取识别区域属性：get\_reco\_prop~~ 26](#_Toc23871)

[2.2.15 获取算法识别参数： get\_alg\_result\_para 27](#_Toc26604)

[2.2.16 设置算法识别参数： set\_alg\_result\_para 28](#_Toc23460)

[2.2.17 获取算法识别参数属性： get\_alg\_result\_prop 28](#_Toc25070)

[2.2.18 设置LED参数： set\_led\_para 30](#_Toc5291)

[2.2.19 获取LED参数： get\_led\_para 31](#_Toc11734)

[2.2.20 获取LED属性： get\_led\_prop 32](#_Toc1958)

[2.3 设备硬件参数配置 32](#_Toc7665)

[2.3.1 控制IO输出: ioctl, ioctl\_resp 32](#_Toc29898)

[2.3.2 获取IO输入状态: get\_gpio\_value 33](#_Toc21568)

[2.3.3 获取IO输出状态: get\_gpio\_out\_value 33](#_Toc13902)

[2.3.4 自动聚焦: auto\_focus, auto\_focus\_rsp 33](#_Toc18823)

[2.3.5 获取存储设备信息: get\_diskinfo 34](#_Toc29603)

[2.3.6 获取串口参数: get\_serial\_para 35](#_Toc23084)

[2.3.7 设置串口参数: set\_serial\_para 35](#_Toc1897)

[2.3.8 设置客户定制sn序列号 36](#_Toc28375)

[2.3.9 获取客户定制SN序列号 36](#_Toc30218)

[2.3.10 设置Gpio口锁定状态 36](#_Toc18215)

[2.3.11 获取gpio口锁定状态 37](#_Toc544)

[2.4 设备连接 37](#_Toc24314)

[2.4.1 通知在线消息:response\_online, response\_online\_rsp 37](#_Toc5389)

[2.4.2 主动断开连接消息:close\_socket 38](#_Toc27423)

[2.5 配置透明通道: ttransmission 38](#_Toc9594)

[请求命令参数 38](#_Toc4200)

[结果返回 39](#_Toc21135)

[2.6 白名单 39](#_Toc20597)

[2.6.1 脱机检查 39](#_Toc25122)

[2.6.2 操作白名单：white\_list\_operator 42](#_Toc16816)

[2.7 识别结果加密 45](#_Toc307)

[2.7.1 获取加密方式：get\_ems 45](#_Toc5203)

[2.7.2 获取用户密码：get\_encrypt\_key 46](#_Toc24939)

[2.7.3 重新设置用户密码：reset\_encrypt\_key 47](#_Toc13421)

[2.7.4 修改用户密码：change\_encrypt\_key 48](#_Toc25468)

[2.7.5 开启是否加密：enable\_encrypt 49](#_Toc6464)

[2.7.6 设置设备有效时间：device\_active\_settings 49](#_Toc15761)

[2.7.7 获取设备有效时间：device\_active\_settings 50](#_Toc5237)

[2.7.8 设置用户私有数据：set\_user\_data 50](#_Toc17970)

[2.7.9 获取用户私有数据：get\_user\_data 51](#_Toc18417)

[2.8 设备组网：dg\_json\_request 51](#_Toc3906)

[2.8.1 得到当前设备vzid：get\_cdvzid 52](#_Toc24403)

[2.8.2 得到当前设备名称：get\_current\_device\_name 53](#_Toc6663)

[2.8.3 得到在线设备信息，不含自己：get\_ovzid 53](#_Toc21090)

[2.8.4 得到在线设备信息，含自己：online\_devices 55](#_Toc29788)

[2.8.5 得到所有连接设备信息：get\_avzid 55](#_Toc24568)

[2.8.6 得到当前组网内所有设备信息：get\_agdi 56](#_Toc20602)

[2.8.7 得到当前设备记录size：current\_records\_size 56](#_Toc11558)

[2.8.8 得到入口设备记录：records\_sparate\_input 57](#_Toc29689)

[2.8.9 得到出口设备记录：records\_sparate\_output 59](#_Toc4781)

[2.8.10 使能设备组网：enable\_devicegroup 62](#_Toc32764)

[2.8.11 设置设备组网类型及参数：set\_device\_type\_enable 63](#_Toc13052)

[2.8.12 查找车牌信息：search\_plate 63](#_Toc25513)

[2.8.13 获取组网匹配模式：get\_device\_match\_mode 66](#_Toc1356)

[2.8.14 设置组网匹配模式: set \_device\_match\_mode 66](#_Toc4945)

[2.8.15 得到组网共享IO：get\_group\_shared\_io 67](#_Toc9890)

[2.8.16 设置组网共享IO：set\_group\_shared\_io 67](#_Toc12798)

[2.8.17 组网识别结果消息 enable\_dg\_result 68](#_Toc1968)

[2.8.18 清除组网数据：reset\_database 71](#_Toc2106)

[2.8.19 得到当前组网内部所有设备配置：get\_group\_cfg 72](#_Toc1774)

[2.8.20 设置当前组网内部所有设备配置：set\_group\_cfg 76](#_Toc14926)

[2.8.21 根据ID获取组网图片：get\_img\_by\_id 76](#_Toc16490)

[2.9 用户登录 77](#_Toc337)

[2.9.1 请求开始登录:start\_login 77](#_Toc26906)

[2.9.2 登录认证: login\_authentication 78](#_Toc1460)

[2.10 语音协议 78](#_Toc31157)

[注 意：此处的语音为相机设备自带的语音，非外接； 79](#_Toc12084)

[请求命令参数如下格式： 79](#_Toc19808)

[回复格式为： 79](#_Toc7512)

[2.10.1 获取当前语音文件列表：playserver\_json\_request 79](#_Toc15021)

[2.10.2 设置语音默认参数：playserver\_json\_request 80](#_Toc21733)

[2.10.3 获取语音默认参数：playserver\_json\_request 80](#_Toc8646)

[2.10.4 播放语音参数：playserver\_json\_request 81](#_Toc20976)

[2.10.5 请求语音对讲：start\_talk 81](#_Toc187)

[2.10.6 监听语音对讲：request\_talk 82](#_Toc27505)

[2.11 屏显协议 82](#_Toc26428)

[注 意： 82](#_Toc7303)

[2.11.1 设置LED显示内容：set\_led\_show 82](#_Toc11604)

[2.11.2 获取LED显示内容：get\_led\_show 83](#_Toc10448)

[2.11.3 获取LED数据传输使用串口号：get\_led\_serial\_port 83](#_Toc11492)

[2.11.4 设置LED数据传输使用串口号：set\_led\_serial\_port 84](#_Toc31869)

[2.11.5 字段信息以及相关枚举 84](#_Toc31715)

[2.12 设置用户自定义OSD 86](#_Toc28173)

[请求命令参数 86](#_Toc21937)

[结果返回 87](#_Toc30416)

[2.13 用户自定义设备信息 87](#_Toc5123)

[2.13.1 获取自定义信息: get\_custom\_user\_info 87](#_Toc24743)

[2.13.2 设置用户自定义信息: set\_custom\_user\_info 88](#_Toc23603)

[2.14 用户通过R3获取伴侣机Rg的相关配置信息 88](#_Toc6809)

[2.14.1 通过R3获取指定伴侣机的配置信息：user\_get\_mate\_info 88](#_Toc3058)

[2.14.2 通过R3获取伴侣机的Rtsp代理信息:user\_get\_rtsp\_info 89](#_Toc13110)

[2.14.3 通过R3获取伴侣机的语音代理信息:user\_request\_talkback 90](#_Toc2225)

[2.14.4 通过Tcp获取日志指定的日志:log\_search 90](#_Toc11994)

[2.14.5 设置Rg人车滞留事件上报使能:stay\_event\_enable 91](#_Toc28482)

[2.15 RG设备Alg参数获取与设置 92](#_Toc24913)

[2.15.1 获取RG绊线绑定算法参数 92](#_Toc19833)

[2.15.2 设置RG绊线绑定算法参数 92](#_Toc28022)

[2.15.3 获取车通行折返事件 93](#_Toc7634)

[2.15.4 设置车通行折返事件 94](#_Toc25264)

[2.15.5 非车滞留事件获取 95](#_Toc15732)

[2.15.6 非车滞留事件设置 96](#_Toc28545)

[2.15.7 获取人滞留事件 96](#_Toc5885)

[2.15.8 设置人滞留事件 97](#_Toc6542)

[2.15.9 获取车滞留事件参数 98](#_Toc20733)

[2.15.10 设置车滞留事件参数 99](#_Toc11962)

[2.15.11 获取布防时间 100](#_Toc1297)

[2.15.12 设置布防时间 100](#_Toc81)

## 车牌识别

### 配置推送数据方式: ivsresult

**请求命令参数**

**{**

**"cmd": "ivsresult",**

**"id" : "123",**

**"enable": true,**

**"format": "json",**

**"image": true,**

**"image\_type": 0**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| enable | 是否允许推送识别结果，默认值：false   * true 表示允许推送识别结果 * false 表示不推送识别结果 |
| format | 推送识别结果数据格式，默认值：json   * bin 表示识别结果数据格式为二进制数据格式 * json表示识别结果数据格式为json数据格式 |
| image | 识别结果是否包含图片，默认值：true   * true 表示识别结果包含图片 * false 表示识别结果不包含图片 |
| image\_type | 识别的图片类型，默认值：0   * 0 表示返回识别结果全图 * 1 表示只返回车牌区域小图 * 2 表示返回两种图片 |

当配置完成后（配置为允许推送识别结果的情况enable：ture），并且在服务器端有ivsresult识别结果产生后，tcp server会主动推送ivsresult识别结果给客户端，推送的识别结果格式如下

**结果返回**

**{**

**"cmd" : " ivsresult",**

**"id" : "123",**

**"state\_code" : 200**

**}**

**配置完成后，接收到识别结果为配置值：**

正常情况下，一个完整的识别结果（返回2种图片的情况）推送消息格式如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **‘V’** | **‘Z’** | **0** | **0** | **长度（数据包的大小）** |
| **识别结果字符串+字符串结束符‘\0’** | | | | |
| **全图的jpg格式数据** | | | | |
| **小图片（车牌区域图片）的jpg格式数据** | | | | |

例如下面就是一个正常的JSON识别推送结果:

{

**"PlateResult" : {**

**"bright" : 0,**

**"carBright" : 0,**

**"carColor" : 0,**

**"colorType" : 1,**

**"colorValue" : 0,**

**"confidence" : 99,**

**"direction" : 0,**

**"license" : "青PTW3Z3",**

**“enable\_encrypt”:0,**

**"location" : {**

**"RECT" : {**

**"bottom" : 392,**

**"left" : 690,**

**"right" : 834,**

**"top" : 350**

**}**

**},**

**"timeStamp" : {**

**"Timeval" : {**

**"sec" : 1458882234,**

**"usec" : 921325**

**}**

**},**

**"timeUsed" : 0,**

**"triggerType" : 1,**

**"type" : 1,**

**"plate\_true\_width" : 1,**

**"plate\_distance" : 50,**

**"fake\_plate" : 1,**

**"car\_location" : {**

**"RECT" : {**

**"bottom" : 392,**

**"left" : 690,**

**"right" : 834,**

**"top" : 350**

**}**

**},**

**"car\_brand" : {**

**"brand" : 1,**

**"type" : 2,**

**"year" : 2017**

**},**

**"featureCode" : "asdgfb4AGD",**

**},**

**"active\_id" : 0,**

**"clipImgSize" : 1103,**

**"cmd" : "ivs\_result",**

**"fullImgSize" : 51566,**

**"id" : 0,**

**"imageformat" : "jpg",**

**"timeString" : "2016-03-25 13:03:54"**

**}**

其中各字段的含义如下表所示：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| license | 车牌号码（汉字为GB2312编码） |
| enable\_encrypt | 车牌是否加密（0不加密，1加密） |
| colorValue | 车牌颜色 |
| colorType | 车牌颜色序号，详见车牌颜色定义LC\_X |
| type | 车牌类型，详见车牌类型定义LT\_X |
| confidence | 车牌可信度 |
| bright | 亮度评价 |
| direction | 运动方向，详见运动方向定义 DIRECTION\_X |
| location：rect | 车牌位置 |
| timeUsed | 识别所用时间 |
| carBright | 车的亮度 |
| carColor | 车的颜色，详见车辆颜色定义LCOLOUR\_X |
| timeStamp | 识别时间点 |
| triggerType | 触发结果的类型,见TH\_TRIGGER\_TYPE\_BIT |
| cmd | 当前指令名称 |
| id | 识别记录的编号 |
| imageformat | 图片格式 |
| timeString | 触发时间字符串，格式如：2015-01-02 03:04:05 |
| fullImgSize | 整幅大图的尺寸（字节数） |
| clipImgSize | 车牌区域图片的尺寸（字节数） |
| active\_id | 车牌加密方式 |
| plate\_true\_width | 车牌真实宽度 |
| plate\_distance | 车牌距离 |
| fake\_plate | 是否是伪车牌 |
| car\_location | 车辆位置 |
| Car\_Brand | 车辆品牌信息 |
| featureCode | 特征码(16位字符串) |

Car\_Brand中个字段信息

|  |  |
| --- | --- |
| brand | 车辆品牌信息 |
| type | 车辆类型 |
| year | 年份 |

另，LC\_X，LT\_X，DIRECTION\_X，LCOLOUR\_X，TH\_TRIGGER\_TYPE\_BIT详细定义请参考头文件VzClientSDK\_LPDefine.h。

如果用户设置的是二进制识别结果，那么将会收到如下的二进制数据：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **‘V’** | **‘Z’** | **2** | **0** | **长度（整个包的大小）** |
| **‘I’** | **‘R’** | **1** | **0** | **长度（识别结果结构体的大小）** |
| **识别结果结构体(TH\_PlateResult)** | | | | |
| **‘I’** | **‘R’** | **2（大图片）** | **0** | **长度（大图像数据的大小）** |
| **全图的图片数据(完整的jpg格式)** | | | | |
| **‘I’** | **‘R’** | **3（小图片）** | **0** | **长度（小图像数据的大小）** |
| **小图片（车牌区域图片）数据(完整的jpg格式)** | | | | |

推送的识别结果结构体定义为TH\_PlateResult，这一个结构体的定义请参考头文件VzClientSDK\_LPDefine.h。

### 获取最近一次识别结果: getivsresult

**请求命令参数**

**{**

**"cmd" : "getivsresult",**

**"image" : true,**

**"format" : "json"**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| image | 是否接收识别结果图片，默认值：false   * true 需要识别结果图片 * false 不需要识别结果图片 |
| format | 推送识别结果数据格式，默认值：json   * bin 表示识别结果数据格式为二进制数据格式 * json表示识别结果数据格式为json数据格式。 |

**结果返回**

这个命令的结果返回值，参见配置推送数据方式里面的识别结果返回值。

### 手动触发车牌识别：trigger

**请求命令参数**

**{**

**"cmd": "trigger"**

**}**

**结果返回**

这个命令的结果返回值，参见配置推送数据方式里面的识别结果返回值。

### 获取记录最大id: get\_max\_rec\_id

主要用于在脱机之后，断线之后使用。

**请求命令参数**

**{**

**"cmd": "get\_max\_rec\_id",**

**"id": "123"**

**}**

**结果返回**

**{**

**"cmd": "get\_max\_rec\_id",**

**"id": “123”,**

**"state\_code":200,**

**"max\_id": 1024**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| max\_id | 识别结果记录中最大的id值 |

### 获取历史记录: get\_record

根据id获取识别记录

**请求命令参数**

**{**

**"cmd": "get\_record",**

**"id": 2,**

**"format": "json",**

**"image": true**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 识别结果记录的id值 |
| format | 推送识别结果数据格式，默认值：json  bin 表示识别结果数据格式为二进制数据格式，当前不支持bin形式。  json表示识别结果数据格式为json数据格式。 |
| image | 识别结果是否包含图片，默认值：true  true 表示识别结果包含图片  false 表示识别结果不包含图片 |

**结果返回**

这个命令的结果返回值，参见配置推送数据方式里面的识别结果返回值。

### 获取记录图片: get\_image、get\_offline\_image

按记录ID获取识别结果对应的图片

**请求命令参数**

**{**

**"cmd": "get\_image",**

**"id": 2**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 识别结果记录的id, 可在识别记录结构体中找到。 |

**结果返回**

**{**

**"cmd": "get\_image",**

**"id": 2,**

**"size": 57344,**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 识别结果记录的id, 可在识别记录结构体中找到。前面的获取脱机记录中的ID也一样 |
| size | 图片的大小 |

注释：这个JSON后面跟了一个57344大小的二进制图片，其中JSON字符串以’\0’结束。

### 抓取当前图片get\_snapshot

**请求命令参数**

**{**

**"cmd":"get\_snapshot",**

**"id":"123456"**

**}**

**结果返回**

**{**

**"cmd":"get\_snapshot",**

**"state\_code":200,**

**"id":"123456",**

**"imgformat":"jpg",**

**"imgdata":"14ewrfswtergfhbfr=="**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30，等于请求时的id |
| state\_code | 状态码 |
| imgformat | 图片格式（jpg） |
| imgdata | 图片数据：经过base64转码后的数据 |

### 获取视频播放的uri: get\_rtsp\_uri

**请求命令参数**

**{**

**"cmd" : "get\_rtsp\_uri",**

**"id" : "132156"**

**}**

**结果返回**

**{**

**"cmd" : " get\_rtsp\_uri ",**

**"id" : "132156",**

**"state\_code" : 200,**

**"uri":"rtsp://192.168.7.7:8557/h264"**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30，等于请求时的id |
| state\_code | 状态码 |
| **uri** | 当前播放视频的URI |

### 获取虚拟线圈参数：get\_virloop\_para

**请求命令参数**

**{**

**"cmd" : "get\_virloop\_para",**

**"id" : "132156"**

**}**

**结果返回**

**{**

**"body" : {**

**"virtualloop" : {**

**"max\_plate\_width" : 400,**

**"min\_plate\_width" : 45,**

**"dir" : 0,**

**"trigger\_gap" : 10,**

**"virtualloop\_num" : 1,**

**"loop" : [ {**

**"id" : 0,**

**"enable" : true,**

**"point\_num" : 4,**

**"point" :**

**[ { "x" : 2400, "y" : 12000 },**

**{ "x" : 13984, "y" : 12000 },**

**{ "x" : 15984, "y" : 14000 },**

**{ "x" : 400, "y" : 14000 } ]**

**}]**

**}**

**},**

**"cmd" : "get\_virloop\_para",**

**"error\_msg" : "success",**

**"id" : "132156",**

**"state\_code" : 200**

**}**

Body中个字段含义

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **virtualloop** | 虚拟线圈 |
| **max\_plate\_width** | 最大车牌尺寸 |
| **min\_plate\_width** | 最小车牌尺寸 |
| **dir** | 运动方向 |
| **trigger\_gap** | 相同车牌的触发时间间隔 |
| **virtualloop\_num** | 虚拟线圈数量 |

**L**oop中字段含义：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **id** | 线圈ID |
| **enable** | 是否可用 |
| **point\_num** | 顶点个数 |
| **Poin** | 顶点坐标   * X横向坐标，从左至右，从0递增 * Y纵向坐标，从上至下，从0递增 |

