

安卓开发包使用说明书

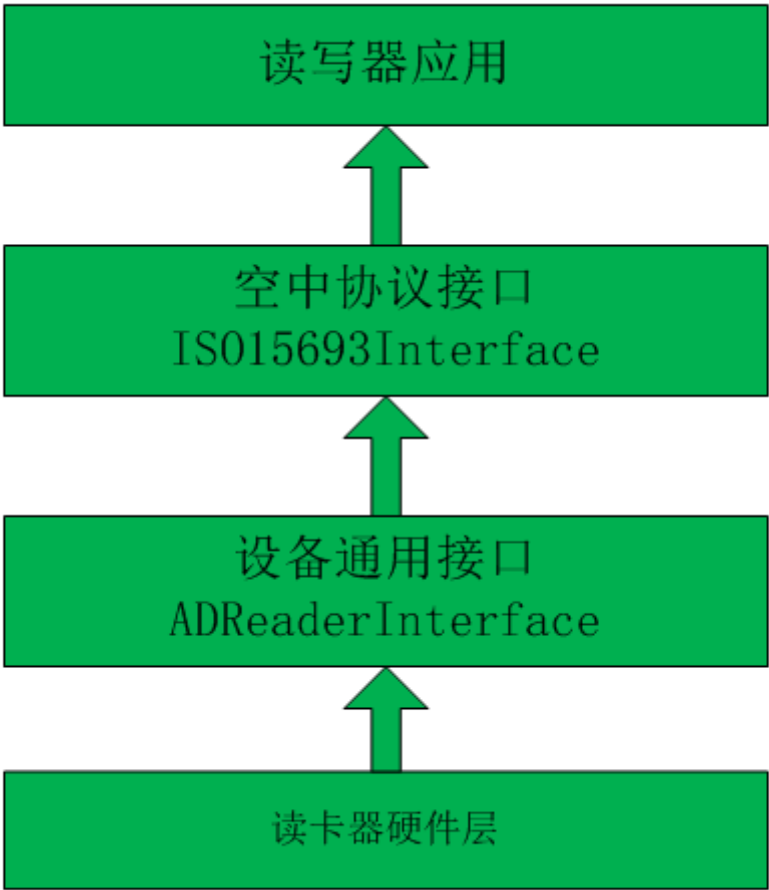
目录

1. 使用指南.....	4
1.1 总体架构.....	4
1.2 调用流程.....	4
1.3 读卡器功能开发流程.....	5
1.3.1 电子标签盘点.....	5
1.3.2 ISO15693 标签内容读写.....	6
1.3.3 获取 RPAN 设备扫描记录.....	7
2. 接口说明.....	8
2.1 读写器通用类——ADReaderInterface.....	8
2.1.1 GetPairBluetooth.....	8
2.1.2 GetSerialPortPath.....	8
2.1.3 EnumerateUsb.....	9
2.1.4 GetUsbDescription.....	9
2.1.5 HasUsbPermission.....	9
2.1.6 RequestUsbPermission.....	10
2.1.7 RDR_Open.....	10
2.1.8 RDR_Close.....	11
2.1.10 RDR_OpenRFTransmitter.....	12
2.1.11 RDR_CloseRFTransmitter.....	13
2.1.12 RDR_GetRFPower.....	13
2.1.13 RDR_SetRFPower.....	14
2.1.14 RDR_LoadFactoryDefault.....	14
2.1.15 RDR_SetCommImmeTimeout.....	15
2.1.16 RDR_ResetCommImmeTimeout.....	15
2.1.17 RDR_CreateInvenParamSpecList.....	16
2.1.18 RDR_GetTagReportCount.....	16
2.1.19 RDR_TagInventory.....	17
2.1.20 RDR_GetTagDataReport.....	18
2.1.21 RPAN_ClearScanRecord.....	18
2.1.23 RPAN_GetRecord.....	19
2.1.24 RPAN_ParseRecord.....	20
2.1.25 RDR_GetReaderLastReturnError.....	20
2.1.26 RDR_GetOverflowTime.....	21
2.1.27 RDR_SetOverflowTime.....	21
2.1.28 RDR_GetAntennaInterfaceCount.....	22
2.1.29 RDR_SetAcessAntenna.....	22
2.2 ISO15693 空中协议操作类——ISO15693Interface.....	23
2.2.1 ISO15693_CreateInvenParam.....	23
2.2.2 ISO15693_ParseTagDataReport.....	24
2.2.3 ISO15693_Connect.....	25
2.2.4 ISO15693_Disconnect.....	25
2.2.5 ISO15693_ReadMultiBlocks.....	26

