

⇨ HTTP 对接协议文档			
文件编号:		当前版本:	V1.1
页数:	xx 页	发布日期:	2021-01-18
文件状态: 初版 <input checked="" type="checkbox"/> 评审通过 <input type="checkbox"/> 正式发布 <input type="checkbox"/>			

作者	李秋燕		
评审人员			
审核人员	彭元华		
批准			

修 订 详 细 记 录

版本	修订内容详述	编制/修订人	批准人	修订日期
V1.0	新建文档	李秋燕		2020.9.5
V1.1	修改 HTTP 推送的注册数据	李秋燕		2020.9.8
V1.2		李秋燕		2021.1.18

目录

目录.....	4
第 1 章 简介	5
第 2 章 接收 HTTP 推送数据.....	6
2.1 HTTP 服务器配置	6
2.2 HTTP 协议	6
2.2.1 推送识别结果的 JSON 数据	6
2.2.2 推送识别结果的 JSON 数据后服务端的回复.....	7
2.2.3 JSON 中的字段解释.....	7
第 3 章 HTTP 心跳.....	9
3.1 网页配置.....	9
3.2 HTTP 推送的注册数据	9
3.2.1 请求强制触发的 JSON 消息	10
3.2.2 请求设备状态控制的 JSON 消息.....	11
3.2.3 请求发送串口数据的 JSON 消息.....	11
3.2.4 请求截图的 JSON 消息	11
第 4 章 循环推送	13
4.1 网页配置.....	13
4.2 协议介绍.....	13

第1章 简介

目前 http 服务包含 http 推送和 http 心跳二部分。

http 推送：推送车牌识别信息和车牌图片。

http 心跳：定时向服务端发送数据，保持连接，并且可以通过回复数据操作设备。

第2章 接收 HTTP 推送数据

2.1 HTTP 服务器配置

打开设备网页，点击“设备参数”，再点击“HTTP 推送”，弹出的网页如下：

车位检测智能相机

实时预览

快速配置

名称和识别区域

外控和指示灯

时间设置

设备参数

网络参数

HTTP推送

HTTP推送

服务器设置

服务器地址: 192.168.10.10

端口号: 8888

超时时间(s): 2

推送识别结果: 启用

发送图片: 小图片 大图片

地址: /devicemanagement/php/plateres

设备心跳设置

设备注册: comet轮询

心跳周期(s): 30

注册地址: /devicemanage/poll.php

循环推送设置

循环推送结果: 启用

循环推送间隔(s): 10

转发服务器

启用转发服务器: 启用

转发服务器地址:

备用服务器设置

备用服务器: 启用

服务器地址: 192.168.1.100

端口号: 1010

超时时间(s): 2

推送识别结果: 启用

发送图片: 小图片 大图片

地址: /devicemanagement/php/plateres

车位设置

获取空车位数: 获取空车位数

获取空车位地址: /devicemanage/get_empty_frame

推送间隔(s): 5

启用屏显: ID1 ID2 ID3 ID4 ID5 ID6

保存配置

1. 服务器地址：设备要连接的 HTTP 服务地址；
2. 端口号：服务器的端口号；
3. 超时时间：HTTP 推送数据的超时时间；如果超时，即推送失败，没有重传机制，不做特殊处理。
4. 启用推送识别结果：是否向服务器推送识别结果；
5. 发送图片：推送的数据里是否含有大图片，小图片；大图片表示推送的是整张图片，小图片是表示推送的车头的图片。
6. 地址：推送到服务器的地址。

2.2 HTTP 协议

2.2.1 推送识别结果的 JSON 数据

```

{
  "Type": "PKResult",           //推送类型
  "s_addr": 1696508096,         //本机的 ip 地址, in_addr.s_addr 的值
                                //注:1696508096 <==> "192.168.30.101"
  "frame_status": DS_FREE,     //见 VZ_DEVICE_STATUS 枚举定义
  "sn": "83e40587-f9601e00",  //设备序列号
  "ibeacon": "ibeacon",       //iBeacon 蓝牙 ID (无效)
  "FrameIndex": 1,            //车位索引号