### 设置虚拟线圈参数：set\_virloop\_para

**请求命令参数**

**{**

**"body" : {**

**"virtualloop" : {**

**"max\_plate\_width" : 400,**

**"min\_plate\_width" : 45,**

**"dir" : 0,**

**"trigger\_gap" : 10,**

**"virtualloop\_num" : 1,**

**"loop" : [ {**

**"id" : 0,**

**"enable" : true,**

**"point\_num" : 4,**

**"point" :**

**[ { "x" : 2400, "y" : 12000 },**

**{ "x" : 13984, "y" : 12000 },**

**{ "x" : 15984, "y" : 14000 },**

**{ "x" : 400, "y" : 14000 } ]**

**}]**

**}**

**},**

**"cmd" : "set\_virloop\_para",**

**"id" : "132156"**

**}**

**结果返回**

**{**

**"cmd" : "set\_virloop\_para",**

**"id" : "132156",**

**"error\_msg" : "success",**

**"state\_code" : 200**

**}**

**参数含义请参考2.2.9上方；**

### 获取虚拟线圈属性：get\_virloop\_prop

**请求命令参数**

**{**

**"cmd" : "get\_virloop\_prop",**

**"id" : "132156"**

**}**

**结果返回**

**{**

**"body" :{**

**"virtualloop" :{**

**"plate\_width" :{**

**"max" : 600,**

**"min" : 45,**

**"default\_max" : 400,**

**"default\_min" : 25**

**},**

**"trigger\_gap" :{**

**"max" : 255,**

**"min" : 0,**

**"default" : 10**

**},**

**"dir" :{**

**"default" : 0,**

**"types" :**

**[{"content" : "双向", "type" : 0},**

**{"content" : "从上至下", "type" : 1},**

**{"content" : "从下至上", "type" : 2}]**

**}**

**}**

**}**

**"cmd" : "get\_virloop\_prop",**

**"error\_msg" : "success",**

**"id" : "132156",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **plate\_width** | 车牌宽度   * max 最大宽度 * min最小宽度 * default\_max默认最大宽度 * default\_min默认最小宽度 |
| **trigger\_gap** | 相同车牌的触发时间间隔   * max 最长间隔时间 * min最短间隔时间 * default默认间隔时间 |
| **dir** | 运动方向   * content”双向”, type,0 * content”从上至下”,type,1 * content”从下至上”,type2 * default默认值 |

### 获取识别区域参数：get\_reco\_para

**请求命令参数**

**{**

**"cmd" : "get\_reco\_para",**

**"id" : "132156"**

**}**

**结果返回**

**{**

**"body" : {**

**"recognition\_area" : {**

**"polygon\_num" : 1,**

**"polygon" :**

**[ {**

**"id" : 1,**

**"enable" : true,**

**"point\_num" : 4,**

**"point" :**

**[ { "x" : 4096, "y" : 4096 },**

**{ "x" : 12288, "y" : 4096},**

**{ "x" : 13926, "y" : 13926 },**

**{ "x" : 2457, "y" : 13926 } ]**

**}]**

**}**

**},**

**"cmd" : "get\_reco\_para",**

**"id" : "132156",**

**"error\_msg" : "success",**

**"state\_code" : 200**

**}**

**recognition\_area识别区域中字段含义**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **polygon\_num** | 规则数量 |
| **enable** | 是否可用 |
| **point\_num** | 顶点个数 |
| **Poin** | 顶点坐标   * X横向坐标，从左至右，从0递增 * Y纵向坐标，从上至下，从0递增 |

### 设置识别区域参数：set\_reco\_para

**请求命令参数**

**{**

**"body" : {**

**"recognition\_area" : {**

**"polygon\_num" : 1,**

**"polygon" :**

**[ {**

**"id" : 1,**

**"enable" : true,**

**"point\_num" : 4,**

**"point" :**

**[ { "x" : 4096, "y" : 4096 },**

**{ "x" : 12288, "y" : 4096},**

**{ "x" : 13926, "y" : 13926 },**

**{ "x" : 2457, "y" : 13926 } ]**

**}]**

**}**

**},**

**"cmd" : "set\_reco\_para",**

**"id" : "132156"**

**}**

**结果返回**

**{**

**"cmd" : "set\_reco\_para",**

**"id" : "132156",**

**"error\_msg" : "success",**

**"state\_code" : 200**

**}**

**参数字段含义参见2.2.13上方**

### ~~获取识别区域属性：get\_reco\_prop~~

**~~请求命令参数~~**

**~~{~~**

**~~"cmd" : "get\_reco\_prop",~~**

**~~"id" : "132156"~~**

**~~}~~**

**~~结果返回~~**

**~~{~~**

**~~"body" : {~~**

**~~"polygon\_num" : { "max" : 1, "min" : 0 },~~**

**~~"polygon" : [ {~~**

**~~"id" : 1,~~**

**~~"enable" : true,~~**

**~~"point\_num" : 4,~~**

**~~"point" :~~**

**~~[ { "x" : 4096, "y" : 4096 },~~**

**~~{ "x" : 12288, "y" : 4096 },~~**

**~~{ "x" : 13926, "y" : 13926},~~**

**~~{ "x" : 2457, "y" : 13926 } ]~~**

**~~} ]~~**

**~~},~~**

**~~"cmd" : "get\_reco\_prop",~~**

**~~"id" : "132156",~~**

**~~"error\_msg" : "success",~~**

**~~"state\_code" : 200~~**

**~~}~~**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **polygon\_num** | 规则数量   * Max最大数量 * Min最小数量 |

**Polygon中各字段含义**

|  |  |
| --- | --- |
| **id** | 规则ID |
| **enable** | 是否可用 |
| **point\_num** | 顶点个数 |
| **Poin** | 顶点坐标   * X横向坐标，从左至右，从0递增 * Y纵向坐标，从上至下，从0递增 |

### 获取算法识别参数： get\_alg\_result\_para

**请求命令参数**

**{**

**"cmd" : "get\_alg\_result\_para",**

**"id" : "132156"**

**}**

**结果返回**

**{**

**"cmd" : " get\_alg\_result\_para",**

**"id" : "132156",**

**"state\_code" : 200,**

**"body" : {**

**“snap\_resolution” : 0,**

**"snap\_image\_quality" : 100,**

**"out\_result\_type" : 1,**

**"recognition\_type" : 1,**

**"province" : 1,**

**"run\_mode":0 ,**

**"alg\_version" : "2016LPRALG",**

**"time\_zone" : 0,**

**"reco\_dis" : 0**

**}**

**}**

**字段含义**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **snap\_resolution** | 抓拍图片分辨率 |
| **snap\_image\_quality** | 抓拍图片质量，单位百分比; |
| **out\_result\_type** | 输出类型; |
| **recognition\_type** | 识别类型 |
| **province** | 预设省份 |
| **run\_mode** | 运行模式 |
| **alg\_version** | 算法版本 |
| **time\_zone** | 时区 |
| **reco\_dis** | 识别距离 0: 1-6米 1: 大于6米 |

### 设置算法识别参数： set\_alg\_result\_para

**请求命令参数**

**{**

**"cmd" : "set\_alg\_result\_para",**

**"id" : "132156",**

**"body" : {**

**"snap\_resolution" : 0,**

**"snap\_image\_quality" : 100,**

**"out\_result\_type" : 1,**

**"recognition\_type" : 1,**

**"province" : 1,**

**"run\_mode":0 ,**

**"alg\_version" : "2016LPRALG",**

**"time\_zone" : 0,**

**"reco\_dis" : 0**

**}**

**}**

**结果返回**

**{**

**"cmd" : "set\_alg\_result\_para",**

**"id" : "132156",**

**"state\_code" : 200**

**}**

**字段含义**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **snap\_resolution** | 抓拍图片分辨率 |
| **snap\_image\_quality** | 抓拍图片质量，单位百分比; |
| **out\_result\_type** | 输出类型; |
| **recognition\_type** | 识别类型 |
| **province** | 预设省份 |
| **run\_mode** | 运行模式 |
| **alg\_version** | 算法版本 |
| **time\_zone** | 时区 |
| **reco\_dis** | 识别距离 0: 1-6米 1: 大于6米 |

### 获取算法识别参数属性： get\_alg\_result\_prop

**请求命令参数**

**{**

**"cmd" : "get\_alg\_result\_prop",**

**"id" : "132156"**

**}**

**结果返回**

**{**

**"cmd" : "get\_alg\_result\_prop",**

**"id" : "132156",**

**"state\_code" : 200,**

**"body" :**

**{**

**"snap\_resolution" :**

**{**

**"default" : 10,**

**"types" : [**

**{ "content" : "720\*576", "type" : 6 },**

**{ "content" : "1280\*720", "type" : 9 },**

**{ "content" : "1920\*1080", "type" : 10 } ]**

**},**

**"snap\_image\_quality" :**

**{**

**"max" : 100,**

**"min" : 10,**

**"default" : 70**

**},**

**"out\_result\_type" :**

**{**

**"default" : 16,**

**"types" : [**

**{ "content" : "稳定识别触发", "type" : 1 },**

**{ "content" : "虚拟线圈", "type" : 2 },**

**{ "content" : "IO1触发", "type" : 16},**

**{ "content" : "IO2触发", "type" : 32 },**

**{ "content" : "IO3触发", "type" : 64 }**

**]**

**},**

**"recognition\_type" :**

**{**

**"default" : 60282,**

**"types" : [**

**{ "content" : "蓝牌", "type" : 1 },**

**{ "content" : "黄牌", "type" : 24},**

**{ "content" : "黑牌", "type" : 4 },**

**{ "content" : "教练车", "type" : 8192},**

**{ "content" : "警车", "type" : 32 },**

**{ "content" : "武警车", "type" : 32832 },**

**{ "content" : "军车", "type" : 768 },**

**{ "content" : "港澳", "type" : 18432},**

**{ "content" : "使馆车", "type" : 1024}**

**]**

**},**

**"province" :**

**{**

**"default" : 0,**

**"types" : [**

**{ "content" : "无", "type" : 255 },**

**{ "content" : "京", "type" : 0 },**

**{ "content" : "津", "type" : 1 },**

**{ "content" : "冀", "type" : 2 },**

**{ "content" : "晋", "type" : 3 },**

**{ "content" : "蒙", "type" : 4 },**

**{ "content" : "辽", "type" : 5 },**

**{ "content" : "吉", "type" : 6 },**

**{ "content" : "黑", "type" : 7 },**

**{ "content" : "沪", "type" : 8 },**

**{ "content" : "苏", "type" : 9 },**

**{ "content" : "浙", "type" : 10 },**

**{ "content" : "皖", "type" : 11 },**

**{ "content" : "闽", "type" : 12 },**

**{ "content" : "赣", "type" : 13 },**

**{ "content" : "鲁", "type" : 14 },**

**{ "content" : "豫", "type" : 15 },**

**{ "content" : "鄂", "type" : 16 },**

**{ "content" : "湘", "type" : 17 },**

**{ "content" : "粤", "type" : 18 },**

**{ "content" : "桂", "type" : 19 },**

**{ "content" : "琼", "type" : 20 },**

**{ "content" : "渝", "type" : 21 },**

**{ "content" : "川", "type" : 22 },**

**{ "content" : "贵", "type" : 23 },**

**{ "content" : "云", "type" : 24 },**

**{ "content" : "藏", "type" : 25 },**

**{ "content" : "陕", "type" : 26 },**

**{ "content" : "甘", "type" : 27 },**

**{ "content" : "青", "type" : 28 },**

**{ "content" : "宁", "type" : 29 },**

**{ "content" : "新", "type" : 30 },**

**{ "content" : "港", "type" : 31 },**

**{ "content" : "澳", "type" : 32 },**

**{ "content" : "台", "type" : 33 },**

**{ "content" : "警", "type" : 34 },**

**{ "content" : "使", "type" : 35 },**

**{ "content" : "WJ", "type" : 36 },**

**{ "content" : "领", "type" : 37 },**

**{ "content" : "学", "type" : 38 } ]**

**},**

**"run\_mode" :**

**{**

**"default" : 0,**

**"types" : [**

**{ "content" : "默认", "type" : 0 },**

**{ "content" : "停车场", "type" : 1 },**

**{ "content" : "卡口", "type" : 2 } ]**

**},**

**"time\_zone" : { "default" : 0 },**

**"reco\_dis" :**

**{**

**"default" : 0,**

**"types" : [**

**{ "content" : "2-4m", "type" : 0 },**

**{ "content" : "6-8m", "type" : 1 },**

**{ "content" : "4-6m", "type" : 2}**

**]**

**}**

**}**

**}**

### 设置LED参数： set\_led\_para

**请求命令参数**

**{**

**"body" : {**

**"led\_level" : 2,**

**"led\_mode" : 3,**

**"time\_ctrl" : [**

**{**

**"id" : 0,**

**"led\_level" : 2,**

**"time\_begin" : "00:00:00",**

**"time\_end" : "02:03:34",**

**"timectrl\_enable" : true**

**},**

**{**

**"id" : 1,**

**"led\_level" : 6,**

**"time\_begin" : "02:03:34",**

**"time\_end" : "09:02:23",**

**"timectrl\_enable" : true**

**}**

**]**

**},**

**"cmd" : "set\_led\_para",**

**"id" : "132156"**

**}**

结果返回

**{**

**"cmd" : "set\_led\_para",**

**"id" : "132156",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **led\_mode** | LED模式   * 0 智能 * 1 常亮 * 2 常灭 * 3时间段 |
| **led\_level** | LED等级[0,5] |
| **time\_ctrl** | 时间段控制 |
| **id** | 时间段ID |
| **time\_begin** | 时间段起始时间 |
| **time\_end** | 时间段结束时间 |
| **timectrl\_enable** | 时间段使能（0去使能 1使能） |

### 获取LED参数： get\_led\_para

**请求命令参数**

**{**

**"cmd" : "get\_led\_para",**

**"id" : "132156"**

**}**

**结果返回**

**{**

**"body" : {**

**"led\_level" : 2,**

**"led\_mode" : 3,**

**"time\_ctrl" : [**

**{**

**"id" : 0,**

**"led\_level" : 2,**

**"time\_begin" : "00:00:00",**

**"time\_end" : "01:02:23",**

**"timectrl\_enable" : true**

**},**

**{**

**"id" : 1,**

**"led\_level" : 6,**

**"time\_begin" : "01:02:24",**

**"time\_end" : "09:02:23",**

**"timectrl\_enable" : true**

**}**

**]**

**},**

**"cmd" : "get\_led\_para",**

**"id" : "132156",**

**"state\_code" : 200**

**}**

### 获取LED属性： get\_led\_prop

**请求命令参数**

**{**

**"cmd" : "get\_led\_prop",**

**"id" : "132156"**

**}**

**结果返回**

**{**

**"cmd" : " get\_led\_prop ",**

**"id" : "132156",**

**"state\_code" : 200,**

**"body" :**

**{**

**"led\_mode" :**

**{**

**"default" : 0,**

**"types" : [**

**{ "content" : "智能", "type" : 0 },**

**{ "content" : "常亮", "type" : 1 },**

**{ "content" : "常灭", "type" : 2 },**

**{ "content" : "时间段", "type" : 3 }**

**]**

**},**

**"led\_level" :**

**{**

**"default" : 4,**

**"max" : 5,**

**"min" : 0**

**}**

**},**

**}**

## 设备硬件参数配置

### 控制IO输出: ioctl, ioctl\_resp

**请求命令参数示例**

**{**

**"cmd" : "ioctl",**

**"id" : "132156",**

**"delay" : 500,**

**"io" : 0,**

**"value" : 2**

**}**

**{**

**"cmd" : "ioctl\_resp",**

**"id" : "132156",**

**"delay" : 500,**

**"io" : 0,**

**"value" : 2**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| io | 对应的输出IO编号，必需值，正常情况下只有0，1两个输出编号 |
| value | 输出IO的状态值   * 0 断 * 1 通 * 2 先通后断（一般做开闸使用） |
| delay | 先通后断的延迟时间 |

**结果返回**

**当发送CMD为：**ioctl\_resp时有如下返回：

**{**

**"cmd" : " ioctl\_resp ",**

**"id" : "132156",**

**"state\_code" : 200**

**}**

### 获取IO输入状态: get\_gpio\_value

**请求命令参数示例**

**{**

**"cmd" : "get\_gpio\_value",**

**"id" : "123",**

**"gpio" : 0**

**}**

**结果返回**

**{**

**"cmd" : "get\_gpio\_value",**

**"id" : "123",**

**"state\_code" : 200,**

**"gpio" : 0,**

**"value" : 1**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| gpio | 对应的输入IO编号，必需值，正常情况下只有0，1两个输入编号 |
| value | 输入IO的状态值   * 0 低电平 * 1 高电平 |

### 获取IO输出状态: get\_gpio\_out\_value

**请求命令参数示例**

**{**

**"cmd" : "get\_gpio\_out\_value",**

**"id" : "123",**

**"gpio" : 0**

**}**

**结果返回**

**{**

**"cmd" : "get\_gpio\_out\_value",**

**"id" : "123",**

**"state\_code" : 200,**

**"gpio" : 0,**

**"value" : 1**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| gpio | 对应的输入IO编号，必需值，正常情况下只有0，1两个输入编号 |
| value | 输入IO的状态值   * 0 低电平 * 1 高电平 |

### 自动聚焦: auto\_focus, auto\_focus\_rsp

**请求命令参数**

**{**

**"cmd": "auto\_focus"**

**}**

**{**

**"cmd": "auto\_focus\_rsp",**

**"id" : "123"**

**}**

**结果返回**

可观察到设备有自动聚焦的动作发生

当cmd为：auto\_focus\_rsp时，返回如下：

**{**

**"cmd": " auto\_focus\_rsp ",**

**"id" : "123",**

**"state\_code" : 200**

**}**

### 获取存储设备信息: get\_diskinfo

**请求命令参数**

**{**

**"cmd" : "get\_diskinfo",**

**"id":"123"**

**}**

**结果返回**

**{**

**"body" : [**

**{**

**"devname" : "/dev/mmcblk0",**

**"devparts" : [**

**{**

**"formatpercent" : 0,**

**"partname" : "/dev/mmcblk0p1",**

**"partspace" : {**

**"left" : 7602,**

**"total" : 7932,**

**"used" : 330**

**},**

**"partstate" : 5**

**}**

**],**

**"devtotal" : 7948,**

**"devtype" : 0**

**}**

**],**

**"cmd" : "get\_diskinfo",**

**"error\_msg" : "success",**

**"id" : "123",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **devname** | 当前磁盘名称 |
| **formatpercent** | 格式化百分比，此值不用关注 |
| **left** | 剩余磁盘空间 |
| **total** | 磁盘总空间 |
| **used** | 已使用磁盘空间 |
| **partstate** | 分区状态   1. 错误 2. 未格式化 3. 未挂载 4. 正在格式化 5. 正在删除文件 6. 正常工作 7. 只读 8. 慢速磁盘 9. 磁盘损坏 |
| **devtype** | 磁盘类型   1. SD卡 2. HD卡 |
| **Devtotal** | 设备总内存 |

### 获取串口参数: get\_serial\_para

**请求命令参数**

**{**

**"cmd" : "get\_serial\_para",**

**"id":"123",**

**"serial\_port":0**

**}**

**结果返回**

**{**

**"body" : {**

**"baud\_rate" : 9600,**

**"data\_bits" : 8,**

**"parity" : 0,**

**"stop\_bits" : 1**

**},**

**"cmd" : "get\_serial\_para",**

**"id" : "123",**

**"serial\_port" : 1,**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **serial\_port** | 串口号 |
| **baud\_rate** | 波特率：  2400/4800/9600/19200/38400/57600/115200 |
| **data\_bits** | 数据位：固定8 |
| **parity** | 校验位：   1. 无校验 2. 奇校验 3. 偶校验 |
| **stop\_bits** | 停止位：   1. 停止位1 2. 停止位2 |

