Histone 服务程序技术规范

开发文档

青岛海信智能商用系统股份有限公司

2019 年 09 月

版本控制信息

版本	日期	更新内容	参与者	备注
V1.06.J919	2019.9.19	1.打印机增加表格打印接口 2.兼容二代发袋机		
	2020.03.13	新增身份证模块 api		

文档审阅信息

版本	审阅人	角色	审阅日期	确认（填姓名）	备注

文档批准信息

版本	批准人	角色	批准日期	确认（填姓名）	备注

目录

1、 目的 7
2、 术语 7
2.1 服务（Service） 7
2.2 Android 接口定义语言(Android Interface definition language） 7
2.3 厂商服务应用（Service Application） 7
3、 常量及定义 7
4、 服务程序概述 7
5、 设备服务引擎模块声明 8
5.1 模块说明 8
5.2 包路径 8
5.3 文件名 8
5.4 AIDL 接口清单 8
5.4.1 获取服务版本号(getSDKVersion) 8
5.4.2 配置服务 (setServiceConifg) 9
5.4.3 获取服务对象(getService) 9
6、 打印机模块声明 9
6.1 模块说明 9
6.2 包路径 9
6.3 AIDL 文件名 9
6.4 接口清单 9
6.4.1 打印机配置(setConfig) 9
6.4.2 获取状态(checkStatus) 10
6.4.3 开钱箱(openCashBox) 10
6.4.4 获取打印机描述(getDescriptions) 10
6.4.5 扩展指令集(addCommand) 10
6.4.6 切纸(addCut) 11
6.4.7 行走纸(addFeedLine) 11
6.4.8 像素走纸(addFeedDots) 11
6.4.9 双倍字体(addTextSizeDouble) 11
6.4.10 正常字体(addTextSizeNormal) 12
6.4.11 打印文本(addText) 12
6.4.12 表格形式打印文本(addTextsAutoWrap) 12
6.4.13 设置对齐方式(setAlign) 12
6.4.14 打印图片(addImage) 13
6.4.15 打印 QR 码(addQRCode) 13
6.4.16 打印条码(addBarcode) 13
6.4.17 开始构建布局(beginTranscation) 13
6.4.18 结束构建布局(endTranscation) 14
6.4.19 开始打印(commit) 14
6.4.20 注册状态监听器(registerStatusListener) 14
6.4.21 取消注册状态监听器(unRegisterStatusListener) 14
6.4.22 获取支持的最大区域(getPageModeMaxArea) 15
6.4.23 设置区域坐标和宽度、高度 (setPageModePrintArea) 15

6.4.24 设置文本打印方向(setPageModeDirection) 15
6.4.25 设置文本打印绝对位置（setPageModeVerticalPostion) 16
6.4.26 设置文本打印绝对位置（setPageModeHorizontalPostion) 16
6.4.27 进入标签模式及标签打印、进纸（pageModePrint) 16
6.4.28 校验标签（pageModeCheckLabel) 16
7、 主板钱箱模块声明 17
7.1 模块说明 17
7.2 包路径 17
7.3 AIDL 文件名 17
7.4 接口清单 17
7.4.1 打开钱箱(openCashBox) 17
7.4.2 获取钱箱状态(getCashBoxStatus) 17
8、 磁卡模块声明 18
8.1 模块说明 18
8.2 包路径 18
8.3 AIDL 文件名 18
8.4 接口清单 18
8.3.1 磁卡初始化(openMsr) 18
8.3.2 设置超时时间(setTimeout) 18
8.3.3 注册磁卡监听器(setMsrListener) 18
8.3.4 停止检测(stopCheck) 19
8.3.5 释放磁卡资源(closeMsr) 19
8.5 磁条事件监听器 19
9、 扫描窗模块声明 20
9.1 模块说明 20
9.2 包路径 20
9.3 AIDL 文件名 20
9.4 接口清单 20
9.4.1 初始化扫描窗(initScanner) 20
9.4.2 销毁扫描窗资源(destoryScanner) 20
9.4.3 设置扫描数据监听器(setScanListener) 20
9.4.4 取消扫描数据监听(removeScanListener) 21
10、 非接模块声明 21
10.1 模块说明 21
10.2 包路径 21
10.3 AIDL 文件名 21
10.4 接口清单 21
10.4.1 非接初始化(piccOpen) 21
10.4.2 非接释放(piccClose) 21
10.4.3 设置超时(setTimeout) 22
10.4.4 设置事件监听(setPiccListener) 22
10.4.5 停止检卡(stopCheck) 22
10.4.6 M 卡认证(piccMfAuthBlock) 22
10.4.7 M 卡读数据(piccMfReadBlock) 23

10.4.8 M 卡写数据(piccMfWriteBlock) 23
10.4.9 M 卡初值(piccMfInitValue) 23
10.4.10 M 卡加值(piccMfAddValue) 23
10.4.11 M 卡减值(piccMfReduceValue) 24
10.4.12 非接检卡(piccDetect) 24
10.4.13 发送 APDU 命令(isoCommand) 25
10.4.14 获取非接卡片类型(piccGetCardType) 25
10.5 26
监听事件 26
11、 串口模块声明 26
11.1 模块说明 26
11.2 包路径 26
11.3 AIDL 文件名 26
11.4 接口清单 26
11.4.1 打开串口设备(openUart) 26
11.4.2 关闭串口设备(closeUart) 27
11.4.3 读串口数据(readUart) 27
11.4.4 写串口数据(writeUart) 27
11.4.5 清除串口数据(flushUart) 27
11.4.6 设置串口波特率(setBaudateRate) 28
12、 发袋机模块声明 28
12.1 模块说明 28
12.2 包路径 28
12.3 AIDL 文件名 28
12.4 接口清单 28
12.4.1 自动发袋(autoOutBag) 28
12.4.2 重置历史出袋数量(clearBaggerHistoryTotal) 28
12.4.3 获取出袋历史数量(getBaggerHistoryTotal) 29
12.4.4 获取出袋通道状态(getBaggerStatus) 29
12.4.5 设置出袋通道监听器(setBaggerListener) 29
13、 指示灯模块声明 30
13.1 模块说明 30
13.2 包路径 30
13.3 AIDL 文件名 31
13.4 接口清单 31
13.4.1 点亮指示灯(setLight) 31
13.4.2 获取指示灯颜色(getLightStatus) 31
13.4.3 控制蜂鸣器(buzzerCtrl) 32
14、 物理按键模块声明 32
14.1 模块说明 32
14.2 包路径 32
14.3 AIDL 文件名 32
14.4 接口清单 32
14.4.1 获取按键灯状态(getKeyLightStatus) 32