```

```

"FrameName": "frame1",           //车位名称, UTF8 编码
"ParkingStatus": PS_NOHEAD_NOPLATE, //车位状态, 参考 VZ_PARK_STATUS 定义
"License": "J| 123342",         //车牌号, UTF8 编码
"LicenseType": 2,               //车牌类型, 见下方定义
"Confidence": 100,              //置信度: 0-100 (无效)
"OccupyLine": 1,                //1 为占线, 0 为不占线
"BDTime": {                      //车位识别的时间
    "BDYear": 1970,
    "BDMon": 1,
    "BDDay": 1,
    "BDHour": 1,
    "BDMin": 1,
    "BDSec": 1,
},
"SmallImageData": base64 编码后的结果 //开启小图片才推送才有 SmallImage* 两个字段
"SmallImageLength": base64 编码前的长度
"Base64SmallImageLength": 16476,     //Base64 后的小图长度
"ImageData": base64 编码后结果      //开启大图推送, 才有 ImageData* 两个字段
"ImageDataLength": base64 编码前的长度
"Base64ImageLength": 215796,        //Base64 后的大图长度
}

```

2.2.2 推送识别结果的 JSON 数据后服务端的回复

```

{
    "Type" : "PKResult" , //推送类型
    "state" : 200,
    "error_msg" : "All done"
}

```

2.2.3 JSON 中的字段解释

1. 设备状态

```

typedef enum {
    DS_FREE = 0,           //空闲, 相机看的几个车位上都没有车
    DS_BUSY,               //占用, 相机看的所有车位上都有车
    DS_RESERVE,            //预留 暂时没用
    DS_ALARM,              //告警, 相机看的所有车位中, 至少有一个车位占线
} VZ_DEVICE_STATUS;

```

2. 停车位状态

```

typedef enum
{

```

```

PS_NOHEAD_NOPLATE, //无车头-无车牌, 没有车
PS_HEAD_NOPLATE,   //有车头-无车牌, 没有车牌的车 (无牌车)
PS_NOHEAD_PLATE,   //无车头-有车牌, 只检测到车牌, 没有检测到车头, 但认为有车
PS_HEAD_PLATE      //有车头-有车牌, 有车牌的车

```

```
}VZ_PARK_STATUS;
```

注：无车头-无车牌，认为当前车位无车，即出场状态。其他状态都是入场状态。入场带车牌，出场不一定带车牌。

3.车牌类型定义

```

#define LT_UNKNOWN 0 //未知车牌
#define LT_BLUE 1 //蓝牌小汽车
#define LT_BLACK 2 //黑牌小汽车
#define LT_YELLOW 3 //单排黄牌
#define LT_YELLOW2 4 //双排黄牌 (大车尾牌, 农用车)
#define LT_POLICE 5 //警车车牌
#define LT_ARMPOL 6 //武警车牌
#define LT_INDIVI 7 //个性化车牌
#define LT_ARMY 8 //单排军车牌
#define LT_ARMY2 9 //双排军车牌
#define LT_EMBASSY 10 //使馆车牌
#define LT_HONGKONG 11 //香港进出中国大陆车牌
#define LT_TRACTOR 12 //农用车牌
#define LT_COACH 13 //教练车牌
#define LT_MACAO 14 //澳门进出中国大陆车牌
#define LT_ARMPOL2 15 //双层武警车牌
#define LT_ARMPOL_ZONGDUI 16 // 武警总队车牌
#define LT_ARMPOL2_ZONGDUI 17 // 双层武警总队车牌
#define LT_CA 18 // 民航车牌
#define LT_NEW_ENERGY 19 // 新能源车牌
#define LT_NEW_ENERGY_2 20 // 新能源车牌大
#define LT_EMERGENCY 21 // 应急

```

4.车位状态上报到 HTTP 服务器:

- (1) 强制触发时，不管车位状态有没有变化，每个车位的状态都会上报到 HTTP 服务器；
- (2) 自动触发时（即相机内部推送），只会上报状态有变化的车位。

第3章 HTTP 心跳

HTTP 心跳分为二种，普通心跳和 comet 心跳。普通心跳，用于保持连接。Comet 心跳用于保持连接和操作设备。普通心跳的周期是在网页上可以配置，Comet 心跳是固定的，目前设备的长轮询的超时时间为 30s，发送长轮询间隔时间为 10ms。

