

# 车牌识别一体机

## HTTP 通讯协议说明手册

V20250318001

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
1.8.0	2018. 08. 23	1. 增加HTTP断线重传功能	20180823180
1.8.1	2021. 12. 30	1、文件排版更新	20211230181
1.8.2	2022. 11. 8	1、文件排版更新	20221108182
1.8.3	2023. 06. 09	增加rule_id、clean_time、plates字段	20230609183
1.8.4	2023. 10. 10	R产品文档合并	20231010184
1.8.5	2023. 10. 31	补充视频osd接口文档	20231031185
1.8.6	2023. 11. 29	补充危险品字段	20231129186
1.8.7	2024. 01. 25	lcd配置更新	20240125187
1.8.8	2024. 04. 24	识别结果内容详细等级字段更新	20240424188
1.8.9	2024. 05. 11	增加道闸状态字段	20240511189
1.9.0	2024. 06. 05	设置图片OSD协议支持动态叠加	20240605190
1.9.1	2024. 06. 06	增加道闸连接状态，功能开关状态上报	20240606191
1.9.2	2024. 08. 13	1. 白名单操作新增加reply_url和msg_id可选字段 2. 新增白名单回复接口（3.7.5.2） 3. 新增时间同步接口（3.7.10） 4. 新增IO常开/常闭操作接口（3.7.11）	20240813130
1.9.3	2024. 11. 20	1. 增加record_uuid、event_channel字段（3.1.1） 2. 截图业务，新增chnlid字段（3.7.3）	20241120160
1.9.4	2024. 12. 12	1. 触发类型新增85和86	20241212115
1.9.5	2024. 12. 24	1. 增加防回退协议（3.7.13）	20241224116
1.9.6	2025. 01. 21	1. 新增车辆颜色（米、橙、紫）	20250121117
1.9.7	2025. 03. 18	1. 增加协议返回说明（3.7.8.7）	20250318001

## 目录

1. 范围 .....	1
2. 协议概述 .....	2
2.1. 数据类型说明 .....	2
2.2. HTTP 服务器配置 .....	2
3. 协议定义 .....	5
3.1. 车牌识别结果推送 .....	5
3.1.1. 识别结果结构定义 .....	5
3.1.2. 断线重传 .....	13
3.1.3. 墨系列相机 .....	16
3.2. 端口触发信息推送 .....	17
3.3. 串口数据推送 .....	18
3.4. 截图数据 .....	18
3.5. 道闸状态推送 .....	19
3.6. 设备注册 .....	20
3.6.1. 普通心跳 .....	20
3.6.2. comet 轮询 .....	21
3.7. 业务处理 .....	21
3.7.1. 控制 IO 开闸 .....	21
3.7.2. 控制串口推送 485 数据 .....	22
3.7.3. 截图 .....	22
3.7.4. 手动触发识别 .....	23
3.7.5. 白名单 .....	25
3.7.6. 系统预置语音播报 .....	29
3.7.7. IO 控制 .....	30
3.7.8. LCD 配置 .....	30
3.7.9. OSD 配置 .....	59
3.7.10. 时间同步 .....	78
3.7.11. IO 常开/常闭控制 .....	79
3.7.12. 防回退版本控制 .....	79
4. 系统码表 .....	81
4.1. 触发类型 .....	81
4.2. 车牌类型 .....	81
4.3. 车牌颜色 .....	82
4.4. 车身颜色 .....	83
4.5. 运动方向 .....	83
4.6. 车辆类型 .....	83
4.7. 开闸类型 .....	84
5. 常见问题 .....	85
5.1. Q: 设备注册是什么？ .....	85
5.2. Q: 设置好了，请求收不到，什么问题？ .....	85
5.3. Q: 请求收到了，但没有数据（数据格式不对）？ .....	85

5.4. Q: 设备注册又是什么格式? .....	85
5.5. Q: 如何回复请求开闸? .....	86
5.6. Q: 回复中 <b>content</b> 能不能是中文? .....	86
5.7. Q: 能否使用 <b>ssl</b> 连接发送, 我们的中心服务器是 <b>ssl</b> 的? .....	86
5.8. Q: 怎么获取截图? .....	86
5.9. Q: 为什么相同车牌返回了两次结果? .....	86
5.10. Q: 中心服务器, 收到推送结果, 回复给一体机, 但看不到返回的具体内容, 如何调试中心服务器? .....	87
5.11. Q: 多台相机同时使用 <b>comet</b> 轮询时, 中心服务器压力过大, 导致服务器响应不过来? .....	87

# 1. 范围

本文档对 R3、R5、R5D、RX、RXIV、X2 设备对外提供的 HTTP 接口进行定义描述。

## 2. 协议概述

### 2.1. 数据类型说明

本协议采用 JSON 数据格式进行交互。

此章节定义协议中用到的各类数据类型。

表 1 数据类型定义表

数据类型标识	备注
int	整数
uint	无符号整数
float	浮点数
double	浮点数
string(N)	字符串。其中 N 表示字符串的最大长度，单位：字节
base64	base64 编码格式的字符串
bool	布尔值： true , false
obj	JSON 对象
time_t	时间戳，单位：秒
msecond	时间戳，单位：毫秒
array	JSON 数组。数组中默认存放的是 obj，若是其它类型需要备注说明，如 array<float> 表示数组中存放的是浮点数数组。

### 2.2. HTTP 服务器配置

1. 使用 HTTP 推送首先需要用户建立一个 HTTP 服务器，同时将这台 HTTP 服务器的地址配置给相机一体机；
2. 当一体机有识别结果后（或者其他需要推送的内容时），就会往指定的服务器地址发送 HTTP 协议消息；
3. 在一体机网页，登录后，点击菜单栏->高级设置->HTTP 推送，进入到 HTTP 推送的设置界面(**不同版本稍有区别**)；
4. 接收 HTTP 推送的服务器，配置包括地址（可以填 ip 地址或者域名），端口号，是否开启 ssl 连接，ssl 端口号，和超时时间设置。请根据架设的服务器的情况进行配置；



图 1 设备 HTTP 配置界面

5. HTTP 服务器可配置 1 个主服务器，最多 3 个备选服务器；若勾选了主服务器优先，则相机只往主服务器推送，当且仅当主服务器断开连接时，才会往备选服务器推送；若没勾选主服务优先，则会同时往主/备服务器推送数据；
6. HTTP 推送具体配置，即 HTTP 需要推送的内容，包括车牌识别结果、IO 触发、串口 485 数据，需要配置相应推送的 url；
7. HTTP 心跳分为取消心跳、普通心跳、comet 轮询；普通心跳相机定时往主服务器推送心跳，不处理主服务器的业务回复；comet 轮询则一直与服务器推送心跳交互，并且服务器可在回复消息中携带业务处理；
8. HTTP 脱机检查，当开启脱机检查时，相机会对 HTTP 进行脱机检查；脱机检查分为心跳检查以及识别

结果检查；心跳检查则为普通心跳定时检查；识别结果检查为当产生识别结果时，推送识别结果后需要在检查时间内收到服务器的回复，否则相机会置状态为脱机；当相机处于脱机时，会进行脱机相关的业务处理；

9. 配置重发次数，最多可配置 4 次，即当产生推送时，若没推送成功，相机会对推送数据进行重发；

## 3. 协议定义

### 3.1. 车牌识别结果推送

#### 3.1.1. 识别结果结构定义

**支持设备：**目前所有 R 系列

开启推送车牌识别结果,同时配置推送 url 后, 当有车牌识别结果产生时, 相机会按图中的配置会发送消息到:

<http://192.168.1.106/devicemanagement/php/plateresult.php>;

**注：**RG 作为辅设备发送过来的车牌结果会转换为 R5 的结果输出，协议与 R5 一致。

**请注意：**在增加 HTTP 重传功能后, 相对于旧版本, 车牌识别结果推送消息增加了 3 个字段, 分别为: plateid、isoffline、gioouts;

识别包含在 AlarmInfoPlate 对象中, 此对象结构如表 2 所示。

表 2 车牌识别结果推送协议定义表

层级	字段编码	数据类型	备注	内容详细等级				
				全部 (拓展)	全部	较 详 细	较 简 略	简 略
1	channel	string	默认通道号(预留)	√	√	√	√	√
2	deviceName	string	设备名称	√	√	√	√	√
3	ipaddr	string	设备 ip 地址	√	√	√	√	√
4	result	obj	实际数据	√	√	√	√	√
4. 1	PlateResult	obj	车牌识别结果信息	√	√	√	√	√
4. 1. 1	plate_true_width	string	车牌真实宽度	√				
4. 1. 2	plate_distance	int	车牌距离	√				
4. 1. 3	is_fake_plate	int	是否伪车牌, 0: 真实车牌, 1: 伪车牌	√				
4. 1. 4	car_location	obj	车头位置信息	√				
4. 1. 4. 1	RECT	obj	位置为矩形区域	√				
4. 1. 4. 1. 1	bottom	int	车牌在图片中位置	√				

4. 1. 4. 1. 2	left	int	车牌在图片中位置	√				
4. 1. 4. 1. 3	right	int	车牌在图片中位置	√				
4. 1. 4. 1. 4	top	int	车牌在图片中位置	√				
4. 1. 5	car_brand	obj	车辆品牌信息	√				
4. 1. 5. 1. 1	brand	int	车辆品牌	√				
4. 1. 5. 1. 2	year	int	车系编码	√				
4. 1. 5. 1. 3	type	int	车辆类型。取值参见表 8。	√				
4. 1. 6	feature_code	string	车辆特征码	√				
4. 1. 7	bright	int	亮度评价（预留）	√	√	√		
4. 1. 8	carBright	int	车身亮度（预留）	√	√	√		
4. 1. 9	carColor	int	车身颜色（预留）	√	√	√		
4. 1. 10	colorType	int	车牌颜色，参见表 5。	√	√	√	√	√
4. 1. 11	clean_time	int	R5 未使用	√	√	√	√	√
4. 1. 12	colorValue	int	车牌颜色(车牌颜色的字符串值,该字段数据无效)	√	√	√		
4. 1. 13	confidence	int	识别结果可信度 1-100	√	√	√		
4. 1. 14	direction	int	车的行进方向。取值参见表 7;	√	√	√	√	√
4. 1. 15	imagePath	string	弃用	√	√	√	√	√
4. 1. 16	imageFile	string	识别大图片内容经过 base64 后的字符串, 需要开启发送大图片	√	√	√	√	√
4. 1. 17	imageFileLen	int	识别大图片内容长度 , 注意不是 base64 后的长度, 需要开启发送大图片	√	√	√	√	√
4. 1. 18	imageFragmentFile	string	识别车牌小图片内容经过 base64 后的字符串, 需要开启发送小图片	√	√	√	√	√
4. 1. 19	imageFragmentFileLen	int	识别小图片内容长度 , 注意不是 base64 后的长度, 需要开启发送小图片	√	√	√	√	√

4. 1. 20	is_danger	int	是否是危险车牌，0: 否, 1: 是, -1: 该功能未启用（目前仅 E5 支持该字段）	√	√	√	√	√
4. 1. 21	license	string	车牌号字符串, 如 "京 AAAAAA"	√	√	√	√	√
4. 1. 22	license_ext_type	int	新式小型车牌扩展, 仅在车牌为小型车牌的时候有效	√	√	√	√	√
4. 1. 23	location	obj	车牌在图片中位置	√	√			
4. 1. 23. 1	RECT	obj	位置为矩形区域	√	√			
4. 1. 23. 1.	bottom	int	车牌在图片中位置	√	√			
4. 1. 23. 1.	left	int	车牌在图片中位置	√	√			
4. 1. 23. 1.	right	int	车牌在图片中位置	√	√			
4. 1. 23. 1.	top	int	车牌在图片中位置	√	√			
4. 1. 24	timeStamp	obj	识别结果对应帧的时间戳	√	√	√	√	
4. 1. 24. 1	Timeval	obj	时间戳结构体类型	√	√	√	√	
4. 1. 24. 1.	decday	int	时间, 天	√	√	√	√	
4. 1. 24. 1.	dechour	int	时间, 小时	√	√	√	√	
4. 1. 24. 1.	decmin	int	时间, 分钟	√	√	√	√	
4. 1. 24. 1.	decmon	int	时间, 月	√	√	√	√	
4. 1. 24. 1.	decsec	int	时间, 秒	√	√	√	√	
4. 1. 24. 1.	decyear	int	时间, 年	√	√	√	√	
4. 1. 24. 1.	sec	int	从 1970 年 1 月 1 日到对应帧的秒	√	√	√	√	
4. 1. 24. 1.	usec	int	从 1970 年 1 月 1 日到对应帧的毫秒	√	√	√	√	
4. 1. 25	timeUsed	int	识别所用时间	√	√			
4. 1. 26	triggerType	int	当前结果的触发类型, 参见表 3。	√	√	√	√	√
4. 1. 27	type	int	车牌类型。取值范围参见表 4。	√	√	√	√	√
4. 1. 28	plateid	int	识别结果车牌 ID	√	√	√	√	√
4. 1. 29	plates	array	车牌列表	√	√	√	√	√
4. 1. 29. 1	color	int	车牌颜色, 参见表 5。	√	√	√	√	√