14.4.2 设置指示灯状态(setKeyLight) 32
14.4.3 设置按键事件监听器(setKeyListener) 33
15、 秤模块声明 33
15.1 模块说明 34
15.2 包路径 34
15.3 AIDL 文件名 34
15.4 接口清单 34
15.4.1 秤初始化(open) 34
15.4.2 释放秤资源 (close) 34
15.4.3 获取重量(getWeight) 34
15.4.4 归零(setZeroWeight) 34
15.4.5 去皮扣重(setTareWeight) 35
15.4.6 去皮扣重(exitTareWeight) 35
15.4.7 配置秤(config) 36
15.5 事件监听 36
16、 身份证模块声明 37
16.1 模块说明 37
16.2 包路径 37
16.3 AIDL 文件名 37
16.4 接口清单 37
16.4.1 打开设备(openIdCard) 37
16.4.2 关闭设备(closeIdCard) 37
16.4.3 设置超时时间(setTimeout) 38
16.4.4 注册监听器(setIdCardListener) 38
16.4.5 停止检测(stopCheck) 38
16.4.6 监听事件 38

1、 目的

为整合企业资源，提高POS 终端软件开发效率，特制订此规范。

2、 术语

2.1 服务（Service）

Android 系统中的四大组件之一

该组件几乎与Activity 的级别一致，但只能运行于后台，且可以和其他组件进行交互。

以下方式可启动服务(Service)：

（Context）ctx.bindService()

2.2 Android 接口定义语言(Android Interface definition language）

由于Android 系统中的进程之间不能共享内存，因此，需要提供一些机制在不同进程之间进行数据通信。

为了使其他的应用程序也可以访问本应用程序提供的服务，Android 系统采用了远程过程调用（Remote Procedure Call，RPC）方式来实现。

2.3 厂商服务应用（Service Application）

一种遵循本规范服务层接口的服务应用，该应用由海信设备厂商开发。能够驱动海信硬件设备（磁条或非接读卡器、蜂鸣等），该应用默认集成在POS产品中，供客户开发调用使用。

3、 常量及定义

对外外设串口编号、打印机设备串口名称定义，若如下接口不满足需要，可根据开发文档自行封装。

串口的设备节点为：/dev/ttySAC0，/dev/ttySAC3

4、 服务程序概述

服务名称：

com.histonepos.psdk

使用方法：

通过绑定com.histonepos.psdk服务，获取连接成功之后返回的实例，即可使

用服务程序的所有接口。

示例如下，仅供参考：

```
SmartService.getInstance().register(this, new IServiceConnector() {
    @Override
    public void onServiceDisconnected() {
        // TODO Auto-generated method stub
        GlobalDefines.cashbox = null;
    }

    @Override
    public void onServiceConnected(IServiceManager serviceManager) {
        // TODO Auto-generated method stub
        Log.i("PosPad_Activity", "onServiceConnected");
        GlobalDefines.sm    = serviceManager;
    }
});
```

应用退出时释放服务资源:

```
SmartService.getInstance().unregister();
```

5、 设备服务引擎模块声明

5.1 模块说明

用于管理所有内置设备接口，如磁条卡、打印机、非接卡、扫描器、钱箱等。

5.2 包路径

com.histonepos.npsdk.api

5.3 文件名

IServiceManager.aidl

5.4 AIDL接口清单

5.4.1 获取服务版本号(getSDKVersion)

接口原型	String getSDKVersion()	
接口功能	获取版本号	
参数说明	入参	无
	出参	无

返回值	成功 版本号 eg.V1.00.J509 失败 返回NULL
补充说明	

5.4.2 配置服务 (setServiceConifg)

接口原型	Void setServiceConfig(in android.os.Bundle extra)	
接口功能	配置服务	
参数说明	入参	Data
	出参	无
返回值	无	
补充说明	接口预留	

5.4.3 获取服务对象(getService)

接口原型	IBinder getService(int sType);	
接口功能	返回各个模块服务的对象	
参数说明	入参	sType：参考Const常量定义
	出参	无
返回值	成功 返回sType对应的服务模块对象 失败 返回NULL	
补充说明		

6、 打印机模块声明

6.1 模块说明

用于操作打印机设备。

注意：构建小票的流程：

printer.beginTranscation-->printer.add-->printer.endTranscation->printer.commit须要放到子线程中，否则过长的小票可能导致主线程卡顿。

6.2 包路径

com.histonepos.npsdk.printer

6.3 AIDL文件名

IPrinterService.aidl

6.4 接口清单

6.4.1 打印机配置(setConfig)

接口原型	Void setConfig(Bundle data)

接口功能	打印机配置	
参数说明	入参	Data：设置打印机宽度等
	出参	无
返回值	无	
补充说明		

6.4.2 获取状态(checkStatus)

