



赞同科技股份有限公司

背夹接口统一规范文档

版权声明:

本文档的版权归北京赞同股份有限公司所有，未经授权不得复制和引用。本文档若有改动，恕不另行通知。

本书约定:

1、通用格式约定


宋体：正文采用小四号宋体。在本文中出现的本文所描述的产品名称以及自定义名词采用 5 号宋体加下划线表示


楷体：警告、说明等提示等内容一律用楷体，并且在内容前后增加线条与正文隔离。

屏幕显示描述：对于来自屏幕的显示信息，采用如下格式：英文Courier New，中文宋体，文字大小为小5号；信息中夹杂的用户从终端输入的信息，使用加粗字体表示。

2、各类标志

本书还采用各种醒目标志来表示在操作过程中应该特别注意的地方，这些标志意义如下：

：警告、危险、提醒操作中应注意的事项。

：说明、注意、提醒操作中应注意的事项。

编制说明:

编制部门:	研究院	密 级:	内部绝密
编 制 人:	李俊逸	编制日期:	2019.03.05
审 核 人:		审核日期:	
文件编号:		文件版本:	V1.6.1

变 更 记 录

序号	修改内容	页号	修改人	修改日期
1.	文档出版定制		任仲瑞	2019-01-17
2.	兼容多方厂商的外设 SDK		李俊逸	2019-02-20
3.	规范变量名字，统一使用小驼峰，修改灌注密钥方法的返回值，从 JSONObject 改成 int		李俊逸	2019-03-20
4.	兼容多厂商 SDK，通过包名与类名区分厂商		张子扬	2019-04-23
5.	增加获取设备模块信息接口 (getDeviceState)		李俊逸	2019-05-20
6.	修改 getICCARQC()接口的入参		李俊逸	2019-07-02
7.	增加了读取 ic 基本信息和 ARQC 的方法 4.6		陶宏晖	2019-07-03
8.				
9.				
10.				
11.				
12.				
13.				
14.				
15.				
16.				
17.				
18.				
19.				
20.				
21.				
22.				
23.				
24.				

注：对该文件内容增加、删除或修改均需填写此变更记录，详细记载变更信息，以保证其可追溯性。

目 录

第一章 产品概述.....	1
1.1. 编写目的.....	1
1.2. 包名.....	1
1.3. 目标.....	1
第二章 设备连接.....	错误! 未定义书签。
2.1. 连接设备.....	6
2.2. 关闭设备.....	7
2.3. 获取电池信息.....	7
2.4. 获取背夹序列号.....	8
第三章 身份证.....	10
3.1. 读取身份证信息.....	10
3.2. 读取港澳台居民居住证信息.....	11
3.3. 读取外国人永久居留身份证信息.....	13
第四章 IC 卡.....	14
4.1. IC 卡读取基本信息.....	14
4.2. IC 卡上电.....	16
4.3. IC 卡下电.....	17
4.4. IC 卡读取 ARQC.....	18
4.5. IC 卡读取交易明细.....	19
第五章 磁条卡.....	25
5.1. 读取 MC 信息.....	25
第六章 指纹.....	25
6.1. 获取上传指纹状态.....	26
6.2. 获取指纹信息.....	26
第七章 密码键盘.....	27
7.1. 下载主密钥.....	27
7.2. 下载工作密钥.....	27
7.3. 读取密码.....	29
第八章 电子签名.....	32
8.1. 启动手写屏.....	35
8.2. 获取手写屏数据.....	35
8.3. 关闭手写屏.....	36
8.4. 获取手写屏签名图片.....	错误! 未定义书签。
第九章 声音和灯控.....	36
9.1. 发起语音.....	36

9.2. 发起蜂鸣.....	37
9.3. LED 灯控制.....	38
第十章 错误码.....	38

第一章 产品概述

1.1.编写目的

本文档对移动外设的驱动接口进行统一的定义，用于在封装多个厂商驱动时，保证不同厂商提供的接口无冲突,方便移动应用开发时快速适应不同厂商设备的接入与扩展。

1.2.包名

jar 包名称:

类名: XxxDeviceProtocol(XXX 为厂商名, 采用大驼峰,如 ZantongDeviceProtocal)

类包名: com.external.xxx.device (如行方对报名有额外要求, 以行方为准)

