

海康威视萤石 IPC 互联互通 SDK 使用说明

修订历史			
版本	修订内容	修订人	修订日期
V1.0.0	创建	张 振	2015/06/05
V1.0.1	修订格式	陈冠兰	2015/06/19
V2.0.0	修改互联互通 SDK2.0 接口	赖程鹏	2015/09/28
V2.0.1	增加获取 wifi 信息, ap 列表, 配置 wifi 功能	赖程鹏	2015/10/16
V2.1.0	1.修订格式; 2.修改 HIK_SDK_DEVICE_NAME 结构体名称为 HIK_SDK_IPC_NAME; 3. hik_sdk_cmd 接口的出参 out_param 内存改由 SDK 分配, 由用户释放, 传入时为字符指针的地址。	赖程鹏	2015/11/17
V2.1.1	第三方设备认证授权接口的参数调整, 由 SignString 改为 AccessToken	赖程鹏	2016/02/02

目录

1. SDK 对系统平台要求.....	4
1.1 硬件要求.....	4
1.2 软件要求.....	4
2. 基础概念	4
2.1 图片的 uuid.....	4
2.2 AccessToken.....	4
2.3 主要流程.....	4
2.3.1 采用 TCP Socket 方式通信.....	5
2.3.2 采用 Unix Socket 方式通信	5
2.3.3 采用串口方式通信.....	5
2.3.4 互联互通过程的时序图.....	5
3. 互联互通 SDK API	7
3.1 错误码.....	7
3.2 数据结构.....	7
3.3 接口函数.....	8
3.3.1 初始化 SDK.....	8
3.3.2 查询设备.....	9
3.3.3 授权.....	10
3.3.4 删除授权.....	11
3.3.5 发送控制命令.....	11
3.3.6 注册通知消息接收的回调函数.....	15
3.3.7 结束 SDK.....	16
3.4 回调函数.....	16

1. SDK 对系统平台要求

1.1 硬件要求

1. 提供模块或设备的唯一标识，如 MAC；
2. 可读写的文件系统。

1.2 软件要求

1. 支持 pthread、libc 库；
2. 提供存储接口，如：文件操作接口；
3. 提供 TCP/IP socket 接口。

2. 基础概念

2.1 图片的 uuid

由 IPC 抓图成功后，会生成一个 uuid。如果 SDK 给 IPC 下发抓图命令，IPC 会将此 uuid 返回给 SDK，SDK 宿主设备根据此 uuid 获取图片。

2.2 AccessToken

IPC 能够响应 SDK 发出的抓图和录像之前，需要对 SDK 进行授权，SDK 使用平台下发的 AccessToken 字符串进行授权。AccessToken 形式如下：
at.9mhy078mddmacfvi478n791lawng6xbp-2rrmlu3tuf-00c7x04-worpcic7f。