### 设置串口参数: set\_serial\_para

**请求命令参数**

**{**

**"body" : {**

**"baud\_rate" : 37100,**

**"data\_bits" : 8,**

**"parity" : 1,**

**"stop\_bits" : 2**

**},**

**"cmd" : "set\_serial\_para",**

**"id" : "123",**

**"serial\_port" : 0**

**}**

**结果返回**

**{**

**"cmd" : "set\_serial\_para",**

**"error\_msg" : "Sucess",**

**"id" : "123",**

**"state\_code" : 200**

**}**

### 设置客户定制sn序列号

**请求命令参数**

**{**

**"cmd":"set\_oem\_sn\_info",**

**"id":"1324567",**

**"body":{**

**"oem\_sn":"1234-5689"**

**}**

**}**

**结果返回**

**{**

**"cmd" : "set\_oem\_sn\_info",**

**"error\_msg" : "Sucess",**

**"id" : "1324567",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **oem\_sn** | 客户定制的序列号 |

### 获取客户定制SN序列号

**请求命令参数**

**{**

**"cmd":"get\_oem\_sn\_info",**

**"id":"1324567"**

**}**

**结果返回**

**{**

**"cmd":"get\_oem\_sn\_info",**

**"id":"1324567",**

**"state\_code" : 200,**

**"body":{**

**"oem\_sn":"1234-5678"**

**}**

**}**

### 设置Gpio口锁定状态

**请求命令参数**

**{**

**"cmd":"set\_io\_lock\_status",**

**"id":"1234568",**

**"body":[**

**{**

**"ioout":0,**

**"status":0**

**}**

**]**

**}**

**结果返回**

**{**

**"cmd" : "set\_io\_lock\_status",**

**"error\_msg" : "Sucess",**

**"id" : "1234568",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **ioout** | ioout输出口 0或者1 |
| **status** | 0 解锁 1高电平锁定 2 低电平锁定 |

### 获取gpio口锁定状态

**请求命令参数**

**{**

**"cmd":"get\_io\_lock\_status",**

**"id":"1234569"**

**}**

**结果返回**

**{**

**"body" : [**

**{**

**"ioout" : 0,**

**"status" : 1**

**}**

**],**

**"cmd" : "get\_io\_lock\_status",**

**"error\_msg" : "Sucess",**

**"id" : "1234569",**

**"state\_code" : 200**

**}**

### 获取交通灯状态【熵基R5D定制】

**请求命令参数**

**{**

**"cmd" : "get\_traffic\_lights"**

**}**

**结果返回**

**{**

**"body" : {**

**"enable" : 1,**

**"green\_gpio" : 1,**

**"red\_gpio" : 0,**

**"time" : 15000**

**},**

**"cmd" : "get\_traffic\_lights",**

**"error\_msg" : "Succeed",**

**"id" : "999999",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| enable | 是否启用交通灯控制 0：不启用 1：启用 |
| **green\_gpio** | 绿灯控制IO口 |
| **red\_gpio** | 红灯控制IO口 |
| **time** | 开闸绿灯延时时间(单位：ms) |

### 设置交通灯功能【熵基R5D定制】

**请求命令参数**

**{**

**"cmd" : "set\_traffic\_lights",**

**"body" : {**

**"enable" : 1,**

**"green\_gpio" : 1,**

**"red\_gpio" : 0,**

**"time" : 15000**

**}**

**}**

**结果返回**

**{**

**"cmd" : "set\_traffic\_lights",**

**"error\_msg" : "Succeed",**

**"id" : "999999",**

**"state\_code" : 200**

**}**

## 设备连接

### 通知在线消息:response\_online, response\_online\_rsp

**请求命令参数**

**{**

**"cmd": "response\_online",**

**"id" : "123"**

**}**

**{**

**"cmd": "response\_online\_rsp",**

**"id" : "123"**

**}**

**结果返回**

**当发送cmd为：**response\_online\_rsp时有如下返回：

**{**

**"cmd": "response\_online",**

**"id" : "123",**

**"state\_code" : 200**

**}**

### 主动断开连接消息:close\_socket

**请求命令参数**

**无请求，服务器主动发送给客户端，发送完此命令后服务器会主动断开客户端。**

**服务器发送给客户端**

**{**

**"cmd": "close\_socket ",**

**"state\_code" : 129,**

**"error\_msg" : "close socket"**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| cmd | 命令，固定为**close\_socket** |
| state\_code | 错误码：   * 129 应用层异步发送缓慢出现堵包 * 130 应用层写数据时连接断开 * 131 应用层连续出现错误数据断开连接 * 132 应用层接收包数据小于0（底层读错误） * 其他 系统层错误，对应的code参考linux下的error code |
| error\_msg | 错误信息 |

## 配置透明通道: ttransmission

**请求命令参数**

**{**

**"cmd": "ttransmission",**

**"id" : "999999",**

**"subcmd": "send",**

**"datalen": 6,**

**"data": "QUJDREVG",**

**"comm": "rs485-3"**

**}**

**{**

**"cmd": "ttransmission",**

**"id" : "999999",**

**"subcmd": "init",**

**"data": "rs485-3"**

**}**

**{**

**"cmd": "ttransmission",**

**"id" : "999999",**

**"subcmd": "uninit"**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| subcmd | 子命令   * init 初始化 * uninit 取消初始化 * send 发送数据 |
| datalen | 数据长度，实际数据长度（即编码前的数据长度） |
| data | 字符串数据, 经过base64编码后的数据 |
| comm | comm 口   * rs485-1 * rs485-2 |
| id | 32位无符号整数，保留字段 |

**结果返回**

如果初始化错误，则返回:

**{**

**"cmd": "ttransmission",**

**"id" : "999999",**

**"subcmd": "init",**

**"state\_code" : 400,**

**"response": "failed"**

**}**

如果初始化成功，则返回:

**{**

**"cmd": "ttransmission",**

**"id" : "999999",**

**"state\_code" : 200,**

**"subcmd": "init",**

**"response": "ok"**

**}**

如果取消初始化失败，则返回

**{**

**"cmd": "ttransmission",**

**"id" : "999999",**

**"subcmd": "uninit",**

**"state\_code" : 400,**

**"response": "failed"**

**}**

如果取消初始化成功，则返回:

**{**

**"cmd": "ttransmission",**

**"id" : "999999",**

**"state\_code" : 200,**

**"subcmd": "uninit",**

**"response": "ok"**

**}**

发送数据没有任何返回值。

|  |  |
| --- | --- |
| **字段名** | **说明** |
| subcmd | 子命令   * init 初始化 * uninit 取消初始化 * send 发送数据 |
| response | 返回结果 |

## 白名单

### 脱机检查

#### 注册脱机功能: reg\_offline\_check

将当前TCP客户端注册为脱机检测的响应终端。

**请求命令参数**

**{**

**"cmd" : "reg\_offline\_check",**

**"id" : "999999",**

**"interval" : 2**

**}**

**结果返回**

**{**

**"cmd": "offline",**

**"id" : "999999",**

**"state\_code" : 200,**

**"response": "ok"**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| interval | 脱机响应的超时时间（秒为单位），如果超时未响应，则设备转为脱机状态。 |
| response | 返回结果 |

#### 取消脱机注册: reg\_offline\_check

**请求命令参数**

**{**

**"cmd" : "reg\_offline\_check",**

**"id" : "999999",**

**"sucmd" : "cancel"**

**}**

**结果返回**

#### 注册脱机事件：register\_offline\_event

**请求命令参数示例**

**{**

**"cmd" : "register\_offline\_event",**

**"id" : "132156",**

**"body":**

**{**

**"register\_status": 1**

**}**

**}**

**结果返回**

**{**

**"cmd" : "register\_offline\_event",**

**"id" : "132156",**

**"state\_code" : 200,**

**"error\_msg":"Sucess"**

**}**

**未注册脱机，返回错误：**

**{**

**"cmd" : "register\_offline\_event",**

**"id" : "132156",**

**"state\_code" : 400,**

**"error\_msg":"The session do not reg OfflineCheck,please reg it first"**

**}**

**当脱机状态改变且register\_status!=0时，收到消息：**

**{**

**"cmd" : "offline\_status\_change",**

**"body":**

**{**

**"offline\_status": 0**

**}**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **register\_status** | 注册状态，当发生改变且**register\_status!=0**时，会收到改变响应消息； |

**备注**：注册脱机事件之前，需要先注册脱机功能，否则失败；

#### 获取脱机状态：get\_offline\_status

**请求命令参数示例**

**{**

**"cmd" : "get\_offline\_status",**

**"id" : "132156"**

**}**

**结果返回：**

**{**

**"cmd" : "get\_offline\_status",**

**"id" : "132156",**

**"state\_code" : 200,**

**"body":**

**{**

**"offline\_status": 1**

**}**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **offline\_status** | 当前脱机状态；0：脱机，1：在线 |

#### 注册OpenSDK监听推送：register\_push\_channel

**请求命令参数示例**

**{**

**"cmd" : "register\_push\_channel",**

**"id" : "132156",**

**"body":{**

**"register\_status": 1**

**}**

**}**

**结果返回**

**{**

**"cmd" : "register\_push\_channel",**

**"id" : "132156",**

**"state\_code" : 200,**

**"error\_msg" :"Success"**

**}**

**有推送消息且register\_status!=0时，会收到推送消息：**

**{**

**"cmd" : "opensdk\_push\_message",**

**"body":{**

**XXXXXXX**

**}**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **register\_status** | 注册openSDK状态，当有推送消息且**register\_status!=0**时，会收到推送消息； |

#### 请求OpenSDK Push：push\_msg\_to\_opensdk

**请求命令参数示例**

**{**

**"cmd" : "push\_msg\_to\_opensdk",**

**"id" : "132156",**

**"body":{**

**Xxxxxxx**

**}**

**}**

**结果返回**

**{**

**"cmd" : "push\_msg\_to\_opensdk",**

**"id" : "132156",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| 字段名 | 说明 |
| id | 序列字符串（唯一），字符串长度 < 30，等于请求时的id |
| state\_code | 状态码 |

### 操作白名单：white\_list\_operator

#### 增加或者更新白名单：update\_or\_add

**请求命令参数示例**

**{**

**"cmd": "white\_list\_operator",**

**"id":"123",**

**"operator\_type": "update\_or\_add",**

**"dldb\_rec": {**

**"create\_time": "2015-10-10 12:30:40",**

**"enable\_time": "2015-10-20 12:30:40",**

**"overdue\_time": "2016-10-20 12:30:40",**

**"enable": 1,**

**"plate": "京A12345",**

**"time\_seg\_enable": 1,**

**"seg\_time": "2016-10-20 12:30:40",**

**"need\_alarm": 1,**

**"vehicle\_code": "3254ASFDSFSD",**

**"vehicle\_comment": "HELOO woradf",**

**"customer\_id": 144413212**

**}**

**}**

结果返回

先根据车牌号进行查找，如果在白名单中找到数据，则为更新记录。如果没有在白名单中找到数据则为添加数据。

其中时间要特别注意，必须是标准的格式：“XXXX-XX-XX XX:XX:XX”。这样才能够和数据库相互对应。

|  |  |
| --- | --- |
| **字段名** | **说明** |
| operator\_type | 子命令   * update\_or\_add 增加或更新白名单 * delete 删除白名单 * Select 白名单查询 |
| create\_time | 白名单创建时间 |
| enable\_time | 白名单生效时间 |
| overdue\_time | 白名单失效时间 |
| enable | 是否启动这条规则 |
| plate | 车牌号 |
| time\_seg\_enable | 是否启用时间段，如果是，那么下面seg\_time，就会生效 |
| seg\_time | 如果上面启动了时间段。这个字段就开始生效了。最长1024个字符 |
| need\_alarm | 是否需要报警 |
| vehicle\_code | 用户自定义代码，是一个字符串。最长32个字符，不允许重复 |
| vehicle\_comment | 用户自定义的注释，也是一个字符串。最长32个字符 |
| customer\_id | 用户自己定义ID，是一个整数，数据库内不保证唯一性 |

增加或者更新返回值:

**{**

**"cmd": "white\_list\_operator",**

**“id”:”123”,**

**state\_code" : 200,**

**"operator\_type": "update\_or\_add",**

**"state": "succeed"**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| operator\_type | 子命令   * update\_or\_add 增加或更新白名单 * delete 删除白名单 * select 白名单查询 |
| state | 返回结果 |

#### 以车牌号查询白名单：plate

**请求命令参数**

**{**

**"cmd" : "white\_list\_operator",**

**"id" : "999999",**

**"operator\_type" : "select",**

**"plate" : "川A07273",**

**"sub\_type" : "plate"**

**}**

**结果返回**

这个查询是模糊查询，如果车牌号为空的话，那么就会查询到所有的记录。如果只有一部分，那么就可能查询到相似的车牌信息。

|  |  |
| --- | --- |
| **字段名** | **说明** |
| operator\_type | 子命令   * update\_or\_add 增加或更新白名单 * delete 删除白名单 * Select 白名单查询 |
| sub\_type | 子子命令   * plate 根据车牌号进行查询 |
| plate | 车牌号 |

以车牌号查询白名单返回结果：

**{**

**"cmd" : "white\_list\_operator",**

**"id" : "999999",**

**"state\_code" : 200,**

**"dldb\_rec" : [**

**{**

**"create\_time" : "2016-03-25 14:23:58",**

**"customer\_id" : 144413212,**

**"enable" : 6,**

**"enable\_time" : "2015-12-25 11:00:00",**

**"index" : 1,**

**"need\_alarm" : 1,**

**"overdue\_time" : "2015-1-20 11:00:00",**

**"plate" : "川A07273",**

**"seg\_time" : "2015-1-20 11:00:00",**

**"time\_seg\_enable" : 1,**

**"vehicle\_code" : "1",**

**"vehicle\_comment" : "HELOO woradf"**

**}**

**],**

**"operator\_type" : "select",**

**"state" : "succeed"**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| operator\_type | 子命令   * update\_or\_add 增加或更新白名单 * delete 删除白名单 * select 白名单查询 |
| index | 此条记录在数据库中的id |
| create\_time | 白名单创建时间 |
| enable\_time | 白名单生效时间 |
| overdue\_time | 白名单失效时间 |
| enable | 是否启动这条规则 |
| plate | 车牌号 |
| time\_seg\_enable | 是否启用时间段，如果是，那么下面seg\_time，就会生效 |
| seg\_time | 如果上面启动了时间段。这个字段就开始生效了。最长1024个字符 |
| need\_alarm | 是否需要报警 |
| vehicle\_code | 用户自定义代码，是一个字符串。最长32个字符，不允许重复 |
| vehicle\_comment | 用户自定义的注释，也是一个字符串。最长32个字符 |
| customer\_id | 用户自己定义ID，是一个整数，数据库内不保证唯一性 |
| state | 返回结果状态 |

#### 白名单删除：delete

**请求命令参数**

**{**

**"cmd":"white\_list\_operator",**

**"id" : "999999",**

**"operator\_type":"delete",**

**"plate ":"京A12345"**

**}**

**结果返回**

通过匹配车牌号删除白名单，如果车牌号为空，则清空整个白名单，否则只删除当前匹配成功的记录。

白名单删除返回结果：

**{**

**"cmd":"white\_list\_operator",**

**"id" : "999999",**

**"operator\_type":"delete",**

**"state":"succeed",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| operator\_type | 子命令   * update\_or\_add 增加或更新白名单 * delete 删除白名单 * select 白名单查询 |
| state | 返回结果状态 |
| plate | 车牌号 |

## 识别结果加密

从安全角度考虑或者不希望第三方获取到一些信息，可调用如下接口对识别结果加密。

当前加密功能一共提供了四个接口get\_ems、reset\_encrypt\_key、change\_encrypt\_key、enable\_encrypt。用户如果使用这些接口传输自己的明文密码的话，将会非常的不安全，所有的信息都是以明文的形式进行传输。恶意用户很容易通过抓包就能够得到设备的加密密码甚至主密码，所以对这几个接口传输的密码信息进行了加密处理。

接口中部分字段的解释：

active\_id：这一个字段指明当前选择的加密字段为多少。其中如果为0则为不加密，如果为其它的则为加密

signature：是一个十六位的字符串，全部的值以`0 - 9 A - Z a -z`字符构成。用户在未来的设置密码的时候，应该将这个字符串作为“消息”并结合秘钥（encrpty\_key或者prime\_key）使用`HMAC-SHA1`算法进行加密，再进行`Base64`编码。然后传输给设备，设备才能够开通或者关闭加密或者修改密码或者重置密码。调用加密功能的这四个接口都会得到一个signature，且每一次用户得到的值都会不一样。signature一旦更新了之后，用户必须确保使用新的signature，才能够进行接下来的认证。

**注意：**用户密码长度应该限制在[4,16]位之前，不然会失败

可借助如下网址中的加密工具来验证设备的加密功能相关接口的正确性

HMAC-SHA算法加密：<http://1024tools.com/hmac>

Base64编码：<http://tomeko.net/online_tools/hex_to_base64.php?lang=en>

AES 128位算法：<http://aes.online-domain-tools.com/>

识别结果加密一般流程：

1. 获取设备支持的加密方式：get\_ems
2. 用户可修改密码：对出厂时的默认用户密码进行修改或者对已经修改过的用户密码再次修改：change\_encrypt\_key
3. 用户可重置密码：用户在忘记用户密码时也可根据主密钥重新设置用户密码：主密钥需要向我们销售要：reset\_encrypt\_key
4. 根据用户密码开启对识别结果的加密：enable\_encrypt

使用主密钥+用户密码加密的原因：如果只使用一个密码（用户密码），当用户忘记这个密码时，需要重置成默认密码，假如默认密码是“000000”，那么代表着所有设备重置一下，密码都变为“000000”。大家都知道这个情况后，设备也将不再安全。例如A用户的设备，为了防止B用户获取到A用户设备上的一些信息，A用户对他的设备返回结果加了密，B用户在不知道A用户密码的情况下，无法解密加密后的文件，就无法获取到有用的信息。但是当B知道了设备重置一下，密码就会变成“000000”，那么B用户可向A用户的设备发送重置密码命令，然后再修改成自己的密码，就可用自己的密码解密加密后的文件，就获取到了A用户设备上的有用信息。主密钥由我们统一管理，且每台设备的主密钥唯一

### 获取加密方式：get\_ems

**请求命令参数**

**{**

**"cmd" : "get\_ems",**

**"id" : "999999"**

**}**

**结果返回**

**{**

**"active\_id" : 1,**

**"cmd" : "get\_ems",**

**"ems" : [**

**{**

**"m\_id" : 0,**

**"name" : "none encrypt"**

**},**

**{**

**"m\_id" : 1,**

**"name" : "aes-128-ecb-padding-zeros"**

**}**

**],**

**"signature" : "BNjnUHLkgPaLH3sI",**

**"id" : "999999",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| active\_id | 当前加密方式的索引 |
| ems | 加密方式列表   * **m\_id**加密方式的索引 * **name**加密方式的名字 |
| signature | 一个十六位的字符串（用于对传输的密码进行加密，每调用一次get\_ems/reset\_encrypt\_key/change\_encrypt\_key/enable\_encrypt四个接口中的任意一个接口, signature都会变，重置/修改密码和开启加密需采用最新的signature） |
| state\_code: | 状态码   * 200：成功 |

### 获取用户密码：get\_encrypt\_key

用户根据主密钥获取用户密码；

**请求命令参数**

**{**

**"cmd" : "get\_encrypt\_key",**

**"id" : "999999",**

**"prime\_key" : "Vg0MgpeBCOzzCxbsQ68V2NHd0Zk="**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| prime\_key | 主密钥字串（加密之后的密码）   * prime\_key的加密方法：由HMAC-SHA算法再加上原始的prime\_key，对signature进行加密生成数据，再进行Base64生成的字符串。Signature可通过get\_ems/reset\_encrypt\_key/change\_encrypt\_key/enable\_encrypt四个接口获取，需采用最新的signature |