2.2.6	IS015693_WriteMultipleBlocks.....	27
2.2.7	IS015693_LockMultipleBlocks.....	27
2.2.8	IS015693_WriteDSFID.....	28
2.2.9	IS015693_LockDSFID.....	28
2.2.10	IS015693_WriteAFI.....	28
2.2.11	IS015693_LockAFI.....	29
2.2.12	IS015693_GetSystemInfo.....	29
2.2.13	NXPICODESLI_EableEAS.....	30
2.2.13	EASNXPICODESLI_DisableEAS.....	30
2.2.14	EASNXPICODESLI_EASCheck.....	31
2.2.15	NXPICODESLI_LockEAS.....	31
2.2.16	NXPICODESLI_GetRandomAndSetPassword.....	32
2.2.17	NXPICODESLI_WritePassword.....	33
2.2.18	NXPICODESLI_Enable64BitPwd.....	34
2.2.19	NXPICODESLI_LockPassword.....	34
2.2.20	NXPICODESLI_PasswordProtect.....	35
2.2.22	NXPICODESLI_LockPageProtection.....	37
2.3	IS01443A 空中协议操作类——IS01443AInterface.....	38
2.3.1	SpecAIPInvenParamIS01443A.....	38
2.3.2	IS01443A_ParseTagDataReport.....	38
2.3.3	MFCL_Connect.....	39
2.3.4	MFCL_Authenticate.....	40
2.3.5	MFCL_ReadBlock.....	40
2.3.6	MFCL_WriteBlock.....	41
2.3.7	MFCL_FormatValueBlock.....	41
2.3.8	MFCL_Restore.....	42
2.3.8	MFCL_Increment.....	42
2.3.8	MFCL_Decrement.....	43
2.3.9	ULTRALIGHT_Connect.....	43
2.3.10	ULTRALIGHT_ReadMultiplePages.....	44
2.3.11	ULTRALIGHT_WriteMultiplePages.....	44
2.3.12	IS01443A_Disconnect.....	45
3.	连接串详细说明.....	46
3.1	概述.....	46
3.2	串口连接串说明.....	46
3.3	蓝牙连接串说明.....	48
3.4	网络连接串说明.....	49
3.5	USB 连接串说明.....	50
4.	错误代码表.....	51

1. 使用指南

1.1 总体架构

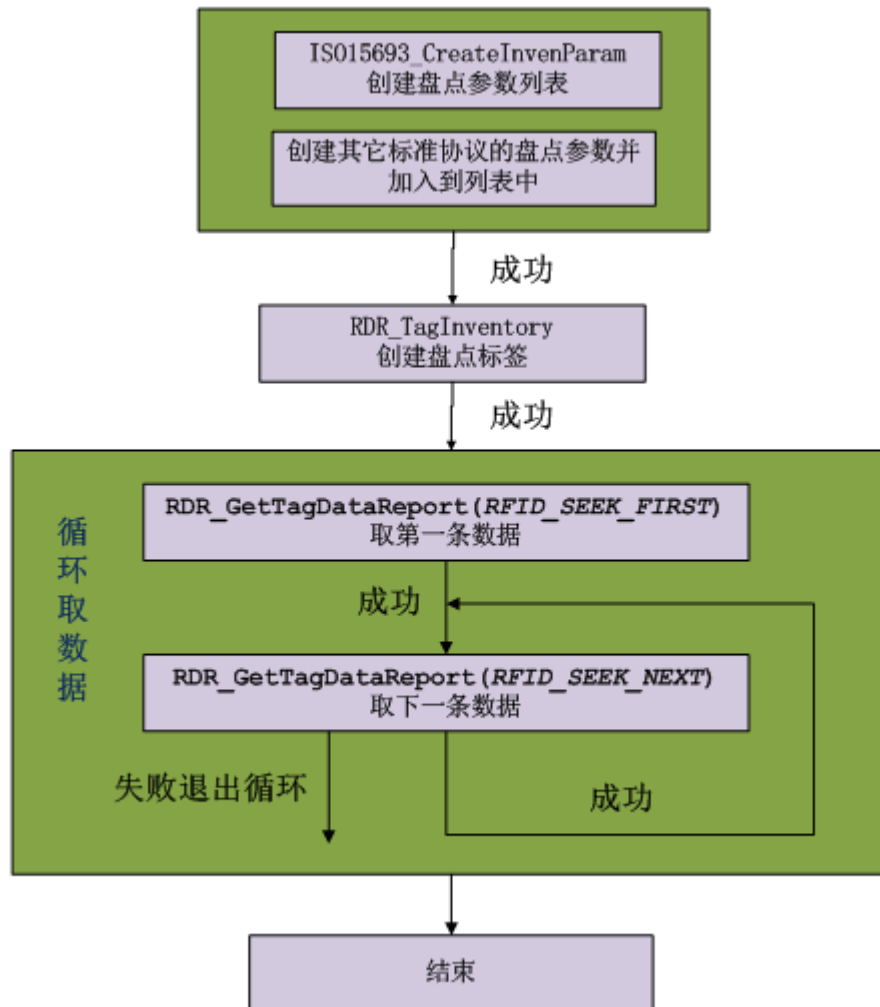


1.2 调用流程

第一步	把开发包里面的 AnReaderLib.jar 文件、armeabi 文件夹和 armeabi-v7a 文件夹复制到安卓工程目录下的 libs 文件夹内。
第二步	调用封装类 ADReaderInterface 和 IS015693Interface 相关接口，操作读写器设备。