2.3 主要流程

运行互联互通 SDK 的设备与 IPC 可以采用三种方式进行通信：TCP Socket、Unix Socket 和串口，不同通信方式其流程略有不同。

2.3.1 采用 TCP Socket 方式通信

1. SDK 初始化
2. SDK 局域网发现 IPC 设备
3. SDK 授权
4. SDK 发送控制命令
5. SDK 接收 IPC 通知消息

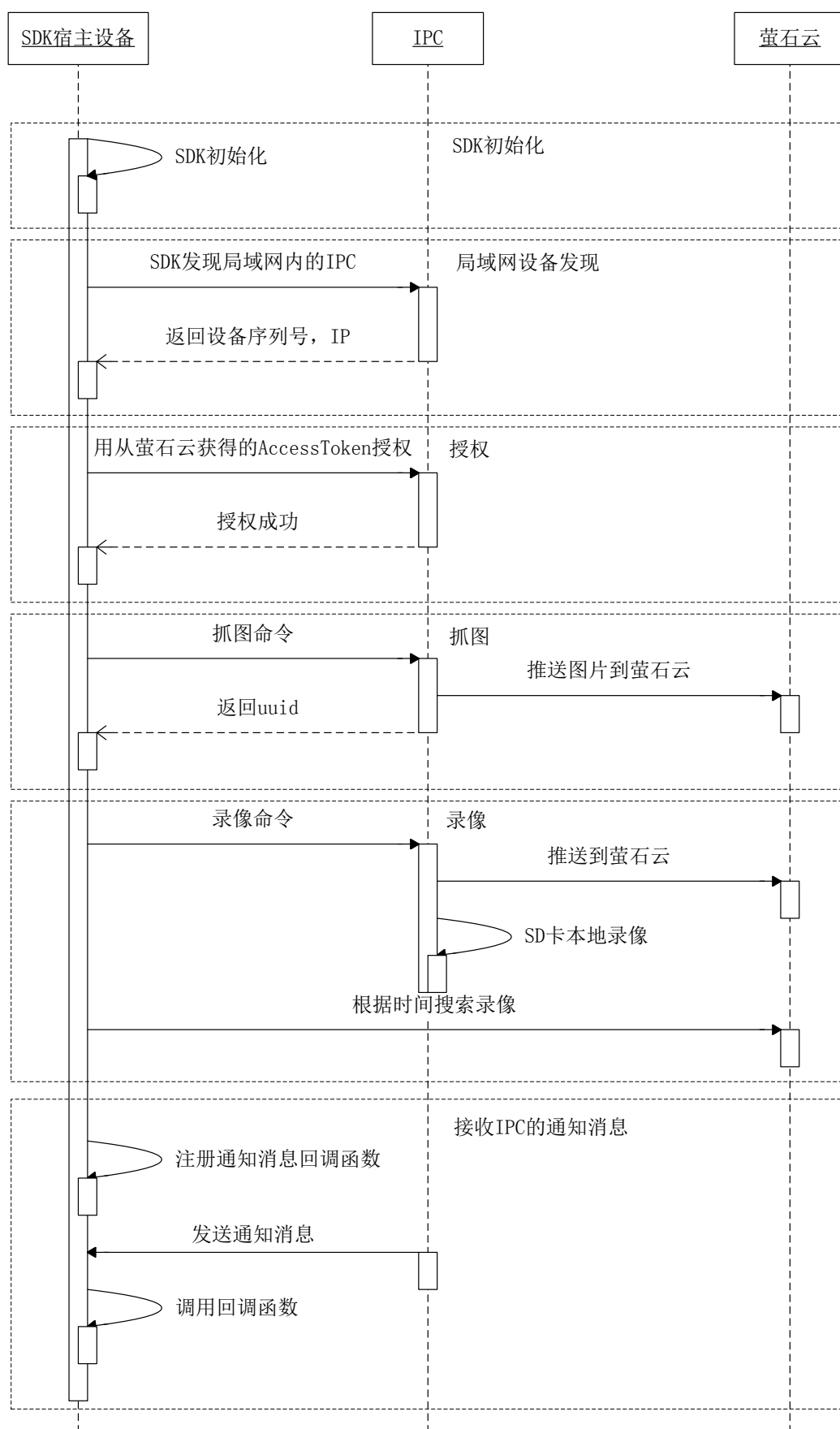
2.3.2 采用 Unix Socket 方式通信

1. SDK 初始化
2. SDK 发送控制命令
3. SDK 接收 IPC 通知消息

2.3.3 采用串口方式通信

1. SDK 初始化
2. SDK 发送控制命令
3. SDK 接收 IPC 通知消息

2.3.4 互联互通过程的时序图



3. 互联互通 SDK API

3.1 错误码

HIK_SDK_NO_ERROR	0x00000000	///< 无错误
HIK_SDK_ALREADY_INITED	0x00200001	///< SDK 已经初始化
HIK_SDK_DEVICE_NOT_FOUND	0x00200002	///< 当前局域网内未找到该设备
HIK_SDK_CONNECT_TO_DEVICE_ERROR	0x00200003	///< 连接设备错误
HIK_SDK_SEND_REQ_ERROR	0x00200004	///< 发送请求错误
IK_SDK_RECV_RSP_ERROR	0x00200005	///< 接收响应错误
HIK_SDK_DEVICE_HAS_AUTHORIZED	0x00200006	///< 设备已经被认证
HIK_SDK_AUTHORIZE_TO_SERVER_ERROR	0x00200007	///< 向服务器认证错误
HIK_SDK_KEY_VERIFY_ERROR	0x00200008	///< 秘钥校验错误
HIK_SDK_DEAUTHORIZE_ERROR	0x00200009	///< 删除认证错误
HIK_SDK_NO_SESSION_FOUND	0x0020000A	///< 未找到认证会话
HIK_SDK_SESSION_OUT_OF_DATE	0x0020000B	///< 认证会话过期
HIK_SDK_CAPTURE_PICTURE_ERROR	0x0020000C	///< 抓图超时错误
HIK_SDK_VIDEO_RECORD_ERROR	0x0020000D	///< 录像错误
HIK_SDK_OLD_DEVICE_FIRMWARE	0x0020000E	///< 设备固件版本不兼容
HIK_SDK_PARAM_ERROR	0x0020000F	///< 参数错误
HIK_SDK_UNKNOW_ERROR	0x00200100	///< 未知错误

3.2 数据结构

```
// IPC 的设备序列号:
typedef struct
{
    char serial[16];
}HIK_SDK_IPC_NAME;

// 互联互通 SDK 与 IPC 的通信方式:
typedef enum
{
    // tcp socket 方式通信, SDK 与 IPC 在局域网中