4. 1. 29. 2	imageFragmentFile	string	识别车牌小图片内容经过 base64 后的字符串，需要开启发送小图片	√	√	√	√	√
4. 1. 29. 3	imageFragmentFileLen	int	识别小图片内容长度，注意不是 base64 后的长度，需要开启发送小图片	√	√	√	√	√
4. 1. 29. 4	binImageFile	string	识别车牌二值化小图片内容经过 base64 后的字符串，需要开启发送小图片(分辨率 120*20，图像格式为 BMP/1bit, 不含 BMP 文件头)	√	√	√	√	√
4. 1. 29. 5	binImageFileLen	int	识别二值化小图片内容长度，注意不是 base64 后的长度，需要开启发送小图片	√	√	√	√	√
4. 1. 29. 6	license	string	加密采用 gbk 编码，不加密采用 utf8 编码。	√	√	√	√	√
4. 1. 29. 7	plate_width	int	车牌宽度	√	√	√	√	√
4. 1. 29. 8	pos	object	车牌坐标	√	√	√	√	√
4. 1. 29. 9	type	int	车辆类型。取值参见表 8。	√	√	√	√	√
4. 1. 30	isoffline	int	设备离线状态，0：在线，1：离线	√	√	√	√	√
4. 1. 31	gioouts	array	开闸输出信息（新增）	√	√	√	√	√
4. 1. 31. 1	ionum	int	IO OUT 序号，当前最大 4 个 IOout	√	√	√	√	√
4. 1. 31. 2	ctrltype	int	开闸类型，参见表 9。	√	√	√	√	√
5	rule_id	int	算法检测框的 ID 0：虚拟线圈 1：识别区域 2：辅线圈	√	√	√	√	√

			3 布防区域(多线圈时使用)					
6	serialno	string	设备序列号，设备唯一	√	√	√	√	√
7	user_data	string	用户自定义数据	√	√	√	√	√
8	oemserialno	string	OEM 标识，需要在 Tcp 协议中设置了客户定制 sn 才会生效，否则不会发送此字段到服务器。	√	√	√	√	√
4.1.32	begin_time	uint32	事件开始时间，秒	√	√	√	√	√
4.1.33	end_time	uint32	事件结束时间，秒	√	√	√	√	√
4.1.34	start_time	uint64	事件触发时间，毫秒					
4.1.35	unique_id	string	事件唯一 id，格式为 sn_start_time，如： "12345678-87654321_159728586500"					
4.1.36	event_channel	int	事件通道(目前仅 X2 产品有效)，0-表示拼接相机通道； 1-表示车头相机通道。					
4.1.37	record_uuid	uint64	记录关联 id (目前仅 X2 产品有效)，如： 1732073278970					

数据内容：JSON 格式 (utf8 编码)

```
{
  "AlarmInfoPlate": {
    "channel": 0,
    "deviceName": "R5-V1.0",
    "ipaddr": "192.168.55.131",
    "result": {
      "id": 1732073278970
    }
  }
}
```

```
"PlateResult":{  
    "bright":0,  
    "carBright":0,  
    "carColor":255,  
    "car_brand":{  
        "brand":255,  
        "type":255,  
        "year":65535  
    },  
    "car_location":{  
        "RECT":{  
            "bottom":0,  
            "left":0,  
            "right":0,  
            "top":0  
        }  
    },  
    "clean_time":0,  
    "colorType":4,  
    "colorValue":0,  
    "confidence":96,  
    "direction":0,  
    "feature_code":true,  
    "imageFile": "",  
    "imageFileLen":0,  
    "imageFragmentFile": "",  
    "imageFragmentFileLen":0,  
    "gioouts": [  
        {  
            "ctrltype":0,  
            "ionum": "0"  
        },  
        {  
            "ctrltype":0,  
            "ionum": "1"  
        },  
        {  
            "ctrltype":0,  
            "ionum": "2"  
        }  
    ],  
    "is_fake_plate":0,  
    "isoffline":0,  
    "license": "\u7ca4Z9065\u6fb3",
```

```
"license_ext_type":0,  
"location":{  
    "RECT":{  
        "bottom":674,  
        "left":1002,  
        "right":1263,  
        "top":565  
    }  
},  
"plate_distance":0,  
"plate_true_width":25,  
"plateid":21085,  
"plates": [  
    {  
        "color":4,  
        "imageFragmentFile": "",  
        "imageFragmentFileLen":0,  
        "binImageFile": "",  
        "binImageFileLen":0,  
        "license": "\u7ca4Z9065\u6fb3",  
        "plate_width":281,  
        "pos":{  
            "bottom":674,  
            "left":1002,  
            "right":1263,  
            "top":565  
        },  
        "type":14  
    }  
],  
"timeStamp":{  
    "Timeval":{  
        "decday":9,  
        "dechour":16,  
        "decmin":39,  
        "decmon":6,  
        "decsec":54,  
        "decyear":2023,  
        "sec":1686299994,  
        "usec":687772  
    }  
},  
"timeUsed":0,  
"triggerType":4,
```

```

    "type":14,
    "begin_time": 1725871942,
    "end_time": 1725871952,
    "start_time": 1686299994687,
    "record_uuid": 1732073278970,

    "event_channel": 0,
    "unique_id": "dac8d1d3-248ce33e_1686299994687"
}

},
"rule_id":1,
"serialno":"fe71e106-57263272",
"user_data":""}

}
}

```

根据内容详细等级，数据的内容会有所不同，“全部”将回传全部内容，简略的只包含其中部分重要内容，请根据需要选择详细度。开启报警发送图片后，json 中的将包含图像数据；

包含全部(扩展)内容的 json 数据如下，选项“全部”去掉了车头等扩展信息，“较详细”去掉了“location”，“timeUsed”，选项“较简略”还去掉了“confidence”，“bright”，“carBright”，“carColor”，“colorValue”，选项“简略”进一步去掉了“timeStamp”；

**请注意：增加 HTTP 重传功能后，需要使用此功能，服务器的回复中，必须携带 plateid 字段，和相机推送过来的数值保持一致；**

中心服务器回复内容：

```
{
  "Response_AlarmInfoPlate": {
    "info": "ok",
    "plateid": 123,
    "channelNum": 0,
    "manualTrigger": "ok",
    "TriggerImage": {
      "port": 80,
      "snapImageRelativeUrl": "/devicemanagement/php/receivedeviceinfo.php",
      "snapImageAbsolutelyUrl": "http://192.168.1.106/devicemanagement/php/receivedeviceinfo.php"
    },
    "is_pay": "true",
    "serialData": [
      {
        "serialChannel": 0,
        "data": "...",
        "dataLen": 123
      }
    ]
  }
}
```

```

        "serialChannel": 1,
        "data": "...",
        "dataLen": 123
    }
]
}
}

```

层级	字段编码	数据类型	备注
1.1	info	string	回复 ok 开闸
1.2	plateid	int	当前车牌 id
1.3	channelNum	int	回复开闸端口号，若无，则默认为 0
1.4	manualTrigger	string	回复 ok 进行手动触发
1.5	TriggerImage		可选，不触发截图可不添加该字段
1.5.1	port	int	回复截图内容端口号（可选，不填则默认使用 http 页面配置端口）
1.5.2	snapImageRelativeUrl	string	回复截图内容相对路径（可选，不触发截图可不添加该字段）
1.5.3	snapImageAbsolutelyUrl	string	回复截图内容绝对路径（可选，不触发截图可不添加该字段）
1.6	is_pay	string	
1.7	serialData	arry	回复串口数据可以发送到相应串口
1.7.1	serialChannel	int	串口号
1.7.2	data	string	回复数据
1.7.3	dataLen	int	数据长度

### 3.1.2. 断线重传

支持设备：目前所有 R 系列

- 当 HTTP 服务器因为某些原因，导致相机与服务器断线以后，相机会把推送失败的识别结果记为离线记录，当服务器重新连接上以后，相机根据配置判断是否需要推送离线记录，同时发送离线记录；
- 配置在网页配置，开启断线重传功能，**注意当取消断线重传功能时，会清空当前相机的离线记录；**
- 相机推送识别结果，相对于旧版本的推送消息，**新增三个字段：** plateid, isoffline, gioouts, 脱机记录 isoffline 的值为 1；
- 服务器回复相机识别结果时，**在线记录需要将 plateid 字段值回复到相机消息中；**
- 服务器回复离线识别结果时，**需要回复是否继续接收离线记录以及接收到最新的 plateid；**
- 离线脱机记录理论上最大支持 9000 条离线记录的重新推送；

7. 在推送离线记录的过程中，如果发生新的识别结果，优先推送新的识别结果，此时未完成推送的离线记录，会直接终止处理，当新识别结果推送完成之后，方重新开始推送离线记录；

**注意：**当推送离线记录，相机还未收到服务器的回复时，产生了新的识别结果，相机会终止上一条离线记录的推送处理，直接推送新的识别结果，当新的识别结果推送完毕后，再推送一条离线记录，故服务器此时有可能会收到两条一模一样的离线记录，服务器可根据 plateid 进行过滤；

```
{  
    "AlarmInfoPlate":{  
        "channel":0,  
        "deviceName":"IVS",  
        "ipaddr":"192.168.1.100",  
        "result":{  
            "PlateResult":{  
                "bright":0,  
                "carBright":0,  
                "carColor":0,  
                "colorType":0,  
                "confidence":0,  
                "direction":0,  
                "imagePath":"xxxxx.jpg",  
                "license":"_无_",  
                "location":{  
                    "RECT":{  
                        "bottom":0,  
                        "left":0,  
                        "right":0,  
                        "top":0  
                    }  
                },  
                "timeStamp":{  
                    "Timeval":{  
                        "decday":8,  
                        "dechour":10,  
                        "decmin":26,  
                        "decmon":6,  
                        "decsec":28,  
                        "decyear":2018,  
                        "sec":1441815171,  
                        "usec":672241  
                    }  
                },  
                "timeUsed":0,  
                "triggerType":4,  
            }  
        }  
    }  
}
```

```

    "type":0,
    "plateid":123,
    "isoffline":0,
    "gioouts":[
        {
            "ionum":1,
            "ctrltype":0
        }
    ]
},
"serialno":"eff50e18-e3d3862b"
}
}

```

字段名称	类型	取值范围	是否必须	描述
plateid	unsigned int	无	否	识别结果车牌 ID
isoffline	int	[0,1]	是	设备离线状态, 0: 在线, 1: 离线
ionum	int	[0, 3]	是	IO OUT 序号, 当前最大 4 个 IOout
ctrltype	int	[0, 2]	是	开闸类型: HTTP_IO_OUT_STATUS

服务器回复:

```

{
    "Response_AlarmInfoPlate": {
        ...// 其他数据
        "ContinuePushOffline": {
            "plateid": 123,
            "continue": 1
        }
    }
}

```

字段名称	类型	取值范围	是否必须	描述
plateid	unsigned int	无	是	推送的离线车牌记录 ID

字段名称	类型	取值范围	是否必须	描述
continue	unsigned int	[0,1]	是	是否继续推送离线记录，0：否，1：是

注意：只有当服务器回复了离线记录消息，以及 continue 字段为 1,时，才继续推送下一条离线记录；

### 3.1.3. 墾系列相机

针对垦系列的相机，HTTP 推送协议增加车辆信息以及伪车牌信息：

```
{
    "AlarmInfoPlate" :
    {
        ...// 其他字段
        "result" :
        {
            "PlateResult" :
            {
                ...//
                "plate_true_width" : 80,
                "plate_distance" : 40,
                "is_fake_plate" : 1,
                "car_location" :
                {
                    ...
                    "RECT" : {
                        "bottom" : 545,
                        "left" : 871,
                        "right" : 1365,
                        "top" : 391
                    }
                },
                "car_brand" :
                {
                    ...
                    "brand" : 6
                    "year" : 2018
                    "type" : 1
                },
                "feature_Code" : "asf"
            }
        }
    }
}
```

层级	字段名称	类型	取值范围	是否必须	描述
1.1	plate_true_width	int	[10, 1000]	是	车牌真实宽度
1.2	plate_distance	int	[0, 1000]	是	车牌距离
1.3	is_fake_plate	int	[0, 1]	是	是否伪车牌, 0: 真实车牌, 1: 伪车牌
1.4	car_location	无	无	是	车头位置
1.5	car_brand	无	无	无	车辆品牌
1.5.1	brand	int	[0, 128]	是	车辆品牌
1.5.2	year	int	[0, 65535]	是	车辆年份
1.5.3	type	int	[0, 128]	是	车辆类型
1.6	feature_Code	string	长度 [0, 20]	是	车辆特征码

## 3.2. 端口触发信息推送

支持设备：目前所有 R 系列

当开启时，如果在输入输出页面->车牌触发方式里，开启了外部输入 1 触发或者 2 触发，输入有变化时，会推送 json 格式数据，内容如下：

数据内容：JSON 格式（utf8 编码）

```
{
  "AlarmGioIn": {
    "deviceName": "IVS",
    "ipaddr": "192.168.109.40",
    "result": {
      "TriggerResult": {
        "source": 3,
        "value": 1
      }
    }
  }
}
```

```
"serialno":"d85b1269-8f942256"  
}  
}
```

