PSAM 接入说明文档

版本	描述	日期	作者
V1.0	添加接入 PSAM 控制说明	2023/2/4	郑俊科
V1.1	新增支持机型：Falcon 1、Swift 2 Pro、Lark 1	2023/4/8	郑俊科

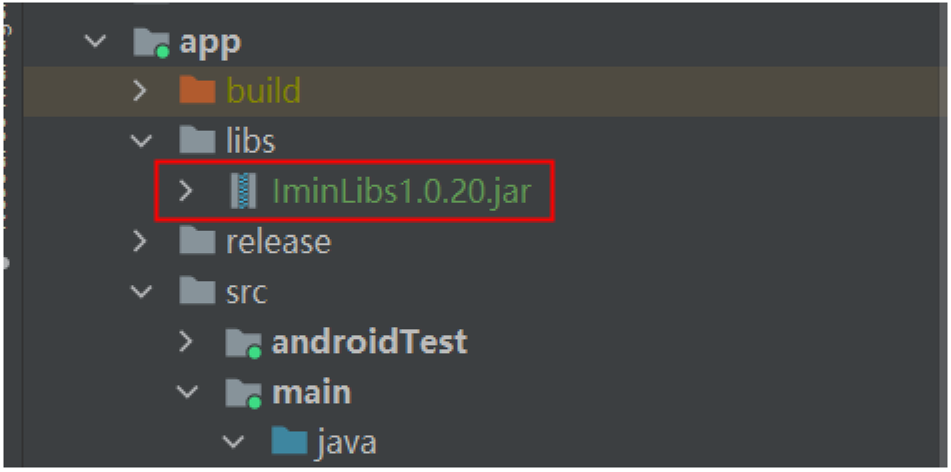
1.简介

该文档用于介绍如何接入读取 PSAM，面向开发者
针对机型

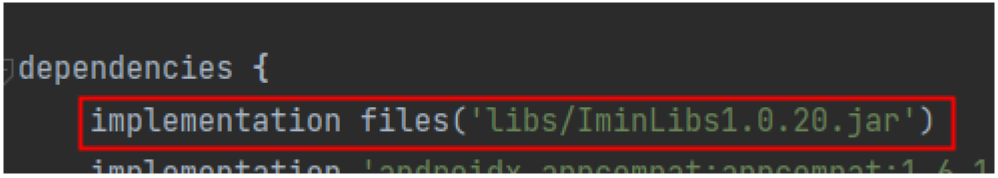
序号	机型
1	Swift 2
2	Falcon 1、Swift 2 Pro、Lark 1

2.接入步骤

1) 引入 IminLib.jar，在 app 目录下 libs 目录下添加 IminLibs.jar，最新 jar 已变为 IminLibs1.0.21.jar