    HIK_SDK_TRANSFER_MODE_TCP_SOCKET = 0x01,
    // 串口方式通信, SDK 与 IPC 串口相连
    HIK_SDK_TRANSFER_MODE_SERIAL = 0x02,
    // unix socket 方式通信, SDK 运行在 IPC 上
    HIK_SDK_TRANSFER_MODE_UNIX_SOCKET = 0x04
}HIK_SDK_TRANSFER_MODE;

//互联互通 SDK 与 IPC 的通信参数
```

```
typedef struct
{
    // 通信方式
    HIK_SDK_TRANSFER_MODE trans_mode;
    // 串口设备名称, 如"/dev/ttyS0"
    char serial_name[32];
}HIK_SDK_TRANSFER_PARAM;

//第三方设备参数
typedef struct
{
    // 厂商信息
    char manufacturer[32];
    // 设备型号
    char model[32];
    // 设备序列号
    char device_id[32];
}HIK_SDK_DEVICE_INFO;

//接收到 IPC 通知消息时调用的回调函数的原型
typedef int (*HIK_SDK_ON_RECV_NOTIFY)(char *msg, char *data, int len);
```

3.3 接口函数

3.3.1 初始化 SDK

1) 功能描述

启动 SDK。在调用 SDK 业务 API 之前, 必须先调用该 API。

2) API 定义

```
///
/// @brief 初始化 HIK_SDK
/// @param device_info 第三方设备（运行 SDK 的设备）相关信息
/// @param trans_param 通信方式参数
/// @return 错误码
///
int hik_sdk_init(HIK_SDK_DEVICE_INFO device_info, HIK_SDK_TRANSFER_PARAM trans_param);
```

3) 错误码

错误码	说明
-----	----

HIK_SDK_NO_ERROR	返回成功
HIK_SDK_ALREADY_INITED	SDK 已经初始化
HIK_SDK_PARAM_ERROR	传入参数错误

4) 用例

```
HIK_SDK_DEVICE_INFO dev_info;
strcpy(dev_info.manufacturer, "third_party_manufacturer");
strcpy(dev_info.model, "device_model");
strcpy(dev_info.device_id, "123456789");

HIK_SDK_TRANSFER_PARAM trans_param;
trans_param.trans_mode = HIK_SDK_TRANSFER_MODE_SERIAL;
strcpy(trans_param.serial_name, "/dev/ttyS0");

hik_sdk_init(dev_info, trans_param);
```