接口原型	Int checkStatus()	
接口功能	获取打印机状态	
参数说明	入参	无
	出参	无
返回值	正常 正值(>=0) 异常 负值(<0)，具体参见PrinterConsts中的常量	
补充说明		

6.4.3 开钱箱(openCashBox)

接口原型	Void openCashBox()	
接口功能	开钱箱	
参数说明	入参	无
	出参	无
返回值	无	
补充说明		

6.4.4 获取打印机描述(getDescriptions)

接口原型	Bundle getDescriptions();	
接口功能	获取打印机描述	
参数说明	入参	无
	出参	无
返回值	Bundle	
补充说明	预留	

6.4.5 扩展指令集(addCommand)

接口原型	Void addCommand(byte[] data)	
接口功能	扩展指令集	
参数说明	入参	Data:指令
	出参	无

返回值	无
补充说明	

6.4.6 切纸(addCut)

接口原型	Void addCut()	
接口功能	切纸	
参数说明	入参	无
	出参	无
返回值	无	
补充说明		

6.4.7 行走纸(addFeedLine)

接口原型	void addFeedLine(int line);	
接口功能	行走纸	
参数说明	入参	Line：行数
	出参	无
返回值	无	
补充说明		

6.4.8 像素走纸(addFeedDots)

接口原型	void addFeedDots(int dots);	
接口功能	点走纸	
参数说明	入参	Dots:走纸点数
	出参	无
返回值	无	
补充说明		

6.4.9 双倍字体(addTextSizeDouble)

接口原型	void addTextSizeDouble();	
接口功能	双倍字体	
参数说明	入参	无
	出参	无
返回值	无	
补充说明		

6.4.10 正常字体(addTextSizeNormal)

接口原型	void addTextSizeNormal();	
接口功能	常规字体	
参数说明	入参	无
	出参	无
返回值	无	
补充说明		

6.4.11 打印文本(addText)

接口原型	void addText(String data, int pattern);	
接口功能	打印文本	
参数说明	入参	Data:字符 pattern:对齐方式 0左对齐 1居中 2 右对齐
	出参	无
返回值	无	
补充说明		

6.4.12 表格形式打印文本(addTextsAutoWrap)

接口原型	void addTextsAutoWrap(in int[] weights, in int[] aligns, in String[] datas);	
接口功能	打印文本	
参数说明	入参	Weights：各列的大小占比 aligns：各列内容在本列内的对齐方式 Datas：各列内容
	出参	无
返回值	无	
补充说明	列宽比须为最简整数比，且各值加和等于2或者4或者8。如，1.0:1.0:2.0"，否则抛出IllegalArgumentException	

6.4.13 设置对齐方式(setAlign)

接口原型	Void setAlign(int align);	
接口功能	对图片，条码，二维码等设置对齐方式	
参数说明	入参	align 对齐方式 0：居左 1:居中 2：居左 请参考demo程序编写
	出参	无

返回值	无
补充说明	如未调用该接口对图片，条码，二维码进行对齐方式设置，则默认居中。

6.4.14 打印图片(addImage)

接口原型	void addImage(in Bitmap data);	
接口功能	打印图片	
参数说明	入参	Data:bitmap位图
	出参	无
返回值	无	
补充说明	如果未调用setAlign设置对齐方式，则默认居中。	

6.4.15 打印QR码(addQRCode)

接口原型	void addQRCode(String qrcode,int mode);	
接口功能	打印QR码	
参数说明	入参	qrcode:内容
	出参	Mode:放大倍数
返回值	无	
补充说明	如果未调用setAlign设置对齐方式，则默认居中。	

6.4.16 打印条码(addBarcode)

接口原型	void addBarcode(String barcode);	
接口功能	打印条码	
参数说明	入参	barcode:条码内容
	出参	无
返回值	无	
补充说明	如果未调用setAlign设置对齐方式，则默认居中。	

6.4.17 开始构建布局(beginTranscation)

接口原型	void beginTranscation();	
接口功能	标明开始构建一个小票布局	
参数说明	入参	无
	出参	无
返回值	无	

补充说明	注意：小票构建流程需要遵循以下调用流程，为了保证代码按以下流程执行，小票构建过程不支持多线程。 1. bengingTranscation() 2. addXXX() 3. endTranscation() 4. commit()

6.4.18 结束构建布局(endTranscation)

接口原型	int endTranscation();	
接口功能	结束构建小票布局	
参数说明	入参	无
	出参	无
返回值	返回 transId，标示文档Id	
补充说明		

6.4.19 开始打印(commit)

接口原型	int commit(int transId);	
接口功能	开始打印	
参数说明	入参	transId - 打印的内容文档标示，由 endTranscation()返回
	出参	无
返回值	0：发送成功 -1：发送失败。	
补充说明		

6.4.20 注册状态监听器(registerStatusListener)

接口原型	void registerStatusListener(IPrinterStatusListener statusChange);	
接口功能	注册打印机状态变化的监听器	
参数说明	入参	statusChange:回调函数
	出参	无
返回值	无	
补充说明		

6.4.21 取消注册状态监听器(unRegisterStatusListener)

接口原型	void unRegisterStatusListener();	
接口功能	取消注册打印机状态变化的监听器	
参数说明	入参	无
	出参	无

返回值	无
补充说明	

注意：以下接口只适用于标签打印机，普通热敏打印机请勿调用

6.4.22 获取支持的最大区域(getPageModeMaxArea)

接口原型	String getPageModeMaxArea ();	
接口功能	以逗号分隔的两个数值dx,dy,返回“0,0”时说明此型号没有适配页模式	
参数说明	入参	无
	出参	无
返回值	以逗号分隔的两个数值 dx,dy,返回“0,0”时说明此型号没有适配页模式	
补充说明		

