



北京圆志科信 射频读写器 应用手册



RW210Ex

地址：北京市通州区通胡大街 78 号京贸中心 1004D

电话：010-64389905

传真：010-89524306

Web: <http://www.yzrfid.com>

0.1 声明

本说明书是为了让用户更好的选择北京圆志科信电子科技有限公司的产品而提供的开发资料，不转让属于北京圆志科信电子科技有限公司或者第三者所有的知识产权，用户在确定使用本产品前，请根据自己实际需求对产品性能及其使用安全性等方面进行相应评估，北京圆志科信电子科技有限公司不承担因评估不当而造成的直接或间接损失，也不承担因此而带来的任何法律或经济责任。

北京圆志科信电子科技有限公司致力于为用户提供不断完善的服务与产品，保有对产品及其相应说明书更新的权利，如有变动，恕不另行通知，在确定购买此产品时，请预先联系北京圆志科信电子科技有限公司以确认是否为最新版本。

本说明书所有权归北京圆志科信电子科技有限公司所有，未经许可，不得翻印或复制全部或部分本资料内容。

0.2 更改历史记录

版本	描述	日期
V1.0	PDF 版本第一版发布	2017.12.29
V1.1	文档格式进行了调整, 转化为简介+chm 方式	2018.03.05
V1.2	在原有基础上增加对 RW210ECN 的描写	2018.07.25
V1.3	增加了对 2 字节长度的描述, 适用那些每帧数据超过 255 字节的特殊应用, 需要注意的是 2 字节长度与 1 字节长度模块的固件不同, 故型号不同, 2 字节长度的模块型号在原默认 1 字节的型号上增加 L2	2018.12.25
V1.4	DII 描述增补对 RW210EH 读卡器端口连接的描述	2021.02.05
V1.5	增加读卡器外观图片	2021.03.12

目 录

0.1 声明	2
0.2 更改历史记录	3
1. 概述	8
1.1 RW210Ex IC 卡读写器简介	8
1.2 产品型号及之间的区别	8
1.3 主要特点	9
1.4 技术指标	10
2. 硬件连接	11
2.1 读卡器外观	11
2.1.1 RS232 串口读卡器	11
2.1.2 USB 读卡器	11
2.1.3 RS485 读卡器/TTL232 读卡器	12
2.1.4 外壳尺寸图	12
2.2 读卡器硬件连接	13
2.2.1 RS232 串口连接	13
2.2.2 RS485 连接	13
2.2.3 TTL232 连接	14
2.2.4 USB 非免驱读卡器连接	15
2.2.5 USB 免驱读卡器连接	16
2.3 上电状态	16
3. 通过读卡器操作卡片步骤简述	17
3.1 MifareOne 卡操作流程	17
3.2 Mifare Ultralight / Ntag 卡操作流程	17
3.3 ISO14443 TYPE A T=CL CPU 卡操作流程	18
3.4 Mifare Desfire EV0/EV1/EV2 卡操作流程	18
3.5 ISO7816-4 卡操作流程 (可支持多个卡)	19
3.6 ISO7816-4 卡操作流程 (只支持 1 个卡)	19
3.7 ISO14443 TYPE B T=CL CPU 卡操作流程	20
3.8 ISO14443 TYPE B ST 卡操作流程	20
3.9 ISO14443 TYPE B 二代证卡操作流程	20
3.10 Felica 卡操作流程	21
3.11 ISO15693 卡操作流程	21
4. 动态库函数说明	21
4.1 DLL 库应用说明	21
4.2 系统函数指令	22
通用函数功能: 获取动态库版本号	22
通用函数功能: DES 算法加密函数	22
通用函数功能: DES 算法解密算法函数	22
通用函数功能: MAC 计算函数	22