5) 注意事项

使用 TCP Socket 和 Unix Socket 方式进行通信时, 结构体 HIK_SDK_TRANSFER_PARAM 的 serial_name 可不赋值, 当使用串口方式时, serial_name 必须赋值。

3.3.2 查询设备

1) 功能描述

发现局域网内支持互联互通的 IPC 设备。

2) API 定义

```
///
/// @brief 查询设备接口, 得到局域网内查找到的所有 IPC
/// @param num 查找到的 IPC 数量
/// @param device_name 查找到的 IPC 序列号
/// @return 查找到的 IPC 数量
///
int hik_sdk_query_device(int *num, HIK_SDK_IPC_NAME *device_name)
```

3) 用例

```
int num = 0;
HIK_SDK_IPC_NAME device_name[32];
hik_sdk_query_device(&num, device_name);
```

4) 注意事项

1. 最多发现 32 个设备。

2. 设备发现不需要授权。

3.3.3 授权

1) 功能描述

在互联互通 SDK 和 IPC 能正常通信之前，此 SDK 设备必须经过授权。

2) API 定义

```
///  
/// @brief 授权认证接口  
/// @param access_token 授权 AccessToken，从萤石开放平台获得  
/// @param device_name 需要关联的 IPC 序号  
/// @return 错误码  
///  
int hik_sdk_session_authorize(const char *access_token, HIK_SDK_IPC_NAME  
*device_name)
```

3) 错误码

错误码	说明
HIK_SDK_NO_ERROR	返回成功
HIK_SDK_DEVICE_NOT_FOUND	当前局域网内未找到该设备
HIK_SDK_CONNECT_TO_DEVICE_ERROR	连接设备错误
HIK_SDK_SEND_REQ_ERROR	发送请求错误
HIK_SDK_RECV_RSP_ERROR	从设备接收响应错误
HIK_SDK_DEVICE_HAS_AUTHORIZED	该设备已经被认证
HIK_SDK_AUTHORIZE_TO_SERVER_ERROR	向服务器认证错误
HIK_SDK_KEY_VERIFY_ERROR	秘钥认证错误

4) 用例

```
char copy[512] = {0};  
HIK_SDK_IPC_NAME authorize_device_name;  
  
strcpy(authorize_device_name.serial, "123456789");  
strcpy(copy, access_token);  
ret = hik_sdk_session_authorize(copy, &authorize_device_name);
```

5) 注意事项

1. 如果一台 SDK 宿主设备要与 2 台以上不同 IPC 互联互通，则需要分别授权。
2. 授权时会建立一个授权文件，需要有可写的文件系统。
3. 授权有效期为 7 天，到期之后需要删除授权，并重新授权。

3.3.4 删除授权

1) 功能描述

删除之前建立的授权。

2) API 定义

```
///  
/// @brief 删除授权接口  
/// @param device_name 关联的 IPC 序列号  
/// @return 错误码  
///  
int hik_sdk_session_deauthorize(HIK_SDK_IPC_NAME *device_name)
```

3) 错误码

错误码	说明
HIK_SDK_NO_ERROR	返回成功
HIK_SDK_DEVICE_NOT_FOUND	当前局域网内未找到该设备
HIK_SDK_CONNECT_TO_DEVICE_ERROR	连接设备错误
HIK_SDK_SEND_REQ_ERROR	发送请求错误
HIK_SDK_DEAUTHORIZE_ERROR	删除认证错误

4) 用例

```
hik_sdk_session_deauthorize(&authorize_sevice_name);
```

5) 注意事项

无。

3.3.5 发送控制命令

1) 功能描述

SDK 向 IPC 发送控制命令，命令类型由入参的 JSON 格式中的“cmd”字段决定，目前支持的命令有“capture_picture”、“record_video”、“reset_device”、“smart_config”、“get_wifi_info”、“get_wifi_list”、“config_wifi”。其中重启设备、一键配置、获取 wifi 信息，获取 ap 列表和配置 wifi 目前仅支持模块设备 CM2，并且只能工作于串口方式和 Unix Socket 方式。