6.4.23 设置区域坐标和宽度、高度 (setPageModePrintArea)

接口原型	int setPageModePrintArea(int x, int y, int dx, int dy);	
接口功能	设置区域坐标和宽度、高度	
参数说明	入参	x,y 一般是0,0, dx 打印区域的宽度（标签宽mm*8） dy 打印区域的高度（标签高mm*8）
	出参	无
返回值	0	
补充说明		

6.4.24 设置文本打印方向(setPageModeDirection)

接口原型	int setPageModeDirection (int direction);	
接口功能	设置文本打印方向，用于文本打印	
参数说明	入参	Direction 0 – 从左到右 1- 从下到上 2- 从右到左 3- 从上到下
	出参	无
返回值	0	
补充说明		

6.4.25 设置文本打印绝对位置（setPageModeVerticalPostion）

接口原型	Int setPageModeVerticalPostion (int dots);	
接口功能	设置文本打印绝对位置（与打印方向同向，y方向）	
参数说明	入参	Dots
	出参	无
返回值	0	
补充说明		

6.4.26 设置文本打印绝对位置(setPageModeHorizontalPostion)

接口原型	Int setPageModeHorizontalPostion (int dots);	
接口功能	设置文本打印绝对位置（与打印方向垂直向，x方向）	
参数说明	入参	Dots （毫米mm*8）
	出参	无
返回值	0	
补充说明		

6.4.27 进入标签模式及标签打印、进纸（pageModePrint）

接口原型	Int pageModePrint (int pm);	
接口功能	进入标签模式及标签打印、进纸	
参数说明	入参	pm 0 进入标签模式且初始化参数 1 打印并保存标签（或进一张标签） 2 打印并清除数据 3 清空数据
	出参	无
返回值	0	
补充说明	首先要进入标签模式（初始化）,设置标签区域，添加文本或addimage位图数据（推荐位图打印）、打印	

6.4.28 校验标签（pageModeCheckLabel）

接口原型	Int pageModePrint (int pm);	
接口功能	校验标签	
参数说明	入参	无
	出参	无
返回值	0	
补充说明	第一次使用标签打印机或者新类型的纸（比如60*40换成60*30）需要校验纸张，让打印机识别纸张，否则会导致进纸不正确	

7、 主板钱箱模块声明

7.1 模块说明

用于操作的钱箱设备。

7.2 包路径

com.histonepos.npsdk.cashbox

7.3 AIDL文件名

ICashBoxService.aidl

7.4 接口清单

7.4.1 打开钱箱(openCashBox)

接口原型	int openCashBox(int id);	
接口功能	打开钱箱	
参数说明	入参	Id:1:：钱箱1　　2：钱箱2
	出参	无
返回值	成功　0 失败　负值	
补充说明		

7.4.2 获取钱箱状态(getCashBoxStatus)

接口原型	int getCashBoxStatus();	
接口功能	获取钱箱状态	
参数说明	入参	无
	出参	无

返回值	成功 0 钱箱开 1：钱箱关 失败 负值
补充说明	非打印机控制钱箱接口

8、 磁卡模块声明

8.1 模块说明

用于操作磁卡设备。

8.2 包路径

com.histonepos.npsdk.msr

8.3 AIDL文件名

IMsrService.aidl

8.4 接口清单

8.3.1 磁卡初始化(openMsr)

接口原型	int openMsr()	
接口功能	磁卡初始化	
参数说明	入参	无
	出参	无
返回值	成功 0 失败 负值	
补充说明		

8.3.2 设置超时时间(setTimeout)

接口原型	void setTimeout(int t);	
接口功能	设置超时	
参数说明	入参	t 单位秒
	出参	无
返回值	无	
补充说明		

8.3.3 注册磁卡监听器(setMsrListener)

接口原型	void setMsrListener(IMsrEventListener listener);

接口功能	注册磁卡监听器,并开始检卡	
参数说明	入参	listener：磁卡回调
	出参	无
返回值	无	
补充说明		

8.3.4 停止检测(stopCheck)

接口原型	void stopCheck();	
接口功能	停止检测	
参数说明	入参	无
	出参	无
返回值	无	
补充说明	超时未到时可以 停止检测	

8.3.5 释放磁卡资源(closeMsr)

接口原型	int closeMsr();	
接口功能	释放磁卡资源	
参数说明	入参	无
	出参	无
返回值	成功 0 失败 负值	
补充说明		

8.5 磁条事件监听器

```
IMsrEventListener {

/*respCode 成功= 0, 其它失败

 成功时 Bundle 保存
        KEY (String)：”1” 或“2”或”3”
       例如第一轨 trackResult .getByteArray(“1”)
         */
void onTrack(int respCode,in android.os.Bundle trackResult);

  /**
    * 检卡超时
   */
   void onTimeout();
}
```

9、 扫描窗模块声明

9.1 模块说明

用于操作扫描窗外设。

9.2 包路径

com.histonepos.npsdk.scanner

9.3 AIDL文件名

IScannerService.aidl

9.4 接口清单

9.4.1 初始化扫描窗(initScanner)

接口原型	int initScanner();	
接口功能	初始化扫描窗	
参数说明	入参	无
	出参	无
返回值	成功 >= 0 失败 < 0	
补充说明		

9.4.2 销毁扫描窗资源(destoryScanner)

接口原型	int destoryScanner();	
接口功能	销毁扫描窗资源	
参数说明	入参	无
	出参	无
返回值	成功 >= 0 失败 < 0	
补充说明		