1.3 读卡器功能开发流程

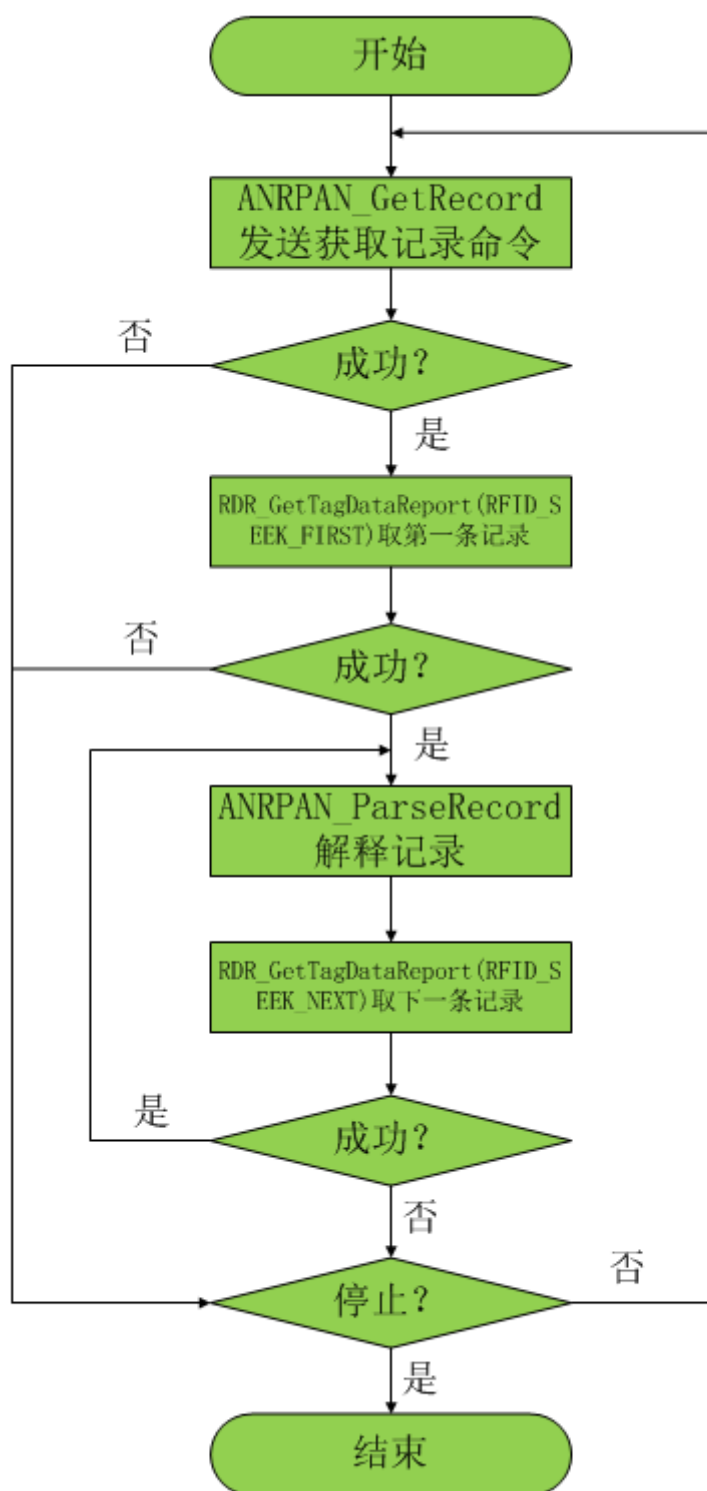
1.3.1 电子标签盘点



1.3.2 ISO15693 标签内容读写



1.3.3 获取 RPAN 设备扫描记录



2. 接口说明

2.1 读写器通用类——ADReader Interface

2.1.1 GetPairBluetooth

定义	<code>Public static ArrayList<BluetoothCfg> GetPairBluetooth()</code>
说明	获取已经配对的蓝牙设备列表
参数	无
返回	如果成功，蓝牙设备列表；失败时返回 null
备注	对 BluetoothCfg 实例，可以通过 String GetName() 和 String GetAddr() 方法获取蓝牙设备的名称和地址。

2.1.2 GetSerialPortPath

定义	<code>public static String[] GetSerialPortPath()</code>
说明	枚举设备上所有的串口
参数	无
返回	串口路径列表
备注	

2.1.3 EnumerateUsb

定义	public static int EnumerateUsb(Context context)		
说明	枚举 USB 设备		
参数	名称	类型	备注
	context	Context	上下文实例
返回	USB 设备数量		
备注			

2.1.4 GetUsbDescription

定义	public static String GetUsbDescription(int idx)		
说明	获取 USB 接口的描述		
参数	名称	类型	备注
	idx	int	枚举到的 USB 设备索引号， 从 0 开始
返回	USB 接口描述		
备注			

2.1.5 HasUsbPermission

定义	public static boolean HasUsbPermission(String usbDes)		
说明	判断是否对 USB 设备有操作权限		
参数	名称	类型	备注
	usbDes	String	USB 设备描述。当为空时，

			表示第一个枚举到的 USB 设备。
返回	true:有操作权限； false:无操作权限。		
备注			

2.1.6 RequestUsbPermission

定义	public static boolean RequestUsbPermission(String usbDes)		
说明	请求 USB 设备操作权限		
参数	名称	类型	备注
	usbDes	String	USB 设备描述。当为空时，表示第一个枚举到的 USB 设备。
返回	true:成功； false:失败。		
备注			

2.1.7 RDR_Open

定义	public int RDR_Open(String conStr)		
说明	打开读写器设备		
参数	名称	类型	备注
	conStr	String	见“连接串详细说明”部分
返回	见错误代码表		

备注	<p>1. 通过蓝牙方式打开设备时，要在 AndroidManifest.xml 文件加入蓝牙操作权限：</p> <pre><uses-permission android:name="android.permission.BLUETOOTH_ADMIN" /> <uses-permission android:name="android.permission.BLUETOOTH" /></pre> <p>2. 通过网络方式打开设备时，要在 AndroidManifest.xml 文件加入网络操作权限：</p> <pre><uses-permission android:name="android.permission.ACCESS_WIFI_STATE" /> <uses-permission android:name="android.permission.INTERNET" /></pre> <p>3. 通过 USB 方式打开设备时，首先要枚举所接的 USB 设备。</p> <p>4. 采用 USB 连接方式时，只支持 Android 5.1.1 以上版本。</p>
----	---