2) API 定义

```

///
/// @brief 发送命令至 IPC
/// @param device_name 接收命令的 IPC 序列号
/// @param in_param 入参，JSON 格式，依据不同命令而不同
/// @param out_param 出参，JSON 格式，依据不同命令而不同
/// @return 错误码
///
int hik_sdk_cmd(const HIK_SDK_IPC_NAME *device_name, const char *in_param, char
**out_param);

```

3) 错误码

错误码	说明
HIK_SDK_NO_ERROR	返回成功
HIK_SDK_DEVICE_NOT_FOUND	当前局域网内未找到该设备
HIK_SDK_CONNECT_TO_DEVICE_ERROR	连接设备错误
HIK_SDK_NO_SESSION_FOUND	未找到认证授权会话
HIK_SDK_SESSION_OUT_OF_DATE	认证授权会话过期
HIK_SDK_SEND_REQ_ERROR	发送请求错误
HIK_SDK_RECV_RSP_ERROR	从设备接收响应错误
HIK_SDK_DEVICE_HAS_AUTHORIZED	该设备已经被认证
HIK_SDK_AUTHORIZE_TO_SERVER_ERROR	向服务器认证错误
HIK_SDK_KEY_VERIFY_ERROR	秘钥认证错误
HIK_SDK_UNKNOW_ERROR	未知错误

4) 入参 JSON 格式说明

JSON 字段	说明
cmd	命令

入参的 JSON 格式中必须包含 cmd 字段，根据命令不同，可能有其它字段。

1. 不带参数控制命令

功能	cmd 字段
抓图	capture_picture
录像	record_video
重置设备	reset_device
wifi 一键配置	smart_config
获取 wifi 信息	get_wifi_info
获取 ap 列表	get_wifi_list

抓图命令的入参 JSON 示例如下所示：

```

{
  "cmd": "capture_picture"
}

```

2. 带参数控制命令

功能	cmd 字段	ssid 字段	pwd 字段
配置 wifi	config_wifi	所要连接的 wifi 名称	wifi 密码

配置 wifi 命令的入参 JSON 示例如下所示：

```
{
  "cmd": "config_wifi",
  "ssid": "123456",
  "pwd": "123456789"
}
```

5) 出参 JSON 格式说明

JSON 字段	说明
err_code	错误码
err_msg	错误信息
ret_data	返回数据（根据不同命令而不同）

record_video、reset_device、smart_config、config_wifi 命令返回的 JSON 示例：

```
{
  "err_code": 0,
  "err_msg": "success",
  "ret_data": {}
}
```

capture_picture 命令返回的 JSON 示例：

```
{
  "err_code": 0,
  "err_msg": "success",
  "ret_data": {
    "uuid": "2a376e9e-65ea-11e5-8000-e40524d611af"
  }
}
```

get_wifi_info 命令返回的 JSON 示例：

```
{
  "err_code": 0,
  "err_msg": "success",
  "ret_data": {
    "status": "connected",
    "ssid": "123456",
    "signal_strength": 80,
    "security": "wpa2-psk",
    "mac_addr": "AABBCCDDEEFF"
  }
}
```

JSON 字段	说明
---------	----

status	连接情况（connected、disconnected）
ssid	wifi 名称
signal_strength	信号强度（百分比）
security	加密方式（none、wep、wpa-psk、wpa-enterprise、wpa2-psk）
mac_addr	无线网卡 MAC 地址

get_wifi_list 命令返回的 JSON 示例：

```
{
  "err_code": 0,
  "err_msg": "success",
  "ret_data": {
    "wifi_ssid": [
      {
        "ssid": "EZVIZ Wi-Fi",
        "mode": 0,
        "signal_strength": 16,
        "security": "none",
        "ap_address": "44:19:B6:AF:F7:DB"
      }, {
        "ssid": "EZVIZ Free Wi-Fi_yx",
        "mode": 0,
        "signal_strength": 16,
        "security": "none",
        "ap_address": "44:19:B6:AF:F7:DC"
      }
    ]
  }
}
```

