

Android SDK 开发说明文档 v1.2

IPCamerDemo 为网络摄像机基础功能的演示，代码仅作为开发参考，部分功能需要开发者根据文档自行添加。如果代码中出现乱码请将 workspace 语言环境设置为 utf-8。

1,二次开发所必须的相关文件:

jni: Libs/armeabi/libvstc2_jni.so
jni 接口: vstc2.nativecaller 整个目录及文件
jni 回调 service: BridgeService.java

以上相关请不要修改，直接添加到工程中。

2,视频连接流程:

1. 启动服务 BridgeService;
2. 初始化服务器 NativeCaller.PPPPInitial
3. 初始化回调 NativeCaller.Init();
4. 开启 p2p 连接 StartPPPP;
5. p2p 返回在线之后 开启视频流(视频画面) StartPPPPLivestream
- 6 关闭视频流 StopPPPPLivestream
- 7 断开 p2p 连接 StopPPPP
- 8 释放 p2p 资源 NativeCaller.Free();

具体流程请参考 demo 程序.

3, 视频连接流程相关参数说明:

StartPPPP 之后 BridgeService 中 PPPPMsgNotify(String did, int type, int param)

返回摄像机连接过程

did:摄像机序列号

type: 0 摄像机状态返回, 1 当前连接为转发模式

param:

```
public static final int PPPP_STATUS_CONNECTING = 0; //连接中
public static final int PPPP_STATUS_INITIALING = 1; //已连接, 正在初始化
public static final int PPPP_STATUS_ON_LINE = 2; //在线
public static final int PPPP_STATUS_CONNECT_FAILED = 3; // 连接失败
public static final int PPPP_STATUS_DISCONNECT = 4; // 连接已关闭
public static final int PPPP_STATUS_INVALID_ID = 5; //无效UID
public static final int PPPP_STATUS_DEVICE_NOT_ON_LINE = 6; //不在线
public static final int PPPP_STATUS_CONNECT_TIMEOUT = 7; //连接超时
public static final int PPPP_STATUS_WRONGUSER_RIGHTPWD = 8; //密码错误..
public static final int PPPP_STATUS_WRONGPWD_RIGHTUSER = 9; // 密码错误.
public static final int PPPP_STATUS_WRONGPWD_WRONGUSER = 10; // 密码错误.
```

StartPPPPLivestream 之后BridgeService中VideoData (String did, byte[] videobuf, int h264Data, int len, int width, int height, int timestamp, short milistamp, int

sessid, **int** version, **int** originFrameLen)
did:摄像机序列号
videobuf:一帧视频数据
h264Data:0 普清摄像机数据, 1 高清摄像机数据
len:一帧数据大小
width:一帧数据宽
height:一帧数据高
后面几个参数暂时不用处理。

具体视频处理见 demo 中 playactivity

4,移动侦测报警部分:

1, 查询当前摄像机报警设置

NativeCaller.PPPPGetSystemParams(did, 4);

BridgeService 中 Callback_AlarmParams 返回当前摄像机的报警相关参数, 其中 motion_armed (0 关闭移动侦测, 1 开启移动侦测)

2, 设置报警信息

NativeCaller.PPPPAlarmSetting 其中 motion_armed (0 关闭移动侦测, 1 开启移动侦测), 其他参数可按照上面接口返回的传过去即可

其他详细参数说明见 TIP-Camera-CGI.pdf 中 get_params.cgi 的报警参数组对应参数说明

3, 报警接口返回

当摄像机设置为开启移动侦测时, 摄像机收到移动侦测会在 BridgeService 中的 Callback_AlarmNotify 中返回提示

```
public void Callback_AlarmNotify(String did, int alarmtype) {}
```

did:摄像机序列号

alarmtype: 1 移动侦测, 2 IO 报警返回

5,本地录像:

1, 普清摄像机录像:

即为 VideoData 中 h264Data=0 的时候可以直接将摄像机视频数据返回的每一帧保存即可。

2, 高清录像

高清录像需要请求原始数据, 所以和普清不一样 在录像的时候需要调用 NativeCaller.RecordLocal (did, 1);

返回数据接口为 BridgeService 中的 Callback_H264Data, 然后将每一帧视频数据保存即可;

6,联动部分:

联动部分所有指令都是通过透传 cgi 指令完成, 透传接口为 NativeCaller.TransferMessage(did, cmd, 1);

其中 cmd 为 cgi 指令.