其中，TriggerResult 中：

source=0 代表是 IO 输入 1;  
source=1 代表是 IO 输入 2;  
source=2 代表是 IO 输入 3;  
source=3 代表是 IO 输入 4;  
source=4 代表输入 TCP 触发输入;  
value 表示触发时输入的状态;  
其他参数的含义和车牌识别结果推送中相同。

中心服务器回复内容：任意合法数据，相机不对回复消息做业务处理

### 3.3. 串口数据推送

支持设备：目前所有 R 系列

当开启串口数据推送时，配置了 url，在相机收到 485 数据时，会主动往服务器地址推送 485 数据；

数据内容：JSON 格式（utf8 编码）

```
{  
    "SerialData":{  
        "channel" : 0,                      //通道号，当前为 0  
        "serialno" : "cead13eb-1a198cd7", //设备序列号  
        "ipaddr" : "192.168.1.100",       //设备 ip  
        "deviceName" : "IVS",             //设备名称  
        "serialChannel" : 0,              //串口的通道号，通道 0 为 485 口 1，通道 1 根据  
                                         //跳线方式为 485 口 2 或者 232  
        "data": "Y2guY29tFw==",          //串口数据，采用 base64 编码  
        "dataLen" : 7                   //串口数据实际长度  
    }  
}
```

中心服务器回复内容：任意合法数据，相机不对回复消息做业务处理

### 3.4. 截图数据

支持设备：目前所有 R 系列

用户在 comet 轮询或者收到识别结果的回复字段有获取截图时，设备会进行当前视频截图并上传，imageFile 字段为图片 base64 后的编码，imageFileLen 为编码前的图片长度

数据内容：JSON 格式（utf8 编码）

```
//推送截图数据
{
    "ipaddr": "192.168.1.100",
    "TriggerImage":
    {
        "imageFile": "Y2guY29tFw==",      //图片数据（base64 编码）
        "imageFileLen": 7                //图片数据实际长度
    }
}
```

中心服务器回复内容：任意合法数据，相机不对回复消息做业务处理

### 3.5. 道闸状态推送

当开启道闸状态推送时，配置了 url，在设备连接的道闸状态改变时，会主动往服务器地址推送道闸状态；

数据内容：JSON 格式（utf8 编码）

```
{
    "channel": 0,
    "cmd": "barr_gate_status_notify",
    "deviceName": "IVS",
    "ipaddr": "192.168.55.76",
    "result": {
        "connect_status": 1,
        "enable": 1,
        "gate_ctrl_id": 1,
        "gate_status": 1
    },
    "serialno": "63ccd073-7228f992"
}
```

字段名	类型	说明
gate_ctrl_id	int	道闸控制器型号编号
gate_status	int	道闸当前状态 0：关到位

		1: 开到位 2: 中间状态
connect_status	int	道闸连接状态 0: 未连接 1: 已连接
enable	int	道闸功能开关状态 0: 关闭 1: 开启
channel	int	通道号
deviceName	string	设备名称
ipaddr	string	设备 ip
serialno	string	设备序列号

## 3.6. 设备注册

支持设备：目前所有 R 系列

### 3.6.1. 普通心跳

1. 当相机网页配置设备注册状态为普通心跳时，相机会定时往主服务器推送心跳消息；
2. 当主服务连接正常，开启脱机检查的情况下，相机每隔 5S 左右推送一次心跳；
3. 当主服务连接正常，同时没开启脱机检查的情况下，是 30S 推送一次心跳消息；
4. 当主服务心跳丢失以后，相机每隔 1S 尝试连接一次；
5. 心跳推送使用 HTTP 长连接， POST 方式；

数据内容，使用 multipart/form-data text 类型的数据格式：

```
192.168.109.40-----caa771fe4a61f3d9Content-Disposition: form-data;
name="device_name"IVS-----caa771fe4a61f3d9Content-Disposition: form-data;
name="ipaddr"192.168.109.40-----caa771fe4a61f3d9Content-Disposition:
form-data;
name="port"80-----caa771fe4a61f3d9Content-Disposition: form-data;
name="user_name"admin-----caa771fe4a61f3d9Content-Disposition:
form-data;
name="pass_wd"admin-----caa771fe4a61f3d9Content-Disposition: form-data;
name="serialno"d85b1269-8f942256-----caa771fe4a61f3d9Content-Disposition:
form-data;
name="channel_num"1-----caa771fe4a61f3d9--
```

### 3.6.2. comet 轮询

1. 当开启 comet 轮询之后，相机会一直与 HTTP 服务器进行交互，保持连接请求，相机主动发送设备注册消息，内容与普通心跳内容一致，收到回复时，立即发送下一条消息；
2. 发送设备注册消息，与普通心跳消息保持一致；
3. comet 轮询会根据服务器回复做相应处理；

## 3.7. 业务处理

1. 相机根据服务器的回复消息，进行相应的业务处理；
2. 当前仅支持车牌识别结果的推送回复，以及 comet 轮询的消息回复，相机会根据回复做业务处理；
3. 可以多个消息组合发送例如系统播报语音同时通知 IO[0] 命令如下：

```
{
  "Response_AlarmInfoPlate":{
    "playserver_json_request":{
      "type":"ps_voice_play",
      "voice":"5qyi6L+O5YWJ5Li0",
      "voice_interval":0,
      "voice_volume":100,
      "voice_male":1
    },
    "ivs_ioctl":{
      "delay":500,
      "io":0,
      "value":2
    }
  }
}
```

### 3.7.1. 控制 IO 开闸

支持设备：目前所有 R 系列

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息，可触发开闸

```
{
  "Response_AlarmInfoPlate": {
    {
      "info":"ok",//回复 ok 开闸
      "channelNum":0, //io 编号
      "delay":500 //先通后断的延迟时间
    }
  }
}
```

```
// ....其他数据
}
}
```

### 3.7.2. 控制串口推送 485 数据

**支持设备：目前所有 R 系列**

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息，可发送 485 数据

```
{
  "Response_AlarmInfoPlate":
  {
    "serialData": [
      {
        "serialChannel": 0,
        "data": "...",
        "dataLen": 123
      },
      // 数据 1，可以有或者没有，收到后将发送到对应串口
      {"serialChannel": 1,
       "data": "...",
       "dataLen": 123
      },
      // 数据 2，可以有或者没有，收到后将发送到对应串口
    ]
    // ....其他数据
  }
}
```

### 3.7.3. 截图

**支持设备：目前所有 R 系列**

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息，可触发截图：

相机会触发当前的视频截图，然后将截图数据推送到 **snapImageAbsolutelyUrl 字段指定的服务器地址**；

```
{
  "Response_AlarmInfoPlate":
  {
    "TriggerImage" :
    {
      }
```

```

//回复截图内容端口号（可选，不填则默认使用 http 页面配置端口）
"port":80,
//回复截图内容相对路径（可选，不触发截图可不添加该字段）
"snapImageRelativeUrl" : "/devicemanagement/php/receivedeviceinfo.php",
//回复截图内容绝对路径（可选，不触发截图可不添加该字段）

"snapImageAbsolutelyUrl" :"http ://192.168.1.106/devicemanagement/php/receivedeviceinfo.ph
p",
    // 回复截图对应的视频通道（仅 X2 涉及，可选，不填则默认触发车头视频截
图，0-表示拼接视频通道，1-表示车头视频通道）
    "chnlid": 1
}
// ....其他数据
}
}

```

### 3.7.4. 手动触发识别

#### 支持设备：目前所有 R 系列

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息，可触发手动识别：

```
{
  "Response_AlarmInfoPlate":
  {
    "manualTrigger" : "ok",//回复 ok 进行手动触发
    // ....其他数据
  }
}
```

或者仅回复以下数据：

```
{
  "type":"AVS_TRIGGER"
}
```

此时会触发相机的手动识别，设备会将识别结果数据推送至 http 服务端，前提是服务端配置了识别数据的推送；

设置 mqtt oss 参数配置

#### 支持设备：目前所有 R 系列

服务器在收到 comet 轮询时，回复以下结构的消息，可配置 mqtt oss 参数并连接：

```
{  
    "type": "SET_MQTT_OSS_CFG", // 轮询类型  
    "cfg": {  
        "type": "set_mqtt_cfg", // 设置协议  
        "module": "MQTT_CONFIG_ALL", // 模块  
        "body": {  
            "enable": 1, // 是否使能, 0: 不启用 1: 启用  
            "base_cfg": {  
                "server_addr": "192.168.25.167",  
                "server_port": 1883,  
                "server_username": "admin",  
                "server_pwd": "admin",  
                "clientId": "{sn}", // 可固定写"{}"字符串, 设备内部自动获取  
                "timeout": 10,  
                "heart_interval": 30,  
                "check_off_line": 0, // 是否启用脱机检测  
                "is_sn": 0, // 主题是否自带 sn  
                "mqtt_type": 1 // 1.自建 mqtt 2.阿里物联网  
            },  
            "oss_cfg": {  
                "oss_id": 1,  
                "oss_info": [  
                    {  
                        "oss_addr": "", // oss 地址  
                        "oss_username": "", // oss 用户名  
                        "oss_pwd": "admin", // oss 密码  
                        "oss_bucket": "",  
                        "oss_id": 1,  
                        "oss_key": "", // oss 访问 key  
                        "oss_secret": "", // oss 访问密钥  
                        "oss_download_addr": "" // oss 下载地址  
                    }  
                ],  
                "pic_mode": "oss",  
                "picture": {  
                    "bigpic_enable": 0, // 1.上传 0.不上传  
                    "smallpic_enable": 0 // 1.上传 0.不上传  
                }  
            }  
        }  
    }  
}
```

### 3.7.5. 白名单

#### 3.7.5.1. 白名单操作

**支持设备：目前所有 R 系列**

中心服务器在收到 HTTP 推送的识别结果时，在回复的识别结果消息中，携带白名单操作信息

**注意：单条回复消息，最多操作 5 条白名单数据；**

1. 回复消息使用 JSON 格式，格式如下：

```
{  
    "Response_AlarmInfoPlate":{  
        "//////其他数据/////////  
        "white_list_operate":{  
            "operate_type":0,  
            "reply_url":"/whitelist_replay",  
            "msg_id":0,  
            "white_list_data": [  
                {  
                    "plate":"京 A12345",  
                    "enable":1,  
                    "need_alarm":1,  
                    "enable_time":"2018-01-01 11:11:11",  
                    "overdue_time":"2018-01-01 11:11:11"  
                },  
                {  
                    "plate":"川 A12345",  
                    "enable":1,  
                    "need_alarm":1,  
                    "enable_time":"2018-01-01 11:11:11",  
                    "overdue_time":"2018-01-01 11:11:11"  
                }  
            ]  
        }  
    }  
}
```

2. 字段含义如下：

层级	字段名称	类型	取值范围	是否必须	描述
1.1	operate_type	int	[0,1]	是	操作类型(0:增加, 1: 删除)
1.2	white_list_data	无	个数 [1, 5]	是	白名单数组：单次操作，最大支持 5 条
1.3	reply_url	string	确保 / 开始的 url 格式 例如： /test	否	当改值为空的时候，设备收到白名单请求后，将发送白名单回复（3.7.10）到中心服务器
1.4	msg_id	int64	无	否	当设备回复白名单回复（3.7.10）中作为下发消息的唯一标识的字段
1.2.1	plate	string	长度 [7, 15]	是	车牌（GB2312）
1.2.2	enable	int	[0, 1]	是	当前名单是否有效（0: 无效, 1: 有效）
1.2.3	need_alarm	int	[0, 1]	是	当前名单是否为黑名单（0: 否, 1: 黑名单）
1.2.4	enable_time	string	固定格式、 固定长度： 19	否	当前名单生效时间，如： 2018-01-01 11:11:11
1.2.5	overdue_time	string	固定格式、 固定长度：	否	当前名单过期时间，如： 2018-01-01 11:11:11

层级	字段名称	类型	取 值 范 围	是 否 必 须	描述
			19		

### 3. 消息示例：

同时增加两条白名单数据（多条数据，格式一致）：

第一条没有白名单有效时间，则默认白名单永久有效；

第二条有白名单有效时间，表示当前白名单在这个时间段内有效；

```
{
  "Response_AlarmInfoPlate": {
    ...// 其他数据
    "white_list_operate": {
      "operate_type" : 0,
      "reply_url":"/whitelist_replay",
      "msg_id":0,
      "white_list_data": [
        {
          "plate": "京 A12345",
          "enable": 1,
          "need_alarm": 0
        },
        {
          "plate": "川 A12345",
          "enable": 1,
          "need_alarm": 0,
          "enable_time": "2018-01-01 11:11:11",
          "overdue_time": "2019-01-01 11:11:11"
        }
      ]
    }
  }
}
```

### 4. 删除两条白名单的回复消息示例：

```
{
```

```
"Response_AlarmInfoPlate": {  
    ...// 其他数据  
    "white_list_operate": {  
        "operate_type": 1,  
        "reply_url": "/whitelist_replay",  
        "msg_id": 0,  
        "white_list_data": [  
            [  
                {  
                    "plate": "京 A12345"  
                },  
                {  
                    "plate": "川 A12345"  
                }  
            ]  
        ]  
    }  
}
```