通用函数功能: 密钥分散函数.....	23
通用函数功能: 端口连接	23
通用函数功能: 关闭串口	24
通用函数功能: 设置读写卡器非接触工作方式.....	24
通用函数功能: 设置读写卡器天线状态	24
通用函数功能: 设置读写卡器天线状态新指令	25
通用函数功能: 蜂鸣器控制.....	25
通用函数功能: 控制指示灯	26
通用函数功能: 读取读写卡器硬件版本号	26
通用函数功能: 读取读写卡器产品序列号	26
通用函数功能: 设置读写卡器设备号	26
通用函数功能: 获取读写卡器设备号	27
4.3 MifareOne 卡函数指令	27
MF1 卡专用功能: 寻 ISO14443A 卡.....	27
MF1 卡专用功能: ISO14443A 卡防冲突	27
MF1 卡专用功能: ISO14443A 卡选卡	27
MF1 卡专用功能: 验证 MifareOne 卡密钥	28
MF1 卡专用功能: 读取 MifareOne 卡一块数据.....	28
MF1 卡专用功能: 写入 MifareOne 卡一块数据.....	28
MF1 卡专用功能: 将 Mifare One 卡某一块初始化为钱包.....	29
MF1 卡专用功能: 读取 Mifare One 卡钱包值.....	29
MF1 卡专用功能: Mifare One 卡扣款	29
MF1 卡专用功能: Mifare One 卡充值	29
MF1 卡专用功能: Mifare One 卡数据回传	30
MF1 卡专用功能: Mifare One 卡数据传送.....	30
MF1 卡专用功能: Mifare One 卡读取连续 3 块数据.....	30
MF1 卡专用功能: Mifare One 卡写连续 3 块数据.....	30
MF1 卡专用功能: 命令已激活的 ISO14443A 卡进入 HALT 状态.....	31
4.4 Ultralight/Ntag 卡函数指令	31
UL /Ntag 专用功能: 寻 ISO14443A 卡	31
UL/Ntag 专用功能: UltraLight/Ntag 选卡	31
UL/Ntag 专用功能: 读取 Ultralight/Ntag 卡 4 页数据	32
UL/Ntag 专用功能: 向 Ultralight/Ntag 卡中写入一页数据	32
Ntag21x 专用功能: Ntag21x 系列快速读取	32
Ntag21x 专用功能: Ntag21x 系列获取版本号	32
Ntag21x 专用功能: Ntag21x 系列密钥认证	33
Ntag21x 专用功能: Ntag21x 系列读取计数值	33
Ntag21x 专用功能: Ntag21x 系列读取签名	33
NFC 卡专用功能: NFC 类型 2 标签选择扇区	34
UL/Ntag 专用功能: 命令已激活的 ISO14443A 卡进入 HALT 状态	34

4.5 ISO14443A-4 CPU 卡函数指令	34
TYPE A CPU 卡专用功能: 符合 ISO14443A-4 标准的非接触 CPU 卡复位	34
TYPE A CPU 卡专用功能: 向符合 ISO14443A-4 CPU 卡发送 COS 命令	34
TYPE A CPU 卡专用功能: 向符合 ISO14443A-4 CPU 卡发送 PPS 请求	35
TYPE A CPU 卡专用功能: ISO14443A-4 CPU 卡退出激活状态	35
4.6 ISO7816-4 SAM 卡函数指令 (支持多个卡)	35
ISO7816-4 SAM/SIM 卡专用功能: 第 N 号卡复位(支持多卡)	35
ISO7816-4 SAM/SIM 卡专用功能: 向第 N 号卡发送 APDU(COS)指令(支持多卡)	36
ISO7816-4 SAM/SIM 卡专用功能: 向第 N 号卡发送 PPS 请求(支持多卡)	36
4.7 ISO7816-4 SAM 卡函数指令 (只支持 1 个卡)	37
ISO7816-4 SAM/SIM 卡专用功能: 初始化卡片波特率 (只支持 1 个卡)	37
ISO7816-4 SAM/SIM 卡专用功能: 复位卡片 (只支持 1 个卡)	38
ISO7816-4 SAM/SIM 卡专用功能: 发送 APDU(COS)指令 (只支持 1 个卡)	38
4.8 ISO14443B-4 CPU 卡指令	38
TYPE B CPU 卡专用功能: 符合 ISO14443B-4 CPU 卡复位	38
TYPE B CPU 卡专用功能: 向符合 ISO14443B-4 CPU 卡发送 COS 命令	39
TYPE B CPU 卡专用功能: 向符合 ISO14443B-4 CPU 卡退出激活状态	39
TYPE B CPU 卡专用功能: ISO14443 TYPE B 卡休眠	39
4.9 ISO14443B ST 卡指令	40
ST 卡专用功能: 寻卡	40
ST 卡专用功能: ST 卡获取 UID	40
ST 卡专用功能: ST 卡读块	40
ST 卡专用功能: ST 卡写块	40
ST 卡专用指令: SRI512 卡块锁定	41
ST 卡专用指令: SRI4K 卡块锁定	41
ST 卡专用指令: ST 卡退出选择	42
4.10 二代证卡指令	42
二代证专用功能: 读取二代证的 UID 卡号	42
4.11 Felica 卡指令	42
Felica 专用功能: 读取 Felica 的 UID 卡号	42
4.12 ISO15693 卡指令	43
ISO15693 卡专用指令: ISO15693 卡寻卡	43
ISO15693 卡专用指令: ISO15693 卡获取卡片系统信息	43
ISO15693 卡专用指令: ISO15693 卡读块	43
ISO15693 卡专用指令: ISO15693 卡写块	44
ISO15693 卡专用指令: ISO15693 卡写连续多块	45
ISO15693 卡专用指令: ISO15693 卡锁块	45
ISO15693 卡专用指令: 获取 ISO15693 卡块锁定信息	45
ISO15693 卡专用指令: ISO15693 卡写 AFI	46
ISO15693 卡专用指令: ISO15693 卡锁定 AFI	46

