

车牌识别一体机数据传输协议

v-1.7

一、HTTP 推送:

这种方式需要用户建立一个 HTTP 服务器,同时将这台 HTTP 服务器的地址配置给一体机。当一体机有识别结果后(或者其他需要推送的内容时),就会往指定的地址发送 HTTP 命令。

在一体机网页,登录后,点击左侧边栏->基本设置->网络,右边 tab 中心服务器,是推送的设置界面(可能不同版本稍有区别)

基本参数	中心服务器	upnp端口映射	动态域名	ftp	mail
服务器地址	192.168.1.243				
端口	80				
ssl连接	<input checked="" type="checkbox"/> 开启				
ssl端口	443				
超时时间(s)	5				
设备注册	<input checked="" type="checkbox"/> 开启				
地址	/devicemanagement/php/receive				
推送车牌识别结果	<input checked="" type="checkbox"/> 开启				
地址	/devicemanagement/php/platere				
内容详细等级	全部				
发送图片	<input checked="" type="checkbox"/> 开启				
发送小图片	<input checked="" type="checkbox"/> 开启				
推送端口触发信息	<input checked="" type="checkbox"/> 开启				
地址	/devicemanagement/php/gio.php				
推送串口数据	<input checked="" type="checkbox"/> 开启				
地址	/devicemanagement/php/serial.p				

确定 查看日志

1. http 服务器设置

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情况进行配置。

2. 车牌识别结果推送

开启推送车牌识别结果后,有车牌识别结果时,按图中的配置会发送消息到

<http://192.168.1.243/devicemanagement/php/plateresult.php> 这个地址。

数据内容: JSON 格式 (utf8 编码)

```
{
  "AlarmInfoPlate" : {
    "channel" : 0,
    "deviceName" : "IVS",
    "ipaddr" : "192.168.1.100",
    "result" : {
      "PlateResult" : {
        "bright" : 0,
        "carBright" : 0,
        "carColor" : 0,
        "colorType" : 0,
        "colorValue" : 0,
        "confidence" : 0,
        "direction" : 0,
        "imagePath" :
"%2Fmmc%2FVzIPCCap%2F2015_09_09%2F1714224504__%CE%DE_.jpg",
        "license" : "_无_",
        "location" : {
          "RECT" : {
            "bottom" : 0,
            "left" : 0,
            "right" : 0,
            "top" : 0
          }
        },
        "timeStamp" : {
          "Timeval" : {
            "sec" : 1441815171,
            "usec" : 672241
          }
        },
        "timeUsed" : 0,
        "triggerType" : 4,
        "type" : 0
      }
    },
    "serialNo" : "eff50e18-e3d3862b"
  }
}
```

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。

包含全部内容的json数据如下,选项“较详细”去掉了“location”，“timeUsed”，选项“较简略”还去掉了"colorValue"，"confidence"，"bright"，"carBright"，"carColor"，选项“简略”进一步去掉了"timeStamp"

说明：

字段名	含义
AlarmInfoPlate	推送结果为车牌识别结果
serialno	设备序列号
channel	默认通道号（预留）
deviceName	设备名称
ipaddr	设备 ip 地址
result	实际数据
PlateResult	车牌识别信息
license	车牌号字符串，如“京 AAAAAA”
colorValue	（预留）
colorType	车牌颜色 0：未知、1：蓝色、2：黄色、3：白色、4：黑色、5：绿色
type	车牌类型 0：未知车牌、1：蓝牌小汽车、2：黑牌小汽车、3：单排黄牌、4：双排黄牌、5：警车车牌、6：武警车牌、7：个性化车牌、8：单排军车牌、9：双排军车牌、10：使馆车牌、11：香港进出中国大陆车牌、12：农用车牌、13：教练车牌、14：澳门进出中国大陆车牌、15：双层武警车牌、16：武警总队车牌、17：双层武警总队车牌
confidence	识别结果可行度 1-100
bright	（预留）
direction	车的行进方向（预留）
location	车牌在图片中位置
RECT	位置为矩形区域； left\righ\top\bottom:车牌在图片中位置
timeUsed	识别所用时间（预留）
carBright	车身亮度（预留）
carColor	车身颜色（预留）
timeStamp	识别结果对应帧的时间戳
Timeval	时间戳结构体类型
sec/usec	从 1970 年 1 月 1 日到对应帧的秒和毫秒
triggerType	当前结果的触发类型：由以下值取并的结果：1：自动触发类型、2：外部输入触发（IO 输入）、4：