9.4.3 设置扫描数据监听器(setScanListener)

接口原型	void setScanListener(IScanEventListener listener);	
接口功能	设置扫描数据监听	
参数说明	入参	listener - 按键事件回调
	出参	无
返回值	无	
补充说明		

9.4.4 取消扫描数据监听(removeScanListener)

接口原型	void removeScanListener();	
接口功能	取消扫描数据监听，不再接受扫描数据	
参数说明	入参	无
	出参	无
返回值	无	
补充说明		

10、 非接模块声明

10.1 模块说明

用于操作设备的非接设备-----M卡。

10.2 包路径

com.histonepos.npsdk.picccard

10.3 AIDL文件名

IPiccCardService.aidl

10.4 接口清单

10.4.1 非接初始化(piccOpen)

接口原型	`int piccOpen();`	
接口功能	非接初始化	
参数说明	入参	无
	出参	无
返回值	成功 0 失败 负值	
补充说明	注意! 初始化与piccClose是对应的，用完之后需要piccClose	

10.4.2 非接释放(piccClose)

接口原型	`void piccClose();`	
接口功能	磁卡释放	
参数说明	入参	无
	出参	无

返回值	无
补充说明	注意！ 初始化与piccClose是对应的，用完之后需要piccClose

10.4.3 设置超时(setTimeout)

接口原型	`void setTimeout (int t);`	
接口功能	设置超时，单位秒	
参数说明	入参	t 秒
	出参	无
返回值	无	
补充说明		

10.4.4 设置事件监听(setPiccListener)

接口原型	`void setPiccListener (IPiccEventListener listener);`	
接口功能	设置事件监听，并开始检卡	
参数说明	入参	listener
	出参	无
返回值	无	
补充说明		

10.4.5 停止检卡(stopCheck)

接口原型	`void stopCheck ();`	
接口功能	停止检卡	
参数说明	入参	无
	出参	无
返回值	无	
补充说明		

10.4.6 M卡认证(piccMfAuthBlock)

接口原型	`int piccMfAuthBlock(byte type, byte blockaddr,in byte[] keydata, in byte[] serialInfo1);`	
接口功能	认证	
参数说明	入参	@param type 认证类型，使用 KEYA,KEYB 取值定义:0x00:KEYA 0x01:KEYB @param blockaddr 要认证的块号 @param keydata 值(明文传入) 6bytes

		@param serialInfo1 复位信息(设备卡号，不包含长度信息)
	出参	无
返回值	成功 0 失败 其他	
补充说明		

10.4.7 M卡读数据(piccMfReadBlock)

接口原型	int piccMfReadBlock(byte blockaddr, inout BytesArray blockdata);	
接口功能	读数据	
参数说明	入参	@param blockaddr 块号
	出参	@param blockdata 成功时返回的数据 共16字节数据
返回值	成功 0 失败 其他	
补充说明		

10.4.8 M卡写数据(piccMfWriteBlock)

接口原型	int piccMfWriteBlock(byte blockaddr, in byte[] data);	
接口功能	写数据	
参数说明	入参	@param blockaddr 块号 @param data 待写入数据, 必须是 16 个字节的数据，若不足内部补零 0x00
	出参	无
返回值	成功 0 失败 其他	
补充说明		

10.4.9 M卡初值(piccMfInitValue)

接口原型	int piccMfInitValue (byte blockaddr, int value);	
接口功能	初值	
参数说明	入参	@param blockaddr 块号 @param value
	出参	无
返回值	成功 0 失败 其他	
补充说明	998FPM_V30读卡器不支持	

10.4.10 M卡加值(piccMfAddValue)

接口原型	int piccMfAddValue(byte blockaddr, int value);
接口功能	加值

参数说明	入参	@param blockaddr 块号 @param value
	出参	无
返回值	成功 0 失败 其他	
补充说明	998FPM_V30读卡器不支持	

10.4.11 M卡减值(piccMfReduceValue)

接口原型	`int piccMfReduceValue(byte blockaddr, int value);`	
接口功能	减值	
参数说明	入参	@param blockaddr 块号 @param value
	出参	无
返回值	成功 0 失败 其他	
补充说明	998FPM_V30读卡器不支持	

10.4.12 非接检卡(piccDetect)

接口原型	`int piccDetect(byte mode,`
	`out byte []cardType,`
	`out byte []serialInfo,`
	`out byte []cid,`
	`out byte []other);`	

接口功能	非接检卡	
参数说明	入参	@param Mode * 0x00-搜寻 A 型卡和 B 型卡一次, 此模式适用于需要增强多卡检测功能的场合 * 0x01-搜寻 A 型卡和 B 型卡一次；此模式为 EMV 寻卡模式，通常使用该模式 * ‘a’或‘A’-只搜寻 A 型卡一次 * 'b'或'B'-只搜寻 B 型卡一次 * ‘m’或‘M’只搜寻 M1 卡一次
	出参	* @param CardType 存放卡片类型,目前均返回一字节的类型值 0x01:TYPE A 0x02:TYPE B 0x04:ISO15693 暂不支持 0x08:ICODE1 暂不支持 0x10:MIFARE1 0x20:Mifare S50 0x40:Mifare One S70 * @param SerialInfo 存放卡片序列号的缓冲区 * L+V；L：一字节的序列号长度，V：长度为 L 的序列号信息。一般序列号为 4 字节、7 字节或 10 字节