1.3.目标

- ✧ 实现外设调用接口整齐, 便于 Android 层处理统一输入与输出;
- ✧ 方便不同外设扩展调用;

第二章 通信层接口规范

2.1.错误码

```
public static final int ERR_SUCCESS = 0; //成功
```

```
public static final int ERR_OPEN = -1; //设备连接失败
```

```
public static final int ERR_PARAM = -2; // 参数错误
```

```
public static final int ERR_DEV_BUSY = -3; //设备忙
```

```
public static final int ERR_TIMEOUT = -4; //操作超时
```

```
public static final int ERR_BLUETOOTH_NOT_SUPPORT = -5; //设置不支持蓝牙
```

```
public static final int ERR_BLUETOOTH_NO_PAIRIED_DEV = -6; //未进行配置远程设备蓝牙地址，配对列表中也找不到以配对设备
```

```
public static final int ERR_BLUETOOTH_PAIR_FAILE = -7; //配对失败
```

```
public static final int ERR_BLUETOOTH_ENABLE_FAILE = -8; //蓝牙使能失败
```

```
public static final int ERR_BLUETOOTH_CONNECT = -9; //蓝牙连接失败
```

```
public static final int ERR_PACKAGE_WRITE = -10; //写报文失败
```

```
public static final int ERR_PACKAGE_READ = -11; //读报文失败
```

```
public static final int ERR_PACKAGE_READ_CANCEL = -12; //读报文被取消
```

```
public static final int ERR_USB_ENUM_DEV = -13;
```

```
public static final int ERR_USB_FIND_INTERFACE = -14;
```

```
public static final int ERR_USB_ASSIGN_ENDPOINT = -15;
```

```
public static final int ERR_USB_NO_PERMISSION = -16; //请求权限失败
```