## 5. 删除全部白名单的回复消息示例：

```
{  
    "Response_AlarmInfoPlate": {  
        ...// 其他数据  
        "white_list_operate": {  
            "operate_type": 1,  
            "reply_url": "/whitelist_replay",  
            "msg_id": 0,  
            "white_list_data": [  
                [  
                    {  
                        "plate": ""  
                    }  
                ]  
            ]  
        }  
    }  
}
```

### 3.7.5.2. 白名单回复

支持设备：目前所有 R 系列（抓包未发现设备回复了这条消息）

设备在收到中心服务器下发的白名单操作（3.7.5.1）时候，会回复中心服务器本次操作的结果

回复消息使用 JSON 格式，格式如下：

```
{
  "white_list_replay_operate": {
    "msg_id": 0,
    "status": 200
  }
}
```

层级	字段名称	类型	取值范围	是否必须	描述
1.1	white_list_replay_operate	Object	[0,1]	是	
1.1.1	msg_id	int	无	是	中心服务器白名单操作（3.7.5.1）下发标识唯一消息的字段
1.1.2	status	int	无	是	操作结果 成功：200 失败：其他

### 3.7.6. 系统预置语音播报

支持设备：目前所有 R 系列

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息

```
{
  "Response_AlarmInfoPlate": {
    "playserver_json_request": {
      "type": "ps_voice_play",
      "voice": "JXU2QjlyJXU4RkNFJXU1MTQ5JXU0RTM0",
      "voice_interval": 0,
      "voice_volume": 100,
      "voice_male": 1
    }
  }
}
```

字段名	类型	说明
voice	String	语音信息， utf-8/GBK 编码的 BASE64 编码字符串
voice_interval	int	语音文件播放间隔
voice_volume	int	语音文件音量大小[1-100]
voice_male	int	语音类型：0 男声；1 女声
type	string	ps_voice_play 语音播放

### 3.7.7. IO 控制

支持设备：目前所有 R 系列

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息

```
{
  "Response_AlarmInfoPlate":{
    "ivs_ioctrl":{
      "delay":500,
      "io":0,
      "value":2
    }
  }
}
```

字段名	类型	说明
io	Int	Ioout 的 id[0-3] 不同产品 io 个数不一样
value	int	输出 IO 的状态值 0 断 1 通 2 先通后断
delay	int	先通后断的延迟时间(uint32，单位:ms)；当 value 为 2 的时候有效，最大 600*1000

### 3.7.8. LCD 配置

支持设备：仅 R5D

1. 当开启 comet 轮询之后，相机会一直与 HTTP 服务器进行交互，保持连接请求，相机主动发送设备注册消息，内容与普通心跳内容一致，收到回复时，立即发送下一条消息；

2. 发送设备注册消息，与普通心跳消息保持一致；
3. comet 轮询会根据服务器回复做相应处理；

### 3.7.8.1. 接口清单

序号	接口分类	接口名称
1	LCD 屏显配置接口	LCD 屏显配置
2		LCD 屏显配置获取
3	LCD 广告配置接口	半屏配置下发
4		全屏配置下发
5		自定义配置下发
6		LCD 广告配置获取
7		控制忙时半屏显示
8	LCD 屏显控制接口	控制闲时半屏显示
9		控制忙时全屏显示
10		控制闲时全屏显示

说明：

1. 文件下载接口以 HTTP 的方式提供；图片不超过 5MB，视频文件不超过 50MB；资源包整体的大小不超过 100MB；
2. 压缩包直接使用设备导出的 tar 文件，否则可能会有格式问题；

### 3.7.8.2. LCD 屏显配置

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息，（修改屏显协议、屏幕翻转和显示屏尺寸将会自动重启相机）：

```
{
  "Response_AlarmInfoPlate":{
    "type":"set_ad_voice_config",
    "module":"AD_CONFIG_REQUESTION",
    "reply_url":"",
    "body":{
      "media_voice_time_ctrl":[
        {
          "time_begin":"00:00:00",

```

```
        "time_end": "08:00:00",
        "voice_level": 2
    },
    {
        "time_begin": "08:00:00",
        "time_end": "17:00:00",
        "voice_level": 0
    },
    {
        "time_begin": "17:00:00",
        "time_end": "24:00:00",
        "voice_level": 0
    }
],
"led_bright_time_ctrl": [
    {
        "time_begin": "00:00:00",
        "time_end": "08:00:00",
        "level": 2
    },
    {
        "time_begin": "08:00:00",
        "time_end": "17:00:00",
        "level": 2
    },
    {
        "time_begin": "17:00:00",
        "time_end": "24:00:00",
        "level": 2
    }
],
"screen_size": 2,
"text_roll_speed": 1,
"led_bright_mode": 1,
"fan_enable_temprature": 40,
"control_mode": 2,
"factory_mode": 0,
"rotate_mode": 1
}
}
```

层级	参数名字	类型	必填	描述
----	------	----	----	----

1.1	<b>type</b>	string	是	操作类型
1.2	<b>module</b>	string	是	模块名
1.3	<b>reply_url</b>	string	是	回复地址，设备执行操作后，会调用该地址，返回执行结果，POST 方式
1.4	<b>body</b>	object	是	消息对象
1.4.1	<b>media_voice_time_ctrl</b>	array		媒体音量（可分三个时间段）
1.4.1.1	<b>time_begin</b>	string	是	时间
1.4.1.2	<b>time_end</b>	string	是	时间
1.4.1.3	<b>voice_level</b>	int	是	音量，0: 静音，1: 音量 1，2: 音量，3: 音量 3，4: 音量 4，5: 音量 5
1.4.2	<b>led_bright_time_ctrl</b>	array	是	LCD 屏显亮度（可分三个时间段）
1.4.2.1	<b>time_begin</b>	string	是	时间
1.4.2.2	<b>time_end</b>	string	是	时间
1.4.2.3	<b>level</b>	int	是	亮度，0: 熄屏，1: 亮度 1，2: 亮度 2，3: 亮度 3，4: 亮度 4，5: 亮度 5
1.4.3	<b>screen_size</b>	int	是	屏幕尺寸，1: 1366*768, 2: 1920*1080
1.4.4	<b>text_roll_speed</b>	int	是	文字滚动速度，0: 缓慢，1: 正常，2: 快速
1.4.5	<b>led_bright_mode</b>	int	是	LCD 屏幕亮度模式，0: 智能调节，1: 手动调节，2: 关闭
1.4.6	<b>fan_enable_temprature</b>	int	是	开启温度
1.4.7	<b>control_mode</b>	int	是	屏显控制模式，0: 软件控制，1: 相机控制，2: 脱机时相机控制

1.4.8	factory_mode	int	是	屏显协议, 0: A(VZ), 1: B(AK); 填 0 即可;
1.4.9	rotate_mode	int	是	屏幕翻转, 0: 原始显示, 1: 上下翻转

### 3.7.8.3. LCD 屏显配置获取

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息：

```
{
  "Response_AlarmInfoPlate": {
    "type": "get_ad_voice_config",
    "module": "AD_CONFIG_REQUESTION",
    "reply_url": "http://192.168.18.55/reply_url"
  }
}
```

参数名字	类型	最大长度	必填	描述
type	string	32	是	操作类型
module	string	32	是	模块名字
reply_url	string	128	是	推送地址, POST 方式

推送内容如下：

```
{
  "code": 0,
  "result": {
    "type": "get_ad_voice_config",
    "module": "AD_CONFIG_REQUESTION",
    "body": {
      "media_voice_time_ctrl": [
        {
          "time_begin": "00:00:00",
          "time_end": "08:00:00",
          "voice_level": 2
        },
        {
          "time_begin": "08:00:00",
          "time_end": "23:59:59",
          "voice_level": 1
        }
      ]
    }
  }
}
```

```
        "time_end":"17:00:00",
        "voice_level":0
    },
    {
        "time_begin":"17:00:00",
        "time_end":"24:00:00",
        "voice_level":0
    }
],
"led_bright_time_ctrl":[
    {
        "time_begin":"00:00:00",
        "time_end":"08:00:00",
        "level":2
    },
    {
        "time_begin":"08:00:00",
        "time_end":"17:00:00",
        "level":2
    },
    {
        "time_begin":"17:00:00",
        "time_end":"24:00:00",
        "level":2
    }
],
"screen_size":2,
"text_roll_speed":1,
"led_bright_mode":1,
"fan_enable_temprature":40,
"control_mode":2,
"factory_mode":0,
"rotate_mode":1
}
}
```