注意：prime\_key的加密与new\_encrypt\_key的加密可借助上面提到的网址得到结果

**结果返回**

**{**

**"cmd" : "reset\_encrypt\_key",**

**"id" : "999999",**

**"signature" : "Rr17TBjPHNYXJR80",**

**"encrypt\_key" : "OB46x4SOg0ZTXa9BDFGh/i5LAfQ=",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| signature | 一个十六位的字符串（用于对传输的密码进行加密，每调用一次get\_ems/reset\_encrypt\_key/change\_encrypt\_key/enable\_encrypt四个接口中的任意一个接口, signature都会变，重置/修改密码和开启加密需采用最新的signature） |
| state\_code: | 状态码   * 200：成功 * 401：用户没有权限 * 400：用户参数设置错误（有可能密码太长或者太短） * 500：服务器内部错误（内部设置错误） |
| **encrypt\_key** | **encrypt\_key**的加密方法：使用AES 128位算法，使用秘钥prime\_key（原始主密钥），对密码进行加密生成数据，再进行Base64生成的字符串 |

### 重新设置用户密码：reset\_encrypt\_key

用户根据主密钥重新设置用户密码：用户在忘记自己设置的密码时可根据主密钥重新设置自己的密码

**请求命令参数**

**{**

**"cmd" : "reset\_encrypt\_key",**

**"id" : "999999",**

**"new\_encrypt\_key" : "j6NrhO8gWFKU0O8mk8+A/g==",**

**"prime\_key" : "Vg0MgpeBCOzzCxbsQ68V2NHd0Zk="**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| prime\_key | 主密钥字串（加密之后的密码）   * prime\_key的加密方法：由HMAC-SHA算法再加上原始的prime\_key（主密码），对signature进行加密生成数据，再进行Base64生成的字符串。Signature可通过get\_ems/reset\_encrypt\_key/change\_encrypt\_key/enable\_encrypt四个接口获取，需采用最新的signature |
| new\_encrypt\_key | 新设置的用户密码（加密之后的密码）   * new\_encrypt\_key的加密方法：使用AES 128位算法，使用密码prime\_key（原始主密钥），对新的密码进行加密生成数据，再进行Base64生成的字符串 |

注意：prime\_key的加密与new\_encrypt\_key的加密可借助上面提到的网址得到结果

**结果返回**

**{**

**"cmd" : "reset\_encrypt\_key",**

**"id" : "999999",**

**"signature" : "Rr17TBjPHNYXJR80",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| signature | 一个十六位的字符串（用于对传输的密码进行加密，每调用一次get\_ems/reset\_encrypt\_key/change\_encrypt\_key/enable\_encrypt四个接口中的任意一个接口, signature都会变，重置/修改密码和开启加密需采用最新的signature） |
| state\_code: | 状态码   * 200：成功 * 401：用户没有权限 * 400：用户参数设置错误（有可能密码太长或者太短） * 500：服务器内部错误（内部设置错误） |

### 修改用户密码：change\_encrypt\_key

将旧的用户密码修改成新的用户密码或者将默认的用户密码修改成新的用户密码

**请求命令参数**

**{**

**"cmd" : "change\_encrypt\_key",**

**"id" : "999999",**

**"encrypt\_key" : "OB46x4SOg0ZTXa9BDFGh/i5LAfQ=",**

**"new\_encrypt\_key" : "sZuBTdl+Fy0HjgFJ/yDmSA=="**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| encrypt\_key | 旧的用户密码（加密后的）   * encrypt \_key的加密方法：使用HMAC-SHA1算法对原始用户密码加密(encrypt \_key)，加密物料使用signature，再进行Base64生成字符串。 * Signature可通过get\_ems/reset\_encrypt\_key/change\_encrypt\_key/enable\_encrypt四个接口获取，需采用最新的signature |
| new\_encrypt\_key | 新设置的用户密码（加密之后的密码）   * new\_encrypt\_key的加密方法：使用AES 128位算法，使用原本的encrypt \_key（原始用户密码），对新的密码进行加密生成数据，再进行Base64生成的字符串 |

注意：encrypt\_key的加密与new\_encrypt\_key的加密可借助上面提到的网址得到结果

**结果返回**

**{**

**"cmd" : "change\_encrypt\_key",**

**"signature" : "8GZiU0f8u2slTCsp",**

**"id" : "999999",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| signature | 一个十六位的字符串（用于对传输的密码进行加密，每调用一次get\_ems/reset\_encrypt\_key/change\_encrypt\_key/enable\_encrypt四个接口中的任意一个接口, signature都会变，重置/修改密码和开启加密需采用最新的signature） |
| state\_code: | 状态码   * 200：成功 * 401：用户没有权限 * 400：用户参数设置错误（有可能密码太长或者太短） * 500：服务器内部错误（内部设置错误） |

如

**std::string old\_key;**

**std::string encrypt\_key;**

**unsigned char digest\_buffer[20] = {0};**

**HmacSha1((const unsigned char \*) old\_key.c\_str(),old\_key.size(),**

**(const unsigned char \*) signature.c\_str(),signature.size(), digest\_buffer);**

**Base64::EncodeFromArray(digest\_buffer, 20, &encrypt\_key);**

**std::string new\_key;**

**std::string new\_encrypt\_key;**

**AES128\_ECB\_encrypt((const unsigned char \*) old\_key.c\_str(),**

**(const unsigned char \*) new\_key.c\_str(), new\_encrypt\_key);**

**Base64::EncodeFromArray(new\_encrypt\_key.c\_str(),new\_encrypt\_key.size(), & new\_encrypt\_key);**

### 开启是否加密：enable\_encrypt

**请求命令参数**

注意：m\_id : 0 为不加密，encrypt\_key需传入用户密码（加密后的），而非主密钥

**{**

**"cmd" : "enable\_encrypt",**

**"encrypt\_key" : "9k1t5PhnUXlfgJKgHmcswU3PIp8=",**

**"id" : "999999",**

**"m\_id" : 1**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| encrypt\_key | 用户密码（加密后的）   * encrypt \_key的加密方法：使用HMAC-SHA1算法对原始用户密码加密(encrypt \_key)，加密物料使用signature，再进行Base64生成字符串。 * 。Signature可通过get\_ems/reset\_encrypt\_key/change\_encrypt\_key/enable\_encrypt四个接口获取，需采用最新的signature |
| m\_id | 加密方式的索引（m\_id : 0 为不加密） |

注意：encrypt\_key的加密可借助上面提到的网址得到结果

**结果返回**

**{**

**"cmd" : "enable\_encrypt",**

**"signature" : "BUqCmKfiKt7dYEOv",**

**"id" : "999999",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| signature | 一个十六位的字符串（用于对传输的密码进行加密，每调用一次get\_ems/reset\_encrypt\_key/change\_encrypt\_key/enable\_encrypt四个接口中的任意一个接口, signature都会变，重置/修改密码和开启加密需采用最新的signature） |
| State\_code: | 状态码   * 200：成功 * 401：用户没有权限 * 400：用户参数设置错误（m\_id错误） * 500：服务器内部错误（内部设置错误） |

.

### 设置设备有效时间：device\_active\_settings

设置启用授权后，超过相机授权时间后，相机不再推送识别结果；

**请求命令参数**

**{**

**"cmd": "device\_active\_settings",**

**"id" : "123",**

**"body":{**

**"type":"set\_device\_active\_status",**

**"active\_status":1,**

**"active\_time":60,**

**"authentication":"asdgzxb"**

**}**

**}**

**结果返回：**

**{**

**"body" : {**

**"state" : 200,**

**"type" : "set\_device\_active\_status"**

**},**

**"cmd" : "device\_active\_settings",**

**"id" : "123",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **active\_status** | 是否启用相机授权  0 不启用  1 启用  启用授权后，在超过授权时间后，相机不再推送识别结果信息； |
| **active\_time** | 相机授权时间，单位秒； |
| **authentication** | 加密后再经过64位编码的用户密码  使用SHA1算法对用户密码加密，加密物料使用签名，再进行64位编码后的字符串； |

### 获取设备有效时间：device\_active\_settings

**请求命令参数**

**{**

**"cmd": "device\_active\_settings",**

**"id" : "123",**

**"body":{**

**"type":"get\_device\_active\_status"**

**}**

**}**

**结果返回：**

**{**

**"body" : {**

**"active\_status" : 1,**

**"active\_time" : 1200,**

**"signature" : "cR6dxChrE42JufOQ",**

**"state" : 200,**

**"type" : "get\_device\_active\_status"**

**},**

**"cmd" : "device\_active\_settings",**

**"id" : "123",**

**"state\_code" : 200**

**}**

### 设置用户私有数据：set\_user\_data

**请求命令参数**

**{**

**"cmd": "set\_user\_data",**

**"id" : "123",**

**"body":{**

**"data":"MTMyNDU2Nzg5**

**"**

**}**

**}**

**结果返回：**

**{**

**"cmd" : "set\_user\_data",**

**"id" : "123",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **data** | 需要设置的用户数据，经过base64位编码 |

### 获取用户私有数据：get\_user\_data

**请求命令参数**

**{**

**"cmd" : "get\_user\_data",**

**"id":"123"**

**}**

**结果返回：**

**{**

**"body" : {**

**"data" : "MTMyNDU2Nzg5"**

**},**

**"cmd" : "get\_user\_data",**

**"id" : "123",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **data** | 需要设置的用户数据，经过base64位编码 |

## 设备组网：dg\_json\_request

设备组网不同于传统的将一台设备的信息提供给用户，需是自动的将一定区域内的所有设备信息提供给用户，以便应用层用户能够更加方便的接入我们的设备系统，更加快速的开发自己的应用。

设备组网功能主要应用在停车场环境，一个停车场里面的所有设备之间相互连接组成一个网络。当有一辆车进入或者离开整个停车场的时候，设备之间进行信息的共享和同步，并且根据这些数据分析出一个最优的结果，然后传输给用户，以方便应用开发，甚至直接收费。

设备组网协议主要分为两个大部分。第一步是关于识别结果的协议。第二部分是关于各种操作组网的协议。

**请求命令参数如下格式：**

**{**

**"cmd": "dg\_json\_request",**

**"id": "132156",**

**"body": {**

**"type": "%s"**

**...//此处为组网请求细分消息中携带的参数，具体如下**

**}**

**}**

**结果返回**

0. 处理成功：

**{**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"state\_code" : 200,**

**"body" : {**

**"state" : 200,**

**"type" : "get\_cdvzid",**

**……// 具体请求具体回复**

**}**

**}**

1. 格式错误：

**{**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"state\_code" : 400**

**}**

2. 服务器处理超时错误：

**{**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"state\_code" : 408**

**}**

3. 服务器内部错误：

**{**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"state\_code" : 500**

**}**

### 得到当前设备vzid：get\_cdvzid

我们当前将每一个设备的名称定义为vzid ，用来标记这一台设备

**请求命令参数**

**{**

**"cmd": "dg\_json\_request",**

**"id": "132156",**

**"body": {**

**"type": "get\_cdvzid"**

**}**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30 |
| body | 需要传递给组网模块的数据 |
| type | 操作组网模块的命令   * get\_cdvzid 获取当前设备vzid |

**结果返回**

**{**

**"body" : {**

**"state" : 200,**

**"type" : "get\_cdvzid",**

**"vzid" : {**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.7.22",**

**"name" : "bHk=&MTkyLjE2OC43LjIy#",**

**"sn" : "56a8872e-2c5f7863",**

**"type" : "input"**

**}**

**},**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30，等于请求时的id |
| body | 组网模块处理后返回的数据 |
| state\_code | 状态码 |

Body内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| type | 操作组网模块的命令  get\_cdvzid 获取当前设备vzid |
| vzid | 设备的名称，vzid说明如下 |
| state | 状态码 |

vzid字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| enable\_group | 开启组网功能   * true 开启 * false 关闭 |
| ip\_addr | 设备ip |
| name | 设备名字 |
| sn | 设备序列号 |
| type | 设备类型   * input 入口设备 * output 出口设备 |

### 得到当前设备名称：get\_current\_device\_name

**请求命令参数**

**{**

**"cmd": "dg\_json\_request",**

**"id": "132156",**

**"body": {**

**"type": "get\_current\_device\_name"**

**}**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30 |
| body | 需要传递给组网模块的数据 |

Body内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| type | 操作组网模块的命令   * **get\_current\_device\_name**得到当前设备名称 |

**结果返回**

返回结果为当前设备vzid的信息，和2.8.1返回结果一致，请参考；

### 得到在线设备信息，不含自己：get\_ovzid

**请求命令参数**

**{**

**"cmd": "dg\_json\_request",**

**"id": "132156",**

**"body": {**

**"type": "get\_ovzid"**

**}**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30 |
| body | 需要传递给组网模块的数据 |

Body内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| type | 操作组网模块的命令   * get\_ovzid 得到所有已连接的设备信息 |

**结果返回**

**{**

**"body": {**

**"state": 200,**

**"type": "get\_ovzid",**

**"vzids": [**

**{**

**"enable\_group": true,**

**"ip\_addr": "192.168.2.29",**

**"name": "2.29 TEST",**

**"sn": "0f692718-95bc0185",**

**"type": "output"**

**},**

**{**

**"enable\_group": true,**

**"ip\_addr": "192.168.2.77",**

**"name": "2.77 TEST",**

**"sn": "ef69b450-21bf3bff",**

**"type": "input"**

**},**

**{**

**"enable\_group": true,**

**"ip\_addr": "192.168.2.79",**

**"name": "2.79 TEST",**

**"sn": "c9455af2-68daf655",**

**"type": "output"**

**},**

**{**

**"enable\_group": true,**

**"ip\_addr": "192.168.2.9",**

**"name": "2.9 TEST",**

**"sn": "037de919-76c16e36",**

**"type": "output"**

**}**

**]**

**},**

**"cmd": "dg\_json\_request",**

**"id": "132156",**

**"state\_code": 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30，等于请求时的id |
| body | 组网模块处理后返回的数据 |
| state\_code | 状态码 |

Body内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| type | 操作组网模块的命令   * get\_ovzid得到所有已连接的设备信息 |
| vzid | 设备信息列表（由多条设备信息组成），一条设备信息，即vzid，vzid上面已经提到过了 |

### 得到在线设备信息，含自己：online\_devices

**请求命令参数**

**{**

**"cmd": "dg\_json\_request",**

**"id": "132156",**

**"body": {**

**"type": "online\_devices"**

**}**

**}**

|  |  |
| --- | --- |
| 字段名 | 说明 |
| type | 操作组网模块的命令   * **online\_devices** 得到所有已连接的设备信息 |

**结果返回：**

**{**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"state\_code" : 200,**

**"body" :**

**{**

**"state" : 200,**

**“type" : "online\_devices",**

**"online\_devices" : [**

**{**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.7.22",**

**"name" : "bHk=&MTkyLjE2OC43LjIy#",**

**"sn" : "56a8872e-2c5f7863",**

**"type" : "input"**

**},**

**{**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.7.22",**

**"name" : "bHk=&MTkyLjE2OC43LjIy#",**

**"sn" : "56a8872e-2c5f7863",**

**"type" : "input"**

**}**

**]**

**}**

**}**

|  |  |
| --- | --- |
| 字段名 | 说明 |
| type | 操作组网模块的命令   * **online\_devices** 得到所有已连接的设备信息 * **online\_devices中的数据结构与上面vzids的数据一致，请参考** |

### 得到所有连接设备信息：get\_avzid

**请求命令参数**

**{**

**"cmd": "dg\_json\_request",**

**"id": "132156",**

**"body": {**

**"type": "get\_avzid"**

**}**

**}**

**结果返回：**

结果返回以及字段信息和上面2.8.3一致，请参考；

### 得到当前组网内所有设备信息：get\_agdi

**请求命令参数**

**{**

**"cmd": "dg\_json\_request",**

**"id": "132156",**

**"body": {**

**"type": "get\_agdi"**

**}**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| type | 操作组网模块的命令   * **get\_agdi**得到当前组网内所有设备信息 |

**结果返回：**

结果返回以及字段信息和上面2.8.3一致，请参考；

### 得到当前设备记录size：current\_records\_size

**请求命令参数**

**{**

**"cmd": "dg\_json\_request",**

**"id": "132156",**

**"body": {**

**"type": "current\_records\_size"**

**}**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| type | 操作组网模块的命令   * **current\_records\_size**得到当前设备记录的size |

**结果返回：**

* + - 1. 入口设备：

**{**

**“body” : {**

**“state” : 200,**

**“type” : “current\_records\_size”,**

**“input\_records\_size”:1024,**

**“vzid” : {**

**“enable\_group” : true,**

**“ip\_addr” : “192.168.7.22”,**

**“name” : “bHk=&MTkyLjE2OC43LjIy#”,**

**“sn” : “56a8872e-2c5f7863”,**

**“type” : “input”**

**}**

**},**

**“cmd” : “dg\_json\_request”,**

**“id” : “132156”,**

**“state\_code” : 200**

**}**

* + - 1. 出口设备：

**{**

**“body” : {**

**“state” : 200,**

**“type” : “current\_records\_size”,**

**“output\_records\_size”:1024,**

**“vzid” : {**

**“enable\_group” : true,**

**“ip\_addr” : “192.168.7.22”,**

**“name” : “bHk=&MtkyLjE2OC43LjIy#”,**

**“sn” : “56a8872e-2c5f7863”,**

**“type” : “output”**

**}**

**},**

**“cmd” : “dg\_json\_request”,**

**“id” : “132156”,**

**“state\_code” : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **output\_records\_size** | * 出口设备记录的size |
| **input\_records\_size** | * 入口设备记录的size |
| **Vzid** | * 同时返回当前设备vzid信息；vzid信息参考前面； |

### 得到入口设备记录：records\_sparate\_input

由于设备记录有很多，一般情况下很难一次性传输出来，所以，在得到设备记录的时候必须指定一个范围。在最开始的时候如果不知道记录有

多少条，可以随便指定一个可以传输的范围，32最好。然后在回复的消息中，会告诉当前设备有多少条记录。这样的分页机制，让用户可以更好的开发自己的应用

**请求命令参数**

**{**

**"body" : {**

**"input\_begin\_pos" : 0,**

**"input\_end\_pos" : 8,**

**"type" : "records\_sparate\_input"**

**},**

**"cmd" : "dg\_json\_request",**

**"id" : "123456789"**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30 |
| body | 需要传递给组网模块的数据 |
| type | 请求回应消息 |

Body内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| type | 操作组网模块的命令   * **records\_sparate\_input**获取入口设备记录 |
| input\_begin\_pos | 获取记录的开始位置 |
| input\_end\_pos | 获取记录的结束位置 |

