java sdk 接口使用说明文档

目录

1. 封面

本手册将对设备网络 SDK 的 API 接口进行介绍，为正确对接出入口相机，请阅读本手册。

2. 前言

本节介绍本文档的内容、对应的车辆检测算法版本、适用的读者对象等。

SDK

SDK 名称	简称	sdk 版本
设备网络 SDK	ice_ipcsdk.jar	V1.4.0.82

读者对象

本文档（本指南）主要适用于以下工程师：

- 平台开发人员

3. 修订记录

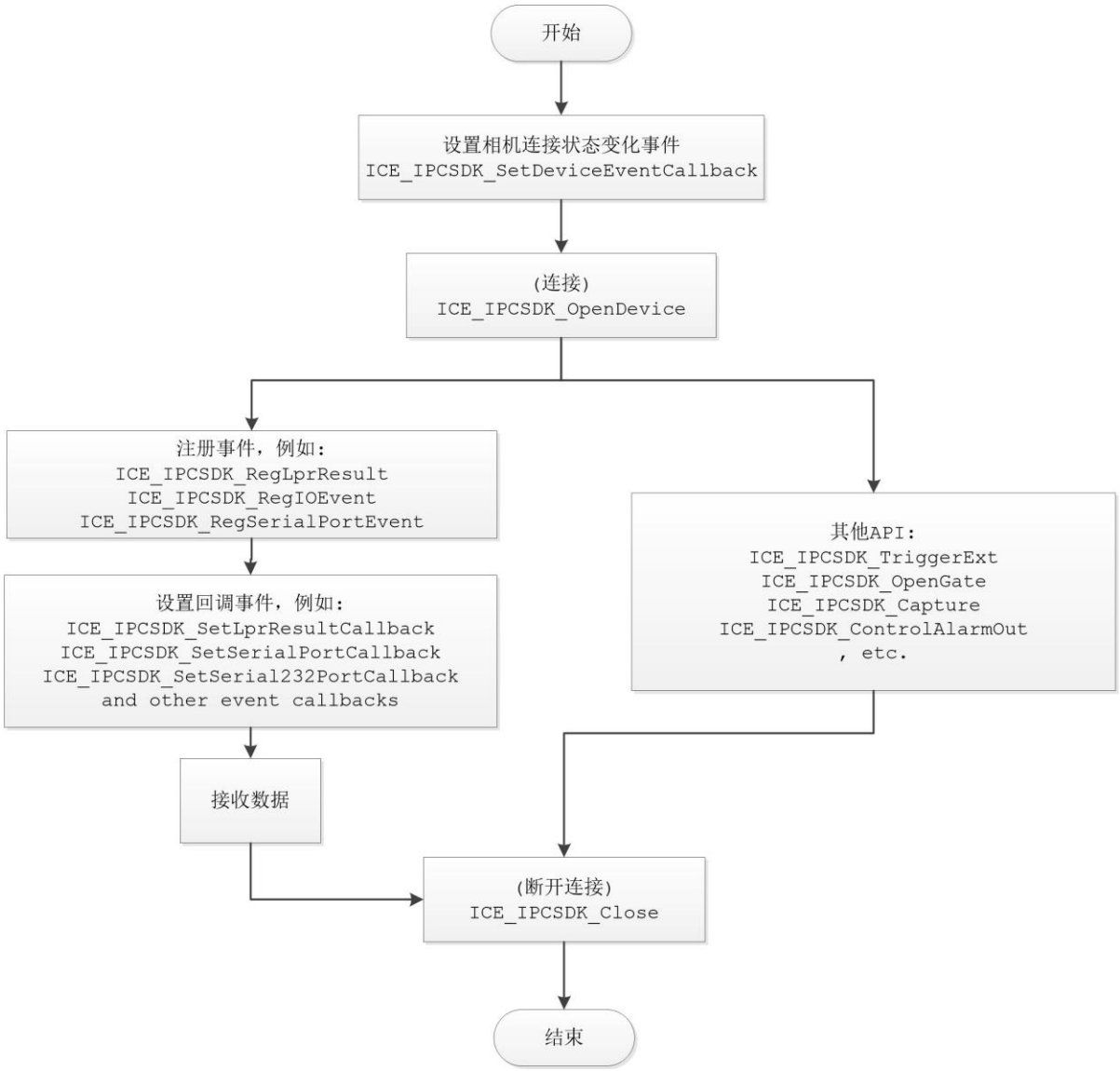
修订日期	修订版本	修订说明
2021-10-09	V1.0.0	第一版

4. 编程说明

4.1 开发环境

SDK 支持系统：windows32 或 64 系统；Linux 操作系统
支持语言：java,android

4.2 接口调用流程


开始
设置相机连接状态变化事件
ICE_IPCSDK_SetDeviceEventCallback
(连接)
ICE_IPCSDK_OpenDevice
注册事件，例如：
ICE_IPCSDK_RegLprResult
ICE_IPCSDK_RegIOEvent
ICE_IPCSDK_RegSerialPortEvent
设置回调事件，例如：
ICE_IPCSDK_SetLprResultCallback
ICE_IPCSDK_SetSerialPortCallback
ICE_IPCSDK_SetSerial232PortCallback
and other event callbacks
接收数据
其他API：
ICE_IPCSDK_TriggerExt
ICE_IPCSDK_OpenGate
ICE_IPCSDK_Capture
ICE_IPCSDK_ControlAlarmOut
, etc.
(断开连接)
ICE_IPCSDK_Close
结束