### 3.7.8.4. 半屏配置下发

屏幕配置方式有三种，半屏、全屏、自定义，采用自定义方式时，只能通过压缩包的方式导

入；

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息：

```
{  
    "Response_AlarmInfoPlate":{  
        "type":"set_ad_config",  
        "module":"AD_CONFIG_REQUESTION",  
        "reply_url": "",  
        "body":{  
            "ad_type":2,  
            "package_url":"http://192.168.18.77:9090/shareFile/res.tar",  
            "ad_config":{  
                "ad_source":"local",  
                "ad_mode":2,  
                "ad_group":{  
                    "group_id":0,  
                    "scenes": [  
                        {  
                            "scene_name":"YnVzeV9oYWxm",  
                            "scene_id":0,  
                            "scene_mode":2,  
                            "scene_type":"car_out",  
                            "scene_max_duration":30,  
                            "video_play_count":2,  
                            "background_type":1,  
                            "background_color": "",  
                            "background_image":"1.png",  
                            "scene_pos":{  
                                "x":0,  
                                "y":307,  
                                "width":360,  
                                "height":333  
                            },  
                            "scene_info":{  
                                "elem_list": [  
                                    {  
                                        "elem_id":1,  
                                        "elem_enable":1,  
                                        "elem_name":"dynamic_text1",  
                                        "elem_type":"dynamic_text",  
                                        "elem_sub_type":"car_type",  
                                        "elem_pos":{  
                                            "x":0,  
                                            "y":47,  
                                            "width":360,  
                                            "height":333  
                                        }  
                                    }  
                                ]  
                            }  
                        }  
                    ]  
                }  
            }  
        }  
    }  
}
```

```
        "height":28
    },
    "elem_text_content":"6Z2Z5oCB5paH5pys",
    "elem_bg_color":"rgba(255,255,255,0)",
    "elem_fg_color":"#941594",
    "elem_font_family":0,
    "elem_font_bold":0,
    "elem_font_size":28,
    "elem_align":"center",
    "extern_text":"",
    "extern_text_enable":0
},
{
    "elem_id":2,
    "elem_enable":1,
    "elem_name":"dynamic_text2",
    "elem_type":"dynamic_text",
    "elem_sub_type":"car_id",
    "elem_pos":{
        "x":0,
        "y":122,
        "width":360,
        "height":24
    },
    "elem_text_content":"6Z2Z5oCB5paH5pys",
    "elem_bg_color":"rgba(255,255,255,0)",
    "elem_fg_color":"#8f388f",
    "elem_font_family":0,
    "elem_font_bold":0,
    "elem_font_size":24,
    "elem_align":"center",
    "extern_text":"",
    "extern_text_enable":0
},
{
    "elem_id":3,
    "elem_enable":1,
    "elem_name":"dynamic_text3",
    "elem_type":"dynamic_text",
    "elem_sub_type":"custom",
    "elem_pos":{
        "x":0,
        "y":194,
        "width":360,
```

```
        "height":24
    },
    "elem_text_content":"6Z2Z5oCB5paH5pys",
    "elem_bg_color":"rgba(255,255,255,0)",
    "elem_fg_color":"#c777c7",
    "elem_font_family":0,
    "elem_font_bold":0,
    "elem_font_size":24,
    "elem_align":"center",
    "extern_text":"",
    "extern_text_enable":0
},
{
    "elem_id":4,
    "elem_enable":1,
    "elem_name":"dynamic_text4",
    "elem_type":"dynamic_text",
    "elem_sub_type":"car_in_time",
    "elem_pos":{
        "x":0,
        "y":265,
        "width":360,
        "height":20
    },
    "elem_text_content":"6Z2Z5oCB5paH5pys",
    "elem_bg_color":"rgba(255,255,255,0)",
    "elem_fg_color":"#5c1b5c",
    "elem_font_family":0,
    "elem_font_bold":0,
    "elem_font_size":20,
    "elem_align":"center",
    "extern_text":"",
    "extern_text_enable":0
},
{
    "elem_id":5,
    "elem_enable":0,
    "elem_name":"dynamic_text5",
    "elem_type":"dynamic_text",
    "elem_sub_type":"car_in_time",
    "elem_pos":{
        "x":0,
        "y":319,
        "width":360,
```

```
        "height":14
    },
    "elem_text_content":"6Z2Z5oCB5paH5pys",
    "elem_bg_color":"rgba(255,255,255,0)",
    "elem_fg_color":"#ffffff",
    "elem_font_family":0,
    "elem_font_bold":0,
    "elem_font_size":14,
    "elem_align":"center",
    "extern_text":"",
    "extern_text_enable":0
}
]
}
},
{
    "scene_name":"ZnJlZV9oYWxm",
    "scene_id":1,
    "scene_mode":2,
    "scene_type":"idle",
    "scene_max_duration":30,
    "video_play_count":2,
    "background_type":1,
    "background_color":"",
    "background_image":"0_MA2435.jpg",
    "scene_pos":{
        "x":0,
        "y":307,
        "width":360,
        "height":333
    },
    "scene_info":{
        "elem_list":[
            {
                "elem_id":4,
                "elem_enable":1,
                "elem_name":"dynamic_text4",
                "elem_type":"dynamic_text",
                "elem_sub_type":"parking_space_left",
                "elem_pos":{
                    "x":0,
                    "y":72,
                    "width":360,
                    "height":14
                }
            }
        ]
    }
}
```

```
        },
        "elem_text_content":"6Z2Z5oCB5paH5pys",
        "elem_bg_color":"rgba(255,255,255,0)",
        "elem_fg_color":"#ffffff",
        "elem_font_family":0,
        "elem_font_bold":0,
        "elem_font_size":14,
        "elem_align":"center",
        "extern_text":"",
        "extern_text_enable":1
    },
    {
        "elem_id":5,
        "elem_enable":1,
        "elem_name":"dynamic_text5",
        "elem_type":"dynamic_text",
        "elem_sub_type":"parking_space_left",
        "elem_pos":{
            "x":0,
            "y":159,
            "width":360,
            "height":14
        },
        "elem_text_content":"6Z2Z5oCB5paH5pys",
        "elem_bg_color":"rgba(255,255,255,0)",
        "elem_fg_color":"#ffffff",
        "elem_font_family":0,
        "elem_font_bold":0,
        "elem_font_size":14,
        "elem_align":"center",
        "extern_text":"",
        "extern_text_enable":1
    },
    {
        "elem_id":6,
        "elem_enable":1,
        "elem_name":"dynamic_text6",
        "elem_type":"dynamic_text",
        "elem_sub_type":"parking_space_left",
        "elem_pos":{
            "x":0,
            "y":246,
            "width":360,
            "height":14
        }
    }
}
```

```
        },
        "elem_text_content":"",
        "elem_bg_color":"rgba(255,255,255,0)",
        "elem_fg_color":"#d117d1",
        "elem_font_family":0,
        "elem_font_bold":0,
        "elem_font_size":14,
        "elem_align":"center",
        "extern_text":"NjY2",
        "extern_text_enable":0
    }
]
}
},
{
    "scene_name":"YWQ=",
    "scene_id":2,
    "scene_mode":1,
    "scene_type":"ad",
    "scene_max_duration":30,
    "video_play_count":2,
    "background_type":2,
    "background_color":"",
    "background_image":"",
    "scene_pos":{
        "x":0,
        "y":0,
        "width":360,
        "height":307
    },
    "scene_info":{
        "elem_list":[
            {
                "elem_id":0,
                "elem_name":"video0",
                "elem_type":"video",
                "elem_pos":{
                    "x":0,
                    "y":0,
                    "width":360,
                    "height":307
                },
                "picture_max_duration":1,
                "video_play_count":1,
            }
        ]
    }
}
```

```
        "elem_res_list": [
            {
                "path": "2233_hisi.mp4"
            }
        ],
        {
            "elem_id": 1,
            "elem_name": "static_pic1",
            "elem_type": "static_pic",
            "elem_pos": {
                "x": 0,
                "y": 0,
                "width": 360,
                "height": 307
            },
            "picture_max_duration": 10,
            "video_play_count": 10,
            "elem_res_list": [
                {
                    "path": "89756.jpg"
                }
            ]
        },
        {
            "elem_id": 2,
            "elem_name": "video2",
            "elem_type": "video",
            "elem_pos": {
                "x": 0,
                "y": 0,
                "width": 360,
                "height": 307
            },
            "picture_max_duration": 1,
            "video_play_count": 1,
            "elem_res_list": [
                {
                    "path": "lego_hisi.mp4"
                }
            ]
        },
        {
            "elem_id": 3,
```

```
        "elem_name":"video3",
        "elem_type":"video",
        "elem_pos":{
            "x":0,
            "y":0,
            "width":360,
            "height":307
        },
        "picture_max_duration":1,
        "video_play_count":1,
        "elem_res_list":[
            {
                "path":"iphone1_mp4.mp4"
            }
        ]
    }
},
{
    "scene_name":"cXJjb2RI",
    "scene_id":3,
    "scene_mode":1,
    "scene_type":"qr_code",
    "scene_max_duration":30,
    "video_play_count":2,
    "background_type":0,
    "background_color":"rgba(255,255,255,1)",
    "background_image":"",
    "scene_pos":{
        "x":0,
        "y":0,
        "width":360,
        "height":307
    },
    "scene_info":{
        "elem_list":[
            {
                "elem_id":6,
                "elem_enable":1,
                "elem_name":"static_pic6",
                "picture_max_duration":300,
                "elem_type":"dynamic_pic",
                "elem_sub_type":"fee-qrcode",
            }
        ]
    }
}
```

```
"elem_pos":{  
    "x":36,  
    "y":10,  
    "width":287,  
    "height":287  
},  
"elem_bg_color":"rgba(255,255,255,0)",  
"elem_res_list": [  
    {  
        "path":""  
    }  
],  
"elem_res_path":"https://img.zcool.cn/community/015c635a51eafea8012180c5e03fa5.gif"  
}  
}  
]  
}  
}  
}  
]  
}  
}  
}  
}  
}  
}  
}  
}  
}
```

层级	参数名字	类型	最大长度	必填	描述
1.1	type	string	32	是	操作类型
1.2	module	string	32	是	模块名字
1.3	reply_url	string	255	是	回复地址, POST 方式
1.4	body	object		是	消息对象
1.4.2	package_url	string	255	否	压缩包路径, 使用自定义方式下发配置的时候, 需要有此字段
1.4.3	ad_type	int		是	0:自定义广告, 1:全屏广告, 2:半屏广告

1.4.4	ad_config				用于广告配置，图片、视频文件，需要通过 tar 压缩包（用设备导出，或保持与设备导出的压缩包格式相同）下发
1.4.4.1	ad_source	string	32	是	资源位置，填"local"即可；
1.4.4.2	ad_mode	int		是	屏显模式，1：全屏，2：半屏
1.4.4.3	ad_group	object		是	
1.4.4.3.1	group_id	int		是	分组 id
1.4.4.3.2	scenes	array	5	是	场景数组
1.4.4.3.2.1	scene_name	string	64	是	场景名
1.4.4.3.2.2	scene_id	int		是	场景 id
1.4.4.3.2.3	scene_mode	int		是	场景模式，预留，填 0 即可
1.4.4.3.2.4	scene_type	string	32	是	场景类型
1.4.4.3.2.5	scene_max_duration	int		是	场景最大播放时长，单位秒
1.4.4.3.2.6	video_play_count	int		是	视频播放次数
1.4.4.3.2.7	background_type	int		是	背景类型，0：背景色，1：背景图，2：不启用
1.4.4.3.2.8	background_color	string	32	是	背景色，eg: "#941594"
1.4.4.3.2.9	background_image	string	96	是	背景图片名
1.4.4.3.2.10	scene_pos	object		是	位置
1.4.4.3.2.10.1	x	int		是	x 轴坐标
1.4.4.3.2.10.2	y	int		是	y 轴坐标
1.4.4.3.2.10.3	width	int		是	宽度
1.4.4.3.2.10.4	height	int		是	高度
1.4.4.3.2.11	scene_info	object			场景资源，文本和视频、图像 是不同的数据结构（数组中可能 存在不同类型的对象）
文本					

1.4.4.3.2.11.1	elem_list	array	10	否	文本
1.4.4.3.2.11.1.1	elem_id	int		是	元素 id
1.4.4.3.2.11.1.2	elem_enable	int		是	是否启用
1.4.4.3.2.11.1.3	elem_name	string	32	是	元素名字
1.4.4.3.2.11.1.4	elem_type	string	16	是	元素类型 static_pic: 静态图片, dynamic_pic: 动态图片, video: 视频, static_text: 静态文字, dynamic_text: 动态文字
1.4.4.3.2.11.1.5	elem_sub_type	string	32	是	元素子类型, car_id: 车牌号, datetime : 日期时间, car_in_time: 入场时间, car_out_time: 离场时间, car_type : 车类型, parking_space_left: 剩余车位 / , custom : 自定义, fee-qrcode: 动态二维码
1.4.4.3.2.11.1.6	elem_pos	object		是	位置
1.4.4.3.2.11.1.6.1	x	int		是	x 轴坐标
1.4.4.3.2.11.1.6.2	y	int		是	y 轴坐标
1.4.4.3.2.11.1.6.3	width	int		是	宽度
1.4.4.3.2.11.1.6.4	height	int		是	高度
1.4.4.3.2.11.1.7	elem_text_content	string	96	是	文字内容
1.4.4.3.2.11.1.8	elem_bg_color	string	32	是	背景色
1.4.4.3.2.11.1.9	elem_fg_color	string	32	是	文字颜色
1.4.4.3.2.11.1.10	elem_font_family	int		是	字体
1.4.4.3.2.11.1.11	elem_font_bold	int		是	是否加粗
1.4.4.3.2.11.1.12	elem_font_size	int		是	字号
1.4.4.3.2.11.1.13	elem_align	string	16	是	对齐方式
1.4.4.3.2.11.1.14	extern_text	string	96	是	扩展文字

1.4.4.3.2.11.1.15	extern_text_enable	int		是	是否启用扩展文字
视频、图片					
1.4.4.3.2.11.1	elem_list	array		是	图片、视频
1.4.4.3.2.11.2	elem_id	int		是	元素 id
1.4.4.3.2.11.3	elem_enable	int		是	是否启用
1.4.4.3.2.11.4	elem_name	string		是	元素名称
1.4.4.3.2.11.5	elem_type	string		是	元素类型 static_pic: 静态图片, dynamic_pic: 动态图片, video: 视频, static_text: 静态文字, dynamic_text: 动态文字
1.4.4.3.2.11.6	elem_sub_type	string		是	元素子类型, car_id: 车牌号, datetime : 日期时间 , car_in_time : 入场时间 , car_out_time : 离场时间 , car_type : 车类型 , parking_space_left: 剩余车位 / , custom : 自定义 , fee-qrcode: 动态二维码
1.4.4.3.2.11.7	elem_pos	object		是	元素字类型
1.4.4.3.2.11.7.1	x	int		是	x 轴坐标
1.4.4.3.2.11.7.2	y	int		是	y 轴坐标
1.4.4.3.2.11.7.3	width	int		是	宽度
1.4.4.3.2.11.7.4	height	int		是	高度
1.4.4.3.2.11.8	picture_max_duration	int		是	图片场景最长显示时间, 单位秒, 范围: 3~30
1.4.4.3.2.11.9	video_play_count	int		是	播放次数, 1~10
1.4.4.3.2.11.10	elem_res_list	array		是	文件路径列表
1.4.4.3.2.11.10.1	path	string		是	文件路径

场景名	base64 编码	必填	资源数组类型
busy_half	YnVzeV9oYWxm	是	文本

free_half	ZnJlZV9oYWxm	是	文本
ad	YWQ=	是	视频、图片
qrcode	cXJjb2Rl	是	图片

响应格式需严格按照以上的场景格式；

### 3.7.8.5. LCD 广告配置获取

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息：

```
{
    "Response_AlarmInfoPlate":{
        "type":"get_ad_config",
        "module":"AD_CONFIG_REQUESTION",
        "reply_url":"",
        "body":{
            "ad_type":2
        }
    }
}
```

推送格式：

具体内容和 lcd 配置一致；

```
{
    "code":0,
    "msg":"",
    "result":{
        "ad_config":{

        },
        "ad_type":0
    }
}
```

### 3.7.8.6. 全屏配置下发

json 的定义与半屏配置相同，但是包含的场景数组有区别；

场景名	base64 编码	必填	资源数组类型
-----	-----------	----	--------

full	ZnVsbA==	是	文本+图片
ad	YWQ=	是	图片、视频

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息：

```
{
  "Response_AlarmInfoPlate":{
    "type":"set_ad_config",
    "module":"AD_CONFIG_REQUESTION",
    "reply_url":"",
    "body":{
      "ad_type":1,
      "package_url":"http://192.168.18.77:9090/shareFile/res.tar",
      "ad_config":{
        "ad_source":"local",
        "ad_mode":1,
        "ad_group":{
          "group_id":0,
          "scenes":[
            {
              "scene_name":"ZnVsbA==",
              "scene_id":0,
              "scene_mode":0,
              "scene_type":"car_out",
              "scene_max_duration":30,
              "video_play_count":2,
              "background_type":1,
              "background_color":"",
              "background_image":"hei.jpg",
              "scene_pos":{
                "x":0,
                "y":0,
                "width":360,
                "height":640
              },
              "scene_info":{
                "elem_list":[
                  {
                    "elem_id":1,
                    "elem_enable":1,
                    "elem_name":"dynamic_text1",
                    "elem_type":"dynamic_text",
                    "elem_sub_type":"car_type",
                    "elem_pos":{


