

**MPOS**  
**安卓 SDK 接口说明**  
**V1.0.1**

**福建魔方电子科技有限公司**

## 目 录

1. 接口函数说明 .....	2
1.1. 初始化连接模式 .....	2
1.2. 清理销毁 .....	3
1.3. 连接设备 .....	3
1.4. 断开连接 .....	3
1.5. 查询 MPOS 连接状态 .....	3
1.6. 输入密码 .....	4
1.7. 计算 PIN .....	4
1.8. 计算 MAC .....	5
1.9. 下载 KEK .....	错误!未定义书签。
1.10. 下载主密钥 .....	5
1.11. 下载工作密钥 .....	6
1.12. 执行 EMV 流程 .....	7
1.13. 执行 IC 卡二次授权 .....	8
1.14. 获取 MPOS 信息 .....	9
1.15. 获取随机数 .....	9
1.16. 蜂鸣器 .....	10
1.17. 设置终端时间 .....	10
1.18. 获取终端时间 .....	10
1.19. 复位 MPOS .....	11
1.20. 退出 MPOS 等待 .....	11
2. 附录 .....	11
2.1 附录 A commResult 取值及含义 .....	11
2.2 附录 B IC 卡公钥和 AID 格式说明 .....	11
2.3 附录 C jar 包使用步骤简介 .....	14
2.4 附录 D 磁条卡交易一般步骤 .....	15
2.5 附录 E IC 交易一般步骤 .....	错误!未定义书签。
2.6 附录 F 磁道加密算法说明 .....	错误!未定义书签。
2.7 附录 G 标准流程实现方案示例代码 .....	16

## 1. 接口函数说明

## 1.1. 初始化连接模式

函数原型:

```
void Init( Context context , CONNECTMODE type,int ManufacturerID )
```

说明:

初始化连接模式。

参数说明：

参数名	取值含义
context	当前应用上下文
type	type MPOS 连接模式。取值如下： BLUETOOTH: 蓝牙模式连接 AUDIO: 音频模式连接
ManufacturerID	厂商 ID 默认为 0

## 1.2. 清理销毁

函数原型：

**void Destory()**

说明：

应用退出时需要执行。

## 1.3. 连接设备

函数原型：

**ConnectPosResult connectPos( String addr)**

说明：

连接 MPOS。阻塞直到连接上或者超时。

参数说明：

参数名	取值含义
addr	蓝牙模式：蓝牙地址，XX:XX:XX:XX:XX:XX 格式 音频模式：参数暂时不用

返回值说明：

成员变量	取值含义
bConnected	true 连上，false 未连上

## 1.4. 断开连接

函数原型：

**void disconnectPos()**

说明：

断开 MPOS 连接

参数说明：

参数名	取值含义
-----	------

返回值说明：

无

## 1.5. 查询MPOS连接状态

函数原型：

**boolean posConnected()**

### 说明:

查询 MPOS 是否已连接上。

### 参数说明:

参数名	取值含义
-----	------

### 返回值说明:

若已连接, 返回 true; 否则返回 false

## 1.6. 输入密码

### 函数原型:

```
InputPinResult InputPin(
    byte pinMaxLen,
    byte timeout,
    String sPan)
```

### 说明:

输入密码, 返回加密后的 PINBLOCK

### 参数说明:

参数名	取值含义
pinMaxLen	输入密码的最大长度
timeout	超时时间, s 为单位
sPan	主账号

### 返回值说明:

成员变量	取值含义
commResult	通讯结果。取值参见附录 A
keyType	返回按键类型。取值如下: OK: 用户按确认键 CANCEL: 用户按取消键
pinBlock	加密后的 <u>pinblock</u>
pwdLen	输入密码的长度