2.1.8 RDR_Close

定义	public int RDR_Close()
说明	关闭读写器设备
参数	无
返回	见错误代码表
备注	无

2.1.9 RDR_GetReaderInfor

定义	public int RDR_GetReaderInfor(StringBuffer buffer)		
说明	获取读卡器的相关信息		
参数	名称	类型	备注
	buffer	StringBuffer	用于保存返回的设备信息。输出格式： “设备固件版本;设备型号;设备系列号”，三项信息用” ;” 分隔。
返回	见错误代码表		
备注			

2.1.10 RDR_OpenRFTransmitter

定义	public int RDR_OpenRFTransmitter()
说明	打开射频
参数	无
返回	见错误代码表
备注	

2. 1. 11 RDR_CloseRFTransmitter

定义	public int RDR_CloseRFTransmitter()
说明	关闭射频
参数	无
返回	
备注	

2. 1. 12 RDR_GetRFPower

定义	public byte RDR_GetRFPower(Byte mTime)		
说明	获取射频功率		
参数	名称	类型	备注
	mTime	Byte	功率值。1-6 分别表示:0.25w、0.50w、0.75w、1.00w、1.25w、1.50w。返回其它时，表示失败（见错误代码表）。
返回	见错误代码表		
备注			

2.1.13 RDR_SetRFPower

定义	public int RDR_SetRFPower(byte index)		
说明	设置射频功率		
参数	名称	类型	备注
	index	byte	1-6，分别表示:0.25w、0.50w、0.75w、1.00w、1.25w、1.50w。其它值无效。
返回	见错误代码表		
备注			

2.1.14 RDR_LoadFactoryDefault

定义	public int RDR_LoadFactoryDefault()
说明	设备恢复出厂设置
参数	无
返回	见错误代码表
备注	

2. 1. 15 RDR_SetCommuImmeTimeout

定义	public int RDR_SetCommuImmeTimeout()
说明	使与读卡器通信的 API 马上超时退出
参数	无
返回	见错误代码表
备注	

2. 1. 16 RDR_ResetCommuImmeTimeout

定义	public int RDR_ResetCommuImmeTimeout()
说明	如果应用程序为使线程马上退出调用了 API RDR_SetCommuImmeTimeout，那么在线程退出后马上调用该 API，否则下一次命令操作有可能会产生-5的错误。
参数	无
返回	见错误代码表
备注	

2. 1. 17 RDR_CreateInvenParamSpecList

定义	Public static Object RDR_CreateInvenParamSpecList()
说明	创建空中接口协议参数列表数据节点
参数	无
返回	如果成功，则返回数据节点句柄。
备注	

2. 1. 18 RDR_GetTagReportCount

定义	public int RDR_GetTagReportCount()
说明	获取盘点到的标签总数
参数	无
返回	盘点到的标签数目
备注	

2. 1. 19 RDR_TagInventory

定义	public int RDR_TagInventory(byte AIType,byte AntennaIDs[],int mTimeout,Object InvenParamSpecList)		
说明	寻找标签，该函数支持多天线接口和多空中接口协议。找到标签读取到的信息以数据节点的形式保存起来，inventory 完成后，应用程序通过 RDR_GetTagDataReport 获得所有标签的信息。		
参数	名称	类型	备注
	AIType	byte	查询类型:1:新查询。在 Inventory 之前使所有标签进入 Ready 状态；2:继续查询。处于 Quiet 状态的标签不会被读到；
	AntennaIDs	Byte[]	天线列表。当为 null 时，表示用默认天线。
	mTimeout	int	超时时间(ms)。小于或等于 0 时，为设备默认时间。
	InvenParamSpecList	Object	空中协议参数表。由 RDR_CreateInvenParamSpecList 创建。
返回	见错误代码表		

2. 1. 20 RDR_GetTagDataReport

定义	public Object RDR_GetTagDataReport(byte seek)		
说明	获取缓冲区的标签记录。		
参数	名称	类型	备注
	seek	byte	记录位置 1:第一条记录; 2:下一条记录 3:最后一条记录
返回	成功时，返回当前记录内存。失败时，返回 null		
备注			

2. 1. 21 RPAN_ClearScanRecord

定义	public int RPAN_ClearScanRecord()
说明	清空 RPAN 设备的记录缓冲区数据
参数	无
返回	见错误代码表
备注	只支持 RPAN 设备

2. 1. 22 RPAN_SetTime

定义	public int RPAN_SetTime(int year, int month, int day, int hour, int min,int sec)		
说明	设置 RPAN 设备的显示时间		
参数	名称	类型	备注
	year	int	年份
	month	int	月份
	day	int	日期
	hour	int	时
	min	int	分
	sec	int	秒
返回	见错误代码表		
备注			

2. 1. 23 RPAN_GetRecord

定义	public int RPAN_GetRecord(byte flg)		
说明	获取设备的扫描记录		
参数	名称	类型	备注
	flg	int	标志位。当上一次采集失败时，该值为 0x00. 否则，该值为 0x01
返回	见错误代码表		
备注			