4.3 使用建议

SDK 使用时，除引入 ice_ipcsdk.jar 外，请同步引入相关的库 zip4j_1.3.2.jar，json 解析依赖的库（fastjson-1.2.70.jar）以及 Java-WebSocket-1.3.8.jar。

5. 接口定义

5.1 设备连接

5.1.1 设备连接

5.1.1.1 连接相机

```
int ICE_IPCSDK_OpenDevice(String strIP)
```

参数：
1.相机 ip（输入）

返回值：
1 连接成功
0 连接失败

5.1.1.2 连接相机(使用密码)

```
int ICE_IPCSDK_OpenDevice_Passwd(String strIP, String strPasswd,
String strReserve)
```

参数：

1．相机 ip（输入）
2．相机密码（输入）
3．预留参数（输入）

返回值：

1 连接成功
0 连接失败

5.1.1.3 断开连接

```
void ICE_IPCSDK_Close()
```

参数：
无

返回值：
无

5.1.2 设备连接状态

5.1.2.1 查询设备是否在线

```
boolean ICE_IPCSDK_GetStatus()
```

参数：
无

返回值：
布尔值，是否在线。

5.1.2.2 设置相机事件回调

```
public void ICE_IPCSDK_SetDeviceEventCallBack(IDeviceEventCallback
callback)
```

参数：

1．相机事件回调，详情见 IDeviceEventCallback(See 5.1.2.3)（输入）

返回值：

无

5.1.2.3 相机事件回调

```
public interface IDeviceEventCallback {
       public void ICE_IPCSDK_OnDeviceEvent(
           String strIP,
           int nEventType,
           int nEventData1,
           int nEventData2,
           int nEventData3,
           int nEventData4
       );
   }
```

参数：

1.相机 ip（输出）
2.事件类型 0：离线 1：在线（输出）
3.预留参数（输出）
4.预留参数（输出）
5.预留参数（输出）
6.预留参数（输出）

5.2 车牌识别

5.2.1 手动软触发

5.2.1.1 手动触发车牌识别（软触发）

```
int ICE_IPCSDK_TriggerExt()
```

参数：
无

返回值：
0 失败
1 成功

5.2.2 识别数据接收

5.2.2.1 注册车牌识别事件

```
int ICE_IPCSDK_RegLprEvent(boolean bNeedPic)
```

参数：

1.是否需要图片（输入）

返回值：

0 失败

1 成功

5.2.2.2 设置车牌识别事件回调

```
void ICE_IPCSDK_SetLprResultCallback(ILprResultCallback callback)
```

参数：

1.车牌识别事件回调，详情见 ILprResultCallback（输入）

返回值：

无

5.2.2.3 车牌识别事件

```
public interface ILprResultCallback {
       public void ICE_IPCSDK_LprResult(
           String strIP,
           LprResult lprResult,
           byte[] bData,
           int iFullPicOffset,
           int iFullPicLen,
           int iPlatePicOffset,
           int iPlatePicLen,
           int iReserve1,
           int iReserve2
       );
   }
```

参数：

1.相机 ip（输出）
2.识别结果（输出）
3.图片数据（输出）
4.全景图偏移量（输出）
5.全景图大小（输出）
6.车牌图偏移量（输出）
7.车牌图大小（输出）
8.预留参数（输出）
9.预留参数（输出）

LprResult：

```
public class LprResult {
   //plateInfo
   private String strPlateNum = "";//车牌号
   private String strPlateColor = "";//车牌颜色
   private boolean bFalsePlate = false;//是否为虚假车牌
   private float fConfidence = 0;//打分值
   private int iPlateType = 0;//车牌类型
   private int iRectLeft = 0;//矩形框
   private int iRectTop = 0;
   private int iRectRight = 0;
   private int iRectBottom = 0;

   //vehicleInfo
   private String strVehicleType = "";//车辆类型
```

```
private String strVehicleColor = "";//车身颜色
private String strVehicleDir = "";//车辆方向
private String strLogName = "";//品牌
private String strSubLogName = "";//子品牌
private String strProductYearName = "";//生产年代

private String strLprState = "";//结果状态(在线实时结果/断网续传结果)
private int iHadPlate = 0;//是否有车牌
private String strTriggerType = "";//触发类型
private String strPlateState = "";//车牌状态
private String strBwList = "";//黑白名单
private String strCapTime = "";//抓拍时间

private int iFullPicLen = 0;
private int iPlatePicLen = 0;
public String getStrPlateNum() {
    return strPlateNum;
}
public String getStrPlateColor() {
    return strPlateColor;
}
public boolean isbFalsePlate() {
    return bFalsePlate;
}
public float getfConfidence() {
    return fConfidence;
}
public int getiPlateType() {
    return iPlateType;
}
public int getiRectLeft() {
    return iRectLeft;
}
public int getiRectTop() {
    return iRectTop;
}
public int getiRectRight() {
    return iRectRight;
}
public int getiRectBottom() {
    return iRectBottom;
}
public String getStrVehicleType() {
    return strVehicleType;
```

```
    }
    public String getStrVehicleColor() {
        return strVehicleColor;
    }
    public String getStrVehicleDir() {
        return strVehicleDir;
    }
    public String getStrLogName() {
        return strLogName;
    }
    public String getStrSubLogName() {
        return strSubLogName;
    }
    public String getStrProductYearName() {
        return strProductYearName;
    }
    public String getStrLprState() {
        return strLprState;
    }
    public int getiHadPlate() {
        return iHadPlate;
    }
    public String getStrTriggerType() {
        return strTriggerType;
    }
    public String getStrPlateState() {
        return strPlateState;
    }
    public String getStrBwList() {
        return strBwList;
    }
    public String getStrCapTime() {
        return strCapTime;
    }
    public int getiFullPicLen() {
        return iFullPicLen;
    }
    public int getiPlatePicLen() {
        return iPlatePicLen;
    }
}
```