## 1.7. 计算PIN

### 函数原型:

```
CalPinResult CalPin(byte[] pindata, String sPan)
```

### 说明:

输入密码, 返回加密后的 PINBLOCK

### 参数说明:

参数名	取值含义
pindata	输入密码的原数据
sPan	主账号

### 返回值说明:

成员变量	取值含义
commResult	通讯结果。取值参见附录 A

pinBlock	加密后的 <u>pinblock</u>
----------	----------------------

### 1.8. 计算MAC

函数原型:

```
CalMacResult CalMac(
    CommEnum.MACALG macAlg,
    byte[] msg,
    int msgLen)
```

说明:

根据保存在 MPOS 的工作密钥, 计算传入的报文的 MAC

参数说明:

参数名	取值含义
macAlg	mac 算法. 取值如下: UCB X99 EBC
msg	待计算的报文缓冲区
msgLen	报文实际长度

返回值说明:

成员变量	取值含义
commResult	通讯结果。取值参见附录 A
macvalue	计算出来的 mac

### 1.9. 下载主密钥

函数原型:

```
LoadMainKeyResult
LoadMainKey(
    CommEnum.MAINKEYENCRYPT encrypt,
    CommEnum.KEYINDEX mainKeyIndex,
    CommEnum.MAINKEYTYPE mainkeytype,
    byte[]mainKeyD1,
    byte[]mainKeyD2,
    byte[]kvc)
```

说明:

下载主密钥到终端

参数说明:

参数名	取值含义
encrypt	加密方式. 取值如下: KEK: 采用 KEK 加密 MAINKEY: 采用原主密钥加密

	KEK_SN: 采用 KEK 和 S N 双重加密
mainKeyIndex	主密钥索引。取值如下: INDEX0~ INDEX9, 分别对应索引 0~索引 9
mainkeytype	主密钥类型。取值如下: SIGNALE: 单倍长 DOUBLE: 双倍长
mainKeyD1	主密钥密文前 8 个字节
mainKeyD2	主密钥密文后 8 个字节。若不存在, 填充全 0
kvc	校验值

#### 返回值说明:

成员变量	取值含义
commResult	通讯结果。取值参见附录 A
loadResult	true 为成功, false 为失败

### 1. 10. 下载工作密钥

#### 函数原型:

LoadWorkKeyResult

```
LoadWorkKey(
    CommEnum.KEYINDEX mainKeyIndex,
    CommEnum.WORKKEYTYPE keyType,
    byte[] keystr,
    int len)
```

#### 说明:

下载工作密钥到终端

#### 参数说明:

参数名	取值含义
mainKeyIndex	主密钥索引。取值如下: INDEX0~ INDEX9, 分别对应索引 0~索引 9
keyType	主密钥类型。取值如下: SIGNALE: 单倍长 数据长度 24=PIK(8)+KVC(4)+MIK(8)+KVC(4) DOUBLE: 双倍长 数据长度 40=PIK(16)+KVC(4)+MIK(16)+KVC(4) DOUBLEMAG: 双倍长带磁道加密 数据长度 60=PIK(16)+KVC(4)+MIK(16)+KVC(4)+TIK(16)+KVC(4)
keystr	工作密钥缓冲区。POS 中心向 MPOS 终端约定的新工作密钥, 长度为 024 或 040 或 060
len	工作密钥实际长度

#### 返回值说明:

成员变量	取值含义
commResult	通讯结果。取值参见附录 A

loadResult	true 为成功, false 为失败
------------	---------------------

## 1. 11. 执行读卡流程

函数原型:

**ReadCardResult ReadCard (ReadCardParam param)**

说明:

下载工作密钥到终端

参数说明 **ReadCardParam:**

参数名	取值含义
setTransName	交易描述
setAmount	交易金额
setTransType	交易类型
setCardmode	用卡方式
setCardTimeout	用卡超时时间
setEmvexectype	EMV 执行方式
setForceonline	是否强制联机
setOtherAmount	设置其他金额
setPinInput	是否启用 pin 输入
setPinMaxLen	输入 PIN 最大长度
setPinTimeout	输入 PIN 超时时间
setTags	IC 卡数据 TAG 列表
setAllowfallback	允许 fallback
setAuthAmount	Ic 卡授权金额
setEcashpermit	电子现金支持