public static final int ERR_INNER_RECIVE_BUF_NOT_FULL = -17; //厂商内部缓冲区不足

public static final int ERR_SIGN_STAR_FAIL = -18; //启动签名失败

public static final int ERR_SIGN_NO_SIGN = -19; //客户未签名

public static final int ERR_CANCLE = -20; //操作被取消

public static final int ERR_OPERATOR_FAIL = -21; //操作失败

public static final int ERR_GET_DEV_FILE_INOF = -22; //从设备获取文件的信息失败

public static final int ERR_GET_DEV_FILE_MD5_CHECK = -23; //md5 校验失败

public static final int ERR_OPEN_TIP_FAIL = -24; //开卡机动画失败

public static final int ERR_MAG_CARD_READ = -25; //读磁卡失败

public static final int ERR_MAG_CARD_NO_DATA = -26; //3 个磁道都没有数据

public static final int ERR_MAG_CARD_UNKOWN = -27; //未知错误，刷卡超时

public static final int ERR_ILOAD_FILE_FAIL = -28; //下载文件失败

public static final int ERR_LOOP_THIS_INS = -29; //继续执行这条指令

public static final int ERR_FINGER_CONNECT = -30; //指纹仪连接异常

public static final int ERR_IC_NOT_SUPPORT_CONTACT = -31; //不支持接触用户卡

public static final int ERR_IC_NOT_FIND_CARD = -32; //接触用户卡未插到位

public static final int ERR_IC_CONTACT_POWER_ON_FAIL = -33; //接触用户卡上电失败


```
public static final int ERR_IC_NOT_SUPPORT_PSAM = -34;//不支持 PSAM 卡
```

```
public static final int ERR_IC_PSAM_POWER_ON_FAIL = -35;//PSAM 卡上电失败
```

```
public static final int ERR_IC_NOT_SUPPORT_NOCONTACT = -36;//不支持非接触用户  
卡
```

```
public static final int ERR_IC_NOCONTACT_POWER_ON_FAIL = -37;//非接触用户卡激  
活失败
```

```
public static final int ERR_IC_NOCONTACT_NO_FIND = -38;//等待卡进入感应区超时
```

```
public static final int ERR_IC_NOCONTACT_FIND_MUL = -39;//有多张卡在感应区
```

```
public static final int ERR_IDCARD_FIND_CARD = -40;//身份证寻卡失败
```

```
public static final int ERR_START_UPGRADE = -41;//启动升级失败
```

```
public static final int ERR_DELETE_FILE = -42;//删除文件失败
```

```
public static final int ERR_DRIVER_EXCEPTION = -43;//厂商驱动内部调用发生异常，  
异常将写入 log 文件
```

```
public static final int ERR_ADD_TASK = -44;//添加任务失败,设备忙
```

```
public static final int ERR_IDCARD_PARSE = -45;//身份证信息解析错误
```

```
public static final int ERR_PACKAGE = -46;//报文错误
```

```
public static final int ERR_IC_FINDCARD = -47;//自动找 IC 卡失败
```

```
public static final int ERR_IC_NO_FOUND_SFI = -48;
```

```
public static final int ERR_IC_SFI_LEIGN = -49;
```

```
public static final int ERR_IC_MALLOC = -50;

public static final int ERR_IC_DOL_NO_FOUND = -51;

public static final int ERR_IC_DOL_DATA = -52;

public static final int ERR_IC_DOL_DATA_NO_FOUND = -53;

public static final int ERR_IC_APP_INIT = -54;

public static final int ERR_IC_LOG_NO_ENTRY = -55;

public static final int ERR_IC_RECORD_NO_FOUND = -56;

public static final int ERR_IC_GET_GAC_TOO_MUCH = -57;

public static final int ERR_IC_CMD = -58;

public static final int ERR_IC_GAC_RESP = -59;

public static final int ERR_IC_AUTH_DATA = -60;

public static final int ERR_IC_TRADE_ACC = -61;

public static final int ERR_IC_NO_SCRIPT = -62;

public static final int ERR_IC_WRITE_SCRIPT = -63;

public static final int ERR_IC_SCRIPT_CMD = -64;


public static final int ERR_ZMK_NOT_EXIST = -65; //主密钥不存在

public static final int Err_WKEY_NOT_EXIST = -66; //工作密钥不存在

public static final int ERR_MAC_KEY_NOT_EXIST = -67; //mac 密钥不存在

public static final int ERR_INIT_PIN_NOT_EXIST = -68; //初始密钥不存在

public static final int ERR_GET_DEVICE_NO = -69; //获取设备序列号失败

public static final int ERR_AUTO_FIND_CARD_VALUE_MSG = -70; //找到磁卡
```

```
public static final int ERR_PACKET_CHECK = -71;//报文校验失败
```

```
public static final int ERR_PIN_KEY_CHECK = -72;//密钥校验失败
```

```
public static final int ERR_ID_81M_OPEN = -73;//打开吸入式身份证阅读器失败
```

```
public static final int ERR_FINGER_PRINT_NOT_MATCH = -74;//需要采集的指纹仪与  
背夹连接的指纹仪不匹配
```

```
public static final int ERR_IDCARD_GEN_CARD = -75;//合成正反面照错误
```

```
public static final int ERR_PIN_PWD_NOT_MATCH = -76;//两次录入密码不一致
```

```
public static final int ERR_GET_DEV_FILE_BUF = -77;//缓冲区不足
```

```
public static final int ERR_MODULE_EXCEPTION = -999;//调用模块外设异常
```

2.2.连接设备

方法名: `int openDevice(Context context, int type);`

入参说明:

参数	输入/输出	说明
<code>context</code>	输入	上下文对象
<code>type</code>	输入	0: USB 连接 1: 蓝牙连接

出参说明

返回值类型	说明
int	成功返回 0，失败返回错误码

2.3.关闭设备

方法名：int closeDevice(Context context, int type);

入参说明：无

出参说明：

返回值类型	说明
int	成功返回 0，失败返回错误码

2.4.获取电池信息

方法名：int[] getBatteryInfo();

入参说明：

参数	输入/输出	长度	说明
无			

出参说明：

返回指类型	说明
-------	----

Int[]	<p>batteryInfo[0] :</p> <p>0: 无电池</p> <p>1: 有电池</p> <p>batteryInfo[1] : 电池电流方向</p> <p>0: 放电</p> <p>1: 充电</p> <p>batteryInfo[2] : 电量百分比</p>
-------	--

2.5.获取背夹序列号

方法名: String getDeviceSerialID();

入参说明: 无

出参说明:

返回值类型	返回值说明
String	<p>成功: 返回各厂商自定义的设备序列号。该序列号应该同设备外观上注明的设备编号相同;</p> <p>失败: 错误码。</p>

2.6. 设备是否连接

方法名: int isConnect();

入参说明: 无

出参说明:

返回值类型	返回值说明
int	成功: 0 失败: 错误码

2.7. 检测设备模块信息

方法名: JSONObject getState(String moduleName, int timeout)

入参说明:

参数	输入/输出	说明
moduleName	输入	模块名称: 读卡器: CR 指纹仪: FR 签名版: SR 磁条卡: MR 密码键盘: PR
timeout	输入	超时时间

出参说明:

返回值	返回值说明
state	若失败, 返回对应错误码, 若成功, 该值为 0
data	类型: JSONObject, 例: data:{ " CR ": 0,

	<pre> " FR ": -1, } </pre>
--	--

第三章 身份证

3.1. 读取身份证信息

方法名: `JSONObject getIDCardInfo(int timeout, Context context);`

入参说明:

参数	输入/输出	说明
timeout	输入	读取身份证信息超时时间
Context	输入	上下文信息

出参说明:

返回值	返回值说明
code	若失败, 返回对应错误码, 若成功, 该值为 0
cardType	类型: int, 0: 大陆居民 1: 外国剧名, 2: 港澳台居民
chineseName	类型: String, 中文姓名
englishName	类型: String, 英文名
sex	类型: int, 性别, 男: 1, 女: 0

nationality	类型：String，民族
birthday	类型：String，出生日期（YYYYMMDD）
address	类型：String，住址
idcardNO	类型：String，身份证号码（通行证号码）
passportNO	类型：String，签证版本号（外国人身份证信息才有此属性）
organizationCode	类型：String，签发机关
expiryDate	类型：String，有效期限，（格式：YYYYMMDD-YYYYMMDD或YYYYMMDD-长期）
idPhoto	类型：JSONObject，身份证照片(本地存储路径) { “FrontImagePath”：“xxx”， // 正面照片路径 “BackImagePath”：“xxx”， // 背面照片路径 “MergeImagePath”：“xxx”， // 合成照片路径 “AvatarImagePath”：“xxx” // 裁剪后头像照片路径 }
fpId	类型：String，身份证指纹仪信息（1024 字节指纹信息 base64 编码）

3.2.读取港澳台居民居住证信息

方法名：同 3.1

入参说明：同 3.1

出参说明：

返回值	返回值说明
code	若失败，返回对应错误码，若成功，该值为 0
chineseName	姓名
sex	性别
idcardNO	身份证号
birthday	出生日期（YYYYMMDD）
organizationCode	签发机关
address	居住地址
expiryDate	类型：String， 有效期限，（格式：YYYYMMDD-YYYYMMDD 或 YYYYMMDD-长期）
passCheck	通行证号
checkTimes	签发次数
idPhoto	类型：JSONObject，身份证照片(本地存储路径) { “FrontImagePath”：“xxx”， // 正面照片路径 “BackImagePath”：“xxx”， // 背面照片路径 “MergelImagePath”：“xxx”， // 合成照片路径

	<pre> “AvatarImagePath”：“xxx” // 裁剪后头像照片路径 } </pre>
fpld	身份证指纹仪信息（1024 字节指纹信息 base64 编码）
cardType	类型：int， 0：大陆居民 1：外国居民， 2：港澳台居民

3.3.读取外国人永久居留身份证信息

方法名：同 3.1

入参说明：同 3.1

出参说明：

返回值	返回值说明
code	若失败，返回对应错误码，若成功，该值为 0
chineseName	姓名
sex	性别
idcardNO	身份证号
birthday	出生日期（YYYYMMDD）
organizationCode	签发机关
areaNO	国籍或所在地区代码
expiryDate	类型：String， 有效期限，（格式：YYYYMMDD-YYYYMMDD）

	或 YYYYMMDD-长期)
cardVersion	证件版本
idPhoto	类型：JSONObject，身份证照片(本地存储路径) <pre>{ "FrontImagePath" : "xxx", // 正面照片路径 "BackImagePath" : "xxx", // 背面照片路径 "AvatarImagePath" : "xxx" // 裁剪后头像照片路径 }</pre>
fpld	身份证指纹仪信息（1024 字节指纹信息 base64 编码）
cardType	类型：int， 0：大陆居民 1：外国剧名， 2：港澳台居民

第四章 IC 卡

4.1.IC 卡读取基本信息

方法名：

JSONObject getICCardInfo(int icType, String aidList, String tagList,

int timeout);