2. 1. 24 RPAN_ParseRecord

定义	public byte[] RPAN_ParseRecord(Object hReport)		
说明	解释扫描记录		
参数	名称	类型	备注
	hReport	Object	记录内存。由 RDR_GetTagDataReport 获得
返回	成功时，返回扫描到的数据。失败时，返回 null		
备注			

2. 1. 25 RDR_GetReaderLastReturnError

定义	public int RDR_GetReaderLastReturnError()		
说明	获取最后一次寄存器返回的错误信息		
参数	名称	类型	备注
返回	最后一次寄存器返回的错误信息		
备注			

2.1.26 RDR_GetOverflowTime

定义	public int RDR_GetOverflowTime(Integer mTime)		
说明	获取盘点标签时的溢出时间		
参数	名称	类型	备注
	mTime	Integer	保存溢出时间的内存。单位：100ms
返回	见错误代码表		
备注			

2.1.27 RDR_SetOverflowTime

定义	public int RDR_GetOverflowTime(int mTime)		
说明	设备溢出时间		
参数	名称	类型	备注
	mTime	mTime	溢出时间，单位：100ms
返回	见错误代码表		
备注			

2. 1. 28 RDR_GetAntennaInterfaceCount

定义	public int RDR_GetAntennaInterfaceCount()		
说明	获取设备的天线数量		
参数	名称	类型	备注
返回	对应设备的天线数量		
备注			

2. 1. 29 RDR_SetAccessAntenna

定义	public int RDR_SetAccessAntenna(byte AntennaID)		
说明	对于支持多天线的读卡器，如果要读写标签时，需要通过该 API 选定标签所在的天线感应区。		
参数	名称	类型	备注
	AntennaID	byte	天线编号
返回	见错误代码表		
备注			

2.2 IS015693 空中协议操作类——IS015693Interface

2.2.1 IS015693_CreateInvenParam

定义	public static Object IS015693_CreateInvenParam(Object hInvenParamSpecList, byte AntennaID, Boolean en_afi,byte afi, byte slot_type)		
说明	创建 IS015693_Inventory 参数数据节点，用于 RDR_TagInventory 函数		
参数	名称	类型	备注
	hInvenParam SpecList	Object	Inventory 参数内存，由 RDR_CreateInvenParamSp ecList 创建。
	AntennaID	byte	天线 ID 号，0 表示所有的 天 线 接 口 都 要 读 取 IS015693 标准的标签，其 他值为对应的天线口。
	en_afi	Boolean	是否匹配 AFI
	afi	byte	匹配的 AFI 值
	slot_type	byte	时隙类型。0：读卡器默认 时隙;1：1 个时隙;16：16 个时隙
返回	成功时，节点内存。失败时，返回 null		
备注			

2.2.2 IS015693_ParseTagDataReport

定义	<pre>public static int IS015693_ParseTagDataReport(Object hTagReport, IS015693Tag tagData)</pre>		
说明	解析 IS015693 Inventory 的标签数据信息。该数据节点从 RDR_GetTagDataReport 获得,用于 RDR_TagInventory 函数。		
参数	名称	类型	备注
	hTagReport	Object	标签数据信息的内存
	tagData	IS015693Tag	保存标签信息的内存
返回	见错误代码表		
备注	<pre>IS015693Tag 定义如下: public class IS015693Tag { public long aip_id = 0;//协议类型 public long tag_id = 0;//标签类型 public long ant_id = 0;//天线号 public byte dsfid = 0;//DSFID 值 public byte uid[] = new byte[8];//UID 号 }</pre>		

2.2.3 IS015693_Connect

定义	public int IS015693_Connect(ADReaderInterface hr, long tagType,byte address_mode, byte uid[])		
说明	连接 IS015693 标签		
参数	名称	类型	备注
	hr	ADReaderInterface	通用读卡器操作实例
	tagType	long	标签型号 ID 号。1:NXP ICODE SLI 标签
	address_mode	byte	地址模式。0:无地址模式, 不需要传入 uid 号;1:有地址模式, 需要传入 uid 号。
	uid	byte[]	标签的 UID 号
返回	见错误代码表		
备注			

2.2.4 IS015693_Disconnect

定义	public int IS015693_Disconnect()		
说明	断开标签连接		
参数	名称	类型	备注
返回	见错误代码表.		
备注			

2. 2. 5 ISO15693_ReadMultiBlocks

定义	public int ISO15693_ReadMultiBlocks(boolean readSecSta, int blkAddr, int numOfBlksToRead, Integer numOfBlksRead, byte bufBlocks[], Long bytesBlkDatRead)		
说明	读数据块		
参数	名称	类型	备注
	readSecSta	boolean	是否读安全状态位
	blkAddr	int	数据块地址
	numOfBlksToRead	int	数据块数量
	numOfBlksRead	Integer	保存读取成功的数据块数量
	bufBlocks	byte[]	保存读取到的数据
	bytesBlkDatRead	Long	保存读取到的数据长度
返回	见错误代码表		
备注			

2.2.6 IS015693_WriteMultipleBlocks

定义	public int IS015693_WriteMultipleBlocks(int blkAddr, int numOfBlks, byte newBlksData[])		
说明	写数据块		
参数	名称	类型	备注
	blkAddr	int	数据块地址
	numOfBlksToRead	int	数据块数量
	newBlksData	byte[]	需要写入的数据
返回	见错误代码表		
备注			