返回值说明:

成员变量	取值含义
commResult	通讯结果。取值参见附录 A
cardType	用卡类型 * 0x00 用户取消操作 * 0x01 刷卡结束 * 0x02 IC 卡已插入 * 0x04 非接卡
expData	有效期
fallback	是否 Fallback
icData	IC 卡数据
pan	卡号
pansn	卡片序列号
pinblock	密码密文
pinLen	密码长度
plaintlen	加密前长度

randomdata	随机数
serviceCode	服务代码
track2	二磁道数据
track2Len	二磁道长度
track3	三磁道数据
track3Len	二磁道长度

### 示例代码:

```

ReadCardParam param = new ReadCardParam();
//设置金额 单位分
param.setAmount(100);
//设置交易类型
param.setTransType( TRANSTYPE.FUNC_SALE);
//启用输入 PIN
param.setPinInput(true);

//执行 MPOS 读卡流程
ReadCardResult ret = Controler.ReadCard(param);

//操作结果
StringBuilder sb = new StringBuilder();
sb.append("通讯结果:" + ret.commResult.toDisplayName() + "\n");
sb.append("用卡类型:" + ret.getCartTypeName() + "\n");
sb.append("是否 FallBack:" + (ret.fallback ? "是":"否") + "\n");
sb.append("主账号:" + ret.pan + "\n");
sb.append("卡有效期:" + ret.expData + "\n");
sb.append("服务代码:" + ret.serviceCode + "\n");
sb.append("二磁道长度:" + ret.track2Len + "\n");
sb.append("磁道二信息:" + ret.track2 + "\n");
sb.append("三磁道长度:" + ret.track3Len + "\n");
sb.append("磁道三信息:" + ret.track3 + "\n");
sb.append("数据随机数:" + ret.randomdata + "\n");
sb.append("卡片序列号:" + ret.pansn + "\n");
sb.append("IC 卡数据:" + ret.icData + "\n");
sb.append("PIN 长度:" + ret.pinLen + "\n");
sb.append("PIN 密文:" + ret.pinblock + "\n");

```

## 1. 12. 执行IC卡二次授权

### 函数原型:

**EmvDealOnlineRspResult**



```
EmvDealOnlineRsp(  
    boolean bOnlineSucc,  
    byte[] tlvData,  
    int tlvDataLen)
```

说明:

执行二次授权，即 IC 卡联机之后的后续处理

参数说明:

参数名	取值含义
bOnlineSucc	是否联机成功
tlvData	8583 应答报文的 55 域值
tlvDataLen	tlvData 实际数据长度

返回值说明:

成员变量	取值含义
commResult	通讯结果。取值参见附录 A
authResult	二次授权结果。取值如下： SUCC: 交易接受 REJECT: 二次授权交易拒绝 FAIL: 交易失败

### 1. 13. 获取MPOS信息

函数原型:

```
ReadPosInfoResult ReadPosInfo()
```

说明:

获取 MPOS 机身号等

参数说明:

参数名	取值含义
-----	------

返回值说明:

成员变量	取值含义
commResult	通讯结果。取值参见附录 A
sn	机身号
initStatus	MPOS 状态。取值如下： WORKKEYLOAD: 工作密钥已灌装 MAINKEYLOAD: 主密钥已灌装 KEKLOAD: KEK 已修改 DEFAULT: 默认初始状态
customInfo	厂商自定义信息，用于存储特定数据
posVer	MPOS 终端版本号

### 1. 14. 获取随机数

函数原型:

```
GetRandomResult GetRandomNum()
```

说明:

获取随机数

**参数说明:**

参数名	取值含义
-----	------

**返回值说明:**

成员变量	取值含义
commResult	通讯结果。取值参见附录 A
randomNum	MPOS 生成的随机数

**1. 15. 蜂鸣器****函数原型:**

```
BeepResult Beep(  
    int beepCount,  
    int freq,  
    int time,  
    int duration)
```