	软件触发（SDK）、8：虚拟线圈触发
imagePath/ imageFile, imageFileLen	截图的 http 地址路径 开启发送图片后，去掉了 imagePath 项，加入 "imageFile" :base64 编码的图像数据， "imageFileLen" : 图像数据实际长度，这两项内 容。
imageFragmentFile, imageFragmentFileLen	开启发送小图片后，加入" imageFragmentFile " :base64 编码的图像数据," imageFragmentFileLen " : 图像数据实际长度，这两项内容，但没有开启 时不会发送小图片路径

中心服务器回复内容：

```
{
  "Response_AlarmInfoPlate":{
    "info":"ok",//回复 ok 开闸
    "channelNum":0, //回复开闸端口号，若无，则默认为 0
    "manualTigger":"ok",//回复 ok 进行手动触发
    //(可选，不触发截图可不添加该字段)
    "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/r
eceivedeviceinfo.php"
    },
    "is_pay":"true",
    //回复串口数据可以发送到相应串口
    "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
    ]
  },
}
```

3. 端口触发信息推送

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

```
{
  "AlarmGiIn" : {
    "deviceName" : "IVS",
    "ipaddr" : "192.168.1.100",
    "result" : {
      "TriggerResult" : {
        "source" : 1,
        "value" : 1
      }
    },
    "serialno" : "eff50e18-e3d3862b"
  }
}
```

其中，TriggerResult 中 source=0 代表是输入 1，source=1 代表输入 2，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：任意内容。

4. 串口数据推送

```
//推送串口接收到的数据
{
  "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
    "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
  }
}
```

5. 截图数据

```
//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
  }
}
```

用户 comet 轮询回复字段设置截图时，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

中心服务器回复内容：

```

{
  "Response_SerialData":{
    "info":"ok",//回复 ok 开闸
    "manualTigger":"ok",//回复 ok 进行手动触发
    //回复截图内容相对路径（可选）
    "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/r
eceivedeviceinfo.php"
    },
    //回复串口数据可以发送到相应串口
    "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,

```

备注：当前 Comet 轮询中心服务器回复内容支持以下几点

- 1、开闸、协议与车牌推送协议回复内容兼容（可选）
- 2、485 推送、协议与车牌推送协议回复内容兼容（可选）
- 3、截图、可设置截图推送绝对路径、相对路径，都填入时优先选择绝对路径。端口号，不填则默认为 80 端口（可选）
- 4、手动触发、可进行手动触发业务，触发结果使用车牌识别结果推送方式进行推送
- 5、Comet 轮询支持服务器不回复以及 body 字段为空