2.2.7 IS015693_LockMultipleBlocks

定义	public int IS015693_LockMultipleBlocks(int blkAddr, int numOfBlks)		
说明	锁数据块		
参数	名称	类型	备注
	blkAddr	int	数据块地址
	numOfBlks	int	数据块数量
返回	见错误代码表		
备注			

2.2.8 IS015693_WriteDSFID

定义	public int IS015693_WriteDSFID(byte dsfid)		
说明	写 DSFID 值		
参数	名称	类型	备注
	dsfid	byte	DSFID 值
返回	见错误代码表		
备注			

2.2.9 IS015693_LockDSFID

定义	public int IS015693_LockDSFID()		
说明	锁定 DSFID 值		
参数	名称	类型	备注
返回	见错误代码表		
备注			

2.2.10 IS015693_WriteAFI

定义	public int IS015693_WriteAFI(byte afi)		
说明	写 AFI 值		
参数	名称	类型	备注
	afi	byte	AFI 值
返回	见错误代码表		
备注			

2.2.11 IS015693_LockAFI

定义	public int IS015693_LockAFI()		
说明	锁定 AFI 值		
参数	名称	类型	备注
返回	见错误代码表		
备注			

2.2.12 IS015693_GetSystemInfo

定义	public int IS015693_GetSystemInfo(byte uid[], Byte dsfid,Byte afi, Long blkSize,Long numOfBlokS, Byte icRef)		
说明	获取标签信息		
参数	名称	类型	备注
	uid[]	Byte[]	返回的标签 uid 号
	dsfid	Byte	保存返回的 DSFID 值
	afi	Byte	保存返回的 AFI 值
	blkSize	Long	保存返回的块容量
	numOfBlokS	Long	保存返回的块数量
	icRef	Byte	保存返回的 icRef 值
返回	见错误代码表		
备注			

2. 2. 13 NXPICODESLI_EableEAS

定义	public int NXPICODESLI_EableEAS()		
说明	使能 EAS		
参数	名称	类型	备注
返回	见错误代码表		
备注			

2. 2. 13 EASNXPICODESLI_DisableEAS

定义	public int NXPICODESLI_DisableEAS		
说明	禁止 EAS		
参数	名称	类型	备注
返回	见错误代码表		
备注			

2. 2. 14 EASNXPICODESLI_EASCheck

定义	public int NXPICODESLI_EASCheck(Byte EASFlag)		
说明	EAS 状态检测		
参数	名称	类型	备注
	EASFlag	Byte	保存 EAS 标志。该值为 0，EAS 为禁止状态；否则，EAS 为使能状态。
返回	见错误代码表		
备注			

2. 2. 15 NXPICODESLI_LockEAS

定义	public int NXPICODESLI_LockEAS()		
说明	锁定 EAS		
参数	名称	类型	备注
返回	见错误代码表		
备注			

2. 2. 16 NXPICODESLI_GetRandomAndSetPassword

定义	public int NXPICODESLI_GetRandomAndSetPassword (byte pwdNo, long pwd)		
说明	Icode slix 和 Icode slix2 标签密码认证，该函数自动 获取随机数异或。		
参数	名称	类型	备注
	pwdNo	byte	密码标识符。 0x01:读 0x02:写 0x04:私有化 0x08:销毁 0x10:EAS/AFI
	pwd	long	4 个字节认证密码
返回	见错误代码表		
备注			

2. 2. 17 NXPICODESLI_WritePassword

定义	public int NXPICODESLI_WritePassword(byte pwdNo, long pwd)		
说明	修改密码，在该命令发送之前，需先要进行进行密码认证，认证正确而且该密码块没锁才能修改成功。		
参数	名称	类型	备注
	pwdNo	byte	密码标识符。 0x01:读 0x02:写 0x04:私有化 0x08:销毁 0x10:EAS/AFI
	pwd	long	4 个字节认证密码
返回	见错误代码表		
备注			

2. 2. 18 NXPICODESLI_Enable64BitPwd

定义	public int NXPICODESLI_Enable64BitPwd()		
说明	使能 64 位密码保护。		
参数	名称	类型	备注
返回	见错误代码表		
备注			

2. 2. 19 NXPICODESLI_LockPassword

定义	public int NXPICODESLI_LockPassword(byte pwdNo)		
说明	锁定密码块，一旦锁定将无法修改密码。在该命令发送之前，需先认证密钥。		
参数	名称	类型	备注
	pwdNo	byte	密码标识符。 0x01:读 0x02:写 0x04:私有化 0x08:销毁 0x10:EAS/AFI
返回	见错误代码表		
备注			

2. 2. 20 NXPICODESLI_PasswordProtect

定义	public int NXPICODESLI_PasswordProtect(byte bandType)		
说明	使能 EAS 或 AFI 密码保护，一旦使能，修改 EAS 或 AFI 需要先 Set Password 认证。		
参数	名称	类型	备注
	bandType	byte	标签内存区域标识. 0x00:EAS 0x01:AFI
返回	见错误代码表		
备注			

2. 2. 21 NXPICODESLI_ProtectPage

定义	public int NXPICODESLI_ProtectPage(byte PPPointer, byte protSta)		
说明	保护数据块。PPPointer 作为块地址的分水岭，低于该地址的数据块由参数 protSta 的低 4 位表示保护状态，高于或等于该地址的数据块由参数 protSta 的高 4 位表示保护状态。		
参数	名称	类型	备注
	PPPointer	byte	高低数据块的分水岭的数据块地址
	protSta	byte	低 4 位表示低于分水岭的数据块的保护状态;高 4 位表示高于或等于分水岭的数据块的保护状态。 保护状态说明: 0x00:公用, 无读写限制; 0x01:读写由读密码保护; 0x02:写由写密码保护; 0x03:读由读密码保护, 写由写密码保护; 如果使能了 64 位密码, 读写操作都由读密码和写密