入参说明：

参数	输入/输出	长度	说明
icType	输入	1	IC 卡类型，1 接触，2 非接，3 自动识别

aidList	输入		值为金融 IC 卡的应用列表，借记卡为“A000000333010101”，贷记卡为“A000000333010102”，也可以传入空，NULL 或 “”。
tagList	输入		要读取的信息标签，例:”A”
timeout	输入		超时时间

tagList 标签列表:

数据元	来自 IC 卡的数据	标签	标签 ASCII 码
卡号		0x41	A
姓名		0x42	B
证件类型	HEX 值: 00: 身份证 01: 军官证 02: 护照 03: 入境证 04: 临时身份证 05: 其它	0x43	C
证件号码		0x44	D
余额	不带小数点	0x45	E

余额上限	不带小数点	0x46	F
单笔交易限额		0x47	G
交易货币代码		0x48	H
失效日期		0x49	I
IC 卡序列号		0x4A	J
二磁道信息		0x4B	K
一磁道信息		0x4C	L
返回所有信息			ALL

出参说明:

返回值类型	返回值说明
JSONObject	返回读取信息 例: <pre>{ "code" : "0", //是否刷卡成功, 成功为 0, 失败为对应错误码 "A": "0196228231825021494762" }</pre> 格式: 标签 (1 个字节) + 十进制长度 (3 个字节) + 数据 (N 字节) ; (标签为 A, 长度为 019, 内容为 6228231825021494762)

4.2.IC 卡上电

方法名: `int iccPowerOn(int icType, int timeout, byte[] attrs);`

入参说明:

参数	输入/输出	长度	说明
icType	输入		IC 卡类型, 0 接触, 1 非接
timeout	输入		超时时间
attrs	输出		Attrs 数据 hex string

出参说明:

返回值类型	返回值说明
int	成功返回 0, 失败返回错误码

4.3.IC 卡下电

方法名:

```
int iccPowerOff(int icType);
```

入参说明:

参数	输入/输出	长度	说明
icType	输入		IC 卡类型, 0 接触, 1 非接

出参说明:

返回值类型	返回值说明
int	成功返回 0, 失败返回错误码

4.4.IC 卡读取 ARQC

说明：根据柜员交易类型、交易金额、交易时间等交易数据，按 PBOC2.0 规定，对 IC 卡执行一系列操作，生成 ARQC，提供给后台进行 IC 卡联机认证。

方法名：

```
JSONObject getICCARQC(int IcType, String tradeData, String AidList,  
  
);
```

入参说明：

参数	输入/输出	长度	说明
icType	输入		IC 卡类型，0 接触，1 非接
tradeData	输入		交易数据，产生 ARQC 的数据，包含多个标签变量，参考下面标签列表
aidList	输入		应用列表，值为金融 IC 卡的应用列表，借记卡为“A000000333010101”，贷记卡为“A000000333010102”，也可以传入空，NULL 或“”。

标签列表：

数据元	来自终端的数据	标签	标签 ASCII 码
授权金额	不带小数点，单位为分	0x50	P
其它金额	不带小数点，单位为分	0x51	Q
交易货币代码	0x0156 (人民币 CNY)	0x52	R

交易日期	YYMMDD	0x53	S
交易类型	《中国银联银行卡联网联合技术规范 V2.1 第2部分 报文接口规范.pdf》 6.4 域 3 交易处理码	0x54	T
交易时间	时分秒中间无分隔符 如： “110530”	0x55	U
商户名称		0x57	W

出参说明:

返回值类型	返回值说明
JSONObject	<pre>{ "code": 成功返回 0，失败返回错误码， "ARQC": 授权请求密文及其相关数据, 按银联规范中 55 域的数据, ASCII 码的形式, "tradeAppData": 交易应用数据，将在检验 ARPC 时作为传入的数据。长度大于 5K 字节。 }</pre>

4.5.IC 卡读取交易明细

方法名:

```
int getICCTradeDetail(int IcType,String AidList,byte[] tradeDetail);
```

入参说明:

参数	输入/输出	长度	说明
----	-------	----	----

icType	输入		IC 卡类型，0 接触，1 非接
aidList	输入		用列表，值为金融 IC 卡的应用列表，借记卡为“A0000000333010101”，贷记卡为“A0000000333010102”，也可以传入空，NULL 或“”。
tradeDetail	输出		返回的交易日志，每条日志数据包含多个标签变量。数据组合形式为：明细条数（2 位）+明细总长度（3 位）+各条明细数据。各条明细数据组合形式为：标签（1 位）+长度（3 位）+数据

tradeDetail 标签列表

数据元	数据类型	标签	标签 ASCII 码
授权金额	不带小数点,单位为分	0x50	P
其它金额	不带小数点,单位为分	0x51	Q
交易货币代码		0x52	R
交易日期	YYMMDD	0x53	S
交易类型	2 字节	0x54	T
交易时间	HHMMSPDB	0x55	U
终端国家代码		0x56	V
商户名称		0x57	W
应用交易计数器（ATC）	按十进制输出	0x58	X

出参说明:

返回值类型	说明
int	成功返回 0，失败返回错误码

4.6.IC 卡读取全部信息

方法名:

JSONObject getICCardInfoAndARQC(int icType, String aidList, String tagList, String tradeData, int timeout);

入参说明:

参数	输入/输出	长度	说明
icType	输入	1	IC 卡类型，1 接触，2 非接，3 自动识别
aidList	输入		值为金融 IC 卡的应用列表，借记卡为“A0000000333010101”，贷记卡为“A0000000333010102”，也可以传入空，NULL 或 “”。
tagList	输入		要读取的信息标签，例:”A”
tradeData	输入		交易数据，产生 ARQC 的数据，包含多个标签变量，参考下面标签列表

timeout	输入		超时时间
---------	----	--	------

tagList 标签列表:

数据元	来自 IC 卡的数据	标签	标签 ASCII 码
卡号		0x41	A
姓名		0x42	B
证件类型	HEX 值: 00: 身份证 01: 军官证 02: 护照 03: 入境证 04: 临时身份证 05: 其它	0x43	C
证件号码		0x44	D
余额	不带小数点	0x45	E
余额上限	不带小数点	0x46	F
单笔交易限额		0x47	G
交易货币代码		0x48	H
失效日期		0x49	I

IC 卡序列号		0x4A	J
二磁道信息		0x4B	K
一磁道信息		0x4C	L
所有信息	所有交易信息		ALL

出参说明:

返回值类型	返回值说明
JSONObject	<p>返回读取信息</p> <p>例:</p> <pre>{ "code" : "0" ,//是否刷卡成功, 成功为 0, 失败为对应错误码 "ARQC": "" //ARQC 字符串, "tradeAppData": "" //校验 ARQC 字符串, "A": "0196228231825021494762" "55field":{"arqc_9F26": ""}</pre> <p>格式: 标签 (1 个字节) + 十进制长度 (3 个字节) +数据 (N 字节) ; (标签为 A, 长度为 019, 内容为 6228231825021494762)</p> <p>55field 格式为键值对格式</p>

55 域 JSON 对象

返回值	返回值说明
arqc_9F26	应用密文

arqc_9F27	密文信息数据
arqc_9F10	发卡行应用数据
arqc_9F37	随机数(4 字节)
arqc_9F36	ATC
arqc_95	终端验证结果
arqc_9A	终端验证结果
arqc_9C	交易类型
arqc_9F02	授权金额
arqc_5F2A	交易货币代码
arqc_82	应用交互特征
arqc_9F1A	终端国家代码
arqc_9F03	其它金额
arqc_9F33	终端性能
arqc_9F13	上次联机应用交易计数器寄存器
arqc_5A	应用主账号
arqc_5F34	应用 PAN 序列号
arqc_CVR	卡片验证结果

第五章 磁条卡

5.1.读取 MC 信息

方法名:

```
JSONObject getMCCardInfo(int tracks, int timeout);
```

入参说明:

参数	输入/输出	长度	说明
tracks	输入		2: 读取 2 磁道 3: 读取三磁道 23: 读取 23 磁道
timeout	输入		等待刷卡时间(秒)

出参说明:

返回值类型	说明
JSONObject	code - 是否刷卡成功, 成功为0, 失败为对应错误码 pan(String) - 主账号 (卡号) track1(String) - 磁道1数据 track2(String) - 磁道2数据 track3(String) - 磁道3数 serviceCode(String) - 服务 expiredDate(String) - 卡片有效期(yyMM)