返回值：

无

5.2.2.4 注销车牌识别事件

```
int ICE_IPCSDK_UnregLprEvent()
```

参数：

无

返回值：

0 失败

1 成功

5.3 实时视频和抓拍

5.3.1 实时视频

5.3.1.1 mjpeg 码流回调

```
public interface IMJpegCallback_Static {
   public void ICE_IPCSDK_MJpeg(String strIP, byte[] bData, int
length);
}
```

参数：
1.相机 ip（输出）
2.mjpeg 数据（输出）
3.jpg 图片长度（输出）

返回值：无

5.3.1.2 设置 mjpeg 码流回调

```
public void ICE_IPCSDK_SetMJpegallback_Static(IMJpegCallback_Static
callback)
```

参数：

1.mjpeg 码流回调，详情见 IMJpegCallback Static(See 5.3.1.1)（输入）

返回值：无

5.3.2 图片抓拍

5.3.2.1 只抓拍，不识别

```
CaptureResult ICE_IPCSDK_Capture()
```

参数：
无

返回值：
抓拍结果，定义如下：

```
public class CaptureResult {
        public byte[] picdata; //抓拍图片数据
    }
```

失败时，返回 null

5.4 设备管理

5.4.1 获取设备 mac 地址

```
public String ICE_IPCSDK_GetDevID()
```

参数：
无

返回值：
设备的 mac 地址

5.4.2 设置网络参数

```
public int ICE_IPCSDK_SetIPAddr(String strIP, String strMask, String
strGateway)
```

参数：

1.ip（输入）
2.掩码（输入）
3.网关（输入）

返回值：

0 失败
1 成功

5.4.3 获取网络参数

```
public NetworkParam ICE_IPCSDK_GetIPAddr()
```

参数：
无

返回值：
网络参数结果，定义如下：

```
public class NetworkParam {
    private String ip = "";//ip
    private String mask = "";//掩码
    private String gateWay = "";//网关
    public String getIp() {
        return ip;
    }
    public void setIp(String ip) {
        this.ip = ip;
    }
    public String getMask() {
        return mask;
    }
    public void setMask(String mask) {
        this.mask = mask;
    }
    public String getGateWay() {
        return gateWay;
    }
    public void setGateWay(String gateWay) {
        this.gateWay = gateWay;
    }
}
```

获得失败时返回 null

5.4.4 获取相机 UID

```
public String ICE_IPCSDK_GetUID()
```

参数：
无

返回值：
相机 UID,失败时返回 null

5.4.5 获取相机时间

```
public TimeCfgInfo ICE_IPCSDK_GetTime()
```

参数：

1.无

返回值：

null 失败,成功时返回时间信息，时间参数定义如下：

```
public class TimeCfgInfo{
       public int iYear;//年
       public int iMon;//月
       public int iDay;//日
       public int iHour;//时
       public int iMin;//分
       public int iSec;//秒
   }
```

5.4.6 时间同步

```
public int ICE_IPCSDK_SyncTime(int sYear, int bMonth, int bDay, int
bHour, int bMin, int bSec)
```

参数：

1.年（输入）
2.月（输入）
3.日（输入）
4.时（输入）
5.分（输入）
6.秒（输入）

返回值：

0 失败
1 成功

注意：相机为 utc 时间，此接口需要传入 utc 时间

5.5 外设管理

5.5.1 打开道闸

```
public int ICE_IPCSDK_OpenGate()
```

参数：
无

返回值：
0 失败
1 成功

5.5.2 控制开关量输出

```
public int ICE_IPCSDK_ControlAlarmOut(int nIndex)
```

参数：

1.开关量输出索引。（输入）

0：开关量 1 输出，即打开道闸

1：开关量 2 输出

2：开关量 3 输出

3：开关量 4 输出

返回值：

0 失败

1 成功

5.5.3 获取 IO 状态

```
public ICE_IOState ICE_IPCSDK_GetIOState()
```

参数：
无

返回值：
IO 状态，定义如下：

```
public class ICE_IOState{
       public int[] nIOState = new int[4];
    }
```