			码保护。
返回	见错误代码表		
备注			

2. 2. 22 NXPICODESLI_LockPageProtection

定义	public int NXPICODESLI_LockPageProtection(byte pageAddr)		
说明	锁定数据块的保护状态，一旦锁定，就无法解锁。		
参数	名称	类型	备注
	pageAddr	byte	高低数据块的分水岭的数据块地址。
返回	见错误代码表		
备注			

2.3 IS01443A 空中协议操作类——IS01443AInterface

2.3.1 SpecAIPInvenParamIS01443A

定义	public static SpecAIPInvenParamIS01443A IS01443A_CreateInvenParam(Object hInvenParamSpecList, byte AntennaID)		
说明	创建 IS01443A Inventory 参数数据节点，用于 RDR_TagInventory		
参数	名称	类型	备注
	hInvenParamSpecList	Object	Inventory 参数句柄，由 RDR_CreateInvenParamSpecList 创建
	AntennaID	byte	天线 ID 号，0 表示所有的 天线接口都要读取 IS01443A 标准的标签，其 他值为对应的天线口。
返回	见错误代码表		
备注			

2.3.2 IS01443A_ParseTagDataReport

定义	public static int IS01443A_ParseTagDataReport(Object hTagReport, IS01443ATag tagData)		
----	---	--	--

说明	解析 ISO14443A Inventory 的标签数据信息。该数据节点 从 RDR_GetTagDataReport 获得 , 用于 RDR_TagInventory 函数。		
参数	名称	类型	备注
	hTagReport	Object	标签数据信息的节点句柄
	tagData	ISO14443ATag	标签数据信息。
返回	见错误代码表		
备注	ISO14443ATag 定义如下: <pre> public class ISO14443ATag{ public long aip_id=0;//协议编号 public long tag_id=0;//标签编号 public long ant_id=0;//天线编号 public byte uid[]=null;//标签 UID } </pre>		

2.3.3 MFCL_Connect

定义	public int MFCL_Connect(ADReaderInterface hr, byte tagType, byte uid[])		
说明	激活 mifare classic 标签		
参数	名称	类型	备注
	hr	ADReaderInterface	读写器接口

	tagType	byte	标签类型 0: Mifare S50 1: Mifare S70
	uid	Byte[]	标签 UID
返回	见错误代码表		
备注			

2.3.4 MFCL_Authenticate

定义	public int MFCL_Authenticate(byte blkAddr, byte keyType, byte key[])		
说明	认证密钥		
参数	名称	类型	备注
	blkAddr	byte	数据块地址
	keyType	byte	Mifare classic 认证密钥类型. 0:key A 1:key B
	key	byte []	6 个字节的密钥数据
返回	见错误代码表		
备注			

2.3.5 MFCL_ReadBlock

定义	public int MFCL_ReadBlock(byte blkAddr, byte blkData[])		
说明	读数据块		
参数	名称	类型	备注
	blkAddr	byte	数据块地址
	blkData	byte []	所读取的 16 个字节数据
返回	见错误代码表		
备注			

2.3.6 MFCL_WriteBlock

定义	public int MFCL_WriteBlock(byte blkAddr, byte blkData[])		
说明	写数据块		
参数	名称	类型	备注
	blkAddr	byte	数据块地址
	blkData	byte []	将要写入的 16 个字节数据
返回	见错误代码表		
备注			

2.3.7 MFCL_FormatValueBlock

定义	public int MFCL_FormatValueBlock(byte blkAddr, long initValue)		
说明	格式化数值数据块		

参数	名称	类型	备注
	blkAddr	byte	数据块地址
	initValue	long	初始化数值
返回	见错误代码表		
备注			

2.3.8 MFCL_Restore

定义	public int MFCL_Restore(byte blkAddr)		
说明	备份金额，在使用备份金额之前，数据块必须先使用 MFCL_FormatValueBlock 进行数值块格式化。		
参数	名称	类型	备注
	blkAddr	byte	数据块地址
返回	见错误代码表		
备注			

2.3.8 MFCL_Increment

定义	public int MFCL_Increment(byte blkAddr, long val)		
说明	充值，在使用充值之前，数据块必须先使用 MFCL_FormatValueBlock 进行数值块格式化。		
参数	名称	类型	备注
	blkAddr	byte	数据块地址
	val	long	充值的额度
返回	见错误代码表		

备注	
----	--

2.3.8 MFCL_Decrement

定义	public int MFCL_Decrement(byte blkAddr, long val)		
说明	扣值，在使用扣值之前，数据块必须先使用 MFCL_FormatValueBlock 进行数值块格式化。		
参数	名称	类型	备注
	blkAddr	byte	数据块地址
	val	long	扣值的额度
返回	见错误代码表		
备注			

2.3.9 ULTRALIGHT_Connect

定义	public int ULTRALIGHT_Connect(ADReaderInterface hr, byte uid[])		
说明	连接 Utralight 标签		
参数	名称	类型	备注
	hr	ADReaderInterface	读写器接口
	uid	Byte[]	7 个字节序列号
返回	见错误代码表		
备注			