第六章 指纹

6.1 获取上传指纹状态

方法名: `int getFpBuffer(int timeout);`

功能说明: 将信息采集到缓冲区 0, 可用上传特征值方法 `getFpInfo` 获取指纹特征信息。

入参说明:

参数	输入/输出	说明
timeout	输入	超时时间(秒)

出参说明:

返回值类型	说明
int	缓存成功返回 0, 失败返回错误码, 只有缓存成功才能调用获取指纹方法。

6.2 获取指纹信息

方法名:

`JSONObject getFpInfo(int timeout);`

入参说明:

参数	输入/输出	长度	说明
timeout	输入		超时时间(秒)

出参说明:

返回值类型	说明
-------	----

JSONObject	<pre> { "code": 若成功，该值为 0，若失败，该值为对应错误码， // 若成功，返回下列信息 "picture": <指纹特征图片储存路径>, "fpId": <指纹特征值>, "factory": <厂商代码: tc: 天成 zz: 中正>, } </pre>
------------	--

第七章 密码键盘

7.1. 下载主密钥

方法名:

```
int downloadMasterKey(int algorithmType, int timeout, int mKeyIndex, String mKey, String
checkValue);
```

入参说明:

参数	输入/输出	长度	说明
algorithmType	输入		1:3DES; 2: SM4
timeout	输入		超时时间(秒), 预留值为 0
mKeyIndex	输入		主密钥索引, 预留值为 0
mKey	输入		主密钥
checkValue	输入		主密钥校验值

出参说明:

返回值类型	说明
-------	----

int	当等于 0 时表示返回正确，其他情况用错误码表示。
-----	---------------------------

7.2. 下载工作密钥

方法名：

```
int downloadWorkKey (int algorithmType, int timeout, int mKeyIndex, int wKeyIndex, String wKey, String checkValue);
```

入参说明：

参数	输入/输出	长度	说明
algorithmType	输入		0: DES; 1: 3DES; 2: SM4
timeout	输入		超时时间(秒)
mKeyIndex	输入		主密钥索引，预留值为 0
wKeyIndex	输入		工作密钥索引，预留值为 0
wKey	输入		工作密钥
checkValue	输入		校验值

出参说明：

返回值类型	说明
int	当等于 0 时表示返回正确，其他情况用错误码表示。

7.3. 柜外清方式下载主密钥

方法名:

JSONObject downloadTPK(int nKeyType, int nTMKIndex, String strKey)

入参说明:

参数	输入/输出	说明
keyType	输入	主密钥类型: 1——DES 单倍长 2 —— DES 双倍长 3—— DES 三倍长 4 —— SM4
TMKIndex	输入	主密钥索引 0-7
mKey	输入	主密钥

出参说明:

返回值类型	说明
JSONObject	{ “code”: 若成功, 该值为 0, 若失败, 该值为对应错误码, // 若成功, 返回加密后的密码 “checkValue”: <校验值> }

7.4. 柜外清方式下载工作密钥

方法名:

public JSONObject downloadTPK(intk keyType, int TMKIndex, int TPKIndex, String wKey);

入参说明:

参数	输入/输出	说明
----	-------	----

keyType	输入	工作密钥类型: 1——DES 单倍长 2 —— DES 双倍长 3—— DES 三倍长 4 —— SM4
TMKIndex	输入	主密钥索引 0-7
TPKIndex	输入	工作密钥索引 0-1 (DES--0, SM4--1)
wKey	输入	工作密钥密文

出参说明:

返回值类型	说明
JSONObject	{ “code”: 若成功, 该值为 0, 若失败, 该值为对应错误码, // 若成功, 返回加密后的密码 “checkValue”: <校验值> }

7.5.读取会话密钥

方法名:

```
public JSONObject getSessionKey(int pubKeyType, int sessionKeyType, String pubKey)
```

入参说明:

参数	输入/输出	说明
pubKeyType	输入	公钥类型 0--RSA, 1--SM2
sessionKeyType	输入	生成的会话密钥类型

		1——DES 单倍长 2 —— DES 双倍长 3—— DES 三倍长 4 —— SM4
pubKey		公钥

出参说明:

返回值类型	说明
JSONObject	<pre>{ "code": 若成功, 该值为 0, 若失败, 该值为对应错误码, // 若成功, 返回加密后的密码 "password": <结果> }</pre>