数组下标：0 为 IO1 状态，1 为 IO2 状态，2 为 IO3 状态，3 为 IO4 状态

失败时，返回 null

5.5.4 注册 IO 状态变化事件

```
int ICE_IPCSDK_RegGpioEvent()
```

参数：

无

返回值：

0 失败

1 成功

5.5.5 设置 IO 状态变化事件回调

```
void ICE_IPCSDK_SetIOEventCallBack(IIOEventCallback callback)
```

参数：

1.IO 状态变化事件，详情见 IIOEventCallback

返回值：

无

5.5.6 IO 状态变化事件

```
public interface IIOEventCallback {
       public void ICE_IPCSDK_OnIOEvent(
           String strIP,
           int nEventType,
           int nEventData1,
           int nEventData2,
           int nEventData3,
           int nEventData4
       );
    }
```

参数：

1.相机 ip
2.事件类型，当为 0 时，表示 IO 状态变化事件
3.当事件类型为 0 时，表示 IO1 状态
4.当事件类型为 0 时，表示 IO2 状态
5.当事件类型为 0 时，表示 IO3 状态
6.当事件类型为 0 时，表示 IO4 状态

返回值：

无

5.5.7 注销 IO 状态变化事件

```
int ICE_IPCSDK_UnregGpioEvent()
```

参数：

无

返回值：

0 失败

1 成功

5.6 相机配置

5.6.1 黑白名单配置

5.6.1.1 获取黑白名单参数

```
public WhiteListParam ICE_IPCSDK_GetWhiteListParam()
```

参数：
无

返回值：
失败时返回 null
黑白名单配置参数

```
public class WhiteListParam {
    private int nWhiteListMode = 0;     //白名单模式：0表示断网后关联开闸，1表示实时关联开闸,2 不关联开闸
    private int nWhiteListMatch = 0;    //白名单匹配相似，范围（60%-100%），默认：80%
    private int nBlackListMode = 0;     //黑名单关联道闸模式，0 不关联开闸，1 实时关联开闸
    private int nBlackListMatch = 0;    //黑名单匹配相似，范围（60%-100%），默认：80%
    private int nTempListMode = 0;      //临时车关联道闸模式，0 不关联开闸，1 实时关联开闸
    private int ignoreHZ_flag = 0;      //忽略汉字标志，0 不忽略，1 忽略
    private int allow_unmatch_chars_cnt = 0; // 允许不匹配的字符个数范围[0 ~3]
    private int new_version = 1;        //新旧黑白名单配置标志，1：新 0：旧
    private int Jing_mode = 0;          //警车模式，0 不关联开闸，1 实时关联开闸
    private int Army_mode = 0;          //军车模式，0 不关联开闸，1 实时关联开闸
    private int NewEnergy_mode = 0;     //新能源车牌模式，0 不关联开闸，1 实时关联开闸
    private int EmergencyMode = 0;      //应急车牌模式，0 不关联开闸，1 实时关联开闸
```

```
    private int Antifake_mode = 0;  //防伪车模式，1 不关联开闸，0 实时关联
开闸
    public int getnWhiteListMode() {
        return nWhiteListMode;
    }
    public void setnWhiteListMode(int nWhiteListMode) {
        this.nWhiteListMode = nWhiteListMode;
    }
    public int getnWhiteListMatch() {
        return nWhiteListMatch;
    }
    public void setnWhiteListMatch(int nWhiteListMatch) {
        this.nWhiteListMatch = nWhiteListMatch;
    }
    public int getnBlackListMode() {
        return nBlackListMode;
    }
    public void setnBlackListMode(int nBlackListMode) {
        this.nBlackListMode = nBlackListMode;
    }
    public int getnBlackListMatch() {
        return nBlackListMatch;
    }
    public void setnBlackListMatch(int nBlackListMatch) {
        this.nBlackListMatch = nBlackListMatch;
    }
    public int getnTempListMode() {
        return nTempListMode;
    }
    public void setnTempListMode(int nTempListMode) {
        this.nTempListMode = nTempListMode;
    }
    public int getIgnoreHZ_flag() {
        return ignoreHZ_flag;
    }
    public void setIgnoreHZ_flag(int ignoreHZ_flag) {
        this.ignoreHZ_flag = ignoreHZ_flag;
    }
    public int getAllow_unmatch_chars_cnt() {
        return allow_unmatch_chars_cnt;
    }
    public void setAllow_unmatch_chars_cnt(int
allow_unmatch_chars_cnt) {
        this.allow_unmatch_chars_cnt = allow_unmatch_chars_cnt;
```

```
    }
    public int getNew_version() {
        return new_version;
    }
    public void setNew_version(int new_version) {
        this.new_version = new_version;
    }
    public int getJing_mode() {
        return Jing_mode;
    }
    public void setJing_mode(int jing_mode) {
        Jing_mode = jing_mode;
    }
    public int getArmy_mode() {
        return Army_mode;
    }
    public void setArmy_mode(int army_mode) {
        Army_mode = army_mode;
    }
    public int getNewEnergy_mode() {
        return NewEnergy_mode;
    }
    public void setNewEnergy_mode(int newEnergy_mode) {
        NewEnergy_mode = newEnergy_mode;
    }
    public int getEmergencyMode() {
        return EmergencyMode;
    }
    public void setEmergencyMode(int emergencyMode) {
        EmergencyMode = emergencyMode;
    }
    public int getAntifake_mode() {
        return Antifake_mode;
    }
    public void setAntifake_mode(int antifake_mode) {
        Antifake_mode = antifake_mode;
    }


}
```