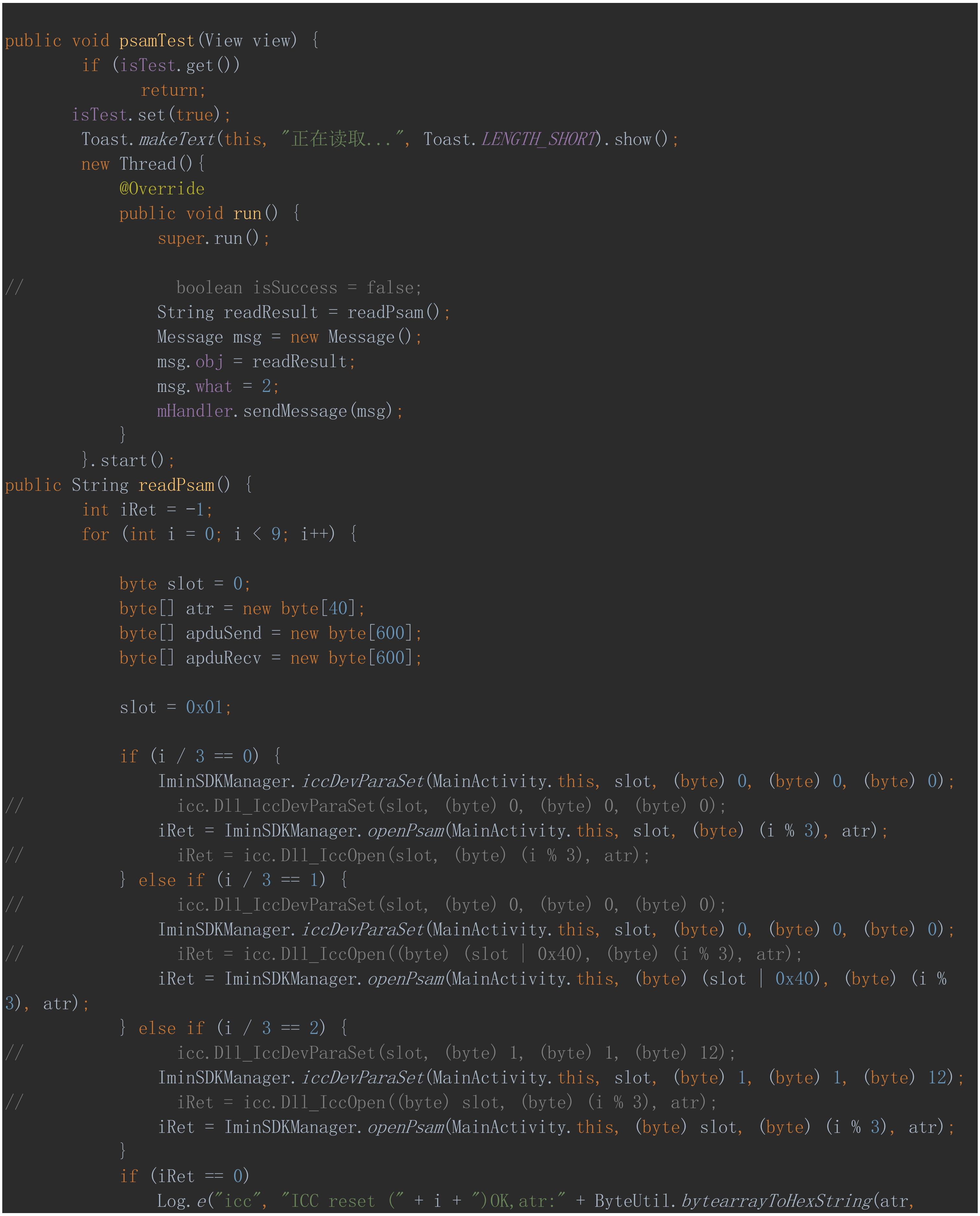
2) 添加 IminLib.jar 的依赖，在 app 目录下的 build.gradle 添加依赖



```
dependencies {
    implementation files('libs/IminLibs1.0.20.jar')
    implementation 'androidx.appcompat:appcompat:1.6.1
```

3) 请求读取 PSAM 状态（前提：已插入 PSAM 卡）

```
private AtomicBoolean isTest = new AtomicBoolean(false);
```