**结果返回**

**{**

**"body" : {**

**"input\_begin\_pos" : 0,**

**"input\_end\_pos" : 8,**

**"input\_records\_size" : 10213,**

**"records" : [**

**{**

**"enter\_ip" : "192.168.5.13",**

**"enter\_name" : "6L2m5bqTMzA=&55u46L6FNg==&MTkyLjE2OC41LjEz#dHJlZURlbW9fNQ==",**

**"enter\_time" : 1494049723,**

**"plate" : "渝AX4462",**

**"state" : 5**

**},**

**{**

**"enter\_ip" : "192.168.5.13",**

**"enter\_name" : "6L2m5bqTMzA=&55u46L6FNg==&MTkyLjE2OC41LjEz#dHJlZURlbW9fNQ==",**

**"enter\_time" : 1494049731,**

**"plate" : "琼ECVWJB",**

**"state" : 6**

**},**

**{**

**"enter\_ip" : "192.168.5.13",**

**"enter\_name" : "6L2m5bqTMzA=&55u46L6FNg==&MTkyLjE2OC41LjEz#dHJlZURlbW9fNQ==",**

**"enter\_time" : 1494049739,**

**"plate" : "豫JDZBHF",**

**"state" : 6**

**},**

**{**

**"enter\_ip" : "192.168.5.13",**

**"enter\_name" : "6L2m5bqTMzA=&55u46L6FNg==&MTkyLjE2OC41LjEz#dHJlZURlbW9fNQ==",**

**"enter\_time" : 1494049754,**

**"plate" : "冀SCNGRR",**

**"state" : 6**

**},**

**{**

**"enter\_ip" : "192.168.5.13",**

**"enter\_name" : "6L2m5bqTMzA=&55u46L6FNg==&MTkyLjE2OC41LjEz#dHJlZURlbW9fNQ==",**

**"enter\_time" : 1494049762,**

**"plate" : "新GMKA3M",**

**"state" : 6**

**},**

**{**

**"enter\_ip" : "192.168.5.13",**

**"enter\_name" : "6L2m5bqTMzA=&55u46L6FNg==&MTkyLjE2OC41LjEz#dHJlZURlbW9fNQ==",**

**"enter\_time" : 1494049770,**

**"plate" : "赣QFWXF5",**

**"state" : 6**

**},**

**{**

**"enter\_ip" : "192.168.5.13",**

**"enter\_name" : "6L2m5bqTMzA=&55u46L6FNg==&MTkyLjE2OC41LjEz#dHJlZURlbW9fNQ==",**

**"enter\_time" : 1494049778,**

**"plate" : "豫X4JK1G",**

**"state" : 6**

**},**

**{**

**"enter\_ip" : "192.168.5.13",**

**"enter\_name" : "6L2m5bqTMzA=&55u46L6FNg==&MTkyLjE2OC41LjEz#dHJlZURlbW9fNQ==",**

**"enter\_time" : 1494049784,**

**"plate" : "豫QWY8KH",**

**"state" : 6**

**},**

**{**

**"enter\_ip" : "192.168.5.13",**

**"enter\_name" : "6L2m5bqTMzA=&55u46L6FNg==&MTkyLjE2OC41LjEz#dHJlZURlbW9fNQ==",**

**"enter\_time" : 1494049793,**

**"plate" : "沪H1JLJ0",**

**"state" : 6**

**}**

**],**

**"state" : 200,**

**"type" : "records\_sparate\_input",**

**"vzid" : {**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.7.22",**

**"name" : "bHk=&MTkyLjE2OC43LjIy#",**

**"sn" : "56a8872e-2c5f7863",**

**"type" : "input"**

**}**

**},**

**"cmd" : "dg\_json\_request",**

**"id" : "123456789",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30，等于请求时的id |
| body | 组网模块处理后返回的数据 |
| state\_code | 状态码 |

Body内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| input\_begin\_pos | 获取记录的开始位置 |
| input\_end\_pos | 获取记录的结束位置 |
| input\_records\_size | 入口设备记录总条数 |
| records | 入口设备记录的列表（由多条入口记录组成） |
| state | 状态码 |
| type | 操作组网模块的命令   * records\_sparate\_input 获取入口设备记录 |
| vzid | 设备的名称，vzid说明如下 |

records内的字段说明:

|  |  |
| --- | --- |
| **字段名** | **说明** |
| enter\_ip | ip地址 |
| enter\_name | 设备名称 |
| enter\_time | 识别时间点 |
| plate | 车牌号码（汉字为GB2312编码） |
| state | 设备的名称，vzid说明如下 |

vzid字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| enable\_group | 开启组网功能   * true 开启 * false 关闭 |
| ip\_addr | 设备ip |
| name | 设备名字 |
| sn | 设备序列号 |
| type | 设备类型   * input 入口设备 * output 出口设备 |

### 得到出口设备记录：records\_sparate\_output

**请求命令参数**

**{**

**"body" : {**

**"output\_records\_begin\_pos" : 0,**

**"output\_records\_end\_pos" : 8,**

**"type" : "records\_sparate\_output"**

**},**

**"cmd" : "dg\_json\_request",**

**"id" : "123456789"**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30 |
| body | 需要传递给组网模块的数据 |

Body内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| type | 操作组网模块的命令   * **records\_sparate\_outpu**获取出口设备记录 |
| output\_records\_begin\_pos | 获取记录的开始位置 |
| output\_records\_end\_pos | 获取记录的结束位置 |

**结果返回**

**{**

**"body" : {**

**"output\_records\_begin\_pos" : 0,**

**"output\_records\_end\_pos" : 8,**

**"output\_records\_size" : 4549,**

**"records" : [**

**{**

**"enter\_ip" : "192.168.4.79",**

**"enter\_name" : "5ryr5q2l5LqR56uv&MTkyLjE2OC40Ljc5#",**

**"enter\_time" : 1465496977,**

**"leave\_ip" : "192.168.7.22",**

**"leave\_name" : "bHk=&MTkyLjE2OC43LjIy#",**

**"leave\_time" : 1465496992,**

**"plate" : "粤Z72D2港",**

**"state" : 0**

**},**

**{**

**"enter\_ip" : "192.168.4.97",**

**"enter\_name" : "5ryr5q2l5LqR56uv&MTkyLjE2OC40Ljk3#",**

**"enter\_time" : 1465497011,**

**"leave\_ip" : "192.168.7.22",**

**"leave\_name" : "bHk=&MTkyLjE2OC43LjIy#",**

**"leave\_time" : 1465497011,**

**"plate" : "粤Z72D2港",**

**"state" : 3**

**},**

**{**

**"enter\_ip" : "192.168.4.97",**

**"enter\_name" : "5ryr5q2l5LqR56uv&MTkyLjE2OC40Ljk3#",**

**"enter\_time" : 1465497051,**

**"leave\_ip" : "192.168.7.22",**

**"leave\_name" : "bHk=&MTkyLjE2OC43LjIy#",**

**"leave\_time" : 1465497051,**

**"plate" : "粤Z72D2港",**

**"state" : 3**

**},**

**{**

**"enter\_ip" : "192.168.4.97",**

**"enter\_name" : "5ryr5q2l5LqR56uv&MTkyLjE2OC40Ljk3#",**

**"enter\_time" : 1468906933,**

**"leave\_ip" : "192.168.7.22",**

**"leave\_name" : "bHk=&MTkyLjE2OC43LjIy#",**

**"leave\_time" : 1468906933,**

**"plate" : "粤Z72D2港",**

**"state" : 3**

**},**

**{**

**"enter\_ip" : "192.168.4.97",**

**"enter\_name" : "5ryr5q2l5LqR56uv&MTkyLjE2OC40Ljk3#",**

**"enter\_time" : 1468906941,**

**"leave\_ip" : "192.168.7.22",**

**"leave\_name" : "bHk=&MTkyLjE2OC43LjIy#",**

**"leave\_time" : 1468906941,**

**"plate" : "粤Z72D2港",**

**"state" : 3**

**},**

**{**

**"enter\_ip" : "192.168.4.79",**

**"enter\_name" : "5ryr5q2l5LqR56uv&MTkyLjE2OC40Ljc5#",**

**"enter\_time" : 1468906962,**

**"leave\_ip" : "192.168.7.22",**

**"leave\_name" : "bHk=&MTkyLjE2OC43LjIy#",**

**"leave\_time" : 1468906967,**

**"plate" : "粤Z72D2港",**

**"state" : 0**

**},**

**{**

**"enter\_ip" : "192.168.4.97",**

**"enter\_name" : "5ryr5q2l5LqR56uv&MTkyLjE2OC40Ljk3#",**

**"enter\_time" : 1468907000,**

**"leave\_ip" : "192.168.7.22",**

**"leave\_name" : "bHk=&MTkyLjE2OC43LjIy#",**

**"leave\_time" : 1468907000,**

**"plate" : "粤Z72D2港",**

**"state" : 3**

**},**

**{**

**"enter\_ip" : "192.168.4.79",**

**"enter\_name" : "5ryr5q2l5LqR56uv&MTkyLjE2OC40Ljc5#",**

**"enter\_time" : 1468907153,**

**"leave\_ip" : "192.168.7.22",**

**"leave\_name" : "bHk=&MTkyLjE2OC43LjIy#",**

**"leave\_time" : 1468907367,**

**"plate" : "粤Z72D2港",**

**"state" : 0**

**},**

**{**

**"enter\_ip" : "192.168.4.97",**

**"enter\_name" : "5ryr5q2l5LqR56uv&MTkyLjE2OC40Ljk3#",**

**"enter\_time" : 1468907448,**

**"leave\_ip" : "192.168.7.22",**

**"leave\_name" : "bHk=&MTkyLjE2OC43LjIy#",**

**"leave\_time" : 1468907448,**

**"plate" : "粤Z0V89港",**

**"state" : 3**

**}**

**],**

**"state" : 200,**

**"type" : "records\_sparate\_output",**

**"vzid" : {**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.7.22",**

**"name" : "bHk=&MTkyLjE2OC43LjIy#",**

**"sn" : "56a8872e-2c5f7863",**

**"type" : "output"**

**}**

**},**

**"cmd" : "dg\_json\_request",**

**"id" : "123456789",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30，等于请求时的id |
| body | 组网模块处理后返回的数据 |
| state\_code | 状态码 |

Body内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| output\_records\_begin\_pos | 获取记录的开始位置 |
| output\_records\_end\_pos | 获取记录的结束位置 |
| output\_records\_size | 出口设备记录总条数 |
| records | 出口设备记录的列表（由多条出口记录组成，一条出口记录由此车牌的入口记录信息+此车牌的出口记录信息组成） |
| state | 状态码 |
| type | 操作组网模块的命令   * **records\_sparate\_output** 获取出口设备记录 |
| vzid | 设备的名称 |

vzid字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| enable\_group | 开启组网功能   * true 开启 * false 关闭 |
| ip\_addr | 设备ip |
| name | 设备名字 |
| sn | 设备序列号 |
| type | 设备类型   * input 入口设备 * output 出口设备 |

### 使能设备组网：enable\_devicegroup

**请求命令参数**

**{**

**"cmd": "dg\_json\_request",**

**"id": "132156",**

**"body": {**

**"type": "enable\_devicegroup",**

**"vzid" : {**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.7.22",**

**"name" : "bHk=&MTkyLjE2OC43LjIy#",**

**"sn" : "56a8872e-2c5f7863",**

**"type" : "input"**

**}**

**}**

**}**

Body内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| type | 操作组网模块的命令   * **enable\_devicegroup 使能设备组网** |
| **vzid** | 需要使能的设备的vzid |
|  | Vzid中的字段意思，参考2.8.1中关于vzid的描述 |

**结果返回**

**{**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"state\_code" : 200**

**"body" :**

**{**

**"state" : 200,**

**"type" : "enable\_devicegroup"**

**}**

**}**

### 设置设备组网类型及参数：set\_device\_type\_enable

**功能和2.8.10一致**

**请求命令参数**

**{**

**"cmd": "dg\_json\_request",**

**"id": "132156",**

**"body": {**

**"type": "set\_device\_type\_enable",**

**"vzid" : {**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.7.22",**

**"name" : "bHk=&MTkyLjE2OC43LjIy#",**

**"sn" : "56a8872e-2c5f7863",**

**"type" : "input"**

**}**

**}**

**}**

Body内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| type | 操作组网模块的命令   * set\_device\_type\_enable **设置设备组网** |
| **vzid** | 需要使能的设备的vzid |
|  | Vzid中的字段意思，参考2.8.1中关于vzid的描述 |

**结果返回**

**{**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"state\_code" : 200**

**"body" :**

**{**

**"state" : 200,**

**"type" : "set\_device\_type\_enable"**

**}**

**}**

### 查找车牌信息：search\_plate

**请求命令参数**

**{**

**"cmd": "dg\_json\_request",**

**"id": "132156",**

**"body": {**

**"type": "search\_plate",**

**"plate":"川A12345",**

**"record\_type": "input",**

**"record\_state": 5**

**}**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| type | 操作组网模块的命令   * **search\_plate 查找车牌信息** |
| **Plate** | 车牌号 |
| **record\_type** | 记录类型  :入场设备;   * input ：入场 * output:出场   select 白名单查询 |
| **record\_state** | 记录状态 |

**结果返回**

**{**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"state\_code" : 200,**

**"body" :**

**{**

**"state" : 200,**

**"type" : "search\_plate",**

**"search\_result":**

**{**

**"record\_type":"output",**

**"output\_record":**

**{**

**"ivs\_result\_param":**

**{**

**"bright" : 0,**

**"carBright" : 0,**

**"carColor" : 0,**

**"colorType" : 1,**

**"colorValue" : 0,**

**"confidence" : 99,**

**"direction" : 0,**

**"license" : "青PTW3Z3",**

**"location" : {**

**"RECT" : {**

**"bottom" : 392,**

**"left" : 690,**

**"right" : 834,**

**"top" : 350**

**}**

**},**

**"timeStamp" : {**

**"Timeval" : {**

**"sec" : 1458882234,**

**"usec" : 921325**

**}**

**},**

**"timeUsed" : 0,**

**"triggerType" : 1,**

**"type" : 1**

**},**

**"active\_id" : 0,**

**"clipImgSize" : 1103,**

**"cmd" : "ivs\_result",**

**"fullImgSize" : 51566,**

**"id" : 0,**

**"imageformat" : "jpg",**

**"timeString" : "2016-03-25 13:03:54"**

**},**

**"device\_name":"asgadg",**

**"state":2**

**},**

**"output\_record":**

**{**

**"ivs\_result\_param":**

**{**

**"bright" : 0,**

**"carBright" : 0,**

**"carColor" : 0,**

**"colorType" : 1,**

**"colorValue" : 0,**

**"confidence" : 99,**

**"direction" : 0,**

**"license" : "青PTW3Z3",**

**"location" : {**

**"RECT" : {**

**"bottom" : 392,**

**"left" : 690,**

**"right" : 834,**

**"top" : 350**

**}**

**},**

**"timeStamp" : {**

**"Timeval" : {**

**"sec" : 1458882234,**

**"usec" : 921325**

**}**

**},**

**"timeUsed" : 0,**

**"triggerType" : 1,**

**"type" : 1**

**},**

**"active\_id" : 0,**

**"clipImgSize" : 1103,**

**"cmd" : "ivs\_result",**

**"fullImgSize" : 51566,**

**"id" : 0,**

**"imageformat" : "jpg",**

**"timeString" : "2016-03-25 13:03:54"**

**},**

**"device\_name":"asgadg",**

**"state":2**

**}**

**},**

**{**

**"record\_type":"input",**

**"input\_record":**

**{**

**"ivs\_result\_param":**

**{**

**// 参考识别结果参数**

**},**

**"device\_name":"asgadg",**

**"state":2**

**}**

**}**

**}**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| type | 操作组网模块的命令   * **search\_plate 查找车牌信息** |

**search\_result中字段含义：**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **record\_type** | * 当前记录是出场还是入场记录 |
| **output\_record** | * 出场记录 |
| **input\_record** | * 入场记录 |
| **ivs\_result\_param** | * 车辆信息，结构体 |
| **State** | * 当前车辆状态 |
| **device\_name** | * 设备名称 |

**ivs\_result\_param中字段含义：**

**参考2.2.1中车牌识别结果中字段含义**

### 获取组网匹配模式：get\_device\_match\_mode

**请求命令参数**

**{**

**"cmd": "dg\_json\_request",**

**"id": "132156",**

**"body": {**

**"type": "get\_device\_match\_mode"**

**}**

**}**

**结果返回**

**{**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"state\_code" : 200**

**"body" :**

**{**

**"state" : 200,**

**"type" : " get\_device\_match\_mode ",**

**"mode":"** **fuzzy\_mode "**

**}**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **mode** | 组网匹配模式：  **fuzzy\_mode：模糊匹配**  exact\_mode：精确匹配 |

### 设置组网匹配模式: set \_device\_match\_mode

**请求命令参数**

**{**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"body" :**

**{**

**"type" : "set\_device\_match\_mode",**

**"mode":"fuzzy\_mode"**

**}**

**}**

**结果返回**

**{**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"state\_code" : 200**

**"body" :**

**{**

**"state" : 200,**

**"type" : "set\_device\_match\_mode"**

**}**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **mode** | 组网匹配模式：  **fuzzy\_mode：模糊匹配**  exact\_mode：精确匹配 |

### 得到组网共享IO：get\_group\_shared\_io

**请求命令参数**

**{**

**"cmd": "dg\_json\_request",**

**"id": "132156",**

**"body": {**

**"type": "get\_group\_shared\_io"**

**}**

**}**

**结果返回**

**{**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"state\_code" : 200**

**"body" :**

**{**

**"state" : 200,**

**"type" : "get\_group\_shared\_io",**

**"value": 1**

**}**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **value** | 当前组网共享的IO的值 |

### 设置组网共享IO：set\_group\_shared\_io

**请求命令参数**

**{**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"body" :**

**{**

**"type" : "set\_group\_shared\_io",**

**"value": 1**

**}**

**}**

**结果返回**

**{**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"state\_code" : 200**

**"body" :**

**{**

**"state" : 200,**

**"type" : "set\_group\_shared\_io"**

**}**

**}**

### 组网识别结果消息 enable\_dg\_result

识别结果消息分成了两种类型一种类型是入口设备的识别结果消息，一种是出口设备的识别结果消息。用户首先需要开启允许接收组网识别结果消息，未来，如果有车牌识别消息到来，用户才会接收到出/入口消息

**请求命令参数**

**{**

**"cmd":"enable\_dg\_result",**

**"enable":true,**

**"id":"123456"**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30 |
| enable | 是否允许接收组网识别消息   * true 允许接收组网识别消息 * false 不接收组网识别消息 |

**结果返回**

**{**

**"cmd" : "enable\_dg\_result",**

**"id" : "123456",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30，等于请求时的id |
| state\_code | 状态码 |

当开启接收组网识别结果后，当组网设备有识别到结果，会主动推送给用户，推送信息如下：

**出口消息**

**{**

**"body": {**

**"record\_type": "output",**

**"input\_record": {**

**"device\_name": {**

**"enable\_group": true,**

**"ip\_addr": "192.168.4.180",**

**"name": "4.180Leave",**

**"sn": "7667931c-7fa08a25",**

**"type": "output"**

**},**

**"ivs\_result\_param": {**

**"bright": 0,**

**"car\_bright": 0,**

**"car\_color": 0,**

**"confidence": 98,**

**"direction": 4,**

**"fragment\_path": "/tmp/app/html/snapshot/lpr/patch\_tri\_snap\_4.jpg",**

**"id": 0,**

**"image\_path": "/tmp/app/html/snapshot/lpr/tri\_snap\_4.jpg",**

**"image\_sd\_path": "/media/mmcblk0p1/VzIPCCap/2015\_11\_12/1534433501\_陕D4FY61.jpg",**

**"location": {**

**"bottom": 309,**

**"left": 965,**

**"right": 1150,**

**"top": 268**

**},**

**"n\_time": 0,**

**"plate": "陕D4FY61",**

**"plate\_color": 1,**

**"plate\_type" : 0,**

**"related\_plate" : "陕D4FY61",**

**"timeval" : {**

**"decday" : 5,**

**"dechour" : 15,**

**"decmin" : 26,**

**"decmon" : 5,**

**"decsec" : 14,**

**"decyear" : 2017,**

**"tv\_sec" : 1493969174,**

**"tv\_usec" : 416656**

**},**

**"trig\_type": 1**

**},**

**"state": 0**

**},**

**"output\_record": {**

**"device\_name": {**

**"enable\_group": true,**

**"ip\_addr": "192.168.4.180",**

**"name": "4.180Leave",**

**"sn": "7667931c-7fa08a25",**

**"type": "output"**

**},**

**"ivs\_result\_param": {**

**"bright": 0,**

**"car\_bright": 0,**

**"car\_color": 0,**

**"confidence": 98,**

**"direction": 4,**

**"fragment\_path": "/tmp/app/html/snapshot/lpr/patch\_tri\_snap\_4.jpg",**

**"id": 0,**

**"image\_path": "/tmp/app/html/snapshot/lpr/tri\_snap\_4.jpg",**

**"image\_sd\_path": "/media/mmcblk0p1/VzIPCCap/2015\_11\_12/1534433501\_陕D4FY61.jpg",**

**"location": {**

**"bottom": 309,**

**"left": 965,**

**"right": 1150,**

**"top": 268**

**},**

**"n\_time": 0,**

**"plate": "陕D4FY61",**

**"plate\_color": 1,**

**"plate\_type" : 0,**

**"related\_plate" : "陕D4FY61",**

**"timeval" : {**

**"decday" : 5,**

**"dechour" : 15,**

**"decmin" : 26,**

**"decmon" : 5,**

**"decsec" : 14,**

**"decyear" : 2017,**

**"tv\_sec" : 1493969174,**

**"tv\_usec" : 416656**

**},**

**"trig\_type": 1**

**},**

**"state": 0**

**}**

**},**

**"cmd": "dg\_plateinfo\_result",**

**"id": "132156",**

**"state\_code": 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30，等于请求时的id |
| state\_code | 状态码 |
| cmd | 操作命令   * dg\_plateinfo\_result组网识别消息结果 |