**说明:**

触发 MPOS 蜂鸣器

**参数说明:**

参数名	取值含义
beepCount	蜂鸣次数
freq	蜂鸣频率, 单位 hz
time	每次蜂鸣时间, 单位 ms
duration	次蜂鸣时间间隔, 单位 ms

**返回值说明:**

成员变量	取值含义
commResult	通讯结果。取值参见附录 A

**1. 16. 设置终端时间****函数原型:**

```
SetDateTimeResult SetDatetime(  
    String datetime)
```

**说明:**

设置 MPOS 的时间。需要确保 MPOS 的时间为当前时间

**参数说明:**

参数名	取值含义
datetime	时间, YYYYMMDDHHMMSS 格式

**返回值说明:**

成员变量	取值含义
commResult	通讯结果。取值参见附录 A

**1. 17. 获取终端时间****函数原型:**

```
GetDateTimeResult GetDatetime()
```

**说明:**

获取 MPOS 的当前时间。

### 参数说明:

参数名	取值含义
-----	------

### 返回值说明:

成员变量	取值含义
commResult	通讯结果。取值参见附录 A
datetime	时间, YYYYMMDDHHMMSS 格式

## 1. 18. 复位MPOS

### 函数原型:

**ResetPosResult ResetPos()**

### 说明:

复位 MPOS 的状态。建议每笔交易前都复位一下 MPOS

### 参数说明:

参数名	取值含义
-----	------

### 返回值说明:

成员变量	取值含义
commResult	通讯结果。取值参见附录 A

## 1. 19. 退出MPOS等待

### 函数原型:

**Void CancelComm()**

### 说明:

在等待 MPOS 数据的时候, 允许提前退出, 比如等待刷卡, 等待输密码。注意一些特殊情况不应允许取消, 比如正在进行二次授权的时候

### 参数说明:

参数名	取值含义
-----	------

### 返回值说明:

无。

## 2. 附录

### 2.1 附录 A commResult 取值及含义

NOERROR	成功
CANCEL	用户取消
CMDNOTSUPPORT	命令功能不支持
PARAMSERR	参数错误
DATALENERR	数据长度错误
TIMEOUT	APK 等待 MPOS 返回超时
CONNFAIL	蓝牙连接错误
CONNDISCONNECT	蓝牙未连接
OTHERERR	其他错误

### 2.2 附录 B IC 卡公钥和 AID 格式说明

当读取指定公钥时, 返回内容参考如下标签列表内容

认证中心公钥类参数采用TLV（tag+length+value）格式表示，具体取值及含义为：

表1 认证中心公钥参数

参数名称	参数属性	参数长度 (byte)	参数 tag 值	参数含义	参数下载时间	参数适用场合
RID	b	5	9F06	与认证中心公钥索引一起标识认证中心的公钥	安装或调整时	交易应用
认证中心公钥索引	b	1	9F22	与 RID 一起标识认证中心的公钥	安装或调整时	交易应用
认证中心公钥有效期	n8	8	DF05	认证中心规定的有效期限	安装或调整时	交易应用
认证中心公钥哈希算法标识	b	1	DF06	标识用于在数字签名方案中产生哈希结果的哈希算法	安装或调整时	交易应用
认证中心公钥算法标识	b	1	DF07	标识使用在认证中心公钥上的数字签名算法	安装或调整时	交易应用
认证中心公钥模	b	变长，最大为248	DF02	公钥模值	安装或调整时	交易应用
认证中心公钥指数	b	1 或 3	DF04	公钥指数	安装或调整时	交易应用
认证中心公钥校验值	b	变长	DF03	验证认证中心公钥用	安装或调整时	交易应用
注： 认证中心公钥校验值的计算内容为RID+认证中心公钥索引+认证中心公钥模+认证中心公钥指数；认证中心公钥校验值的计算方法为SHA-1。						