```

```
        "x":0,
        "y":47,
        "width":360,
        "height":14
    },
    "elem_text_content":"6Z2Z5oCB5paH5pys",
    "elem_bg_color":"rgba(255,255,255,0)",
    "elem_fg_color":"#000000",
    "elem_font_family":0,
    "elem_font_bold":0,
    "elem_font_size":14,
    "elem_align":"center",
    "extern_text":"",
    "extern_text_enable":0
},
{
    "elem_id":2,
    "elem_enable":1,
    "elem_name":"dynamic_text2",
    "elem_type":"dynamic_text",
    "elem_sub_type":"car_id",
    "elem_pos":{
        "x":0,
        "y":108,
        "width":360,
        "height":14
    },
    "elem_text_content":"6Z2Z5oCB5paH5pys",
    "elem_bg_color":"rgba(255,255,255,0)",
    "elem_fg_color":"#000000",
    "elem_font_family":0,
    "elem_font_bold":0,
    "elem_font_size":14,
    "elem_align":"center",
    "extern_text":"",
    "extern_text_enable":0
},
{
    "elem_id":3,
    "elem_enable":1,
    "elem_name":"dynamic_text3",
    "elem_type":"dynamic_text",
    "elem_sub_type":"car_in_time",
    "elem_pos":{
```

```
        "x":0,
        "y":169,
        "width":360,
        "height":14
    },
    "elem_text_content":"6Z2Z5oCB5paH5pys",
    "elem_bg_color":"rgba(255,255,255,0)",
    "elem_fg_color":"#000000",
    "elem_font_family":0,
    "elem_font_bold":0,
    "elem_font_size":14,
    "elem_align":"center",
    "extern_text":"",
    "extern_text_enable":0
},
{
    "elem_id":4,
    "elem_enable":1,
    "elem_name":"dynamic_text4",
    "elem_type":"dynamic_text",
    "elem_sub_type":"custom",
    "elem_pos":{
        "x":0,
        "y":230,
        "width":360,
        "height":14
    },
    "elem_text_content":"6Z2Z5oCB5paH5pys",
    "elem_bg_color":"rgba(255,255,255,0)",
    "elem_fg_color":"#000000",
    "elem_font_family":0,
    "elem_font_bold":0,
    "elem_font_size":14,
    "elem_align":"center",
    "extern_text":"",
    "extern_text_enable":0
},
{
    "elem_id":5,
    "elem_enable":1,
    "elem_name":"dynamic_text5",
    "elem_type":"dynamic_text",
    "elem_sub_type":"custom",
    "elem_pos":{
```

```
        "x":0,
        "y":291,
        "width":360,
        "height":14
    },
    "elem_text_content":"6Z2Z5oCB5paH5pys",
    "elem_bg_color":"rgba(255,255,255,0)",
    "elem_fg_color":"#000000",
    "elem_font_family":0,
    "elem_font_bold":0,
    "elem_font_size":14,
    "elem_align":"center",
    "extern_text":"",
    "extern_text_enable":0
},
{
    "elem_id":6,
    "elem_enable":1,
    "elem_name":"static_pic6",
    "picture_max_duration":300,
    "elem_type":"dynamic_pic",
    "elem_sub_type":"fee-qrcode",
    "elem_pos":{
        "x":54,
        "y":374,
        "width":252,
        "height":252
    },
    "elem_bg_color":"rgba(255,255,255,0)",
    "elem_res_list":[
        {
            "path": ""
        }
    ],
    "elem_res_path": ""
}
]
}
{
    "scene_name":"YWQ=",
    "scene_id":1,
    "scene_mode":0,
    "scene_type":"ad",
```

```
"scene_max_duration":30,
"video_play_count":2,
"background_type":2,
"background_color":"",
"background_image":"",
"scene_pos":{
    "x":0,
    "y":0,
    "width":360,
    "height":640
},
"scene_info":{
    "elem_list":[
        {
            "elem_id":0,
            "elem_name":"video0",
            "elem_type":"video",
            "elem_pos":{
                "x":0,
                "y":0,
                "width":360,
                "height":640
            },
            "picture_max_duration":1,
            "video_play_count":1,
            "elem_res_list":[
                {
                    "path":"2233_hisi.mp4"
                }
            ]
        },
        {
            "elem_id":1,
            "elem_name":"static_pic1",
            "elem_type":"static_pic",
            "elem_pos":{
                "x":0,
                "y":0,
                "width":360,
                "height":640
            },
            "picture_max_duration":10,
            "video_play_count":10,
            "elem_res_list":[]
```

```
{
    "path":"bg_bottom.png"
}
]
}
]
},
"ratio":48
}
}
}
}
```

### 3.7.8.7. 自定义配置下发

与半屏和全屏下发的区别是，可以导入完整的广告包，使用更便捷；

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息：

```
{
  "Response_AlarmInfoPlate":{

    "type":"set_ad_config",

    "module":"AD_CONFIG_REQUESTION",

    "reply_url":"",
    "body":{

      "ad_type":0,
      "package_url":"http://192.168.18.77:9090/shareFile/res.tar"

    }
  }
}
```

请求结果会往 reply 中的地址返回，目前请求内容符合要求都返回 code 为 0，后续会拓展不同请求结果值 code 和提示信息 msg；

```
{"code":0,"msg":"","result":null}
```

### 3.7.8.8. 控制忙时半屏显示

半屏模式忙时文字只能显示在下半屏，上半屏显示二维码。场景有全屏、半屏，忙时和闲时。两两组合：全屏忙时(full)，全屏闲时(full\_ad)，半屏忙时(busy\_half)，半屏闲时(free\_half)；后面 4 条都是在广告在页面配置好的条件下进行控制。

透传数据中的字段，需要和广告配置中的字段一致，才能控制。自定义文字可以配置多个，其他的只能配置一个。

可以控制的类型包括：car\_id 车牌号 / car\_in\_time 入场时间 / car\_out\_time 离场时间 / car\_type 车类型 / parking\_space\_left 剩余车位 / custom 自定义 / qrcode\_text 二维码。

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息：

```
{
  "Response_AlarmInfoPlate": {
    "module": "AD_CONFIG_REQUESTION",
    "type": "ad_push_message",
    "reply_url": "",
    "body": {
      "scene_name": "YnVzeV9oYWxm",
      "custom": ["5Ymp5L2Z6L2m5L2NOiAzMA==", "5Ymp5L2Z6L2m5L2NOiAzMA=="],
      "qrcode_text": "https://www.vzicar.com"
    }
  }
}
```

层级	参数名字	类型	最大长度	必填	描述
1.1	type	string	32	是	操作类型
1.2	module	string	32	是	模块名字
1.3	reply_url	string	128	是	推送地址
1.4	body	object		是	消息体
1.4.1	scene_name	string	32	是	要控制的场景名称 base64 编码， ZnJlZV9oYWxm(free_half)
1.4.2	custom	array		是	自定义文字 utf-8 base64,5Ymp5L2Z6L2m5L2NOiAzMA== (剩余车位: 30) 数组跟网页配置一一对应。比如配置三个就显示前三行
1.4.3	qrcode_text	string	128	是	动态二维码

回复示例：

```
{
  "code":0,
  "msg":"",
}
```

### 3.7.8.9. 控制闲时半屏显示

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息：

```
{
  "Response_AlarmInfoPlate": {
    "module": "AD_CONFIG_REQUEST",
    "type": "ad_push_message",
    "reply_url": "",
    "body": {
      "scene_name": "ZnJlZV9oYWxm",
      "custom": ["5Ymp5L2Z6L2m5L2NOiAzMA==", "5Ymp5L2Z6L2m5L2NOiAzMA=="],
    }
  }
}
```

层级	参数名字	类型	最大长度	必填	描述
	type	string	32	是	操作类型
	module	string	32	是	模块名字
	reply_url	string	128	是	推送地址
	body	object		是	消息体
	scene_name	string	32	是	要控制的场景名称 base64 编码， ZnJlZV9oYWxm(free_half)
	custom	array		是	自 定 义 文 字 utf-8 base64,5Ymp5L2Z6L2m5L2NOiAzMA== (剩余车位: 30) 数组跟网页配置一一 对应。比如配置三个就显示前三行

回复示例：

```
{
  "code":0,
  "msg": ""
}
```

### 3.7.8.10. 控制忙时全屏显示

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息：

```
{
  "Response_AlarmInfoPlate": {
    "module": "AD_CONFIG_REQUESTION",
    "type": "ad_push_message",
    "reply_url": "",
    "body": {
      "scene_name": "YnVzeV9oYWxm",
      "custom": ["5Ymp5L2Z6L2m5L2NOiAzMA==", "5Ymp5L2Z6L2m5L2NOiAzMA=="],
      "qrcode_text": "https://www.vzicar.com"
    }
  }
}
```

层级	参数名字	类型	最 大 长 度	必 填	描 述
1.1	type	string	32	是	操作类型
1.2	module	string	32	是	模块名字
1.3	reply_url	string	128	是	推送地址
1.4	body	object		是	消息体
1.4.1	scene_name	string	32	是	场景名
1.4.2	parking_space_left	string	128	是	剩余车位 utf-8 base64 编码 5Ymp5L2Z6L2m5L2NOiAzMA== (剩余车位: 30)
1.4.3	qrcode_text	string	128	是	动态二维码内容

回复示例：

```
{
```

```

"code":0,
"msg":"",
}

```

### 3.7.8.11. 控制闲时全屏显示

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息：

```

{
  "Response_AlarmInfoPlate": {
    "module": "AD_CONFIG_REQUEST",
    "type": "ad_push_message",
    "reply_url": "",
    "body": {
      "scene_name": "ZnVsbF9hZA=="
    }
  }
}

```

层级	参数名字	类型	最大长度	必填	描述
1.1	type	string	32	是	操作类型
1.2	module	string	32	是	模块名字
1.3	reply_url	string	128	是	推送地址
1.4	body	object		是	消息体
1.4.1	scene_name	string	32	是	场景名

回复示例：

```

{
  "code":0,
  "msg":"",
}

```

### 3.7.9. OSD 配置

#### 3.6.9.1 设置自定义 OSD

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息：

```
{  
    "Response_AlarmInfoPlate": {  
        "set_osd_para": "http_set_osd_para",  
        "body": {  
            "osd_type": 1,  
            "user_osd": {  
                "user_osd_param": [ {  
                    "id": 0,  
                    "display": 1,  
                    "color": 0,  
                    "front_size": 0,  
                    "text": "ZG15MQ=="  
                }, {  
                    "id": 1,  
                    "display": 1,  
                    "color": 0,  
                    "front_size": 1,  
                    "text": "ZG15Mg=="  
                }, {  
                    "id": 2,  
                    "display": 1,  
                    "color": 0,  
                    "front_size": 2,  
                    "text": "ZG15Mw=="  
                }, {  
                    "id": 3,  
                    "display": 1,  
                    "color": 0,  
                    "front_size": 3,  
                    "text": "ZG15NA=="  
                } ],  
                "x_pos": 7,  
                "y_pos": 46  
            },  
            "usr_multi_text": {  
                "display": 1,  
                "color": 0,  
            }  
        }  
    }  
}
```

```
        "front_size": 3,
        "x_pos": 36,
        "y_pos": 28,
        "texts": [
            {
                "enable": 1,
                "text": "ZG15NS0x"
            },
            {
                "enable": 1,
                "text": "ZG15NS0y"
            },
            {
                "enable": 1,
                "text": "ZG15NS0z"
            },
            {
                "enable": 0,
                "text": "T1NEIFR1eHQ="
            }
        ]
    }
}
```

层级	参数名字	类型	最 大 长 度	必填	描述
1.1	set_osd_para	string	32	是	操作类型
1.2	body	object		是	消息体
1.2.1	user_osd	object		是	自定义 osd
1.2.1.1	user_osd_param	array		是	osd 参数列表
1.2.1.1.1	id	int		是	osd 行数, 0~3

1.2.1.1.2	display	int		是	是否显示, 0: 不显示, 1: 显示
1.2.1.1.3	color	int		是	颜色, 0: 白, 1: 红, 2: 蓝, 3: 绿
1.2.1.1.3	front_size	int		是	字体大小
1.2.1.1.4	text	string		是	文字的 base64 编码 (编码前最多 30 个汉字, 英文最多 60 个字母)
1.2.1.2	x_pos	int		是	左上角横坐标 0~100
1.2.1.3	y_pos	int		是	左上角纵坐标 0~100
1.2.1.1	usr_multi_text	object		否	只有高速版本有效
1.2.1.1.1	display	int		是	是否显示, 0: 不显示, 1: 显示
1.2.1.1.2	color	int		是	颜色, 0: 白, 1: 红, 2: 蓝, 3: 绿
1.2.1.1.3	front_size	int		是	字体大小, 从小到大 0~3
1.2.1.1.4	x_pos	int		是	
1.2.1.1.5	y_pos	int		是	
1.2.1.1.6	texts	array	8	是	自定义内容数组
1.2.1.1.6.1	enable	int		是	是否显示
1.2.1.1.6.2	context	string		是	文字的 base64 编码 (编码前最多 30 个汉字, 英文最多 60 个字母)
1.2.2	osd_type	int		是	0: 动态叠加自