5.6.1.2 设置黑白名单参数

```
public int ICE_IPCSDK_SetWhiteListParam(WhiteListParam param)
```

参数：

1．黑白名单配置参数（输入）

```
public class WhiteListParam {
    private int nWhiteListMode = 0;    //白名单模式：0表示断网后关联开闸，1表示实时关联开闸,2 不关联开闸
    private int nWhiteListMatch = 0;   //白名单匹配相似，范围（60%-100%），默认：80%
    private int nBlackListMode = 0;    //黑名单关联道闸模式，0 不关联开闸，1 实时关联开闸
    private int nBlackListMatch = 0;   //黑名单匹配相似，范围（60%-100%），默认：80%
    private int nTempListMode = 0;     //临时车关联道闸模式，0 不关联开闸，1 实时关联开闸
    private int ignoreHZ_flag = 0;     //忽略汉字标志，0 不忽略，1 忽略
    private int allow_unmatch_chars_cnt = 0; // 允许不匹配的字符个数范围[0 ~3]
    private int new_version = 1;       //新旧黑白名单配置标志，1：新 0：旧
    private int Jing_mode = 0;         //警车模式，0 不关联开闸，1 实时关联开闸
    private int Army_mode = 0;         //军车模式，0 不关联开闸，1 实时关联开闸
    private int NewEnergy_mode = 0;    //新能源车牌模式，0 不关联开闸，1 实时关联开闸
    private int EmergencyMode = 0;     //应急车牌模式，0 不关联开闸，1 实时关联开闸
    private int Antifake_mode = 0;  //防伪车模式，1 不关联开闸，0 实时关联开闸
    public int getnWhiteListMode() {
        return nWhiteListMode;
    }
    public void setnWhiteListMode(int nWhiteListMode) {
        this.nWhiteListMode = nWhiteListMode;
    }
    public int getnWhiteListMatch() {
        return nWhiteListMatch;
    }
    public void setnWhiteListMatch(int nWhiteListMatch) {
```

```
        this.nWhiteListMatch = nWhiteListMatch;
    }
    public int getnBlackListMode() {
        return nBlackListMode;
    }
    public void setnBlackListMode(int nBlackListMode) {
        this.nBlackListMode = nBlackListMode;
    }
    public int getnBlackListMatch() {
        return nBlackListMatch;
    }
    public void setnBlackListMatch(int nBlackListMatch) {
        this.nBlackListMatch = nBlackListMatch;
    }
    public int getnTempListMode() {
        return nTempListMode;
    }
    public void setnTempListMode(int nTempListMode) {
        this.nTempListMode = nTempListMode;
    }
    public int getIgnoreHZ_flag() {
        return ignoreHZ_flag;
    }
    public void setIgnoreHZ_flag(int ignoreHZ_flag) {
        this.ignoreHZ_flag = ignoreHZ_flag;
    }
    public int getAllow_unmatch_chars_cnt() {
        return allow_unmatch_chars_cnt;
    }
    public void setAllow_unmatch_chars_cnt(int
allow_unmatch_chars_cnt) {
        this.allow_unmatch_chars_cnt = allow_unmatch_chars_cnt;
    }
    public int getNew_version() {
        return new_version;
    }
    public void setNew_version(int new_version) {
        this.new_version = new_version;
    }
    public int getJing_mode() {
        return Jing_mode;
    }
    public void setJing_mode(int jing_mode) {
        Jing_mode = jing_mode;
```

```
    }
    public int getArmy_mode() {
        return Army_mode;
    }
    public void setArmy_mode(int army_mode) {
        Army_mode = army_mode;
    }
    public int getNewEnergy_mode() {
        return NewEnergy_mode;
    }
    public void setNewEnergy_mode(int newEnergy_mode) {
        NewEnergy_mode = newEnergy_mode;
    }
    public int getEmergencyMode() {
        return EmergencyMode;
    }
    public void setEmergencyMode(int emergencyMode) {
        EmergencyMode = emergencyMode;
    }
    public int getAntifake_mode() {
        return Antifake_mode;
    }
    public void setAntifake_mode(int antifake_mode) {
        Antifake_mode = antifake_mode;
    }


}
```

返回值：

0 失败

1 成功

5.6.2 DNS 配置

5.6.2.1 设置 DNS

```
public int ICE_IPCSDK_SetDNSAddr(String strDNS, String strDNSReserve)
```