		* * @param CID 存放卡片逻辑通道号的缓冲区指针,该通道号由驱动内部分配和指定,取值范围为 0~14 目前仅允许感应区内存在一张卡，故 CID[0]总是返回 0x00 * @param Other 存放详细错误代码、卡片响应信息等内容的缓冲区指针，可为 NULL * Other：错误号（2 字节）+卡片相关信息（L+V) 卡片相关信息具体为： 对于 A 型卡，返回卡片的 ATS(Answer To Select)信息，其长度不超过 62 字节 对于 B 型卡，返回卡片的 ATQB(Answer To Request B)信息，其长度为 12 字节 对于 M1 卡，返回卡片的 ATQA(Answer To Request A)信息，其长度为 2 字节 有关 ATS、ATQB、ATQA 的详细信息请查阅 ISO14443-3、ISO14443-4 的相关部分。 Other[…299]：Other指向的缓冲区大小至少应为300字节
返回值	成功 0 失败 负值	
补充说明	与setPiccListener不可同时使用	

10.4.13 发送APDU命令(isoCommand)

接口原型	byte[] isoCommand(byte[] data)	
接口功能	磁卡	
参数说明	入参	data apdu 指令
	出参	@param apduRecv 从PICC卡返回的数据结构,含SWA SWB等
返回值	返回值里的第一，第二个字节，分别是返回的SWA和SWB。	
补充说明		

10.4.14 获取非接卡片类型(piccGetCardType)

接口原型	Byte piccGetCardType ();	
接口功能	获取非接卡片类型	
参数说明	入参	0x01:TYPE A 0x02:TYPE B 0x04:ISO15693 暂不支持 0x08:ICODE1 暂不支持 0x10:MIFARE1 0x20:Mifare S50 0x40:Mifare One S70

		未执行 piccDetec 时，返回 0x00
	出参	无
返回值	成功 0 失败 其他	
补充说明	此函数应在piccDetect函数调用后使用且返回的piccDetect的参数返回类型一致	

10.5

监听事件

```
interface IPiccEventListener {
    /**
     * 非接卡激活成功
     */
    void onCardActivate(byte[] serialInfo);

    /**
      * 检卡超时
     */
     void onTimeout();
}
```

11、 串口模块声明

11.1 模块说明

用于操作设备的串口设备。

11.2 包路径

com.histonepos.npsdk.uart

11.3 AIDL文件名

IUartDriver.aidl

11.4 接口清单

11.4.1 打开串口设备(openUart)

接口原型	int openUart (int Slot);	
接口功能	打开串口设备	
参数说明	入参	Slot:端口号
	出参	无

返回值	成功 返回文件描述符 错误 返回负值
补充说明	

11.4.2 关闭串口设备(closeUart)

接口原型	void closeUart (int Slot);	
接口功能	关闭串口设备	
参数说明	入参	Slot:端口号
	出参	无
返回值	无	
补充说明		

11.4.3 读串口数据(readUart)

接口原型	int readUart(int Slot,out byte [] value,int len);	
接口功能	读串口数据	
参数说明	入参	Slot：串口编号 Len：期望读取的数据长度
	出参	Value：读取的数据内容
返回值	成功 实际读取的数据长度 失败 负值	
补充说明		

11.4.4 写串口数据(writeUart)

接口原型	int writeUart(int Slot,in byte [] value,int len);	
接口功能	写串口数据	
参数说明	入参	Slot：串口编号 Value：数据内容 Len：需要写入的数据长度
	出参	
返回值	成功 实际写入的数据长度 失败 负值	
补充说明		

11.4.5 清除串口数据(flushUart)

接口原型	int flushUart (int Slot);	
接口功能	清除串口数据	
参数说明	入参	Slot：串口编号

	出参	
返回值	成功 0 失败 负值	
补充说明		

11.4.6 设置串口波特率(setBaudateRate)

接口原型	Int setBaudateRate(int port,int baudrate,boolean ctrlOPen);	
接口功能	设置串口波特率	
参数说明	入参	Port：串口编号 Baudrate：波特率 ctrlOPen：流控
	出参	
返回值	成功 0 失败 负值	
补充说明	openUart之后使用	

12、 发袋机模块声明

12.1 模块说明

用于操作设备的发袋机设备。

12.2 包路径

com.histonepos.npsdk.bagger

12.3 AIDL文件名

IBaggerService.aidl

12.4 接口清单

12.4.1 自动发袋(autoOutBag)

接口原型	boolean autoOutBag(int position, int num);	
接口功能	自动发袋，后续的状态通过 IBaggerEventListener 回调	
参数说明	入参	position - 塑料袋机出袋通道的序号，1,2,3... num - 出袋数量，大于0
	出参	无
返回值	成功 true 错误 false	
补充说明		

12.4.2 重置历史出袋数量(clearBaggerHistoryTotal)

接口原型	boolean clearBaggerHistoryTotal(int position);	
接口功能	发袋机历史出袋数量置0	
参数说明	入参	position - 塑料袋机出袋通道的序号，1,2,3...
	出参	无
返回值	成功 true 错误 false	
补充说明	注意：第二代吐袋机该功能无效！仅第一代吐袋机有效，并且固件中不支持针对某一通道单独清零，无论position为1或2，两个通道数据都会同时清空	

12.4.3 获取出袋历史数量(getBaggerHistoryTotal)

接口原型	int getBaggerHistoryTotal(int position);	
接口功能	获取出袋通道的历史出袋的数量	
参数说明	入参	position - 塑料袋机出袋通道的序号，1,2,3...
	出参	无
返回值	指定通道的历史出袋数量	
补充说明	注意：第二代吐袋机该功能无效！仅第一代吐袋机有效！	

12.4.4 获取出袋通道状态(getBaggerStatus)

接口原型	int getBaggerStatus(int position);	
接口功能	获取出袋通道的状态	
参数说明	入参	position - 塑料袋机出袋通道的序号，1,2,3...
	出参	
返回值	指定通道的状态 1：正常可使用 0：无购物袋 -1：通道门处于开启状态 -2：其他不能正常工作异常 -3：无发袋机	
补充说明		