6.触发相机手动识别

用户 comet 轮询回复字段进行手动识别时，回复数据中需要包含字段如截图：

```
{  
  "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，如下：

推送配置

设备注册 取消心跳 普通心跳 comet轮询

主服务器优先 开启

地址

推送车牌识别结果 开启

地址

内容详细等级

发送图片 发送大图片 发送小图片

推送端口触发信息 开启

地址

推送串口数据 开启

地址

HTTP脱机检查 开启

重发次数

确定

相机推送的识别结果结构体见（以上 2 车牌识别结果推送）

常见问题：

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。

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。

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>

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"19  
2.168.0.100-----cd9a1a32759bContent-Disposition: form-data; name="po  
rt"80-----cd9a1a32759bContent-Disposition: form-data; name="user_na  
me"admin-----cd9a1a32759bContent-Disposition: form-data; name="pas  
s_wd"admin-----cd9a1a32759bContent-Disposition: form-data; name="se  
rialno"fcb68a83-ee8409dd-----cd9a1a32759bContent-Disposition: form-d  
ata; 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

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

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。

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。

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

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。

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。

二、485 推送方式:

使用通过 485 或者 232 串口将识别结果发送到上位机;

数据格式:

0xBB(协议头)+0x88(命令)+DATA0..DATA11+XOR (DATA0 到 DATA11 异或)

说明:

1、波特率: 9600b/s

数据位: 8

奇偶校验: 无

停止位: 1

数据流控制: 无

2、DATA0 为识别类型

0xAA 为可识别, 0xBB 为不可识别

3、DATA1 代表车牌颜色

代码	颜色
00	蓝色
01	黄色
02	黑色
03	白色
04..	其它备用

4、DATA2, DATA3, DATA8, 这几位可能是汉字也可能普通字符, 如果是汉字, 查表如下: (字母则使用 ASCII 码)

注:

- 车牌: WJ12345 (中间不带汉字), 需要把 WJ 分成两个字节;
- 车牌: WJ 粤 12345 (中间带汉字), WJ 查表取得。

编号	汉字	编号	汉字	编号	汉字	编号	汉字
0	京	13	赣	26	陕	39	
1	津	14	鲁	27	甘	40	
2	冀	15	豫	28	青	41	
3	晋	16	鄂	29	宁	42	
4	蒙	17	湘	30	新	43	
5	辽	18	粤	31	港	44	
6	吉	19	桂	32	澳	45	
7	黑	20	琼	33	台	46	
8	沪	21	渝	34	警	47	
9	苏	22	川	35	使		
10	浙	23	贵	36	WJ		
11	皖	24	云	37	领		
12	闽	25	藏	38	学		

5、DATA3...DATA7 代表车牌其它数字或字母 (标准内码)

6、DATA9...DATA11 备用

三、TCP 推送方式:

使用通过 TCP 协议将识别结果发送到上位机，上位机作为 TCP 的 client，使用普通的 socket 通信即可，设备上 TCP 的端口号目前固定为 8131。

1. 基本的数据收发格式:

a) 基本结构

8 字节的包头: “VZ” (2 字节) + 包类型 (1 字节) + 保留位 (1 字节) + 长度 (4 字节)
数据内容

b) 包头格式

1 字节	1 字节	1 字节	1 字节	4 字节
‘V’	‘Z’	类型	reserved	长度

c) 数据内容

- 收发送，一般采用 json 指令；
- 推送结果默认也是 json，可以设置二进制；
- 数据内容附加在包头后面，长度由包头内的长度指定；

d) 包类型，为 0 时表示普通数据包，为 1 时为心跳包。

2. 发送指令采用 json 格式，指令列表如下:

功能	指令格式	说明
获取设备的序列号	<pre>{ "cmd" : "getsn" }</pre>	获取序列号
配置推送数据方式	<pre>{ "cmd" : "ivsresult", "enable" : true, "format" : "bin", "image" : true, "image_type" : 0 }</pre>	<p>enable: 为 true 时，允许有识别结果的推送，为 false 则不进行结果推送；</p> <p>format: 识别结果的数据格式，现在支持两种格式：json 格式和二进制格式(bin)，默认为 json 格式；</p> <p>image: 为 true 时，识别结果包含图片数据，为 false 时，不包含图片数据</p> <p>image_type: 图片的类型: 0 为识别时的全图, 1 为车牌区域的小图, 2 为连个图片都接收</p>
获取最近一次识别结果	<pre>{ "cmd" : "getivsresult", "image" : true }</pre>	Image: 是否接受图片

	} }	
触发车牌识别	{ cmd:trigger }	描述: 以外部触发的方式, 触发识别动作, 稍后返回识别结果
控制 IO 输出	{ cmd:ioc1, io:0, value:2, delay:500 }	描述: 通过 TCP 通道控制 GPIO 的输出 io: 对应的输出 IO 编号; value: IO 的状态, 目前有三种高电平, 低电平, 高状态方波(一般做开闸用) delay: 暂时未用
注册脱机功能	{ cmd:offline, interval:2 }	描述: 将当前 tcp 客户端注册为脱机检测的响应终端; interval: 脱机响应的超时时间
获取脱机记录	{ cmd:get_offline_record }	描述: 主动获取脱机记录, 返回记录列表, 数据只包含识别结果结构体, 不包含图片, 如果需要获取对应的图片, 可以进一步调用“获取记录图片”的接口
获取记录图片	{ cmd:get_offline_image, id:2 }	描述: 按记录 id 取识别结果对应的图片; id: 识别记录的 id, 保存在记录结构体中

● 脱机的说明

若客户端向设备注册脱机, 则白名单功能处理方式如下

- 白名单开启方式为“不开启”, 则设备不做白名单处理
- 白名单开启方式为“开启”, 则设备全权处理白名单功能
- 白名单开启方式为“脱机自动”, 则发生车牌识别时, 设备向 tcp 客户端发送识别结果指令, 如果客户端在约定好的时间内无响应, 则认为客户端已经处于脱机状态, 由设备接管白名单功能; 若设备及时做出回复, 则设备不做白名单处理。
- 白名单开启方式在网页上可配置



- 识别结果的说明
 允许接收图像信息时，上位机采用异步接收的方式，接收到的数据如上面所述的格式：长度+数据内容，其中数据内容包括：
 - 识别结果的 json 字符串
 - 识别时的图像二进制数据信息，图像为完整的 jpg 格式

3. 识别结果 JSON 格式说明

标准头+JSON 格式的识别结果字符串+空字符('\0')+图片数据。

完整的推送数据包格式				
‘V’	‘Z’	0	0	长度（整个包的大小）
识别结果字符串+结束符‘\0’				
图片数据(完整的 jpg 格式)				

识别结果的内容如下所示：