Body内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| record\_type | 消息类型   * output出口消息 * input 入口消息 |
| input\_record | 上面已经提到过了，请自行查找 |
| output\_record | 上面已经提到过了，请自行查找 |

**入口消息**

**{**

**"body": {**

**"record\_type": "input",**

**"input\_record": {**

**"device\_name": {**

**"enable\_group": true,**

**"ip\_addr": "192.168.4.157",**

**"name": "4.157Enter",**

**"sn": "0ba76bc6-e26d52eb",**

**"type": "input"**

**},**

**"ivs\_result\_param": {**

**"bright": 0,**

**"car\_bright": 0,**

**"car\_color": 0,**

**"confidence": 99,**

**"direction": 0,**

**"fragment\_path": "/tmp/app/html/snapshot/lpr/patch\_tri\_snap\_16.jpg",**

**"id": 0,**

**"image\_path": "/tmp/app/html/snapshot/lpr/tri\_snap\_16.jpg",**

**"image\_sd\_path": "/media/mmcblk0p1/VzIPCCap/2015\_11\_12/1533300901\_陕A8T00学.jpg",**

**"location": {**

**"bottom": 590,**

**"left": 587,**

**"right": 696,**

**"top": 571**

**},**

**"n\_time": 0,**

**"plate": "****陕A8T00学",**

**"plate\_color": 13,**

**"plate\_type" : 0,**

**"related\_plate" : "陕A8T00学",**

**"timeval" : {**

**"decday" : 5,**

**"dechour" : 15,**

**"decmin" : 26,**

**"decmon" : 5,**

**"decsec" : 14,**

**"decyear" : 2017,**

**"tv\_sec" : 1493969174,**

**"tv\_usec" : 416656**

**},**

**"trig\_type": 1**

**},**

**"state": 1**

**}**

**},**

**"cmd": "dg\_plateinfo\_result",**

**"id": "132156",**

**"state\_code": 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30，等于请求时的id |
| state\_code | 状态码 |
| cmd | 操作命令  dg\_plateinfo\_result组网识别消息结果 |

Body内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| record\_type | 消息类型  output出口消息  input 入口消息 |
| input\_record | 上面已经提到过了，请自行查找 |

### 清除组网数据：reset\_database

**请求命令参数**

**{**

**"body" : {**

**"type" : "reset\_database"**

**},**

**"cmd" : "dg\_json\_request",**

**"id" : "132156"**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30 |
| body | 需要传递给组网模块的数据 |

Body内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| type | 操作组网模块的命令   * reset\_database 清除组网数据 |

**结果返回**

**{**

**"body" : {**

**"state" : 200,**

**"type" : "reset\_database"**

**},**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30，等于请求时的id |
| body | 组网模块处理后返回的数据 |
| state\_code | 状态码 |

Body内的字段说明:

|  |  |
| --- | --- |
| **字段名** | **说明** |
| type | 操作组网模块的命令  reset\_database 清除组网数据 |
| state | 状态码 |

### 得到当前组网内部所有设备配置：get\_group\_cfg

**请求命令参数**

**{**

**"body" : {**

**"type" : "get\_group\_cfg"**

**},**

**"cmd" : "dg\_json\_request",**

**"id" : "132156"**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30 |
| body | 需要传递给组网模块的数据 |

Body内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| type | 操作组网模块的命令   * get\_group\_cfg 得到当前组网内部所有设备配置 |

**结果返回**

**{**

**"body" : {**

**"group\_cfg" : {**

**"group\_vzids" : [**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.105.39",**

**"name" : "",**

**"sn" : "018cd419-5e0702a4",**

**"type" : "unkown"**

**},**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.103.16",**

**"name" : "",**

**"sn" : "0a0248b7-f9121dba",**

**"type" : "unkown"**

**},**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.8.94",**

**"name" : "",**

**"sn" : "10391362-00e22a20",**

**"type" : "unkown"**

**},**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.4.186",**

**"name" : "6L2m5bqTNQ==&55u46L6FNTk=&MTkyLjE2OC41LjEx#dHJlZURlbW9fMTU=",**

**"sn" : "169d39b6-34645de7",**

**"type" : "input"**

**},**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.4.132",**

**"name" : "",**

**"sn" : "3acd054f-6b3c0af3",**

**"type" : "input"**

**},**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.1.7",**

**"name" : "",**

**"sn" : "702cdcaa-597a919f",**

**"type" : "unkown"**

**},**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.102.40",**

**"name" : "",**

**"sn" : "783f0ea9-90485466",**

**"type" : "unkown"**

**},**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.22.71",**

**"name" : "",**

**"sn" : "7a827083-62ad93cd",**

**"type" : "input"**

**},**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.3.3",**

**"name" : "",**

**"sn" : "7aea8606-8d88dd06",**

**"type" : "unkown"**

**},**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.102.4",**

**"name" : "",**

**"sn" : "a9132b96-4af3b688",**

**"type" : "unkown"**

**},**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.4.137",**

**"name" : "",**

**"sn" : "abf20a38-23f0b33d",**

**"type" : "unkown"**

**},**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.1.187",**

**"name" : "",**

**"sn" : "abf775b8-bf50d0b2",**

**"type" : "unkown"**

**},**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.105.38",**

**"name" : "",**

**"sn" : "af52ed95-8deaeadb",**

**"type" : "unkown"**

**},**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.102.41",**

**"name" : "ODEyNy0t5Yir5Yqo&55u46L6FMQ==&MTkyLjE2OC4xMDIuNDE=#dHJlZURlbW9fNg==",**

**"sn" : "b189c806-faec3304",**

**"type" : "output"**

**},**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.12.251",**

**"name" : "",**

**"sn" : "b1a7806e-326e00a2",**

**"type" : "unkown"**

**},**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.20.10",**

**"name" : "",**

**"sn" : "ed2045b2-30acf3a2",**

**"type" : "unkown"**

**},**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.102.56",**

**"name" : "",**

**"sn" : "f0565a83-fd3be473",**

**"type" : "unkown"**

**},**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.105.22",**

**"name" : "",**

**"sn" : "fcf2e551-a44c1cce",**

**"type" : "unkown"**

**},**

**{**

**"connect\_status" : 5,**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.105.14",**

**"name" : "",**

**"sn" : "ffedb27b-cd76bd51",**

**"type" : "unkown"**

**}**

**]**

**},**

**"state" : 200,**

**"type" : "get\_group\_cfg",**

**"vzid" : {**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.7.22",**

**"name" : "bHk=&MTkyLjE2OC43LjIy#",**

**"sn" : "56a8872e-2c5f7863",**

**"type" : "output"**

**}**

**},**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"state\_code" : 200,**

**"type" : null**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30，等于请求时的id |
| body | 组网模块处理后返回的数据 |
| state\_code | 状态码 |

body内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| group\_cfg | 组网内所有设备配置 |
| state | 状态码 |
| type | 操作组网模块的命令   * get\_group\_cfg 得到当前组网内部所有设备配置 |
| vzid | 设备的名称 |

group\_cfg内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| group\_vzids | 组网内所有设备配置信息)(由多条设备配置信息组成) |

group\_vzids内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| connect\_status | 连接状态 |
| enable\_group | 开启组网功能   * true 开启 * false 关闭 |
| ip\_addr | IP地址 |
| name | 设备的名称 |
| sn | 设备序列号 |
| type | 设备类型   * input 入口设备 * output 出口设备 |

vzid字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| enable\_group | 开启组网功能   * true 开启 * false 关闭 |
| ip\_addr | 设备ip |
| name | 设备名字 |
| sn | 设备序列号 |
| type | 设备类型   * input 入口设备 * output 出口设备 |

### 设置当前组网内部所有设备配置：set\_group\_cfg

**请求命令参数**

**{**

**"cmd": "dg\_json\_request",**

**"id": "132156",**

**"body": {**

**"type": "set\_group\_cfg",**

**"vzid" :**

**{**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.7.22",**

**"name" : "bHk=&MTkyLjE2OC43LjIy#",**

**"sn" : "56a8872e-2c5f7863",**

**"type" : "input"**

**},**

**"group\_cfg":**

**[{**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.1.2",**

**"name" : "bHk=&asdg4231#",**

**"sn" : "1238872e-2c5fasdv",**

**"type" : "input",**

**"connect\_status": 2**

**},**

**{**

**"enable\_group" : true,**

**"ip\_addr" : "192.168.22.12",**

**"name" : "bHk=&456sdg13#",**

**"sn" : "asxc887e-2cafasdv",**

**"type" : "input",**

**"connect\_status": 2**

**}]**

**}**

**}**

**结果返回**

**{**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"state\_code" : 200**

**"body" :**

**{**

**"state" : 200,**

**"type" : "set\_group\_cfg"**

**}**

**}**

### 根据ID获取组网图片：get\_img\_by\_id

**请求命令参数**

**{**

**"body" : {**

**"image\_id" : 79509,**

**"image\_type" : 1,**

**"sn" : "ef69b450-21bf3bff",**

**"type" : "get\_img\_by\_id"**

**},**

**"cmd" : "dg\_json\_request",**

**"id" : "132156"**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30 |
| body | 需要传递给组网模块的数据 |

body内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| image\_id | 图片id |
| image\_type | 图片类型 |
| sn | 设备序列号 |
| type | 操作组网模块的命令   * get\_img\_by\_id 得到当前组网内部所有设备配置 |

**结果返回**

**{**

**"body" : {**

**"image\_id" : 79509,**

**"image\_type" : 1,**

**"state" : 404,**

**"type" : "get\_img\_by\_id"**

**},**

**"cmd" : "dg\_json\_request",**

**"id" : "132156",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 序列字符串（唯一），字符串长度 < 30，等于请求时的id |
| body | 组网模块处理后返回的数据 |
| state\_code | 状态码 |

body内的字段说明：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| image\_id | 图片id |
| image\_type | 图片类型 |
| state | 状态码 |
| type | 操作组网模块的命令   * get\_img\_by\_id 得到当前组网内部所有设备配置 |

## 用户登录

### 请求开始登录:start\_login

**请求命令参数**

**{**

**"cmd" : "start\_login",**

**"id" : "123456"**

**}**

**结果返回**

**{**

**"active\_id":0,**

**"cmd":"start\_login",**

**"device\_sn":"b77b647f-277f2aa9",**

**"ems":[**

**{"m\_id":0,"name":"none encrypt"},**

**{"m\_id":1,"name":aes-128-ecb-padding-zeros},**

**{"m\_id":2,"name":level 2}**

**],**

**"id":"123456",**

**"signature":" Rr17TBjPHNYXJR80",**

**"state\_code":200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| active\_id | 当前加密方式的索引 |
| ems | 加密方式列表   * **m\_id**加密方式的索引 * **name**加密方式的名字 |
| signature | 一个十六位的字符串（用于对传输的密码进行加密，每调用一次get\_ems/reset\_encrypt\_key/change\_encrypt\_key/enable\_encrypt/start\_login四个接口中的任意一个接口, signature都会变，重置/修改密码和开启加密需采用最新的signature） |
| state\_code: | 状态码   * 200：成功 |

### 登录认证: login\_authentication

**请求命令参数**

**{**

**"cmd" : "login\_authentication",**

**"id" : "13245",**

**"authentication" : "12345789"**

**}**

**结果返回**

**{**

**"cmd":"login\_authentication",**

**"id":"13245",**

**"state\_code":200**

**}**

**注：认证失败，直接断开会话连接，需重新建立会话连接**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **authentication** | 用户密码;  使用HMAC-SHA1算法对 用户密码进行加密，加密物料使用signature，再对加密后的数据进行Base64生成的字符串。  Signature可通过get\_ems/reset\_encrypt\_key/change\_encrypt\_key/enable\_encrypt/start\_login接口获取，需采用最新的signature |
| id | 序列字符串（唯一），字符串长度 < 30，等于请求时的id |

**如**

**Std::string user\_pass;**

**Std::string signature;**

**Std::string authentication;**

**unsigned char digest\_buffer[20] = {0};**

**HmacSha1((const unsigned char \*) user\_pass.c\_str(), user\_pass.size(),**

**(const unsigned char \*) signature.c\_str(),signature.size(), digest\_buffer);**

**Base64::EncodeFromArray(digest\_buffer, 20, &authentication);**

## 语音协议

语音协议主要分为两部分，配置和播放；

**注 意：此处的语音为相机设备自带的语音，非外接；**

**请求命令参数：**

**{**

**"cmd": "playserver\_json\_request",**

**"id": "132156",**

**"body": {**

**"type": "%s"**

**...**

**}**

**}**

**结果返回：**

**{**

**"cmd": "playserver\_json\_request",**

**"id": "132156",**

**“state\_code”:200,**

**"body": {**

**"type": "%s"**

**...**

**}**

**}**

### 获取当前语音文件列表：playserver\_json\_request

**请求命令参数：**

**{**

**"cmd" : "playserver\_json\_request",**

**"id":"132156",**

**"body":{**

**"type":"ps\_get\_voice\_info",**

**"voice\_type":3**

**}**

**}**

**结果返回：**

**{**

**"body" : {**

**"type" : "ps\_get\_voice\_info",**

**"voice" : [**

**{**

**"file\_name" : "MC53YXY="**

**},**

**{**

**"file\_name" : "6bKBLndhdg=="**

**},**

**{**

**"file\_name" : "6buRLndhdg=="**

**}**

**],**

**"voice\_type" : 4**

**},**

**"cmd" : "playserver\_json\_request",**

**"id" : "132156",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **type** | 获取语音文件**ps\_get\_voice\_info** |
| **file\_name** | 语音文件名称，base64字符串； |
| **voice\_type** | 文件类型：  用户上传语音目录结构为:/home/admin/video/male,  /home/admin/video/female   1. 未知类型 2. 用户语音男声（用户通过ftp上传至/home/admin目录下.wav文件） 3. 用户语音女声（用户通过ftp上传至/home/admin目录下.wav文件） 4. 系统语音男声 5. 系统语音女声 |

### 设置语音默认参数：playserver\_json\_request

**请求命令参数：**

**{**

**"cmd": "playserver\_json\_request",**

**"id": "132156",**

**"body": {**

**"type" : "ps\_set\_voice\_config",**

**"voice\_defalut\_interval" : 1,**

**"voice\_defalut\_volume" : 1,**

**"voice\_defalut\_male" : 1**

**}**

**}**

**结果返回：**

**{**

**"body" : {**

**"type" : "ps\_set\_voice\_config"**

**},**

**"cmd" : "playserver\_json\_request",**

**"id" : "132156",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **voice\_defalut\_interval** | 语音文件默认播放间隔，单位为毫秒； |
| **voice\_defalut\_volume** | 语音默认音量，范围为[0, 100],现在还没有用，待开放； |
| **voice\_defalut\_male** | 默认语音类型：0 男声；1 女声； |
| **type** | ps\_set\_voice\_config设置语音 |

### 获取语音默认参数：playserver\_json\_request

**请求命令参数：**

**{**

**"cmd" : "playserver\_json\_request",**

**"id":"132156",**

**"body":{**

**"type" : "ps\_get\_voice\_config"**

**}**

**}**

**结果返回：**

**{**

**"body" : {**

**"type" : "ps\_get\_voice\_config",**

**"voice\_defalut\_interval" : 300,**

**"voice\_defalut\_male" : 1,**

**"voice\_defalut\_volume" : 100**

**},**

**"cmd" : "playserver\_json\_request",**

**"id" : "132156",**

**"state\_code" : 200**

**}**

### 播放语音参数：playserver\_json\_request

**请求命令参数：**

**{**

**"cmd" : "playserver\_json\_request",**

**"id":"132156",**

**"body":{**

**"type" : "ps\_voice\_play",**

**"voice" : "agaasg",**

**"voice\_interval" : 1,**

**"voice\_volume" : 1,**

**"voice\_male" : 1**

**}**

**}**

**结果返回：**

**{**

**"cmd" : "playserver\_json\_request",**

**"id":"132156",**

**"state\_code":200,**

**"body":{**

**"type" : " ps\_voice\_play "**

**}**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **voice** | 语音信息，utf-8/GBK编码的BASE64编码字符串 |
| **voice\_interval** | 语音文件播放间隔 |
| **voice\_volume** | 语音文件音量大小 |
| **voice\_male** | 语音类型：0 男声；1 女声 |
| **type** | ps\_voice\_play语音播放 |

### 请求语音对讲：start\_talk

**注意：设置语音对讲后，由相机回复出来的消息，是广播到所有的会话连接；**

**此语音对讲命令只是控制命令，具体的语音对讲信息通过双方协商的端口通信；**

**请求命令参数**

**{**

**"cmd": "start\_talk",**

**"id" : "1234",**

**"body":{**

**"window\_size":640**

**}**

**}**

**结果返回**

**{**

**"body" : {**

**"encode\_type" : 3,**

**"port" : 2707,**

**"status" : 1,**

**"window\_size" : 640**

**},**

**"cmd" : "start\_talk",**

**"id" : "1234",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **encode\_type** | 支持的编码类型  1. PCM  2. G711  3. PCM | G711 |
| **port** | 语音对讲通信端口 |
| **status** | 对讲状态  0 空闲  1 忙碌 |
| **window\_size** | 服务器窗口大小，每次发送数据包的大小 |