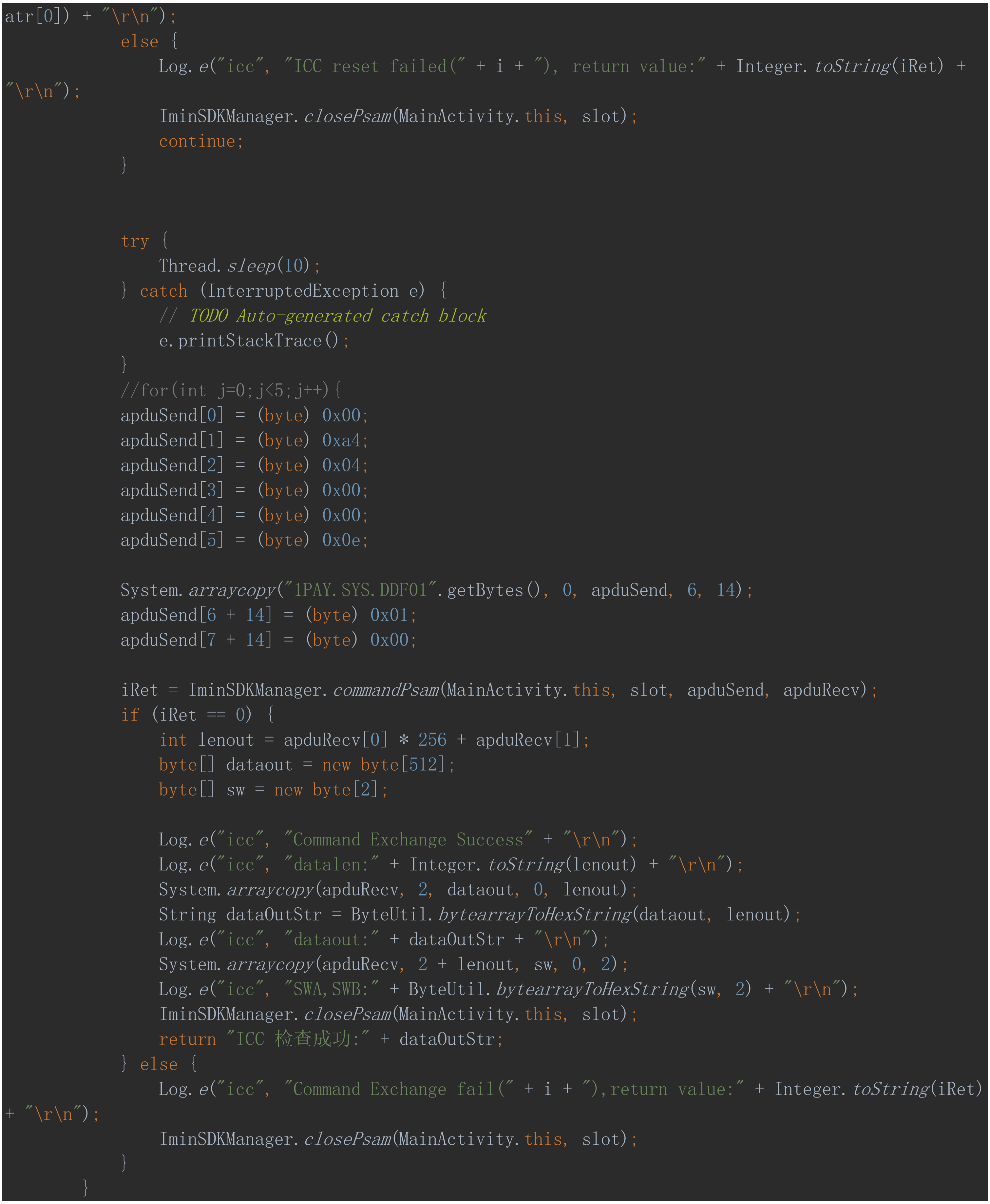
```
public void psamTest(View view) {
        if (isTest.get())
              return;
       isTest.set(true);
        Toast.makeText(this, "正在读取...", Toast.LENGTH_SHORT).show();
        new Thread(){
            @Override
            public void run() {
                super.run();

//                boolean isSuccess = false;
                String readResult = readPsam();
                Message msg = new Message();
                msg.obj = readResult;
                msg.what = 2;
                mHandler.sendMessage(msg);
            }
        }.start();
public String readPsam() {
        int iRet = -1;
        for (int i = 0; i < 9; i++) {

            byte slot = 0;
            byte[] atr = new byte[40];
            byte[] apduSend = new byte[600];
            byte[] apduRecv = new byte[600];

            slot = 0x01;

            if (i / 3 == 0) {
                IminSDKManager.iccDevParaSet(MainActivity.this, slot, (byte) 0, (byte) 0, (byte) 0);
//                icc.Dll_IccDevParaSet(slot, (byte) 0, (byte) 0, (byte) 0);
                iRet = IminSDKManager.openPsam(MainActivity.this, slot, (byte) (i % 3), atr);
//                iRet = icc.Dll_IccOpen(slot, (byte) (i % 3), atr);
            } else if (i / 3 == 1) {
//                icc.Dll_IccDevParaSet(slot, (byte) 0, (byte) 0, (byte) 0);
                IminSDKManager.iccDevParaSet(MainActivity.this, slot, (byte) 0, (byte) 0, (byte) 0);
//                iRet = icc.Dll_IccOpen((byte) (slot | 0x40), (byte) (i % 3), atr);
                iRet = IminSDKManager.openPsam(MainActivity.this, (byte) (slot | 0x40), (byte) (i %
3), atr);
            } else if (i / 3 == 2) {
//                icc.Dll_IccDevParaSet(slot, (byte) 1, (byte) 1, (byte) 12);
                IminSDKManager.iccDevParaSet(MainActivity.this, slot, (byte) 1, (byte) 1, (byte) 12);
//                iRet = icc.Dll_IccOpen((byte) slot, (byte) (i % 3), atr);
                iRet = IminSDKManager.openPsam(MainActivity.this, (byte) slot, (byte) (i % 3), atr);
            }
            if (iRet == 0)