3.1 网页配置

车位检测智能相机

实时预览	HTTP推送	
	<p>服务器设置</p> <p>服务器地址: 192.168.10.10</p> <p>端口号: 8888</p> <p>超时时间(s): 2</p> <p>推送识别结果: <input checked="" type="checkbox"/> 启用</p> <p>发送图片: <input type="checkbox"/> 小图片 <input type="checkbox"/> 大图片</p> <p>地址: /devicemanagement/php/plateres</p>	<p>备用服务器设置</p> <p>备用服务器: <input type="checkbox"/> 启用</p> <p>服务器地址: 192.168.1.100</p> <p>端口号: 1010</p> <p>超时时间(s): 2</p> <p>推送识别结果: <input type="checkbox"/> 启用</p> <p>发送图片: <input type="checkbox"/> 小图片 <input type="checkbox"/> 大图片</p> <p>地址: /devicemanagement/php/plateres</p>
快速配置	<p>设备心跳设置</p> <p>设备注册: comet轮询</p> <p>心跳周期(s): 30</p> <p>注册地址: /devicemange/poll.php</p>	<p>车位设置</p> <p>获取空车位数: <input type="checkbox"/> 获取空车位数</p> <p>获取空车位地址: /devicemange/get_empty_frame</p> <p>推送间隔(s): 5</p> <p>启用屏显: <input type="checkbox"/> ID1 <input type="checkbox"/> ID2 <input type="checkbox"/> ID3 <input type="checkbox"/> ID4 <input type="checkbox"/> ID5 <input type="checkbox"/> ID6</p> <p>保存配置</p>
名称和识别区域		
外控和指示灯		
时间设置		
设备参数		
网络参数		
HTTP推送		
相机设置		
串口设置		
节点设置		
本地配置		
系统维护	<p>循环推送设置</p> <p>循环推送结果: <input checked="" type="checkbox"/> 启用</p> <p>循环推送间隔(s): 10</p> <p>转发服务器</p> <p>启用转发服务器: <input type="checkbox"/> 启用</p> <p>转发服务器地址: </p>	

3.2 HTTP 推送的注册数据

服务端开启 HTTP 业务，以接受设备发送来的注册消息，设备发来的注册消息中包含设备信息的 Json 消息，形式如下：

```
{
  "type": "http_reg",
  "body": {
    "dev_name": IVS,           //设备名称
    "ipaddr": 192.168.7.60,   //设备 IP 地址
    "serialNo": 5d6c1b9c-567056df //设备序列号
  }
}
```

设备收到注册的 Json 消息后，服务端回复如下：

```
{
  "type": "http_reg",
  "state": 200,
  "error_msg": "All done"
```

```
}
```

注册消息中有推送该消息的设备的 IP 地址，服务端在长轮询超时时间内，向该 IP 地址发送带有请求内容的回复消息，便可在对应设备上执行相应的功能。

服务端在接收到一台设备发送的注册消息后，在长轮询超时时间内，构建请求相应业务的 Json 消息，并在回复消息的内容中携带该请求消息。即可完成相应功能。

举例：如果，HTTP 服务器希望相机执行强制触发，可以在收到相机的心跳时，把强制触发的 json 通过心跳的响应结果返回给相机，相机就会将强制触发的结果发送给 HTTP 服务器。

3.2.1 请求强制触发的 JSON 消息

```
{
  "force_trigger": 1,
  "msg": "ok",
  "state": 200,
  "type": "http_reg"
}
```

强制触发后，设备会推送强制识别的 Json 消息到 HTTP 服务端，其形式如下，各字段解释参考第二章：

```
{
  "BDTime" : {
    "BDTDay" : 7,
    "BDTHour" : 17,
    "BDTMin" : 40,
    "BDTMon" : 11,
    "BDTSec" : 11,
    "BDTYear" : 2017
  },
  "ImageData" : "U2NobWlkdA==",
  "Base64ImageLength" : 12,
  "SmallImageData" : "dG9j",
  "Base64SmallImageLength" : 4,
  "Confidence" : 98,
  "FrameIndex" : 2,
  "FrameName" : "frame2",
  "ImageDataLength" : 7,
  "License" : ")|| 5B556",
  "LicenseType" : 3,
  "OccupyLine" : 0,
  "ParkingStatus" : PS_NOHEAD_NOPLATE,
  "SmallImageLength" : 3,
  "TriggerType" : 2,
}
```

```

        "Type" : "PKResult",
        "frame_status" : 1,
        "ibeacon" : "ivs_06221",
        "s_addr" : 1426565312,
        "sn" : "83340587-486c1ed2"
    }