例如要获取摄像机类型

String cmd=" get_status.cgi?loginuse=admin&loginpas=pwd&user=admin&pwd=pwd" ;其中 pwd 为相机的密码然后发送指令即可收到接口返回;

部分指令:

获取布撤防状态

```
"get_sensorstatus.cgi?loginuse=admin&loginpas=" + pwd+ "&user=admin&pwd="
```

布防:

```
"set_sensorstatus.cgi?cmd=0&loginuse=admin&loginpas=" + pwd+ "&user=admin&pwd=" +  
pwd
```

撤防:

```
"set_sensorstatus.cgi?cmd=1&loginuse=admin&loginpas=" + pwd+ "&user=admin&pwd=" +  
pwd
```

获取绑定的传感器:

```
"get_sensorlist.cgi?loginuse=admin&loginpas=" + pwd+ "&user=admin&pwd=" + pwd
```

获取绑定的看守位:

```
"get_sensor_preset.cgi?cmd=1&sensorid=0&loginuse=admin&loginpas="+pwd+"&user=admin&pwd=" + pwd
```

设置看守位:

```
"set_sensor_preset.cgi?sensorid="+ 255 + "&presetid=" + pos +  
"&loginuse=admin&loginpas=" + pwd
```

pos 可以设置 0-5

开关门铃声音:

```
"set_sensorstatus.cgi?cmd=4&doorbell=1&loginuse=admin&loginpas="+pwd+"&user=admin&pwd=" + pwd
```

```
"set_sensorstatus.cgi?cmd=4&doorbell=0&loginuse=admin&loginpas="+pwd+"&user=admin&pwd=" + pwd
```

编辑传感器名称:

```
"set_sensorname.cgi?" + "&sensorid=" +  
id + "&sensorname=" + name + "&loginuse=admin&loginpas=" +  
pwd
```

Id 为返回传感器时候的 index

Name 为要设置的传感器名称

开启对码状态:

```
"set_sensorstatus.cgi?cmd=2&loginuse=admin&loginpas=" + pwd+ "&user=admin&pwd=" +  
pwd
```

关闭对码:

```
"set_sensorstatus.cgi?cmd=3&loginuse=admin&loginpas=" + pwd+ "&user=admin&pwd=" +  
pwd
```

删除传感器:

```
"del_sensor.cgi?" + "&sensorid="+ id + "&loginuse=admin&loginpas=" + pwd+  
"&user=admin&pwd=" + pwd
```

Id 为返回传感器时返回的 index

设置报警预制位:

```
"set_sensor_preset.cgi?sensorid="+ id + "&presetid=" + pos + "&sensorid=" +
```

sensorid
+ "&loginuse=admin&loginpas=" + pwd
Id 为返回传感器时返回的 index
Pos 0-5

传感器类型:

SENSOR_TYPE_DOOR = 0x01, //门磁
SENSOR_TYPE_INFRARED = 0x02, //红外
SENSOR_TYPE_SMOKE = 0x03, //烟感
SENSOR_TYPE_SMELL = 0x04, //气感
SENSOR_TYPE_REMOTE = 0x07, //遥控
SENSOR_TYPE_CAMERA = 0x0A, //摄像头
SENSOR_TYPE_CURTAIN = 0x0B //幕帘

7, NativeCaller 相关接口参数说明:

部分参数详细参数说明见 TIP-Camera-CGI.pdf

//初始化服务器
void PPPPInitial(string svr);
//开始搜索, 创建多播/广播, 默认使用广播
Void StartSearch();
//停止搜索
void StopSearch();
//初始化视频解码器
Void Init();
//释放并关闭解码器, 释放 PPPP 资源
void Free();
//开启 P2P 服务, 传入 UID 、用户名、密码、1、""
int StartPPPP(String, String, String, int,String);
//停止 P2P 服务
int StopPPPP(string);
//请求视频数据, 传入 uid、streamid、substreamid
int StartPPPLLivestream(string , int, int);
//停止视频数据, 传入 uid
int StopPPPLLivestream(string);
//云台控制, 第一个参数 uid , 第二个参数控制命令
int PPPPTZControl(string, int);
//图像传感器参数设置, 三个参数: uid, 摄像头参数类型, 摄像头参数值
int PPPPCameraControl(string, int, int);
//请求 Camera 传来的声音, 传入 uid
int PPPPStartAudio(string);
int PPPPStopAudio(string);
//发送语音数据, 传入 uid

```

int PPPPStartTalk(string);
int PPPPStopTalk(string);
int PPPPTalkAudioData(string, byteArray, int);
//探测网络信息
int PPPPNetworkDetect();
//初始化回调函数环境
int PPPPSetCallbackContext(object context);
//获取系统参数，传入 uid 、消息类型，消息类型在 P2P_API_Define.h 中定义
int PPPPGetSystemParams(string did, int paramType);
//重启设备
int PPPPRebootDevice(string did);
//恢复出厂设置
int PPPPRestorFactory(string did);
//设置设备 wifi 参数, set_wifi.cgi
int PPPPWifiSetting(string did, int enable, string ssid, int channel, int mode,int authtype, int
encryp, int keyformat,
                int defkey, string key1, string key2, string key3, string key4, int key1_bits, int
key2_bits, int key3_bits, int key4_bits, string wpa_psk);
//设置设备基本网络参数, set_network.cgi
int PPPPNetworkSetting(string did, string ipaddr, string netmask, string gateway, string dns1,
string dns2, int dhcp, int port, int rtspport);
//设置设备用户参数，包括访客、操作者、管理员参数， set_users.cgi
int PPPPUserSetting( string did, string user1, string pwd1, string user2, string pwd2, string user3,
string pwd3);
//设置设备日期时间、参数见 set_datetime.cgi
int PPPPDatetimeSetting( string did, int now, int tz, int ntp_enable, string ntp_svr);
//设置设备 DDNS 、参数见 set_ddns.cgi
int PPPPDDNSSetting( string did, int service, string user, string pwd, string host, string proxy_svr,
int ddns_mode, int proxy_port);
//邮箱设置， set_mail.cgi
int PPPPMailSetting( string did, string svr, int port, string user,
                string pwd, int ssl, string sender, string receiver1, string receiver2, string receiver3,
string receiver4);
//Ftp 设置， set_ftp.cgi
int PPPPFtpSetting( string did, string svr_ftp, string user, string pwd, string dir, int port, int mode,
int upload_interval);
//摄像机云台参数设置， set_misc.cgi
int PPPPPTZSetting( string did, int led_mod, int ptz_center_onstart, int ptz_run_times, int
ptz_patrol_rate, int ptz_patrul_up_rate,
                int ptz_patrol_down_rate, int ptz_patrol_left_rate, int ptz_patrol_right_rate, int
disable_preset);
//设置设备报警选项， set_alarm.cgi
int PPPPAlarmSetting( string did, int motion_armed, int motion_sensitivity, int input_armed, int
ioin_level, int iolinkage, int ioout_level,int alarmpresetsit,

```

```

        int mail, int snapshot, int record, int upload_interval, int schedule_enable, int
schedule_sun_0, int schedule_sun_1, int schedule_sun_2,
        int schedule_mon_0, int schedule_mon_1, int schedule_mon_2, int
schedule_tue_0, int schedule_tue_1, int schedule_tue_2, int schedule_wed_0, int
schedule_wed_1, int schedule_wed_2,
        int schedule_thu_0, int schedule_thu_1, int schedule_thu_2, int schedule_fri_0,
int schedule_fri_1, int schedule_fri_2,
        int schedule_sat_0, int schedule_sat_1, int schedule_sat_2);
//设置录像, set_recordsch.cgi
int PPPPSDRecordSetting( string did,
        int record_cover_enable, int record_timer, int record_size, int record_time_enable,
int record_schedule_sun_0, int record_schedule_sun_1, int record_schedule_sun_2,
        int record_schedule_mon_0, int record_schedule_mon_1, int
record_schedule_mon_2, int record_schedule_tue_0, int record_schedule_tue_1, int
record_schedule_tue_2,
        int record_schedule_wed_0, int record_schedule_wed_1, int
record_schedule_wed_2, int record_schedule_thu_0, int record_schedule_thu_1, int
record_schedule_thu_2,
        int record_schedule_fri_0, int record_schedule_fri_1, int record_schedule_fri_2,
int record_schedule_sat_0, int record_schedule_sat_1, int record_schedule_sat_2);
//获取录像文件列表, get_record_file.cgi
int PPPPGetSDCardRecordFileList( string did, int startTime, int endTime);
//开始回放, livestream.cgi
int StartPlayBack( string did, string filename, int offset, int picTag);

int StopPlayBack( string did);
//yuv420 转 rgb565
int YUV420RGB565( byteArray yuv, byteArray rgb, int width, int height);
//解码 h264 的帧
int DecodeH264Frame( byteArray h264, int bIframe, byteArray yuv, int length, intArray size);

int SetBill( string did, string svr, int port, string user, string pwd, int reboottime);
//set_syswifi.cgi
int SetAPPParam( string did, string ssid, string pwd);

//更新固件
int UpgradeFirmware( string did, string svr, string fpath, int ftype);
//xq-06-18
int FormatSD( string did);

//透传 CGI
int TransferMessage( string did, string msg, int len);