当读取指定 AID 时，返回内容参考如下标签列表内容

表2 IC 卡 AID 参数列表

参数名称	参数属性	参数长度 (byte)	参数 tag 值	参数含义	参数下载时间	参数适用场合
AID	b	5—16	9F06	终端支持的借/贷记应用列表，如 ISO/IEC 7816-5 所述，指明应用	安装或调整时	交易应用
应用选择指示符 (ASI)	b	1	DF01	指示应用选择时终端上的 AID 与卡片中的 AID 是完全匹配（长度和内容都必须一样），还是部分匹配（卡片 AID 的前面部分与终端 AID 相同，长度可以更长）。终端支持的应用列表中的每个 AID 仅有一个应用选择指示符。	安装或调整时	交易应用

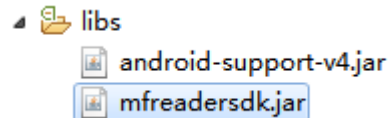
参数名称	参数属性	参数长度 (byte)	参数 tag 值	参数含义	参数下载时间	参数适用场合
应用版本号	b	2	9F09	支付系统给应用分配的版本号	安装或调整时	交易应用
TAC—缺省	b	5	DF11	标识如果交易可以联机完成但终端没有联机交易能力时，拒绝交易的收单行条件	安装或调整时	交易应用
TAC—联机	b	5	DF12	标识联机交易的收单行条件	安装或调整时	交易应用
TAC—拒绝	b	5	DF13	标识不作联机尝试即拒绝交易的收单行条件	安装或调整时	交易应用
终端最低限额	b	4	9F1B	IC 卡消费时终端允许的最低脱机限额	安装或调整时	交易应用
偏置随机选择的 阈值	b	4	DF15	在终端风险管理中用于随机交易选择的值	安装或调整时	交易应用
偏置随机选择的 最大目标百分数	cn(包含 两位有效数字)	1	DF16	用于偏置随机选择的最大目标百分数	安装或调整时	交易应用
随机选择的目标 百分数	cn(包含 两位有效数字)	1	DF17	用于随机选择的目标百分数	安装或调整时	交易应用
缺省 DDOL	b	变长	DF14	卡片中无 DDOL 时用于构造内部认证命令的 DDOL	安装或调整时	交易应用
终端联机 PIN 支持能力	b	1	DF18	指示终端在每个 AID 的要求下是否支持联机 PIN 的输入。	安装或调整时	交易应用  当值为 00000001 时表示支持联机 PIN。当值为 00000000 时表示不支持联机 PIN。
终端电子现金交易限额	cn	6	9F7B	终端使用此数据元（如果存在的话）判断一个交易的处理方式，当授权金额小于该限额时允许电子现金交易，否则设置终端行为代码并根据判断确认交易方式（小额支付参数）。	安装或调整时	交易应用

参数名称	参数属性	参数长度 (byte)	参数 tag 值	参数含义	参数下载时间	参数适用场合
非接触读写器脱机最低限额	cn	6	DF19	在 AID 联合中, 用来指示读写器中非接触交易的最低限额	安装或调整时	交易应用
非接触读写器交易限额	cn	6	DF20	如果非接触交易的金额大于或等于此数值, 则交易终止。允许在其他界面尝试此交易	安装或调整时	交易应用
读写器持卡人验证方法 (CVM) 所需限制	cn	6	DF21	如果非接触交易超过此值, 读写器要求一个持卡人验证方法 (CVM)。	安装或调整时	交易应用