```

服务端接收到该消息，对 Base64 后的大图和小图数据进行 Base64 解码，写入图片文件，即可得到强制触发的图片。

注意：强制触发的结果和推送识别结果，各字段的含义相同，接口地址是同一个。

3.2.2 请求设备状态控制的 JSON 消息

```

//设备状态定义
typedef enum {
    DEVICE_FREE = 0,
    DEVICE_BUSY
} DEVICE_STATUS;

{
    "type": "http_reg"
    "dev_status": {
        "status": 1 //1 占用, 0 空闲
    },
    "msg": "ok",
    "state": 200,
}

```

3.2.3 请求发送串口数据的 JSON 消息

```

{
    "msg": "ok",
    "serial_data": {
        "data": "test http",
        "data_len": len("test http")
    },
    "state": 200,
    "type": "http_reg"
}

```

备注：首先要对发送的数据进行 Base64 编码。

3.2.4 请求截图的 JSON 消息

```

{
    "msg": "ok",
    "screen_shot": {

```

```
    "relative_url": "/F2PlateRecognitionResult/test/nosave"  
  },  
  "state": 200,  
  "type": "http_reg"  
}
```

发送截图请求后，设备会推送一个带有截图信息的 Json 消息到服务端，其中包含推送设备的 IP 地址，所截图片的数据(Base64 编码后)和图片数据的实际长度数据，服务端接收该信息，即可对截图的图片数据进行 Base64 解码，然后写入图片文件里，即得到了截图的数据。

HTTP 服务端收到的设备推送的截图结果的 Json 消息如下：

```
{  
  "Base64ImageLength" : 207916,  
  "ImageData" : "U2NobWlkdA==" , //截图通过 Base64 编码后的数据  
  "ImageDataLength" : 155936  
}
```

备注：上述四种请求既可单独发送，也可以按 Json 格式组合发送。

第4章 循环推送

4.1 网页配置

启用后，以设置的时间间隔，推送当前相机的识别结果到服务器。当前相机启用几个车位，就推送几个车位的信息到服务器。

服务器配置请参考第二章。

车位检测智能相机

The screenshot shows the configuration page for a parking detection camera, specifically the 'HTTP推送' (HTTP Push) section. The interface is divided into several panels:

- 左侧导航栏 (Left Sidebar):** Includes options like '实时预览' (Real-time Preview), '快速配置' (Quick Configuration), '设备参数' (Device Parameters), '网络参数' (Network Parameters), 'HTTP推送' (HTTP Push), '相机设置' (Camera Settings), '串口设置' (Serial Port Settings), '节点设置' (Node Settings), '本地配置' (Local Configuration), and '系统维护' (System Maintenance).
- 主配置区 (Main Configuration Area):**
 - 服务器设置 (Server Settings):** Includes fields for '服务器地址' (Server Address: 192.168.10.10), '端口号' (Port: 8888), '超时时间(s)' (Timeout: 2), '推送识别结果' (Push Recognition Results: checked), '发送图片' (Send Images: Small/Large), and '地址' (Address: /devicemanagement/php/plateres).
 - 设备心跳设置 (Device Heartbeat Settings):** Includes '设备注册' (Device Registration: 普通心跳), '心跳周期(s)' (Heartbeat Cycle: 30), and '注册地址' (Registration Address: /devicemanage/poll.php).
 - 循环推送设置 (Loop Push Settings):** Includes '循环推送结果' (Loop Push Results: checked) and '循环推送间隔(s)' (Loop Push Interval: 10).
 - 转发服务器 (Forwarding Server):** Includes '启用转发服务器' (Enable Forwarding Server: unchecked) and '转发服务器地址' (Forwarding Server Address).
- 备用服务器设置 (Backup Server Settings):** Includes '备用服务器' (Backup Server: unchecked), '服务器地址' (Server Address: 192.168.1.100), '端口号' (Port: 1010), '超时时间(s)' (Timeout: 2), '推送识别结果' (Push Recognition Results: unchecked), '发送图片' (Send Images: Small/Large), and '地址' (Address: /devicemanagement/php/plateres).
- 车位设置 (Parking Settings):** Includes '获取空车位数' (Get Empty Parking Spaces: unchecked), '获取空车位地址' (Get Empty Parking Spaces Address: /devicemanage/get_empty_frame), '推送间隔(s)' (Push Interval: 5), and '启用屏显' (Enable Display: ID1, ID2, ID3, ID4, ID5, ID6).

- 底部 (Bottom):** A blue '保存配置' (Save Configuration) button.

4.2 协议介绍

协议参考第二章。

推送的识别结果是手动触发的结果。