                Log.e("icc", "ICC reset (" + i + ")OK,atr:" + ByteUtil.bytearrayToHexString(atr,
```



```
atr[0]) + "\r\n");
            else {
                Log.e("icc", "ICC reset failed(" + i + "), return value:" + Integer.toString(iRet) +
"\r\n");
                IminSDKManager.closePsam(MainActivity.this, slot);
                continue;
            }


            try {
                Thread.sleep(10);
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            //for(int j=0;j<5;j++){
            apduSend[0] = (byte) 0x00;
            apduSend[1] = (byte) 0xa4;
            apduSend[2] = (byte) 0x04;
            apduSend[3] = (byte) 0x00;
            apduSend[4] = (byte) 0x00;
            apduSend[5] = (byte) 0x0e;

            System.arraycopy("1PAY.SYS.DDF01".getBytes(), 0, apduSend, 6, 14);
            apduSend[6 + 14] = (byte) 0x01;
            apduSend[7 + 14] = (byte) 0x00;

            iRet = IminSDKManager.commandPsam(MainActivity.this, slot, apduSend, apduRecv);
            if (iRet == 0) {
                int lenout = apduRecv[0] * 256 + apduRecv[1];
                byte[] dataout = new byte[512];
                byte[] sw = new byte[2];

                Log.e("icc", "Command Exchange Success" + "\r\n");
                Log.e("icc", "datalen:" + Integer.toString(lenout) + "\r\n");
                System.arraycopy(apduRecv, 2, dataout, 0, lenout);
                String dataOutStr = ByteUtil.bytearrayToHexString(dataout, lenout);
                Log.e("icc", "dataout:" + dataOutStr + "\r\n");
                System.arraycopy(apduRecv, 2 + lenout, sw, 0, 2);
                Log.e("icc", "SWA,SWB:" + ByteUtil.bytearrayToHexString(sw, 2) + "\r\n");
                IminSDKManager.closePsam(MainActivity.this, slot);
                return "ICC 检查成功:" + dataOutStr;
            } else {
                Log.e("icc", "Command Exchange fail(" + i + "),return value:" + Integer.toString(iRet)
+ "\r\n");
                IminSDKManager.closePsam(MainActivity.this, slot);
            }
        }
        return "ICC 检查失败, result:" + iRet;
    }
```

4) 上述代码读取必须要在子线程中添加，iRet 返回不等于 0 的情况可以参考 Android Psam 应用开发指南中对应的返回值