### 2.3 附录 C jar 包使用步骤简介

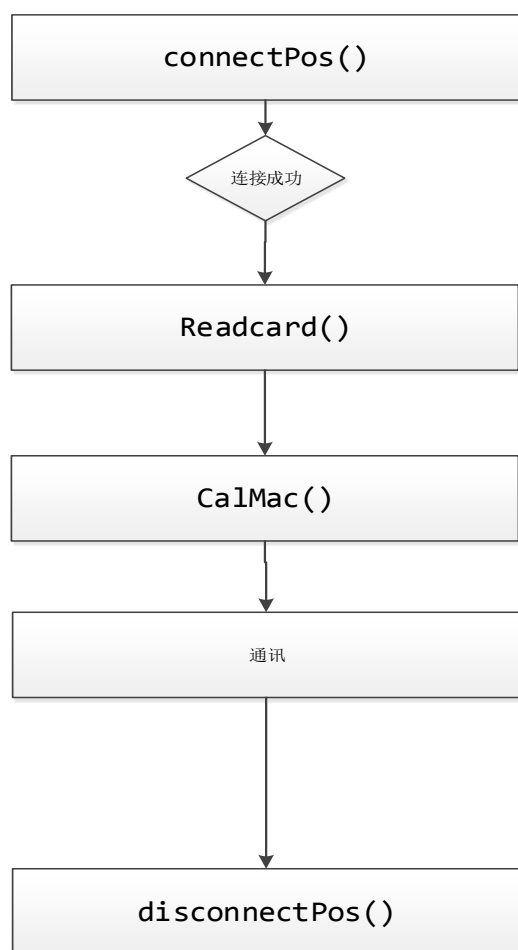
1. 修改 AndroidManifest, 开启 APK 的蓝牙权限:

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
```



2. 添加 mfreadersdk.jar 到项目工程的 libs 文件夹
3. 接口调用:  
所有接口放在 **Controler** 类中, 均为静态函数, 可直接调用:  
**Controler.disconnectPos();**

## 2.4 附录 D 交易一般步骤



## 2.5 附录 E 标准流程实现方案示例代码

### 1. 初始化, 在 Activity 上使用 `Controler.Init` 初始化

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Controler.Init(this, CONNECTMODE.BLUETOOTH ,0);
}

@Override
protected void onDestroy() {
    Controler.Destroy();
    super.onDestroy();
}
```

### 2. 初始化完成后, 启动一个辅助线程, 执行连接设备, 刷卡, 等阻塞式操作

```
new Thread(new Runnable() {
    @Override
    public void run() {
        Controler.connectPos("8C:DE:52:9B:F5:97");
        if( Controler.posConnected() ) {

ReadCardParam param = new ReadCardParam();
//设置金额 单位分
param.setAmount(100);
//设置交易类型
param.setTransType( TRANSTYPE.FUNC_SALE);
//启用输入PIN
param.setPinInput(true);

//执行MPOS读卡流程
ReadCardResult ret = Controler.ReadCard(param);

//操作结果
StringBuilder sb = new StringBuilder();
sb.append( "通讯结果:" + ret.commResult.toDisplayName() + "\n" );
sb.append( "用卡类型:" + ret.getCartTypeName() + "\n" );
sb.append( "是否FallBack:" + (ret.fallback ? "是":"否") + "\n" );
sb.append( "主账号:" + ret.pan + "\n" );
sb.append( "卡有效期:" + ret.expData + "\n" );
sb.append( "服务代码:" + ret.serviceCode + "\n" );
sb.append( "二磁道长度:" + ret.track2Len + "\n" );
sb.append( "磁道二信息:" + ret.track2 + "\n" );
sb.append( "三磁道长度:" + ret.track3Len + "\n" );
```



```
sb.append( "磁道三信息:" + ret.track3 + "\n" );
sb.append( "数据随机数:" + ret.randomdata + "\n" );
sb.append( "卡片序列号:" + ret.pansn + "\n" );
sb.append( "IC卡数据:" + ret.icData + "\n" );
sb.append( "PIN长度:" + ret.pinLen + "\n" );
sb.append( "PIN密文:" + ret.pinblock + "\n" );

    }
    Controler.disconnectPos();
}
}).start();
```

3. 如需要取消操作，调用 **Controler.CancelComm()**:

```
@Override
public void onBackPressed() {
    Controler.CancelComm();
}
```