2.3.10 ULTRALIGHT_ReadMultiplePages

定义	public int ULTRALIGHT_ReadMultiplePages(long pageStart, long pageNum, byte databuf[], Integer nSize)		
说明	读多页		
参数	名称	类型	备注
	pageStart	long	开始页， ultralight 页地址 0-15； ultralightC 页地址 0-47；
	pageNum	long	页数量
	databuf	Byte[]	读到的页数据
	nSize	Integer	读到的数据长度
返回	见错误代码表		
备注			

2.3.11 ULTRALIGHT_WriteMultiplePages

定义	public int ULTRALIGHT_WriteMultiplePages(int pageStart, int pageNum, byte databuf[], int bytesToWrite)		
说明	写多页		
参数	名称	类型	备注
	pageStart	int	开始页， ultralight 页地址 0-15；

			ultralightC 页地址 0-47;
	pageNum	int	页数量
	databuf	byte[]	将要写入的页数据
	bytesToWrite	int	写入的数据长度
返回	见错误代码表		
备注			

2. 3. 12 ISO14443A_Disconnect

定义	public int ISO14443A_Disconnect()		
说明	断开连接的标签		
参数	名称	类型	备注
返回	见错误代码表		
备注			

3. 连接串详细说明

3.1 概述

连接串由参数字段组成，参数之间使用” ;” 分号分隔，每个参数有名称和值组成，中间用” =” 等号分隔。连接串有 2 个基本的参数，RDType(设备驱动的名称或型号)和 CommType(通信接口类型) 。通信接口类型支持串口，蓝牙和网络接口。

3.2 串口连接串说明

3.2.1 参数例子

采用串口连接 M201 设备的连接串为：

```
RDType=M201;CommType=COM;ComPath=/dev/ttyMT1;Baund=38400;Frame=8E1;Addr=255
```

3.2.2 参数说明

序号	名称	值说明	
1	RDType	设备驱动的类型，传入设备名称	
2	CommType	通信接口类型，这里选择“COM”	
		值	说明
		COM	串口
		BLUETOOTH	蓝牙接口
		NET	网络接口
3	ComPath	串口路径，如：/dev/ttyMT1	
4	Baund	波特率，如：“38400”	
5	Frame	帧结构：	
		值	说明
		8E1	8 位数据位，偶校验，1 位停止位
		8O1	8 位数据位，奇校验，1 位停止位
		8N1	8 位数据位，无校验，1 位停止位
6	Addr	总线地址，1-254，255 位广播地址。	

3.3 蓝牙连接串说明

3.3.1 参数例子

采用蓝牙方式连接 RPAN 设备的连接串为：

RDType=RPAN;CommType=BLUETOOTH;Name=R-PAN. D835

3.3.2 参数说明

序号	名称	值说明	
1	RDType	设备驱动的类型，传入设备名称	
2	CommType	通信接口类型，这里选择“BLUETOOL”	
		值	说明
		COM	串口
		BLUETOOTH	蓝牙接口
		USB	USB 接口
		NET	网络接口
3	Name	已经成功配对蓝牙名称	

3.4 网络连接串说明

3.4.1 参数例子

采用网络方式连接 RPAN 设备的连接串为：

RDType=RPAN;CommType=NET;RemoteIp=192.168.1.88;RemotePort=4800

3.4.2 参数说明

序号	名称	值说明	
1	RDType	设备驱动的类型，传入设备名称	
2	CommType	通信接口类型，这里选择“NET”	
		值	说明
		COM	串口
		BLUETOOTH	蓝牙接口
		USB	USB 接口
		NET	网络接口
3	RemoteIp	设备的 IP 地址	
4	RemotePort	设备的端口号	

3.5 USB 连接串说明

3.5.1 参数例子

采用 USB 方式连接 RL8000 设备的连接串为:
RDType=RL8000;CommType=USB;Description=0

3.4.2 参数说明

序号	名称	值说明	
1	RDType	设备驱动的类型，传入设备名称	
2	CommType	通信接口类型，这里选择“USB”	
		值	说明
		COM	串口
		BLUETOOTH	蓝牙接口
		USB	USB 接口
		NET	网络接口
3	Description	USB 序列号，从 0 开始	

安卓系统的版本必须在 5.1.1 以上。

4. 错误代码表

错误代码	描述
0	无错误，表示 API 调用成功。
-1	未知错误
-2	与读卡器硬件的通信失败
-3	API 的传入参数有误
-4	API 的传入参数的值不支持，如参数值只能是 0-5，如果传入 6 那么会返回该错误。
-5	超时，发送到读卡器的命令，在设定时间内等不到数据返回。
-6	API 申请内存失败
-7	功能未开启
-8	设备不支持指令
-9	保留
-10	保留
-11	保留
-12	读卡器返回的数据包长度有误
-13	保留
-14	保留
-15	保留
-16	保留
-17	读卡器返回操作失败标识数据包，可用 API RDR_GetReaderLastReturnError 获取该失败的错误代码。
-18	保留
-19	保留
-20	保留
-21	Inventory 的停止触发器发生，举个例子：假设设定 1 秒为 Inventory 的最大读卡时间，如果在 1 秒钟内还没读完所有的标签，读卡器会终止 Inventory，那么 API 会返回该错误告诉应用程序，可能还有标签没读完。
-22	标签操作命令不支持
-23	保留
-24	保留
-25	TCP socket 错误，API 返回该错误表明 TCP 连接已断开。
-26	应用层传入的缓冲区太小。
-27	与读卡器返回的数据有误。