12.4.5 设置出袋通道监听器(setBaggerListener)

接口原型	void setBaggerListener(int position, IBaggerEventListener listener);	
接口功能	设置出袋通道各种事件的回调	
参数说明	入参	position - 塑料袋机出袋通道的序号，1,2,3... listener - 出袋事件回调，可以为null，表示清除回调 @see IBaggerEventListener
	出参	
返回值	无	

补充说明	IBaggerEventListener 回调事件
	/**
	* 通道状态发生变化回调
	*
	* @param newStatus 通道新的状态
	* 常量参看{@link#getBaggerStatus(int)} 的返回
	* @param msg 状态详细描述信息或原因
	*/
	void onBaggerStatusChanged(int newStatus, String msg);
	
	/**
	* 开始单次出袋事件
	*/
	void onBaggerOutputStart();
	
	/**
	* 完成单次出袋事件
	*/
	void onBaggerOutputEnd();
	
	/**
	* 出袋完成事件，这是一个总的事件回调，和 {@link #onBaggerOutputEnd()} 不冲突
	*
	* <pre>
	* 如：一个出单袋指令发出去或手动拉出一个袋子，会收到一次 onBaggerOutputEnd 和一次 onBaggerCharge
	* 一个出多袋的指令发出去之后，会收到多次 onBaggerOutputEnd，但只会收到一次 onBaggerCharge
	* </pre>
	* @param num 出袋的数量
	*/
	void onBaggerCharge(int num);

13、 指示灯模块声明

13.1 模块说明

用于操作设备的指示灯设备。

13.2 包路径

com.histonepos.npsdk.light

13.3 AIDL文件名

ILightService.aidl

13.4 接口清单

13.4.1 点亮指示灯(setLight)

接口原型	boolean setLight(String status);	
接口功能	点亮灯颜色	
参数说明	入参	status - 定制固件的灯状态码 “HM_LIGHT_OFF” - 灯灭, “HM_LIGHT_ON_BLUE” : 蓝灯亮, "HM_LIGHT_ON_RED" : 红灯亮, "HM_LIGHT_BLINK_BLUE" : 蓝灯闪烁, "HM_LIGHT_BLINK_RED" : 红灯闪烁, status - 普通固件的灯状态码 “NORMAL_LIGHT_ON_ORANGE” : 橙灯亮, "NORMAL_LIGHT_ON_RED" : 红灯亮, "NORMAL_LIGHT_ON_GREEN" : 绿灯亮, “NORMAL_LIGHT_OFF_ORANGE” : 橙灯灭, "NORMAL_LIGHT_OFF_RED" : 红灯灭, "NORMAL_LIGHT_OFF_GREEN" : 绿灯灭, “NORMAL_LIGHT_BLINK_ORANGE” : 橙灯闪, "NORMAL_LIGHT_BLINK_RED" : 红灯闪, "NORMAL_LIGHT_BLINK_GREEN" : 绿灯闪,
	出参	无
返回值	成功 true 错误 false	
补充说明	定制固件的灯如下： • 1、闪烁间隔 400ms。 • 2、各指令间互斥。例如 当前蓝灯亮，当接收到"ON_RED"后，切换到红灯亮	

13.4.2 获取指示灯颜色(getLightStatus)

接口原型	String getLightStatus();	
接口功能	获取灯颜色	
参数说明	入参	无
	出参	无
返回值	顶灯状态码，见 setLight(java.lang.String) 参数	
补充说明	暂不支持	

13.4.3 控制蜂鸣器(buzzerCtrl)

接口原型	boolean buzzerCtrl(String status);	
接口功能	控制蜂鸣器	
参数说明	入参	status -状态码 “BUZZER_ON” 蜂鸣器响 “BUZZER_OFF”关闭蜂鸣器
	出参	无
返回值	成功 true 错误 false	
补充说明	普通固件的灯才有蜂鸣器，定制固件的灯不支持此操作	

14、 物理按键模块声明

14.1 模块说明

用于操作设备的物理按键设备。

14.2 包路径

com.histonepos.npsdk. physicalkey

14.3 AIDL文件名

IPhysicalKeyService.aidl

14.4 接口清单

14.4.1 获取按键灯状态(getKeyLightStatus)

接口原型	String getKeyLightStatus(int keyType);	
接口功能	获取按键灯状态	
参数说明	入参	keyType - 按键类型 1：帮助按钮
	出参	无
返回值	灯状态码，见 setKeyLight(int, String) status 定义	
补充说明	暂不支持	

14.4.2 设置指示灯状态(setKeyLight)

接口原型	boolean setKeyLight(int keyType, java.lang.String status);	
接口功能	设置按键灯状态	
参数说明	入参	keyType - 按键类型 1：帮助按钮 status - 按键灯状态

		“BUTTON_OFF” ：按钮灭, “BUTTON_ON” ：按钮亮, "BUTTON_BLINK"：按钮闪烁
	出参	无
返回值	成功 true 错误 false	
补充说明		

14.4.3 设置按键事件监听器(setKeyListener)

接口原型	void setKeyListener(int keyType, IEventListener listener);	
接口功能	设置按键事件监听器	
参数说明	入参	keyType - 按键类型 1：帮助按钮 listener - 按键事件回调，可以为null，表示清除回调 @see IEventListener
	出参	无
返回值	无	

补充说明	IEventListener 回调事件
	/**
	* 按钮点击事件
	*
	* @param keyType 按键类型
	* <pre>
	* 1：帮助按钮
	* </pre>
	*/
	void onClick(int keyType);
	