					定义 osd，叠加 内容不存 flash 1: 静态叠加自 定义 osd，叠加 内容存 flash
--	--	--	--	--	---

### 3.6.9.2 设置图片 OSD (E5 1.0)

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息：

```
{
  "Response_AlarmInfoPlate": {
    "module": "AVS_PIC OSD_MODULE",
    "type": "AVS_SET_PIC OSD_PRM",
    "is_dynamic": 0,
    "body": {
      "pos": 2,
      "font": {
        "color": 0,
        "size": 2
      },
      "separation": 0,
      "content": [
        {
          "id": 0,
          "enable": 1,
          "x_pos": 2,
          "y_pos": 64,
          "configJson": {
            ...
          }
        },
        {
          "id": 1,
          "enable": 1,
          "x_pos": 2,
          "y_pos": 71,
          "configJson": {
            ...
          }
        },
        {
          "id": 2,
          ...
        }
      ],
      ...
    }
  }
}
```

```
        "enable": 1,
        "x_pos": 2,
        "y_pos": 86,
        "configJson": {
            "type": 0, // 时间格式, 0: 12 小时制; 1: 24 小时制
            "ms_enable": 1 //是否精确到毫秒
        }
    },
    {
        "id": 3,
        "enable": 1,
        "x_pos": 2,
        "y_pos": 78,
        "configJson": {
            "text": "YWRkcmVzczw=="
        }
    },
    {
        "id": 4,
        "enable": 0,
        "x_pos": 0,
        "y_pos": 0,
        "configJson": {
        }
    },
    {
        "id": 5,
        "enable": 1,
        "x_pos": 54,
        "y_pos": 35,
        "configJson": {
        }
    },
    {
        "id": 6,
        "enable": 1,
        "x_pos": 55,
        "y_pos": 52,
        "configJson": {
        }
    },
}
```

```
{  
    "id": 7,  
    "enable": 1,  
    "x_pos": 55,  
    "y_pos": 43,  
    "configJson": {  
        "  
    }  
},  
{  
    "id": 8,  
    "enable": 1,  
    "x_pos": 54,  
    "y_pos": 3,  
    "configJson": {  
        "text": "ZG15MQ=="  
    }  
},  
{  
    "id": 9,  
    "enable": 1,  
    "x_pos": 54,  
    "y_pos": 13,  
    "configJson": {  
        "text": "ZG15Mg=="  
    }  
},  
{  
    "id": 10,  
    "enable": 1,  
    "x_pos": 54,  
    "y_pos": 22,  
    "configJson": {  
        "text": "ZG15Mw=="  
    }  
}  
],  
"content_order": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
}  
}  
}
```

层级	参数名称	参数类型	长度	是否必须	描述
----	------	------	----	------	----

1. 1	module	string		是	模块名字
1. 2	cmd	string		是	控制命令
1. 3	type	string		是	消息类型
1. 4	is_dynamic	int		否	0: 静态图片 OSD 信息叠加, 写入 flash; 1: 动态平台 OSD 信息叠加, 不写 flash; 无此字段时默认为静态 OSD 叠加
1. 5	body	object		是	消息体
1. 5. 1	pos	int		是	叠加位置, 0: 图片外上, 1: 图片外下, 2: 图片内, 3: 关闭
1. 5. 2	font	object		是	字体
1. 5. 2. 1	color	int		是	字体颜色, 0: 白色, 1: 红色, 2: 蓝色, 3: 绿色
1. 5. 2. 2	size	int		是	字体大小, 0: 16, 1: 24, 2: 32, 3: 48
1. 5. 3	separation	int		是	0: 空格; 1: 竖线 (扩张区域显示时生效)
1. 5. 4	content	array		是	
1. 5. 4. 1	id	int		是	0: 实时结果, 1: 日期, 2: 时间, 3: 地点, 4: 车辆事件(部分支持), 5: 车辆类型, 6: 车标车款, 7: 车身颜色, 8: 自定义内容 1, 9: 自定义内容 2, 10: 自定义内容 3
1. 5. 4. 2	x_pos	int		是	水平位置, 0~100
1. 5. 4. 3	y_pos	int		是	垂直位置, 0~100
1. 5. 4. 4	configJson	object		是	
1. 5. 4. 5	type	int		否	与 id 值有关, 见示例
1. 5. 4. 6	text	string		否	文本的 base64 编码, 编码前最多 30 个汉字或者 60 个英文
1. 5. 4. 7	ms_enable	int		否	精确到毫秒
1. 5. 5	content_order	array		是	根据 id 排序 (扩张区域显示时生效)

### 3.6.9.3 设置图片 OSD (DV500 版本)

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息：

```
{  
    "Response_AlarmInfoPlate": {  
        "type": "AVS_SET_PIC OSD_PRM",  
        "module": "AVS_PIC OSD_MODULE",  
        "body": {  
            "content": [  
                {  
                    "configJson": {  
                        "enable": 0  
                    },  
                    "configJsons": [  
                    ],  
                    "enable": 1,  
                    "id": 0,  
                    "multi_line": 0,  
                    "x_pos": 0,  
                    "y_pos": 0  
                },  
                {  
                    "configJson": {  
                        "enable": 0,  
                        "type": 0  
                    },  
                    "configJsons": [  
                    {  
                        "enable": 1,  
                        "ms_enable": 1,  
                        "text": "",  
                        "type": 0  
                    },  
                    {  
                        "enable": 1,  
                        "ms_enable": 1,  
                        "text": "",  
                        "type": 0  
                    },  
                    {  
                        "enable": 1,  
                        "ms_enable": 1,  
                        "text": "",  
                        "type": 0  
                    }  
                }  
            ]  
        }  
    }  
}
```

```
        "text": "",  
        "type": 1  
    }  
,  
    "enable": 1,  
    "id": 1,  
    "multi_line": 1,  
    "x_pos": 0,  
    "y_pos": 0  
},  
{  
    "configJson": {  
        "enable": 0,  
        "ms_enable": 0,  
        "type": 0  
    },  
    "configJsons": [  
    ],  
    "enable": 1,  
    "id": 2,  
    "multi_line": 0,  
    "x_pos": 0,  
    "y_pos": 0  
},  
{  
    "configJson": {  
        "enable": 0,  
        "text": "T1NEIFRleHQ="  
    },  
    "configJsons": [  
    ],  
    "enable": 1,  
    "id": 3,  
    "multi_line": 0,  
    "x_pos": 0,  
    "y_pos": 0  
},  
{  
    "configJson": {  
        "enable": 0  
    },  
    "configJsons": [  
    ],  
    "enable": 1,
```

```
"id": 4,
"multi_line": 0,
"x_pos": 0,
"y_pos": 0
},
{
"configJson": {
    "enable": 0
},
"configJsons": [
],
"enable": 1,
"id": 5,
"multi_line": 0,
"x_pos": 0,
"y_pos": 0
},
{
"configJson": {
    "enable": 0
},
"configJsons": [
],
"enable": 1,
"id": 6,
"multi_line": 0,
"x_pos": 0,
"y_pos": 0
},
{
"configJson": {
    "enable": 0
},
"configJsons": [
],
"enable": 1,
"id": 7,
"multi_line": 0,
"x_pos": 0,
"y_pos": 0
},
{
"configJson": {
    "enable": 0
},
```

```
"text": "T1NEIFRleHQ="
},
"configJsons": [
{
    "enable": 1,
    "ms_enable": 0,
    "text": "",
    "type": 0
},
{
    "enable": 1,
    "ms_enable": 0,
    "text": "",
    "type": 0
},
{
    "enable": 1,
    "ms_enable": 0,
    "text": "",
    "type": 0
},
{
    "enable": 1,
    "ms_enable": 0,
    "text": "",
    "type": 0
},
{
    "enable": 1,
    "ms_enable": 0,
    "text": "",
    "type": 0
},
{
    "enable": 1,
    "ms_enable": 0,
    "text": "",
    "type": 0
},
{
    "enable": 1,
    "ms_enable": 0,
    "text": "",
    "type": 0
},
{
    "enable": 1,
    "ms_enable": 0,
    "text": "",
    "type": 0
},
{
    "enable": 1,
    "ms_enable": 0,
    "text": "",
    "type": 0
}
]
```

```

        },
        {
            "enable": 1,
            "ms_enable": 0,
            "text": "",
            "type": 0
        }
    ],
    "enable": 1,
    "id": 8,
    "multi_line": 1,
    "x_pos": 0,
    "y_pos": 0
}
],
"content_order": [
    0,
    1,
    3,
    4,
    5,
    6,
    7,
    8,
    9,
    10
],
"font": {
    "color": 0,
    "size": 2
},
"pos": 1,
"separation": 0
}
}
}
}

```

层级	参数名称	参数类型	长度	是否必须	描述
1. 1	module	string		是	模块名字
1. 2	cmd	string		是	控制命令
1. 3	type	string		是	消息类型
1. 4	is_dynamic	int		否	0: 静态图片 OSD 信息叠加, 写入 flash; 1: 动态平台 OSD 信息叠加, 不写

					flash; 无此字段时默认为静态 OSD 叠加
1. 5	body	object	是	消息体	
1. 5. 1	pos	int	是	叠加位置, 0: 图片外上, 1: 图片外下, 2: 图片内, 3: 关闭	
1. 5. 2	font	object	是	字体	
1. 5. 2. 1	color	int	是	字体颜色, 0: 白色, 1: 红色, 2: 蓝色, 3: 绿色	
1. 5. 2. 2	size	int	是	字体大小, 0: 16, 1: 24, 2: 32, 3: 48	
1. 5. 3	separation	int	是	0: 空格; 1: 坚线 (扩张区域显示时生效)	
1. 5. 4	content	array	是		
1. 5. 4. 1	id	int	是	0: 实时结果, 1: 日期, 2: 时间, 3: 地点, 4: 车辆事件(部分支持), 5: 车辆类型, 6: 车标车款, 7: 车身颜色, 8: 自定义内容 1, 9: 自定义内容 2, 10: 自定义内容 3	
1. 5. 4. 2	x_pos	int	是	水平位置, 0~100	
1. 5. 4. 3	y_pos	int	是	垂直位置, 0~100	
1. 5. 4. 4	configJson	object	是		
1. 5. 4. 5	type	int	否	与 id 值有关, 见示例	
1. 5. 4. 6	text	string	否	文本的 base64 编码, 编码前最多 30 个汉字或者 60 个英文	
1. 5. 4. 7	ms_enable	int	否	精确到毫秒	
1. 5. 4. 8	configJsons	array	否	如果 id 为 1, 按顺序, 分别为日期, 时间, 星期设置	
1. 5. 5	content_order	array	是	根据 id 排序 (扩张区域显示时生效)	

### 3. 6. 9. 4 设置图片 OSD (高速实时拆分)

```
{
  "Response_AlarmInfoPlate": {
    "type": "AVS_SET_PIC OSD_PRM",
    "module": "AVS_PIC OSD_MODULE",
  }
}
```

```
"body": {
    "content": [
        {
            "configJson": {
                "enable": 0
            },
            "configJsons": [
                {
                    "enable": 1,
                    "ms_enable": 0,
                    "text": "",
                    "type": 0
                },
                {
                    "enable": 1,
                    "ms_enable": 0,
                    "text": "",
                    "type": 1
                },
                {
                    "enable": 1,
                    "ms_enable": 0,
                    "text": "",
                    "type": 2
                },
                {
                    "enable": 1,
                    "ms_enable": 0,
                    "text": "",
                    "type": 3
                },
                {
                    "enable": 1,
                    "ms_enable": 0,
                    "text": "",
                    "type": 4
                },
                {
                    "enable": 0,
                    "ms_enable": 0,
                    "text": "",
                    "type": 5
                }
            ],
        }
    ]
},
```

```
"enable": 1,
"id": 0,
"multi_line": 1,
"x_pos": 0,
"y_pos": 0
},
{
  "configJson": {
    "enable": 0,
    "type": 0
  },
  "configJsons": [
    {
      "enable": 1,
      "ms_enable": 1,
      "text": "",
      "type": 0
    },
    {
      "enable": 1,
      "ms_enable": 1,
      "text": "",
      "type": 0
    }
  ],
  "enable": 1,
  "id": 1,
  "multi_line": 1,
  "x_pos": 5,
  "y_pos": 5
},
{
  "configJson": {
    "enable": 0,
    "ms_enable": 0,
    "type": 0
  },
  "configJsons": [],
  "enable": 1,
  "id": 2,
  "multi_line": 0,
  "x_pos": 10,
  "y_pos": 10
},
```

```
{  
    "configJson": {  
        "enable": 0,  
        "text": "T1NEIFRleHQ="  
    },  
    "configJsons": [],  
    "enable": 1,  
    "id": 3,  
    "multi_line": 0,  
    "x_pos": 15,  
    "y_pos": 15  
},  
{  
    "configJson": {  
        "enable": 0  
    },  