```

{
  "PlateResult":
  {
    "license":"陕 K70QU0",
    "colorValue":0,
    "colorType":1,
    "type":1,
    "confidence":99,
    "bright":0,
    "direction":4,
    "location":{"RECT":{"left":565,"top":334,"right":807,"bottom":391}},
    "timeUsed":0,
    "carBright":0,
    "carColor":0,
    "timeStamp":{"Timeval":{"sec":1426625222,"usec":622802}},
    "triggerType":4
  },
  "imageEnable":1,
  "cmd":"ivs_result",
  "id":1597,
  "imageformat":"jpg",
  "timeSting":"2015-03-17 20:47:02"
}

```

其中，JSON 中各个字段的含义如下：

字段名	说明
license	车牌号码（汉字为 GB2312 编码）
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
imageEnable	允许带图片
cmd	当前指令名称
id	识别记录的编号
imageformat	图片格式

timeSting	触发时间字符串，格式如：5-01-02 03:04:05
bigImageSize	整幅大图尺寸（字节数）
smallImageSize	车牌区域图片尺寸（字节数）

另，LC_X, LT_X, DIRECTION_X 详细定义请参考头文件 VzClientSDK_LPDefine.h。

4. 识别结果二进制格式的支持

通过发送指令

```
{
    "cmd" : "ivsresult",
    "enable" : true,
    "format" : "bin",
    "image" : true
};
```

可以设定，接收识别结果为二进制格式。收到数据包的格式如下：

标准头+数据块头+二进制的识别结果+数据块头+图片数据。

- 标准头：上述的‘VZ’+ 类型+保留位+长度；
- 数据块头：‘1R’(2 个字节)+ 类型(1 个字节)+ 保留位(1 个字节)+ 后续数据块的长度 4 个字节)，总计 8 个字节；
- 识别结果的格式为结构体 TH_PlateResult, 参考头文件可以查看具体定义，它的数据块类型为 BLOCK_TYPE_BIN_RESULT (值为 0)；
- 图片数据：完整的 jpg 格式的图片数据，它的数据块类型为 BLOCK_TYPE_IMAGE_DATA (数值为 2)。

完整的推送数据包格式				
‘V’	‘Z’	0	0	长度（整个包的大小）
‘I’	‘R’	1(二进制)	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（识别结果结构体的大小）
图片数据(完整的 jpg 格式)				
‘I’	‘R’	3(小图片)	0	长度（识别结果结构体的大小）
图片数据(完整的 jpg 格式)				

5. 心跳包

- 心跳包为固定的 8 字节，即不带实际数据的消息头，其中包类型要设置为 1，长度设置为 0，所以心跳包的内容固定为：565A0100 00000000 (16 进制)；
- 由客户端把心跳包发送给服务端，服务端再返回相同的内容给客户端；
- 心跳包的默认间隔时间为 30 秒，超过 3 次没发送心跳包，服务端会主动断开连接。

6. 注意：

4 个字节保存数据包的长度采用网络字节发送，发送或接收后用 htonl 转换；