参数：

1.首选 DNS（输入）

2.备选 DNS（输入）

返回值：

0 失败

1 成功

5.6.2.2 获取 DNS

```
public DNSParam ICE_IPCSDK_GetDNSAddr()
```

参数：
无

返回值：
获取到的 DNS，定义如下：

```
public class DNSParam {
    private String strDNS = "";//首选DNS
    private String strDNSReserve = "";//备选DNS
    public String getStrDNS() {
        return strDNS;
    }
    public void setStrDNS(String strDNS) {
        this.strDNS = strDNS;
    }
    public String getStrDNSReserve() {
        return strDNSReserve;
    }
    public void setStrDNSReserve(String strDNSReserve) {
        this.strDNSReserve = strDNSReserve;
    }

}
```

获得失败时返回 null

5.6.3 开关量输出配置

5.6.3.1 获取开关量输出

```
public RelayOutputResult ICE_IPCSDK_GetRelayOutput(int nIndex)
```

参数：

1. 开关量索引。（0：开关量 1 ,1：开关量 2, 2:开关量 3, 3：开关量 4）（输出）

返回值：

获取的开关量结果，定义如下：

```
public class RelayOutputResult{
       public byte bIdleState; //开关量输出状态 0：常开 1：常闭
       public int nDelayTime;  //输出时长，单位 s。（-1 表示不恢复，取值范围：
-1,1-10）
    }
```

获得失败时，返回 null

5.6.3.2 设置开关量输出

```
public int ICE_IPCSDK_SetRelayOutputSettings(
    byte bIdleState,
    int nDelayTime,
    int nIndex
    )
```

参数：

1.开关量输出状态 0：常开 1：常闭（输入）
2.输出时长，单位 s。（-1 表示不恢复，取值范围：-1,1-10）（输入）
3.开关量索引。（0：开关量 1， 1：开关量 2， 2：开关量 3， 3:开关量 4）（输入）

返回值：

0 失败
1 成功

5.6.4 串口参数配置

5.6.4.1 获取串口参数

```
public UartParam ICE_IPCSDK_GetUARTCfg()
```

参数：
无

返回值：
串口参数，定义如下：

```
public class Uart{
    public int uartEn;                      //串口是否使能，0 不使能，1 使能，
默认为 1 使用
    public int uartWorkMode;                //预留
    public int baudRate;                    // 波 特 率 ， 默 认 值 为  9600 ，
0:1200、1:2400、2:4800、3:9600、4:19200、5:38400、6:115200
    public int dataBits;                    //数据位，默认值为 8，可选值为：
0:5、1:6、2:7、3:8
    public int parity;                      //校验位，默认值为 无，可选值为：
0:无、1:奇校验、2:偶校验、3:标记、4:空格
    public int stopBits;                    //停止位，默认值为 1，可选值为：
0:1、1:2
    public int flowControl;                 //流控模式,默认值为 无,可选值为：
0:无、1:硬件， 2:Xon，3: Xoff
    public int LEDBusinessType;             // 预留
    public int u32UartProcOneReSendCnt;  // 车牌协议重发次数， 默认值 0，
取值范围[0~3]，0 代表不重传
    public byte screen_mode;                // 屏显模式， 1 屏显 2 透传
}

public class UartParam{
    public Uart uartParam1 = new Uart(); //RS485 串口
    public Uart uartParam2 = new Uart(); //RS485-2 串口
}
```

获取失败时，返回 null.

5.6.4.2 设置串口参数

```
public int ICE_IPCSDK_SetUARTCfg(UartParam param)
```

参数：

1.串口参数，定义如下：（输入）

```
public class Uart{
    public int uartEn;                          //串口是否使能，0 不使能，1 使能，
默认为 1 使用
    public int uartWorkMode;                    //预留
    public int baudRate;                        // 波 特 率 ， 默 认 值 为  9600 ，
0:1200、1:2400、2:4800、3:9600、4:19200、5:38400、6:115200
    public int dataBits;                        //数据位，默认值为 8，可选值为：
0:5、1:6、2:7、3:8
    public int parity;                          //校验位，默认值为 无，可选值为：
0:无、1:奇校验、2:偶校验、3:标记、4:空格
    public int stopBits;                        //停止位，默认值为 1，可选值为：
0:1、1:2
    public int flowControl;                     //流控模式,默认值为 无,可选值为：
0:无、1:硬件， 2:Xon，3：Xoff
    public int LEDBusinessType;                 // 预留
    public int u32UartProcOneReSendCnt;  // 车牌协议重发次数， 默认值 0，
取值范围[0~3]，0 代表不重传
    public byte screen_mode;                    // 屏显模式， 1 屏显 2 透传

public class UartParam{
    public Uart uartParam1 = new Uart(); //RS485 串口
    public Uart uartParam2 = new Uart(); //RS485-2 串口
}
```