7.6.读取密码

方法名:

JSONObject getPassword(String account,int timeout, int wKeyIndex, int encryptType, int times,
int minLength, int maxLength);

入参说明:

参数	输入/输出	说明
timeout	输入	超时时间(秒)
acount	输入	账号

wKeyIndex	输入	工作密钥索引 (DES--0, SM4--1)
encryptType	输入	加密类型: 0-不加密, 1-DES 加密, 2-SM4 加密
times	输入	可输入的次数
minLength	输入	密码最小长度
maxLength	输入	密码最大长度

出参说明:

返回值类型	说明
JSONObject	<pre>{ "code": 若成功, 该值为 0, 若失败, 该值为对应错误码, // 若成功, 返回加密后的密码 "password": <结果> }</pre>

7.7.重置密钥

方法名:

```
int resetPassword(int timeout);
```

入参说明:

参数	输入/输出	说明
timeout	输入	超时时间(秒)

出参说明:

返回值类型	说明
int	成功返回 0，失败返回错误码。

7.8. 获取密钥安全节点

方法名:

JSONObject getKeypadID()

返回值类型	说明
UID(String)	序列号
companyID(String)	厂商名称
SN	设备编号
Code(int)	状态码，成功返回 0，失败返回对应错误码

7.9. 明文转密文

方法名:

JSONObject encryptPin(String password, int wKeyIndex, String account);

参数	输入/输出	说明
----	-------	----

password	输入	明文密码
account	输入	账号
wKeyIndex	输入	工作密钥索引

出参说明:

返回值类型	说明
JSONObject	code:成功返回 0, 失败返回错误码 password(String): 密文

7.10. 获取密钥加密方式

方法名:

JSONObject getKeypadUseAlgorithm()

出参说明:

返回值类型	说明
JSONObject	code:成功返回 0, 失败返回错误码 algorithm(int): 加密类型, 0 为 DES, 1 为 SM4

7.11. 取消键盘输入 PS

方法名:

void cancelReadPin()

第八章 电子签名

8.1.启动手写屏

方法名:

```
int startHWScreen(int timeout);
```

入参说明:

参数	输入/输出	说明
timeout	输入	超时时间(秒)

出参说明:

返回值类型	说明
int	成功返回 0，失败返回错误码。

8.2.获取手写屏数据

方法名:

```
JSONObject getHWScreenData(int timeout);
```

入参说明:

参数	输入/输出	长度	说明
timeout	输入		超时时间(秒)

出参说明:

返回值类型	说明
JSONObject	{ “code”: 若成功，该值为 0，若失败，该值为对应错误码，

	<pre>// 若成功，返回储存到本地的签名图片路径 “picture”:<结果> }</pre>
--	---

8.3.关闭手写屏

方法名:

```
int closeHWScreen ();
```

入参说明: 无

出参说明:

返回值类型	说明
int	成功返回 0，失败返回错误码。

第九章 声音和灯控

9.1.发起语音

方法名:

```
int makeVoice(int voiceType);
```

入参说明:

参数	输入/输出	长度	说明
----	-------	----	----

voiceType	输入		声音类型: 0: 请输入密码 1: 请再次输入密码 2: 密码错误, 请重新输入密码 3: 请放置身份证 4: 请放置 IC 卡 5: 请插入 IC 卡 6: 请刷磁条卡 7: 请签字 8: 请录入指纹 9: 请重新录入指纹 a: 操作成功
-----------	----	--	---

出参说明:

返回值类型	说明
int	成功返回 0, 失败返回错误码。

9.2.发起蜂鸣

方法名:

```
int makeBeep(int times,int timeout);
```

入参说明:

参数	输入/输出	长度	说明
times	输入		次数
timeout	输入		蜂鸣时间,100 毫秒的倍数

出参说明:

返回值类型	说明
int	成功返回 0，失败返回错误码。

9.3.LED 灯控制

方法名:

`int ledOnOrOff(int ledNo,int onOrOff,int color);`

入参说明:

参数	输入/输出	长度	说明
ledNo	输入		IC 灯 0; 磁条卡灯 1; 身份证灯 2; 指纹灯 3
onOrOff	输入		亮灭: 亮 1; 灭 2
color	输入		0

第十章 错误码和混淆

10.1.混淆规则

需要各厂商提供自己的 jar 包的不混淆规则