### 监听语音对讲：request\_talk

**注意：此命令，为相机主动向TCP连接的会话，主动发起的语音对讲请求，需要tcp客户端监听此命令；**

**监听cmd命令参数**

**请求命令参数**

**{**

**"cmd": " request \_talk",**

**"id" : "1234",**

**"state\_code" : 200,**

**“error\_msg”:”ok”**

**}**

**结果返回**

**收到此命令时，TCP客户端作为语音对讲的服务端，也需要回复如下信息：**

**{**

**"body" : {**

**"encode\_type" : 3,**

**"port" : 2707,**

**"status" : 1,**

**"window\_size" : 640**

**},**

**"cmd" : "start\_talk",**

**"id" : "1234",**

**"state\_code" : 200**

**}**

## 屏显协议

**注 意：**

1.仅当led\_enable为1时，led\_content中的内容才有效；

2. line\_content 数组大小与led\_line\_num保持一致；最大不超过4；

3.仅当voice\_mode 不为 0 时，voice\_content中的内容有效；

4.与网页配置互斥，通过TCP配置后，网页配置不再生效，网页配置后，TCP配置失效；

5.请注意此处的语音为外接LED屏显自带的语音；

### 设置LED显示内容：set\_led\_show

**请求命令参数**

**{**

**"cmd" : "set\_led\_show",**

**"id" : "12365",**

**"body" : {**

**"led\_enable" : 1,**

**"led\_content" :{**

**"led\_proto" : 1,**

**"led\_status" : 1,**

**"led\_refresh\_time" : 1,**

**"led\_line\_num" : 4,**

**"line\_content":**

**[{"show\_mode" : 1, "show\_content" : "5qyi6L+O5YWJ5Li0"},**

**{"show\_mode" : 1, "show\_content" : "5qyi6L+O5YWJ5Li0"},**

**{"show\_mode" : 1, "show\_content" : "5qyi6L+O5YWJ5Li0"},**

**{"show\_mode" : 1, "show\_content" : "5qyi6L+O5YWJ5Li0"}]**

**},**

**"voice\_mode" : 1,**

**"voice\_content" : {**

**"voice\_volume":2,**

**"voice\_welcom": 1,**

**"voice\_tag" : 1,**

**"play\_content":"5qyi6L+O5YWJ5Li0"**

**},**

**"car\_info":{**

**"park\_time" : 32,**

**"payment\_amount" : 9,**

**"car\_type" : 1,**

**"car\_plate" : "5qyi6L+O5YWJ5Li0"**

**}**

**}**

**}**

**结果返回**

**{**

**"cmd" : "set\_led\_show",**

**"id" : “12365”,**

**"state\_code":200**

**}**

### 获取LED显示内容：get\_led\_show

**请求命令参数**

**{**

**"cmd" : " get\_led\_show",**

**"id" : “12365”**

**}**

**结果返回**

**{**

**"cmd" : "get\_led\_show",**

**"id" : “12365”,**

**"state\_code":200**

**"body" : {**

**"led\_enable" : 1,**

**"led\_content" :{**

**"led\_proto" : 1,**

**"led\_status" : 1,**

**"led\_refresh\_time" : 1,**

**"led\_line\_num" : 4,**

**"line\_content":**

**[{"show\_mode" : 1, "show\_content" : "5qyi6L+O5YWJ5Li0"},**

**{"show\_mode" : 1, "show\_content" : "5qyi6L+O5YWJ5Li0"},**

**{"show\_mode" : 1, "show\_content" : "5qyi6L+O5YWJ5Li0"},**

**{"show\_mode" : 1, "show\_content" : "5qyi6L+O5YWJ5Li0"}]**

**},**

**"voice\_mode" : 1,**

**"voice\_content" : {**

**"voice\_volume":2,**

**"voice\_welcom": 1,**

**"voice\_tag" : 1,**

**"play\_content":"5qyi6L+O5YWJ5Li0"**

**},**

**"car\_info":{**

**"park\_time" : 32,**

**"payment\_amount" : 9,**

**"car\_type" : 1,**

**"car\_plate" : "5qyi6L+O5YWJ5Li0"**

**}**

**}**

**}**

### 获取LED数据传输使用串口号：get\_led\_serial\_port

**请求命令参数**

**{**

**"cmd" : " get\_led\_serial\_port",**

**"id" : “12365”**

**}**

**结果返回**

**{**

**"cmd" : "get\_led\_serial\_port",**

**"use\_serial\_port" : 1**

**"id" : “12365”,**

**"state\_code":200**

**}**

### 设置LED数据传输使用串口号：set\_led\_serial\_port

**请求命令参数**

**{**

**"cmd" : "set\_led\_serial\_port",**

**"body" : {**

**"use\_serial\_port" : 1**

**},**

**"id" : 12365**

**}**

**结果返回**

**{**

**"cmd" : "set\_led\_serial\_port",**

**"id" : 12365,**

**"state\_code":200**

**}**

### 字段信息以及相关枚举

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 说明 |
| voice\_mode | int | 语音播放播放模式：LED\_VOICE\_PLAY\_MODE |
| voice\_content | 无 | voice\_mode不为0时有效 |
| voice\_volume | int | 音量大小（0 - 9） |
| voice\_welcom | int | 欢迎语： LED\_VOICE\_USER\_DEFINE |
| voice\_tag | int | 结束语：LED\_VOICE\_USER\_DEFINE |
| led\_enable | int | 使能屏显 (0:去使能, 1:使能) |
| led\_content | 无 | led\_enable 为1时，内容有效 |
| led\_line\_num | int | 屏幕支持行数(目前最大支持4行) |
| led\_proto | int | 屏显协议：参考枚举定义： RS485\_CTRL\_PRO |
| led\_status | int | 屏显状态，用户自保存 |
| led\_refresh\_time | int | 屏显内容定时刷新时间[6,15] |
| line\_content | 无 | 屏显每行具体内容,数组大小与led\_line\_num值一致 |
| show\_mode | int | 屏显显示模式，参考枚举： LED\_SCREEN\_SHOW\_MODE |
| show\_content | string | 用户自定义显示（utf-8的字符串经过64位编码之后的内容） |
| play\_content | string | 用户自定义播放（utf-8的字符串经过64位编码之后的内容）仅支持32省份简称+大写字母+数字+军+警+学+十+百+千+万+年+月+日 |
| use\_serial\_port | int | 使用的串口号（1,2，部分设备支持3） |
| state | int | 回复状态；200：成功，400：格式错误，500服务器错误 |
| car\_info | 无 | 用户需要显示车辆信息时，有效 |
| park\_time | int | 停车时间（分钟） |
| payment\_amount | int | 收费金额，元 |
| car\_type | int | 车辆类型：LED\_VOICE\_CAR\_TYPE |
| car\_plate | string | 车牌号：UTF-8经过64位编码 |

**相关枚举定义：**

**1.屏显协议：RS485\_CTRL\_PRO**

enum RS485\_CTRL\_PRO {

ZS\_SERIAL = 0x00, /\* A 臻识科技识别结果输出协议 \*/

YB\_BX5K1 = 0x01, /\* B 仰绑科技BX5K1控制卡协议 \*/

XQ\_EQ2013 = 0x02, /\* C 谐阔EQ系列2013控制卡协议 \*/

KF\_LED = 0X03， /\* D 科发LED控制卡协议 \*/

FK\_LED = 0X04， /\* E 方控智能LED控制卡协议 \*/

};

**2.屏显模式：LED\_SCREEN\_SHOW\_MODE**

enum LED\_SCREEN\_SHOW\_MODE {

LED\_SHOW\_MODE\_NONE = 0x00, // 不显示

LED\_SHOW\_MODE\_USER\_DEFINE = 0x01, // 显示用户自定义内容

LED\_SHOW\_MODE\_SYS\_TIME = 0x02, // 显示时间

LED\_SHOW\_MODE\_FREE\_PARKING = 0x04, // 显示空余车位

LED\_SHOW\_MODE\_CAR\_PLATE = 0x08, // 显示车牌号

LED\_SHOW\_MODE\_CAR\_TYPE = 0x10, // 显示车类型

LED\_SHOW\_MODE\_PARK\_TIME = 0x20, // 显示停车时间

LED\_SHOW\_MODE\_CHARGE\_MONEY = 0x40, // 显示收费金额

LED\_SHOW\_MODE\_MAX,

};

**3.语音播放模式：LED\_VOICE\_PLAY\_MODE**

enum LED\_VOICE\_PLAY\_MODE {

LED\_VOIC\_PLAY\_NONE = 0x00, // 不播放语音

LED\_VOICE\_PLAY\_WELCOM = 0x01, // 欢迎语

LED\_VOICE\_PLAY\_CAR\_TYPE = 0x02, // 车辆类型

LED\_VOICE\_PLAY\_CAR\_PLATE = 0x04, // 车牌号

LED\_VOICE\_PLAY\_PARK\_TIME = 0x08, // 停车时间

LED\_VOICE\_PLAY\_CHARGE\_MONEY = 0x10, // 收费金额

LED\_VOICE\_PLAY\_TAG = 0x20, // 结束语

LED\_VOICE\_PLAY\_USER\_DEFINE = 0x40, // 自定义语音

LED\_VOICE\_PLAY\_MAX,

};

**4.** **播放语音的车辆类型：LED\_VOICE\_CAR\_TYPE**

enum LED\_VOICE\_CAR\_TYPE{

CAR\_TYPE\_MONTH\_RENT = 1, //月租车

CAR\_TYPE\_TEMP\_CAR = 2, // 临时车

CAR\_TYPE\_NO\_PLATE = 3, // 无牌车

CAR\_TYPE\_BLACKLIST = 4, // 黑名单

CAR\_TYPE\_MONTH\_RENT\_EXPIRE = 7, // 月租车到期

CAR\_TYPE\_SPECIAL\_CAR = 9, // 特殊车

CAR\_TYPE\_MAX,

};

**5.播放的欢迎语、结束语**

enum LED\_VOICE\_USER\_DEFINE {

USER\_DEFINE\_VOICE\_NONE = 0, //

USER\_DEFINE\_VOICE\_WELCOM\_1, // 您好

USER\_DEFINE\_VOICE\_WELCOM\_2, // 欢迎光临

USER\_DEFINE\_VOICE\_WELCOM\_3, // 欢迎回家

USER\_DEFINE\_VOICE\_TAG\_1, // 一路顺风

USER\_DEFINE\_VOICE\_TAG\_2, // 一路平安

USER\_DEFINE\_VOICE\_TAG\_3, // 欢迎下次光临

USER\_DEFINE\_VOICE\_MAX,

};

## 设置用户自定义OSD

**请求命令参数**

**{**

**"cmd": "set\_osd\_para",**

**"body": {**

**"user\_osd": {**

**"user\_osd\_param": [{**

**"id": 0,**

**"display": 1,**

**"color": 0,**

**"front\_size": 3,**

**"text": "56ys5LiA6KGM"**

**}, {**

**"id": 1,**

**"display": 1,**

**"color": 1,**

**"front\_size": 3,**

**"text": "c2Vjb25kIGxpbmU="**

**}, {**

**"id": 2,**

**"display": 1,**

**"color": 2,**

**"front\_size": 3,**

**"text": "dGjkuInooYw="**

**}, {**

**"id": 3,**

**"display": 1,**

**"color": 3,**

**"front\_size": 3,**

**"text": "56ys5Zub6KGMT1NEIFRDUCA="**

**}],**

**"x\_pos": 10,**

**"y\_pos": 10**

**}**

**}**

**}**

**结果返回**

**{**

**"cmd" : "set\_osd\_para",**

**"error\_msg" : "Sucess",**

**"id" : "999999",**

**"state\_code" : 200**

**}**

参数解释

|  |  |
| --- | --- |
| **字段名** | **说明** |
| id | 自定义OSD行数，取值０到３，id为０表示第一行，３表示第四行 |
| **display** | 是否显示，１：显示文本　　０：不显示文本 |
| **color** | 文本颜色，取值０到３　　０：白色　１：红色　　２：蓝色　　３：绿色 |
| **front\_size** | 文本字体大小。取值０到３　０：最小字号　　　３：最大字号 |
| **text** | 想要显示的文本经过ｂａｓｅ６４编码后的内容 |
| **x\_pos** | 横坐标百分比，取值范围０到１００　例如取值为０，最左边显示 |
| **y\_pos** | 纵坐标百分比，取值范围０到１００　例如取值为０，最上边显示 |

## 用户自定义设备信息

用户自定义设备信息设置和获取。

### 获取自定义信息: get\_custom\_user\_info

**请求命令参数**

**{**

**"cmd" : "get\_custom\_user\_info",**

**"id" : "123456"**

**}**

**结果返回**

**{**

**"cmd":"** **get\_custom\_user\_info",**

**"body":{**

**"manufacturer":”SJ-1”,**

**"data":{“ver”:1,”is\_zoom”:0,”class”:1}**

**},**

**"id":"123456",**

**"err\_msg":"OK",**

**"state\_code":200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| manufacturer | 字符串标识的协议适配厂家  如果没有配置或厂家不在支持列表返回“unknown” |
| data | 自定义信息（不同厂家结构不同，以SJ-1为例）   * **ver 版本，整数0~3，0表示没有设置。** * **is\_zoom 是否支持变焦，整数0~2，1支持，2不支持，0表示没有设置。** * **class 设备等级（或分类），整数0~15，0表示没有设置。**   **注：值为0（没有设置）的项不会通过广播发送。** |
| id | 序列字符串（唯一），字符串长度 < 30，等于请求时的id |
| err\_msg | 错误信息  命令执行正确返回“OK” |
| state\_code: | 状态码   * 200：成功 * 其他：失败 |

### 设置用户自定义信息: set\_custom\_user\_info

**请求命令参数**

**{**

**"cmd" : "set\_custom\_user\_info ",**

**"id" : "132456",**

**"body":{**

**"manufacturer":”SJ-1”,**

**"data":{“ver”:1,”is\_zoom”:0,”class”:1}**

**}**

**}**

**结果返回**

**{**

**"cmd":"set\_custom\_user\_info",**

**"id":"132456",**

**"state\_code":200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| body | 需要设置的信息  manufacturer：字符串标识的协议适配厂家  data：自定义信息（不同厂家结构不同，以SJ-1为例）   * **ver 版本，整数0~4，0表示没有设置。** * **is\_zoom 是否支持变焦，整数0~2，1不支持，2支持，0表示没有设置。** * **class 设备等级（或分类），整数0~16，0表示没有设置。**   **注：值为0（没有设置）的项不会通过广播发送。** |
| id | 序列字符串（唯一），字符串长度 < 30，等于请求时的id |
| state\_code: | 状态码  200：成功  其他：失败 |

## 用户通过R3获取伴侣机Rg的相关配置信息

### 通过R3获取指定伴侣机的配置信息：user\_get\_mate\_info

**请求命令参数**

**{**

**"cmd" : "user\_get\_mate\_info",**

**"id" : "12345693"**

**}**

**结果返回**

**{**

**"body" : [**

**{**

**"dev\_name" : "SVZT",**

**"device\_type" : "RG",**

**"ip" : "192.168.13.112",**

**"online" : 1,**

**"sn" : "e7cc75b8-d5774474"**

**}**

**],**

**"cmd" : "user\_get\_mate\_info",**

**"error\_msg" : "OK",**

**"id" : "12345693",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **ip** | 伴侣机Ip地址 |
| **dev\_names** | 伴侣机设备名 |
| **sn** | 伴侣机序列号 |
| **device\_type** | 伴侣机设备类型 |
| **online** | 伴侣机是否在线 0 离线 1在线 |
| state\_code: | 状态码   * 200：成功 * 其他：失败 |

### 通过R3获取伴侣机的Rtsp代理信息:user\_get\_rtsp\_info

**请求命令参数**

**{**

**"cmd" : "user\_get\_rtsp\_info",**

**"id" : "12345694",**

**"body":{**

**"sn":"e7cc75b8-d5774474"**

**}**

**}**

**结果返回**

**{**

**"body" : {**

**"proxyurl" : "rtsp://192.168.13.81:8557/afa5852b-6945f0f3\_h264",**

**"proxyurl\_sub1" : "rtsp://192.168.13.81:8557/afa5852b-6945f0f3\_sub\_stream\_1"**

**},**

**"cmd" : "user\_get\_rtsp\_info",**

**"error\_msg" : "OK",**

**"id" : "12345694",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| sn | 伴侣机的序列号 |
| **proxyurl** | 转发代理伴侣机主码流地址 |
| **proxyurl\_sub1** | 转发代理伴侣机子码流地址 |
| state\_code: | 状态码   * 200：成功 * 其他：失败 |

### 通过R3获取伴侣机的语音代理信息:user\_request\_talkback

**请求命令参数**

**{**

**"cmd" :"**user\_request\_talkback**",**

**"id" : "1234569",**

**"body":{**

**"sn":"e7cc75b8-d5774474",**

**"window\_size" : 320**

**}**

**}**

**结果返回**

**{**

**"body" : {**

**"encode\_type" : 3,**

**"port" : 10011,**

**"sampling\_rate" : 16000,**

**"window\_size" : 320**

**},**

**"cmd" : "user\_request\_talkback",**

**"error\_msg" : "OK",**

**"id" : "1234569",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| Sn | 伴侣机序列号 |
| **port** | 转发伴侣机的语音代理端口 |
| **window\_size** | 每次发送数据包的大小 |
| **encode\_type** | 支持的编码类型  1. PCM  2. G711  3. PCM | G711 |
| **sampling\_rate** | 采样率例如：8000表示8khz 12000表示12khz以此类推 |
| state\_code: | 状态码   * 200：成功 * 其他：失败 |

### 通过Tcp获取日志指定的日志:log\_search

**请求命令参数**

**{"cmd":"log\_search",**

**"id":"12345697",**

**"body":{**

**"start\_time":1627315200,**

**"end\_time":1627401599,**

**"qtype":1,**

**"max\_id":0,**

**"min\_id":0,**

**"start\_id":0}**

**}**

**结果返回**

**{**

**"body" : {**

**"logs" : [**

**],**

**"qnode" : {**

**"last\_id" : 658864**

**"max\_id" : 658963,**

**"min\_id" : 647353,**

**"start\_id" : 4294967295**

**}**

**},**

**"err\_msg" : "OK",**

**"state" : 200,**

**"type" : "log\_search"**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **start\_time** | 开始时间时间戳 |
| **end\_time** | 结束时间时间戳 |
| **qtype** | 日志排序方式，默认1表示降序,(0 升序) |
| **max\_id** | 最大的日志id，第一次获取应答包主动赋值，用于控制搜索次数，除了第一包其他的后续包都需要设置，第一包必须是0 |
| **min\_id** | 最小的日志id，第一次获取应答包主动赋值，用于控制搜索次数，除了第一包其他的后续包都需要设置,第一包必须是0 |
| **start\_id** | 第一次默认设置为0，后面是应答包last\_id的值 |
| **last\_id** | 应答包的最后一个日志的id，用于设置下次发送的start\_id，最后一个包是0 |
| **logs** | 详细日志信息 |
| state\_code: | 状态码   * 200：成功 * 其他：失败 |

### 设置Rg人车滞留事件上报使能:stay\_event\_enable

**请求命令参数**

**{**

**"cmd" : "stay\_event\_enable",**

**"id" : "12345701",**

**"body":{**

**"enable": true,**

**"format": "json",**

**"image": true,**

**"image\_type": 0**

**}**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| enable | 是否允许推送识别结果，默认值：false   * true 表示允许推送识别结果 * false 表示不推送识别结果 |
| format | 推送识别结果数据格式，默认值：json  json表示识别结果数据格式为json数据格式 |
| image | 识别结果是否包含图片，默认值：true   * true 表示识别结果包含图片 * false 表示识别结果不包含图片 |
| image\_type | 识别的图片类型，默认值：0   * 0 表示返回识别结果全图 * 1 表示只返回车牌区域小图 * 2 表示返回两种图片 |

**结果返回**

**{**

**"cmd":" stay\_event\_enable",**

**“id”:”12345701”,**

**"state\_code":200,**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| state\_code: | 状态码   * 200：成功 * 其他：失败 |

当配置完成后（配置为允许推送识别结果的情况enable：ture），并且在服务器端有ivsresult识别结果产生后，tcp server会主动推送common\_alarm\_result识别结果给客户端，推送的识别结果格式如下

**配置完成后，接收到识别结果为配置值：**

正常情况下，一个完整的识别结果（返回2种图片的情况）推送消息格式如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **‘V’** | **‘Z’** | **0** | **0** | **长度（数据包的大小）** |
| **识别结果字符串+字符串结束符‘\0’** | | | | |
| **全图的jpg格式数据** | | | | |
| **小图片（车牌区域图片）的jpg格式数据** | | | | |

例如下面就是一个正常的JSON识别推送结果:

**{**

**"cmd":"common\_alarm\_result",**

**"device\_info":{**

**"alg\_chn":0,**

**"chn\_id":0,**

**"dev\_ip":"192.168.19.214",**

**"dev\_port":80,**

**"rule\_id":0**

**},**

**"full\_image":[**

**{**

**"fe\_crop\_info":{**

**"Height":911,**

**"Left":506,**

**"Top":412,**

**"Width":1118**

**},**

**"file\_name":"/nfsroot/test\_big.jpeg",**

**"full\_image\_id":344,**

**"image\_type":0**

**}**

**],**

**"module":"TCP\_BUS\_REQUEST",**

**"results":[**

**{**

**"appname":"lane\_control",**

**"clip\_type":6,**

**"event\_type":81,**

**"grab\_type":65,**

**"lane\_ctrl\_prop":{**

**"car\_stay\_result":{**

**"alarm\_state":1**

**},**

**"person\_stay\_result":{**

**"alarm\_state":1**

**},**

**"non\_car\_stay\_result":{**

**"alarm\_state":1**

**},**

**"lane\_except\_result":{**

**"alarm\_state":1**

**},**

**"image\_id":345**

**},**

**"result\_id":5,**

**"time":"2021-09-13 15:53:56",**

**"type":"common\_alarm\_lane\_control"**

**}**

**],**

**"small\_image":[**

**{**

**"file\_name":"/nfsroot/test\_small.jpeg",**

**"full\_image\_id":345**

**}**

**],**

**"type":"common\_alarm\_result"**

**}**

**//json 后接图片数据，数据排列格式**

**|数据头+图片数据|数据头+图片数据|**

**/\* 图片二进制数据头 \*/**

**static const char NET\_MARK\_0 = 'V';**

**static const char NET\_MARK\_1 = 'Z';**

**typedef struct \_ImageHead {**

**uint8 mark[2]; // [0]='V' [1]='Z'**

**uint16 image\_type; // 图片类型; 0=jpg,1=png**

**uint32 image\_id; // 图片 id**

**uint32 data\_size; // 图片长度；**

**} ImageHead**

其中各字段的含义如下表所示：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| alg\_chn | 算法通道，固定值0 |
| chn\_id | 固定值0 |
| dev\_ip | 设备ip |
| dev\_port | 兼容老RG，固定值80 |
| rule\_id | 规则id |
| fe\_crop\_info | 小图坐标 |
| file\_name | 大图路径 兼容老RG，固定值。 |
| full\_image\_id | 大图id |
| image\_type | 0背景图 1 缩略图 |
| appname | 固定值 |
| clip\_type | 检测框类型：枚举型   * 0：未知 * 1：人脸 * 2：人头 * 3：人形 * 4：车牌 * 5：非机动车 * 6：机动车 * 7：扩展模型 |
| event\_type | 输出事件类型：枚举型   * 0x50： 人滞留 * 0x51： 车滞留 * 0x52： 非车滞留 * 0x53：通行事件 * 0x54： 折返事件 * 0x55： 道闸抬升 * 0x56：闸杆回落 * 0x57： 闸杆异常   注：目前这个字段就可以确定事件类型，其他的事件类型有些是为了兼容老版本，可以不关注。 |
| grab\_type | 算法抓拍事件类型：枚举型   * 0x40：车道布控报警 * 0x41：滞留 * 0x42：车通过 * 0x43： 车折返 * 0x44： 闸杆抬升 * 0x45： 闸杆落下 * 0x46： 闸杆异常 |
| car\_stay\_result | 车滞留事件 兼容老RG保留字段 |
| person\_stay\_result | 人滞留事件 兼容老RG保留字段 |
| non\_car\_stay\_result | 非车滞留事件 兼容老RG新增字段 |
| lane\_except\_result | 栏杆异常事件 兼容老RG新增字段 |
| alarm\_state | 报警状态始终是1兼容老RG新增字段 |
| image\_id | 小图id |
| lane\_ctrl\_prop | 兼容老RG字段 |
| result\_id | 算法输出事件累加id，重启自动清零。 |
| time | 输出事件时间 |
| trigger\_type | 全部是0，兼容老RG。 |

## RG设备Alg参数获取与设置

### 获取RG绊线绑定算法参数

**请求命令参数**

**{**

**"cmd" : "get\_alg\_para",**

**"body":**

**{**

**"rule\_chn" : 2,**

**"alg\_prm\_type":"cross\_line\_bind\_prm"**

**},**

**"id" : "132158"**

**}**

**结果返回**

**{**

**"body" : {**

**"alg\_chn" : 0,**

**"param" : {**

**"cross\_line\_bind\_prm" : {**

**"is\_open" : 0,**

**"relative\_rule\_chn" : 0**

**}**

**},**

**"rule\_chn" : 2**

**},**

**"cmd" : "get\_alg\_para",**

**"error\_msg" : "success",**

**"id" : "132158",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **rule\_chn** | 规则ID   * 0： 车辆滞留规则ID * 1： 人员检测规则ID * 2： 绊线规则ID * 3： 非车检测规则ID * 4： 道闸状态规则检测ID |
| **is\_open** | 使能开关 0 关闭 1打开 |
| **relative\_rule\_chn** | 需要绑定的规则通道，此项参数仅用于绑定车辆通行拌线与车辆通行识别区域,不启用任何实际事件,默认0,RG这个值不可更改。 |
| state\_code: | 状态码   * 200：成功 * 其他：失败 |

### 获取车通行折返事件

**请求命令参数**

**{**

**"cmd" : "get\_alg\_para",**

**"body":**

**{**

**"rule\_chn" : 0,**

**"alg\_prm\_type":"car\_motion\_prm"**

**},**

**"id" : "132158"**

**}**

**结果返回**

**{**

**"body" : {**

**"alg\_chn" : 0,**

**"param" : {**

**"car\_motion\_prm" : {**

**"car\_pass\_event" : {**

**"is\_area1" : 0,**

**"is\_area2" : 0,**

**"is\_open" : 1**

**},**

**"car\_retrace\_event" : {**

**"is\_open" : 1**

**},**

**"is\_open" : 1**

**}**

**},**

**"rule\_chn" : 0**

**},**

**"cmd" : "get\_alg\_para",**

**"error\_msg" : "success",**

**"id" : "132158",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **rule\_chn** | 规则ID   * 0： 车辆滞留规则ID * 1： 人员检测规则ID * 2： 绊线规则ID * 3： 非车检测规则ID * 4： 道闸状态规则检测ID |
| **is\_open** | 使能开关 0 关闭 1打开 |
| **is\_area1，is\_area2** | 通行方向，正向或者反向，1是开启，0是关闭。 |
| **car\_retrace\_event** | 车辆折返事件 |
| **car\_pass\_event** | 车辆通行事件 |
| state\_code: | 状态码   * 200：成功 * 其他：失败 |

### 设置车通行折返事件

**请求命令参数**

**{**

**"cmd" : "set\_alg\_para",**

**"body":**

**{**

**"rule\_chn" : 0,**

**"alg\_prm\_type":"car\_motion\_prm",**

**"param" : {**

**"car\_motion\_prm" : {**

**"car\_pass\_event" : {**

**"is\_area1" : 1,**

**"is\_area2" : 1,**

**"is\_open" : 1**

**},**

**"car\_retrace\_event" : {**

**"is\_open" : 1**

**},**

**"is\_open" : 1**

**}**

**}**

**},**

**"id" : "132158"**

**}**

**结果返回**

**{**

**"cmd" : "set\_alg\_para",**

**"error\_msg" : "success",**

**"id" : "132158",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **rule\_chn** | 规则ID   * 0： 车辆滞留规则ID * 1： 人员检测规则ID * 2： 绊线规则ID * 3： 非车检测规则ID * 4： 道闸状态规则检测ID |
| **is\_open** | 使能开关 0 关闭 1打开 |
| **is\_area1，is\_area2** | 通行方向，正向或者反向，1是开启，0是关闭。 |
| **car\_retrace\_event** | 车辆折返事件 |
| **car\_pass\_event** | 车辆通行事件 |
| state\_code: | 状态码   * 200：成功 * 其他：失败 |

### 非车滞留事件获取

**请求命令参数**

**{**

**"cmd" : "get\_alg\_para",**

**"body":{**

**"rule\_chn" : 3,**

**"alg\_prm\_type" : "nonv\_stay\_prm"**

**},**

**"id" : "132158"**

**}**

**结果返回**

**{**

**"body" : {**

**"alg\_chn" : 0,**

**"param" : {**

**"nonv\_stay\_prm" : {**

**"is\_open" : 1,**

**"nonv\_stay\_time" : {**

**"enable" : 1,**

**"time" : 10**

**},**

**"nonv\_width" : {**

**"max" : 150,**

**"min" : 20**

**}**

**}**

**},**

**"rule\_chn" : 3**

**},**

**"cmd" : "get\_alg\_para",**

**"error\_msg" : "success",**

**"id" : "132158",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **rule\_chn** | 规则ID   * 0： 车辆滞留规则ID * 1： 人员检测规则ID * 2： 绊线规则ID * 3： 非车检测规则ID * 4： 道闸状态规则检测ID |
| **is\_open** | 使能开关 0 关闭 1打开 |
| **nonv\_stay\_time/time** | 判断车滞留时间阈值，单位秒，范围（5~3600） |
| **nonv\_widt** | 车辆宽度 |
| state\_code: | 状态码   * 200：成功 * 其他：失败 |

### 非车滞留事件设置

**请求命令参数**

**{**

**"cmd" : "set\_alg\_para",**

**"body":**

**{**

**"rule\_chn" : 3,**

**"alg\_prm\_type":"nonv\_stay\_prm",**

**"param" : {**

**"nonv\_stay\_prm" : {**

**"is\_open" : 1,**

**"nonv\_stay\_time" : {**

**"enable" : 1,**

**"time" : 10**

**},**

**"nonv\_width" : {**

**"max" : 150,**

**"min" : 20**

**}**

**}**

**}**

**},**

**"id" : "132158"**

**}**

**结果返回**

**{**

**"cmd" : "set\_alg\_para",**

**"error\_msg" : "success",**

**"id" : "132158",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **rule\_chn** | 规则ID   * 0： 车辆滞留规则ID * 1： 人员检测规则ID * 2： 绊线规则ID * 3： 非车检测规则ID * 4： 道闸状态规则检测ID |
| **is\_open** | 使能开关 0 关闭 1打开 |
| **nonv\_stay\_time/time** | 判断车滞留时间阈值，单位秒，范围（5~3600） |
| **nonv\_widt** | 车辆宽度 |
| state\_code: | 状态码   * 200：成功 * 其他：失败 |
| **字段名** | **说明** |

### 获取人滞留事件

**请求命令参数**

**{**

**"cmd" : "get\_alg\_para",**

**"body":{**

**"rule\_chn" : 1,**

**"alg\_prm\_type" : "person\_stay\_prm"**

**},**

**"id" : "132158"**

**}**

**结果返回**

**{**

**"body" : {**

**"alg\_chn" : 0,**

**"param" : {**

**"person\_stay\_prm" : {**

**"is\_open" : 1,**

**"person\_stay\_time" : {**

**"enable" : 1,**

**"time" : 10**

**},**

**"person\_width" : {**

**"max" : 600,**

**"min" : 50**

**}**

**}**

**},**

**"rule\_chn" : 1**

**},**

**"cmd" : "get\_alg\_para",**

**"error\_msg" : "success",**

**"id" : "132158",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **rule\_chn** | 规则ID   * 0： 车辆滞留规则ID * 1： 人员检测规则ID * 2： 绊线规则ID * 3： 非车检测规则ID * 4： 道闸状态规则检测ID |
| **is\_open** | 使能开关 0 关闭 1打开 |
| **person\_stay\_time/time** | 判断人滞留时间阈值，单位秒，范围（5~3600） |
| **person\_width** | 行人宽度 |
| state\_code: | 状态码   * 200：成功 * 其他：失败 |

### 设置人滞留事件

**请求命令参数**

**{**

**"cmd" : "set\_alg\_para",**

**"body":**

**{**

**"rule\_chn" : 1,**

**"alg\_prm\_type":"person\_stay\_prm",**

**"param" : {**

**"person\_stay\_prm" : {**

**"is\_open" : 1,**

**"person\_stay\_time" : {**

**"enable" : 1,**

**"time" : 10**

**},**

**"person\_width" : {**

**"max" : 600,**

**"min" : 50**

**}**

**}**

**}**

**},**

**"id" : "132158"**

**}**

**结果返回**

**{**

**"cmd" : "set\_alg\_para",**

**"error\_msg" : "success",**

**"id" : "132158",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **rule\_chn** | 规则ID   * 0： 车辆滞留规则ID * 1： 人员检测规则ID * 2： 绊线规则ID * 3： 非车检测规则ID * 4： 道闸状态规则检测ID |
| **is\_open** | 使能开关 0 关闭 1打开 |
| **person\_stay\_time/time** | 判断人滞留时间阈值，单位秒，范围（5~3600） |
| **person\_width** | 行人宽度 |
| state\_code: | 状态码   * 200：成功 * 其他：失败 |

### 获取车滞留事件参数

**请求命令参数**

**{**

**"cmd" : "get\_alg\_para",**

**"body":{**

**"rule\_chn" : 0,**

**"alg\_prm\_type" : "car\_stay\_prm"**

**},**

**"id" : "132158"**

**}**

**结果返回**

**{**

**"body" : {**

**"alg\_chn" : 0,**

**"param" : {**

**"car\_stay\_prm" : {**

**"car\_stay\_time" : {**

**"enable" : 1,**

**"time" : 10**

**},**

**"car\_width" : {**

**"max" : 1920,**

**"min" : 50**

**},**

**"is\_open" : 1**

**}**

**},**

**"rule\_chn" : 0**

**},**

**"cmd" : "get\_alg\_para",**

**"error\_msg" : "success",**

**"id" : "132158",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **rule\_chn** | 规则ID   * 0： 车辆滞留规则ID * 1： 人员检测规则ID * 2： 绊线规则ID * 3： 非车检测规则ID * 4： 道闸状态规则检测ID |
| **is\_open** | 使能开关 0 关闭 1打开 |
| **car\_stay\_time/time** | 判断车滞留时间阈值，单位秒，范围（5~3600） |
| **car\_width** | 车辆宽度 |
| state\_code: | 状态码   * 200：成功 * 其他：失败 |

### 设置车滞留事件参数

**请求命令参数**

**{**

**"cmd" : "set\_alg\_para",**

**"body":**

**{**

**"rule\_chn" : 0,**

**"alg\_prm\_type":"car\_stay\_prm",**

**"param" : {**

**"car\_stay\_prm" : {**

**"car\_stay\_time" : {**

**"enable" : 1,**

**"time" : 10**

**},**

**"car\_width" : {**

**"max" : 1920,**

**"min" : 50**

**},**

**"is\_open" : 1**

**}**

**}**

**},**

**"id" : "132158"**

**}**

**结果返回**

**{**

**"cmd" : "set\_alg\_para",**

**"error\_msg" : "success",**

**"id" : "132158",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **rule\_chn** | 规则ID   * 0： 车辆滞留规则ID * 1： 人员检测规则ID * 2： 绊线规则ID * 3： 非车检测规则ID * 4： 道闸状态规则检测ID |
| **is\_open** | 使能开关 0 关闭 1打开 |
| **car\_stay\_time/time** | 判断车滞留时间阈值，单位秒，范围（5~3600） |
| **car\_width** | 车辆宽度 |
| state\_code: | 状态码   * 200：成功 * 其他：失败 |

### 获取道闸事件参数

**请求命令参数**

**{**

**"cmd" : "get\_alg\_para",**

**"body":{**

**"rule\_chn" : 4,**

**"alg\_prm\_type" : "flag\_motion\_prm"**

**},**

**"id" : "132158"**

**}**

**结果返回**

**{**

**"body" : {**

**"alg\_chn" : 0,**

**"param" : {**

**"flag\_motion\_prm" : {**

**"flag\_error\_time" : {**

**"enable" : 1,**

**"time" : 300**

**},**

**"is\_open" : 1**

**}**

**},**

**"rule\_chn" : 4**

**},**

**"cmd" : "get\_alg\_para",**

**"error\_msg" : "success",**

**"id" : "132158",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **rule\_chn** | 规则ID   * 0： 车辆滞留规则ID * 1： 人员检测规则ID * 2： 绊线规则ID * 3： 非车检测规则ID * 4： 道闸状态规则检测ID |
| **is\_open** | 使能开关 0 关闭 1打开 |
| **flag\_error\_time/time** | 判断道闸事件时间阈值，单位秒，范围（5~3600） |
| state\_code: | 状态码   * 200：成功 * 其他：失败 |

### 设置道闸事件参数

**请求命令参数**

**{**

**"body" : {**

**"alg\_prm\_type" : "flag\_motion\_prm",**

**"param" : {**

**"flag\_motion\_prm" : {**

**"flag\_error\_time" : {**

**"enable" : 1,**

**"time" : 600**

**},**

**"is\_open" : 1**

**}**

**},**

**"rule\_chn" : 4**

**},**

**"cmd" : "set\_alg\_para",**

**"id" : "132158"**

**}**

**结果返回**

**{**

**"cmd" : "set\_alg\_para",**

**"error\_msg" : "success",**

**"id" : "132158",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **rule\_chn** | 规则ID   * 0： 车辆滞留规则ID * 1： 人员检测规则ID * 2： 绊线规则ID * 3： 非车检测规则ID * 4： 道闸状态规则检测ID |
| **is\_open** | 使能开关 0 关闭 1打开 |
| **car\_stay\_time/time** | 判断车滞留时间阈值，单位秒，范围（5~3600） |
| **car\_width** | 车辆宽度 |
| state\_code: | 状态码   * 200：成功 * 其他：失败 |

### 获取布防时间

算法暂时不支持

### 设置布防时间

**请求命令参数**

**{**

**"cmd" : "set\_alg\_para",**

**"body":**

**{**

**"rule\_chn" : 3,**

**"alg\_prm\_type":"depolyment\_time",**

**"param":{"depolyment\_time":**

**[{"day":0,"period":[{"end\_time":{"hour":23,"minute":59},**

**"start\_time":{"hour":0,"minute":0}}]},**

**{"day":1,"period":[{"end\_time":{"hour":23,"minute":59},**

**"start\_time":{"hour":0,"minute":0}}]},**

**{"day":2,"period":[{"end\_time":{"hour":23,"minute":59},**

**"start\_time":{"hour":0,"minute":0}}]},**

**{"day":3,"period":[{"end\_time":{"hour":23,"minute":59},**

**"start\_time":{"hour":0,"minute":0}}]},**

**{"day":4,"period":[{"end\_time":{"hour":23,"minute":59},**

**"start\_time":{"hour":0,"minute":0}}]},**

**{"day":5,"period":[{"end\_time":{"hour":23,"minute":59},**

**"start\_time":{"hour":0,"minute":0}}]},**

**{"day":6,"period":[{"end\_time":{"hour":23,"minute":59},**

**"start\_time":{"hour":0,"minute":0}}]}]}**

**},**

**"id" : "132158"**

**}**

**结果返回**

**{**

**"cmd" : "set\_alg\_para",**

**"error\_msg" : "success",**

**"id" : "132158",**

**"state\_code" : 200**

**}**

|  |  |
| --- | --- |
| **字段名** | **说明** |
| **rule\_chn** | 规则ID   * 0： 车辆滞留规则ID * 1： 人员检测规则ID * 2： 绊线规则ID * 3： 非车检测规则ID * 4： 道闸状态规则检测ID |
| **is\_open** | 使能开关 0 关闭 1打开 |
| **day** | 0-6 依次表示周一到周日 |
| **period** | 每天可以设置多个时间段 |
| state\_code: | 状态码   * 200：成功 * 其他：失败 |