	/**
	* 按钮长按2秒以上触发的事件，触发该事件同时不需触发按钮点击事件
	*
	* @param keyType 按键类型
	* <pre>
	* 1：帮助按钮
	* </pre>
	*/
	void onLongClick(int keyType);	

15、 秤模块声明

15.1 模块说明

用于操作设备的串口秤设备。

15.2 包路径

com.histonepos.npsdk.scale

15.3 AIDL文件名

IScaleService.aidl

15.4 接口清单

15.4.1 秤初始化(open)

接口原型	boolean open(String portSetting);	
接口功能	秤初始化	
参数说明	入参	portSetting 端口设置
	出参	无
返回值	true 成功，false失败	
补充说明		

15.4.2 释放秤资源 (close)

接口原型	Boolean close();	
接口功能	释放秤资源	
参数说明	入参	无
	出参	无
返回值	true 成功，false失败	
补充说明		

15.4.3 获取重量(getWeight)

接口原型	void getWeight(IScaleEventListener scaleListener);	
接口功能	获取重量	
参数说明	入参	scaleListener 事件监听
	出参	无
返回值	无	
补充说明		

15.4.4 归零(setZeroWeight)

接口原型	int setZeroWeight();	
接口功能	归零	
参数说明	入参	无
	出参	无
返回值	0-发送命令成功　其它失败	
补充说明		

15.4.5 物理去皮扣重(setTareWeight)

接口原型	int setTareWeight();	
接口功能	去皮扣重	
参数说明	入参	无
	出参	无
返回值	0-发送命令成功　其它失败	
补充说明	使用秤的物理特性进行去皮 注意：此接口与“setDigitalTareWeight(int)”互斥，同时只能使用一种去皮方式，需要先执行消皮（exitTareWeight()）才能切换到另一个去皮方式。	

15.4.6 数字去皮扣重(setTareWeight)

接口原型	int setDigitalTareWeight(int tare);	
接口功能	去皮扣重	
参数说明	入参	Tare，设置的皮重
	出参	无
返回值	0-发送命令成功　其它失败	
补充说明	未使用秤的物理特性去皮，只是在sdk做了去皮处理。 注意： 1. 此接口与“setTareWeight()”互斥，同时只能使用一种去皮方式，需要先执行消皮（exitTareWeight()）才能切换到另一个去皮方式。 2. 非特殊场景，尽量使用“物理去皮扣重”接口，而不是此接口。	

15.4.7 退出去皮扣重(exitTareWeight)

接口原型	int exitTareWeight();	
接口功能	消皮,即取消皮重	
参数说明	入参	无

	出参	无
返回值	0-发送命令成功　其它失败	
补充说明		

15.4.8　配置秤(config)

接口原型	Void config(Bundle type);	
接口功能	配置不同的秤	
参数说明	入参	Bundle 携带秤信息的bundle。详见demo程序
	出参	无
返回值		
补充说明		

15.5　事件监听

```
interface IScaleEventListener{

    /**数据变化事件
     * @param strData　未处理的原始数据（调试用）或调用者自己解析重量
     */
     void　dataMessage(String strData);

    * 重量变化事件
 * @param status
 ST:稳定
UT:不稳定
Flow:超重溢出或者非去皮情况下重量小于-20d
FT:无数据或离线 ER:未知错误
 * @param weightType NW:净重 GW:毛重
 * @param zeroPoint　零点是否为正（"+"为正，"-"为负）
 * @param weightVaule 重量（NW模式下就是净重，GW模式下就是毛重）
 * @param unit 单位 kg或g
 */
     void　weightChange(String status, String weightType,String zeroPoint,String weightVaule,
String unit);
}

/**
     * 重量获取事件
     * @param status
```

ST:稳定
UT:不稳定
Flow:超重溢出或者非去皮情况下重量小于-20d
FT:无数据或离线　其它值:未知错误

```
     * @param netWeight  净重(毛重-皮重)
     * @param tareWeight  皮重        为负值表示不支持皮重获取
     * @param unit  单位  kg或g
     */
        void    onWeightBack(String status, float netWeight, float tareWeight,String unit);
```

16、 身份证模块声明

16.1 模块说明

用于操作设备的身份证模块设备。

16.2 包路径

com.histonepos.npsdk.idcard

16.3 AIDL文件名

IIdCardService.aidl

16.4 接口清单

16.4.1 打开设备(openIdCard)

接口原型	Int openIdCard()	
接口功能	打开设备	
参数说明	入参	无
	出参	无
返回值	0　正常 <0　失败	
补充说明		

16.4.2 关闭设备(closeIdCard)

接口原型	Int closeIdCard()	
接口功能	关闭设备	
参数说明	入参	无
	出参	无
返回值	0　成功 <0　失败	
补充说明		

16.4.3 设置超时时间(setTimeout)

接口原型	void setTimeout(int t);	
接口功能	设置超时	
参数说明	入参	t 单位秒
	出参	无
返回值	无	
补充说明		

16.4.4 注册监听器(setIdCardListener)

接口原型	void setIdCardListener(IIdCardEventListener listener);	
接口功能	注册监听器,并开始检卡	
参数说明	入参	listener：身份证卡回调
	出参	无
返回值	无	
补充说明		

16.4.5 停止检测(stopCheck)

接口原型	void stopCheck();	
接口功能	停止检测	
参数说明	入参	无
	出参	无
返回值	无	
补充说明	超时未到时可以 停止检测	

16.4.6 监听事件

```
interface IIdCardEventListener{
    /*
        respCode : 0 成功，其它值：失败，details = null
    */
    void onResult(int respCode,in IIdCardDetails details);

    /**
     * 检卡超时
    */
    void onTimeout();
}
```