JSON 字段	说明
wifi_ssid	JSON 数组
ssid	wifi 名称
mode	0：普通模式，1：ad-hoc
signal_strength	信号强度（百分比）
security	加密方式（none、wep、wpa-psk、wpa-enterprise、wpa2-psk）
ap_address	ap 的物理地址

6) 用例

```
const char *capture_picture_str = "{\"cmd\":\"capture_picture\"}";
HIK_SDK_IPC_NAME device_name;
char *data = NULL;

strcpy(device_name.serial, "123456789");
int ret = hik_sdk_cmd(&device_name, capture_picture_str, &data);
if(data != NULL)
{
}
```

```
printf("%s\n", data);  
free(data);  
data = NULL;  
}
```

7) 注意事项

1. 如果运行 SDK 的设备与 IPC 采用 TCP Socket 的方式通信，接收控制命令的 IPC 必须为已发现的 IPC。
2. IPC 需要开通云存储。
3. 抓图得到的图像存在萤石云，可通过 uuid 从萤石开放平台获取。
4. 录像命令所录的视频存在萤石云与本地 SD 卡。视频录制的时长根据设备配置情况而定。
5. 出参 out_param 的内存由 SDK 分配，用户必须判断 out_param 是否为空指针，非空则调用 free 进行释放。

3.3.6 注册通知消息接收的回调函数

1) 功能描述

SDK 可接收与之授权过的 IPC 推送的通知消息，当接收到 IPC 发送过来的通知时，SDK 会调用此函数注册的回调函数。

2) API 定义

```
///  
/// @brief 注册通知消息接收的回调函数，当接收到 IPC 推送的通知时会被调用  
/// @param recv_notify_cb 通知消息接收回调函数  
/// @return 错误码  
///  
int hik_sdk_set_notify_callback(HIK_SDK_ON_RECV_NOTIFY recv_notify_cb);
```

3) 用例

```
int recv_notify(char *msg, char *data, int len)  
{  
    if(msg != NULL)  
    {  
        printf("%s\n", msg);  
        free(msg);  
        msg = NULL;  
    }
```

```
}  
if(data != NULL)  
{  
    free(data);  
    data = NULL;  
}  
return 0;  
}  
void sample()  
{  
    hik_sdk_set_notify_callback(recv_notify);  
}
```

4) 注意事项

msg 和 **data** 的内存均由 SDK 分配，用户必须在回调函数中判断 **msg** 和 **data** 是否为空指针，非空则调用 **free** 进行释放。

3.3.7 结束 SDK

1) 功能描述

释放资源，结束线程。

2) API 定义

```
///  
/// @brief 结束 sdk，结束线程，释放资源  
/// @return 错误码  
///  
int hik_sdk_finish();
```

3) 用例

```
hik_sdk_finish();
```

4) 注意事项

无

3.4 回调函数

1) 功能描述

互联互通 SDK 接收到 IPC 的通知消息时，调用回调函数，用户在回调函数中处理通知

消息。回调函数原型为 `int (*HIK_SDK_ON_RECV_NOTIFY)(char *msg, char *data, int len)`，其中 `msg` 为 JSON 格式消息内容，`data` 为数据，`len` 为数据长度。

2) JSON 格式说明

3) JSON 字段	说明
<code>notify_type</code>	通知类型
<code>device_name</code>	发送通知消息的 IPC 序列号
<code>time</code>	通知消息发送时间
<code>msg</code>	通知消息内容（根据不同通知类型而不同）

当通知类型为 `alarm` 时，代表由 IPC 发送的报警信息，此时 `msg` 字段中的 `alarm_type` 字段表示报警信息来源，可能为 `motdetect`（移动侦测）或者 `pir`（人体红外感应）。`alarm_time` 字段表示报警产生的时间。

IPC 发送的移动侦测触发的报警信息的 JSON 示例如下：

```
{
  "notify_type": "alarm",
  "device_name": "123456789",
  "time": "1443443000",
  "msg": {
    "alarm_type": "motdetect",
    "alarm_time": "1443443000"
  }
}
```

4) 注意事项

人体红外报警需 IPC 设备硬件支持。