返回值：

0 失败

1 成功

注意事项：

由于 Uart 类中含有 private 成员变量，暂不支持设置，

请先调用获取接口后，再给需要设置的成员变量赋值后进行设置。

5.7 透明串口

5.7.1 发送透明串口数据

1.透明串口，RS485-1

```
int ICE_IPCSDK_TransSerialPort(byte[] data)
```

参数：

1.串口数据（输入）

返回值：

0 失败

1 成功

2.透明串口，RS232/RS485-2

```
int ICE_IPCSDK_TransSerial232Port(byte[] data)
```

参数：

1.串口数据（输入）

返回值：

0 失败

1 成功

5.7.2 注册透明串口数据上报事件

int ICE_IPCSDK_RegRS485Event(int iRS485Index)

参数：

1.串口索引号（1：RS485-1,2:RS485-2/RS232,3：RS485-1 和 RS485-2/RS232）

返回值：

0 失败

1 成功

5.7.3 设置透明串口数据上报事件回调(RS485-1)

```
void ICE_ICPSDK_SetSerialPortCallback(ISerialPortCallback callback)
```

参数：

1.透明串口数据上报事件，详情见 ISerialPortCallback(See 5.7.5)（输入）

返回值：无

5.7.4 设置透明串口数据上报事件回调(RS485-2/RS232)

```
void ICE_ICPSDK_SetSerial232PortCallback(
       ISerialPortCallback callback)
```

参数：

1.透明串口数据上报事件，详情见 ISerialPortCallback(See 5.7.5)（输入）

返回值：无

5.7.5 透明串口数据上报事件

```
public interface ISerialPortCallback {
       public void ICE_IPCSDK_SerialPort(
           String strIP,
           byte[] bData,
           int nOffset,
           int nLen
       );
    }
```

参数：

1.ip 地址（输出）
2.串口数据内存地址（输出）
3.数据偏移量（输出）
4.串口数据大小（输出）

返回值：

无

5.7.6 注销透明串口数据上报事件

```
int ICE_IPCSDK_UnregRS485Event()
```

参数：

无

返回值：

0 失败

1 成功

5.10 日志配置

5.10.1 配置日志记录参数

```
public void ICE_IPCSDK_LogConfig(int nEnableLog, String filePath)
```

参数：

1.是否记录日志 0：不记录 1：记录（输入）

2.日志记录的路径 日志完整路径为 filePath+ipclog.txt（输入）

日志记录的内容为：

1. 相机离线在线状态，离线原因
2. 接收到车牌识别数据
3. 接收到串口变化数据
4. 接收到串口数据

返回值：

无

5.11 黑白名单

5.11.1 黑白名单接口注意事项

格式：车牌号码+TAB 键+起始有效期+TAB 键+截止有效期+ TAB 键+起始时间+ TAB 键+截止时间+回车键+TAB 键+白名单类型（B/W）

实例：吉 AP889Q 2015/09/21 2015/10/20 00:00:00 23:59:59 W

注意：起始时间必须早于截止时间，这两个时间是指一个时间段，与起始有效期和截止有效期无关。

比如：2015/09/21 2015/10/20 00:00:00 23:59:59 是正确的，

而 2015/09/21 2015/10/20 22:00:00 11:00:00 是错误的时间。

2、在界面显示所有黑白名单的方法

a.调用 ICE_IPCSDK_WhiteListGetCount 获得黑白名单的总数 totalNum

b.for(int i=0; i<totalNum; i++)

```
{
    ICE_IPCSDK_WhiteListGetItem;(第二个参数输入 i，即索引号)
    显示
}
```

5.11.2 黑白名单操作

5.11.2.1 获取黑白名单项

```
WhiteListItem ICE_IPCSDK_WhiteListGetItem(int index)
```

参数：
1.黑白名单索引（0 到（总数-1）之间的整数）（输入）

返回值：
黑白名单项，定义如下：

```
public class WhiteListItem {
    private String strPlate = "";     //车牌号
    private String strDateBegin = ""; //开始日期
    private String strDateEnd = "";   //结束日期
    private String strTimeBegin = ""; //开始时间
    private String strTimeEnd = "";   //结束时间
    private String strType = "";      //类型（'W'白名单/'B'黑名单）
    private String strRemark = "";    //备注
    private String strRsrv3 = "";     //预留
    private String strRsrv4 = "";     //预留

    private int iStatus = 0;

    public String getStrPlate() {
        return strPlate;
    }
    public void setStrPlate(String strPlate) {
        this.strPlate = strPlate;
    }
    public String getStrDateBegin() {
        return strDateBegin;
    }
    public void setStrDateBegin(String strDateBegin) {
        this.strDateBegin = strDateBegin;
    }
    public String getStrDateEnd() {
        return strDateEnd;
    }
    public void setStrDateEnd(String strDateEnd) {
        this.strDateEnd = strDateEnd;
```

```
    }
    public String getStrTimeBegin() {
        return strTimeBegin;
    }
    public void setStrTimeBegin(String strTimeBegin) {
        this.strTimeBegin = strTimeBegin;
    }
    public String getStrTimeEnd() {
        return strTimeEnd;
    }
    public void setStrTimeEnd(String strTimeEnd) {
        this.strTimeEnd = strTimeEnd;
    }
    public String getStrType() {
        return strType;
    }
    public void setStrType(String strType) {
        this.strType = strType;
    }
    public String getStrRemark() {
        return strRemark;
    }
    public void setStrRemark(String strRemark) {
        this.strRemark = strRemark;
    }
    public String getStrRsrv3() {
        return strRsrv3;
    }
    public void setStrRsrv3(String strRsrv3) {
        this.strRsrv3 = strRsrv3;
    }
    public String getStrRsrv4() {
        return strRsrv4;
    }
    public void setStrRsrv4(String strRsrv4) {
        this.strRsrv4 = strRsrv4;
    }
    public int getiStatus() {
        return iStatus;
    }
    public void setiStatus(int iStatus) {
        this.iStatus = iStatus;
    }
}
```