```

```
},
{
  "configJson": {
    "enable": 0
  },
  "configJsons": [],
  "enable": 1,
  "id": 7,
  "multi_line": 0,
  "x_pos": 35,
  "y_pos": 35
},
{
  "configJson": {
    "enable": 0,
    "text": "T1NEIFRleHQ="
  },
  "configJsons": [
    {
      "enable": 1,
      "ms_enable": 0,
      "text": "MTIxMzEy",
      "type": 0
    },
    {
      "enable": 1,
      "ms_enable": 0,
      "text": "MTIxMTIx",
      "type": 0
    },
    {
      "enable": 0,
      "ms_enable": 0,
      "text": "",
      "type": 0
    },
    {
      "enable": 0,
      "ms_enable": 0,
      "text": "",
      "type": 0
    },
    {
      "enable": 0,
      "ms_enable": 0,
      "text": "",
      "type": 0
    }
  ]
}
```

```
        "ms_enable": 0,  
        "text": "",  
        "type": 0  
    },  
    {  
        "enable": 0,  
        "ms_enable": 0,  
        "text": "",  
        "type": 0  
    },  
    {  
        "enable": 0,  
        "ms_enable": 0,  
        "text": "",  
        "type": 0  
    },  
    {  
        "enable": 0,  
        "ms_enable": 0,  
        "text": "",  
        "type": 0  
    },  
    {  
        "enable": 0,  
        "ms_enable": 0,  
        "text": "",  
        "type": 0  
    },  
    {  
        "enable": 1,  
        "id": 8,  
        "multi_line": 1,  
        "x_pos": 40,  
        "y_pos": 40  
    }  
],  
"content_order": [  
    0,  
    1,  
    3,  
    4,  
    5,  
    6,  
    7,  
    8,  
    9,  
    10  
],  
"font": {  
    "color": 0,
```

<pre>         "size": 2     },     "pos": 1,     "separation": 0 } } } } </pre>					
层级	参数名称	参数类型	长度	是否必须	描述
1. 1	module	string		是	模块名字
1. 2	cmd	string		是	控制命令
1. 3	type	string		是	消息类型
1. 4	is_dynamic	int		否	0: 静态图片 OSD 信息叠加, 写入 flash; 1: 动态平台 OSD 信息叠加, 不写 flash; 无此字段时默认为静态 OSD 叠加
1. 5	body	object		是	消息体
1. 5. 1	pos	int		是	叠加位置, 0: 图片外上, 1: 图片外下, 2: 图片内, 3: 关闭
1. 5. 2	font	object		是	字体
1. 5. 2. 1	color	int		是	字体颜色, 0: 白色, 1: 红色, 2: 蓝色, 3: 绿色
1. 5. 2. 2	size	int		是	字体大小, 0: 16, 1: 24, 2: 32, 3: 48
1. 5. 3	separation	int		是	0: 空格; 1: 竖线 (扩张区域显示时生效)
1. 5. 4	content	array		是	
1. 5. 4. 1	id	int		是	0: 实时结果, 1: 日期, 2: 时间, 3: 地点, 4: 车辆事件(部分支持), 5: 车辆类型, 6: 车标车款, 7: 车身颜色, 8: 自定义内容 1, 9: 自定义内容 2, 10: 自定义内容 3
1. 5. 4. 2	x_pos	int		是	水平位置, 0~100
1. 5. 4. 3	y_pos	int		是	垂直位置, 0~100
1. 5. 4. 4	configJson	object		是	
1. 5. 4. 5	type	int		否	与 id 值有关, 见示例
1. 5. 4. 6	text	string		否	文本的 base64 编码, 编码前最多 30 个汉字或者 60 个英文

1.5.4.7	ms_enable	int		否	精确到毫秒
1.5.4.8	configJsons	array		否	如果 id 为 1, 按顺序, 分别为日期, 时间, 星期设置
1.5.5	content_order	array		是	根据 id 排序 (扩张区域显示时生效)

实时结果拆分出小选项框, configJsons 数组存储, 其中按 type 值表示不通选项: 0: 车牌号 1: 颜色 2: 像素 3: 车道号 4: 真伪 5: 危险车

### 3.7.10. 时间同步

#### 支持设备: 目前所有 R 系列

服务器在收到车牌识别结果推送、或者 comet 轮询时, 回复以下结构的消息, 可触发时间同步:

回复消息使用 JSON 格式, 格式如下:

```
{
    "Response_AlarmInfoPlate":{
        "TimeSync":{
            "timestamp":1723518863000
        }
    }
}
```

层级	字段名称	类型	取值范围	是否必须	描述
1	Response_AlarmInfoPlate	Object	无	是	
1.1	TimeSync	Object	无	是	
1.1.1	timestamp	int64	无	是	时间戳 (毫秒)

### 3.7.11. IO 常开/常闭控制

**支持设备:** 目前所有 R 系列

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息，可触发 io 常开/常闭控制：

回复消息使用 JSON 格式，格式如下：

```
{
  "Response_AlarmInfoPlate": {
    "SetIoLockStatus": {
      "ioout": 0,
      "status": 1
    }
  }
}
```

层级	字段名称	类型	取值范围	是否必须	描述
1	Response_AlarmInfoPlate	Object	无	是	
1.1	SetIoLockStatus	Object	无	是	
1.1.1	ioout	int	[0,1,2,3,4...]	是	操作的 gpio 输出口，根据设备硬件支持的最大 gpio 口限制，大部分设备只支持 0 或者 1
1.1.2	status	int	[0,1,2]	是	0 : 解锁 1 : 高电平锁定 2 : 低电平锁定

注释：当设备处于高电平/低电平锁定时候，无法直接切换到另外一个锁定状态，需要先解锁后再进行操作

### 3.7.12. 防回退版本控制

**支持设备:** 目前所有 R 系列

服务器在收到车牌识别结果推送、或者 comet 轮询时，回复以下结构的消息，可触发防回退版本控制：

设置防回退信息，回复消息使用 JSON 格式，格式如下：

```
{
  "Response_AlarmInfoPlate": {
    "SetSoftRollbackCfg": {
      "prohibit_version_rollback": true,
      "time_seconds": 1732982400
    }
  }
}
```

层级	字段名称	类型	取值范围	是否必须	描述
1	prohibit_version_rollback	Boolean	无	是	是否开启防回退功能
1.1	time_seconds	uint64_t	无	是	该值只有在开启功能后才生效。默认值为 0，表示只能向上升级版本，也可以自己下发具体版本时间戳，单位为秒，表示只能升级该时间之后的版本。

注释：设置时间戳需注意，为了统一正式版本和临时版本，统一会将输入版本转为正式版本格式，正式版本格式为年月日形式，转时间戳时会丢失时分秒，建议时间戳设置时按年月日 0 时 0 分 0 秒进行设置，防止设置时间戳有时分秒时，升级版本在转为正式版本时丢失精度，与时间戳比较出现实际结果不符的情况

## 4. 系统码表

### 4.1. 触发类型

表 3 触发类型定义表

触发类型	编码	备注
自动触发类型	1	
外部输入触发 (IO 输入)	2	
软件触发 (SDK)	4	
虚拟线圈触发	8	
车滞留事件	64	
车滞留恢复事件	65	
车折返事件	66	
机动车辆通行事件	67	
跟车事件	68	
车辆拥堵	69	
车辆拥堵解除	70	
人滞留	71	
人滞留解除	72	
人员拥堵	73	
人员拥堵解除	74	
非机动车滞留	75	
非机动车滞留解除	76	
道闸正常	77	
道闸异常	78	
道闸回落	79	
道闸抬升	80	
车头车尾号牌不一致	85	
车牌识别	86	
非机动车拥堵	87	
非机动车拥堵解除	88	

### 4.2. 车牌类型

表 4 车牌类型定义表

车牌类型	编码	备注
未知车牌	0	
蓝牌小汽车	1	

黑牌小汽车	2	
单排黄牌	3	
双排黄牌	4	
警车车牌	5	
武警车牌	6	
个性化车牌	7	
单排军车牌	8	
双排军车牌	9	
使馆车牌	10	
香港进出中国大陆车牌	11	
农用车牌	12	
教练车牌	13	
澳门进出中国大陆车牌	14	
双层武警车牌	15	
武警总队车牌	16	
双层武警总队车牌	17	
民航车牌	18	
新能源车牌	19	
大型新能源	20	
应急车	21	
领事馆	22	
新标准燃油车	23	
新标准新能源	24	
机场车牌	25	
海外车牌(所有海外车牌类型)	26	
假车牌	29	
车标	30	
无牌车	31	

### 4.3. 车牌颜色

表 5 车牌颜色定义表

车牌颜色	编码	备注
未知	0	
蓝色	1	
黄色	2	
白色	3	
黑色	4	
绿色	5	
黄绿色	6	

## 4.4. 车身颜色

表 6 车身颜色定义表

车身颜色	编码	备注
白	0	
银（灰）	1	
黄	2	
粉	3	
红	4	
绿	5	
蓝	6	
棕	7	
黑	8	
灰	9	
米	10	
橙	11	
紫	12	
未知	255	

## 4.5. 运动方向

表 7 运动方向定义表

运动方向	编码	备注
未知	0	
左	1	
右	2	
上	3	
下	4	

## 4.6. 车辆类型

表 8 车辆类型定义表

车辆类型	编码	备注
未知	0Xff	
轿车	0X00	
SUV	0X01	
MPV	0X02	

小型客车	0X03	
大型客车	0X04	
小型货车	0X05	
大型货车	0X06	

## 4.7. 开闸类型

表 9 开闸类型定义表

开闸类型	编码	备注
无开闸	0	
服务器触发开闸	1	
脱机状态白名单开闸	2	

## 5. 常见问题

### 5.1. Q: 设备注册是什么？

A：当开启时，每隔一段时间，一体机会自动发送设备信息到中心服务器，包括设备 ip，端口，序列号等信息；

### 5.2. Q: 设置好了，请求收不到，什么问题？

A：请确保一体机可以访问中心服务器的相应地址。常见的问题如，局域网内，网线是否接好，ip 地址是否冲突，是否在可以访问的网段；中心服务器如果在公网，请确保一体机可以访问公网，需要设置好一体机的网关和 dns 地址。检查中心服务器是否运行；

### 5.3. Q: 请求收到了，但没有数据（数据格式不对）？

A：车牌识别结果推送的请求发送的是 json 数据，http 的 body 内容如；

```
{"AlarmInfoPlate":{...}}
```

接收方法例如：

```
php:  
$doc = file_get_contents("php://input");  
  
java:  
    StringBuffer jb = new StringBuffer();  
    String line = null;  
    try {  
        BufferedReader reader = request.getReader();  
        while ((line = reader.readLine()) != null)  
            jb.append(line);  
    } catch (Exception e) { /*report an error*/ }
```

详细参考：<http://stackoverflow.com/questions/3831680/httpServletRequest-get-post-data>;

### 5.4. Q: 设备注册又是什么格式？

A：设备注册请求发送的数据内容如下：

```
-----cd9a1a32759bContent-Disposition: form-data; name="device_name"IVS-----
```

```
-----cd9a1a32759bContent-Disposition: form-data; name="ipaddr"192.168.0.100-----cd9a1a32759bContent-Disposition: form-data; name="port"80-----cd9a1a32759bContent-Disposition: form-data; name="user_name"admin-----cd9a1a32759bContent-Disposition: form-data; name="pass_wd"admin-----cd9a1a32759bContent-Disposition: form-data; name="serialno"fcb68a83-e8409dd-----cd9a1a32759bContent-Disposition: form-data; name="channel_num"1-----cd9a1a32759b--
```

如所见是 formpost 的格式，接收方法例如：java 使用 request.getQueryString 接收，php 使用\$\_POST 变量接收；

## 5.5. Q：如何回复请求开闸？

A: 回复

```
{"Response_AlarmInfoPlate":{"info":"ok","content":"...","is_pay":"true"}}
```

info 如果是 ok 表示开闸；

## 5.6. Q：回复中 content 能不能是中文？

A: 所有请求都用 utf8 进行编码，回复也用 utf8 即可；

## 5.7. Q：能否使用 ssl 连接发送，我们的中心服务器是 ssl 的？

A: 在设置中设置 ssl 端口（一般是 443），然后选上开启，设置就可以了，注意如果中心服务器不支持 ssl 连接，请不要选择开启该项；

## 5.8. Q：怎么获取截图？

A: 推送的结果中有 "imagePath": "/snapshot/lpr/tri\_snap\_24.jpg"，后面是访问截图的 http 路径，前面加上一体机的网址，就可以得到截图的地址如 [http://192.168.1.100:8080/snapshot/lpr/tri\\_snap\\_24.jpg](http://192.168.1.100:8080/snapshot/lpr/tri_snap_24.jpg)；

## 5.9. Q：为什么相同车牌返回了两次结果？

A: 推送的结果中有一项触发类型 triggerType，可以根据触发类型来过滤结果；

## 5.10. Q: 中心服务器，收到推送结果，回复给一体机，但看不到返回的具体内容，如何调试中心服务器？

A: 在网页上可以查看访问中心服务器的日志，可以看到中心服务器回复给一体机的内容；

## 5.11. Q: 多台相机同时使用 comet 轮询时，中心服务器压力过大，导致服务器响应不过来？

A: comet 轮询机制，为相机循环往中心服务器推送心跳包的数据，当服务器收到推送，回复响应以后，相机立马启动下一心跳包的推送，中间没有定时器等时延；

故当相机数量较大时，同时使用 comet 方式往一个服务器推送，请注意服务器性能；