```

1, HZ:

50hz:NativeCaller.PPPPCameraControl(did, 3, 0);

60hz:NativeCaller.PPPPCameraControl(did, 3, 1);

2, digital zoom:

放大: NativeCaller.PPPPCameraControl([did](#), 18, 1);

停止: NativeCaller.PPPPCameraControl([did](#), 18, 0);

缩小: NativeCaller.PPPPCameraControl(strDID, 17, 1);

停止: NativeCaller.PPPPCameraControl(strDID, 17, 0);

3, wifi 设置:

设置方法

NativeCaller.PPPWifiSetting

参数说明见 cgi 文档中 set_wifi.cgi 参数说明;

流程:

① 请求当前摄像机 wifi 设置参数 NativeCaller.PPPGetSystemParams(did, 4);
服务里面 Callback_WifiParams 方法返回参数 参数说明见 cgi 文档中的
get_params.cgi 里无线参数组;

② 获取摄像机附近的 wifi 信息 NativeCaller.PPPGetSystemParams(did, 20);
服务里返回 **Callback_WifiScanResult** 参数说明:

* @param [did](#) 摄像机[uid](#)
* @param [ssid](#) [wifi](#) [ssid](#)
* @param [mac](#) [wifi](#) [mac](#)
* @param security
0->指WEP-NONE
1->指WEP
2->WPA-PSK TKIP
3->WPA-PSK AES
4->WPA2-PSK TKIP
5->WPA2-PSK AES
* @param [dbm0](#) 信号强度
* @param [dbm1](#)
* @param mode 工作模式, 0->[infra](#) 1->[adhoc](#)
* @param channel 通道
* @param bEnd 1 搜索完毕

③ 选择对应的 w i f i 信息进行设置;

4. s d 卡设置:

设置方法: NativeCaller.PPPPSDRecordSetting

参数见 cgi 文档中 set_recordsch.cgi

其中录像时间是将一天时间分为 3 段, 比如 schedule_sun_0, schedule_sun_1,
schedule_sun_2 每段分为 32 小段, 每段 15 分钟, 录像为 1, 不录为 0

获取当前 sd 卡状态:

`NativeCaller.PPPPGetSystemParams(did, 22)` 服务返回方法
`callBackRecordSchParams` 参数说明见 `cgi` 文档 `get_record.cgi` 中的参数说明。

5.报警设置:

设置方法: `NativeCaller.PPPPAAlarmSetting`

参数说明见 `cgi` 文档中 `set_alarm.cgi`

获取当前摄像机报警设置:

`NativeCaller.PPPPGetSystemParams(did, 4);` 服务返回方法
`CallBack_AlarmParams` 参数说明见 `cgi` 文档中 `get_params.cgi` 里报警参数组;

`PPPPAlarmSetting` 中 `motion_armed=1` 时即为开启移动侦测, 当发生移动侦测报警时
`BridgeService` 中 `CallBack_AlarmNotify(String did, int alarmtype)`会有返回, `did`
为发生移动侦测的摄像机 `uid`, `alarmtype=1` 时为移动侦测报警, `alarmtype=2` 为 IO 联动报警