5.11.2.2 编辑黑白名单

```
public boolean ICE_IPCSDK_WhiteListEditItem_ByNumber(
    String strPlate,
    String strDateBegin,
    String strDateEnd,
    String strTimeBegin,
    String strTimeEnd,
    String strType,
    String strRemark,
    String strRsrv1,
    String strRsrv2
)
```

参数：

1.车牌号（输入）
2.开始日期（输入）
3.结束日期（输入）
4.开始时间（输入）
5.结束时间（输入）
6.名单类型（'W'白名单 'B'黑名单）（输入）
7.备注（输入）
8.保留字段
9.保留字段

返回值：

布尔值，是否成功

5.11.2.3 添加黑白名单

```
public boolean ICE_IPCSDK_WhiteListInsertItem_ByNumber(
    String strPlate,
    String strDateBegin,
    String strDateEnd,
    String strTimeBegin,
    String strTimeEnd,
    String strType,
    String strRemark,
    String strRsrv1,
    String strRsrv2
)
```

参数：

1.车牌号（输入）
2.开始日期（输入）
3.结束日期（输入）
4.开始时间（输入）
5.结束时间（输入）
6.名单类型（'W'白名单 'B'黑名单）（输入）
7.备注（输入）
8.保留字段
9.保留字段

返回值：

布尔值，是否成功

5.11.2.4 删除黑白名单

```
public boolean ICE_IPCSDK_WhiteListDeleteItem_ByNumber(String strPlate)
```

参数：
1.车牌号（输入）

返回值：
布尔值，是否成功

5.11.2.5 查找黑白名单

```
public WhiteListItem ICE_IPCSDK_WhiteListFindItem_ByNumber(String strPlate)
```

参数：
1.车牌号（输入）

返回值：
结构体如下：

```
public class WhiteListItem {
private String strPlate = "";     //车牌号
private String strDateBegin = ""; //开始日期
private String strDateEnd = "";   //结束日期
private String strTimeBegin = ""; //开始时间
private String strTimeEnd = "";   //结束时间
private String strType = "";      //类型（'W'白名单/'B'黑名单）
private String strRemark = "";    //备注
private String strRsrv3 = "";     //预留
private String strRsrv4 = "";     //预留

private int iStatus = 0;

public String getStrPlate() {
    return strPlate;
}
public void setStrPlate(String strPlate) {
    this.strPlate = strPlate;
}
public String getStrDateBegin() {
    return strDateBegin;
}
public void setStrDateBegin(String strDateBegin) {
    this.strDateBegin = strDateBegin;
}
public String getStrDateEnd() {
    return strDateEnd;
}
public void setStrDateEnd(String strDateEnd) {
    this.strDateEnd = strDateEnd;
}
public String getStrTimeBegin() {
    return strTimeBegin;
```

```
    }
    public void setStrTimeBegin(String strTimeBegin) {
        this.strTimeBegin = strTimeBegin;
    }
    public String getStrTimeEnd() {
        return strTimeEnd;
    }
    public void setStrTimeEnd(String strTimeEnd) {
        this.strTimeEnd = strTimeEnd;
    }
    public String getStrType() {
        return strType;
    }
    public void setStrType(String strType) {
        this.strType = strType;
    }
    public String getStrRemark() {
        return strRemark;
    }
    public void setStrRemark(String strRemark) {
        this.strRemark = strRemark;
    }
    public String getStrRsrv3() {
        return strRsrv3;
    }
    public void setStrRsrv3(String strRsrv3) {
        this.strRsrv3 = strRsrv3;
    }
    public String getStrRsrv4() {
        return strRsrv4;
    }
    public void setStrRsrv4(String strRsrv4) {
        this.strRsrv4 = strRsrv4;
    }
    public int getiStatus() {
        return iStatus;
    }
    public void setiStatus(int iStatus) {
        this.iStatus = iStatus;
    }
}
```

5.11.2.6 删除所有白名单项

```
boolean ICE_IPCSDK_WhiteListDeleteAllItems()
```

参数：
无

返回值：
布尔值，是否成功

5.11.3 获取黑白名单总数

5.11.3.1 获取黑白名单项总数

```
int ICE_IPCSDK_WhiteListGetCount()
```

参数：
无

返回值：
失败时，返回-1
白名单项总数