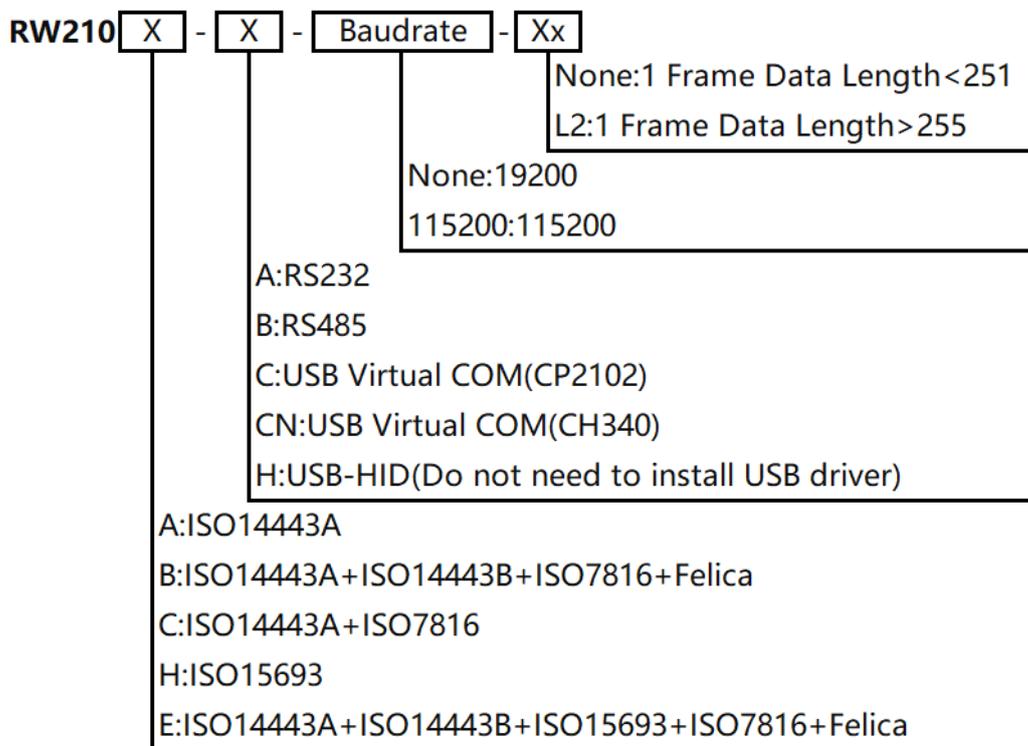
ISO15693 卡专用指令：ISO15693 卡写 DSFID.....	47
ISO15693 卡专用指令：ISO15693 卡锁定 DSFID.....	47
ISO15693 卡专用指令：ISO15693 卡选卡.....	48
ISO15693 卡专用指令：ISO15693 卡复位.....	48
ISO15693 卡专用指令：ISO15693 卡进入静默.....	48
ISO15693 卡专用指令：ISO15693 卡寻多张卡.....	48
ISO15693-NXP 卡专用指令：ISO15693-NXP 卡设置 EAS.....	49
ISO15693-NXP 卡专用指令：ISO15693-NXP 卡 EAS ALARM.....	49
ISO15693-NXP 卡专用指令：ISO15693-NXP 卡复位 EAS.....	50
ISO15693-NXP 卡专用指令：ISO15693-NXP 卡锁定 EAS.....	50

1. 概述

1.1 RW210Ex IC 卡读写器简介

RW210Ex 读写器采用 13.56MHz 非接触射频技术，内嵌飞利浦射频基站。用户不必关心射频基站的复杂控制方法，只需通过简单的接口发送命令或操作函数就可以实现对卡片完全的操作。该系列读写卡器支持 ISO14443-A Mifare One S50, S70, FM11RF08, Ultralight, Ntag203, Ntag213, 215, 216, FM1208, Desfire EV1/EV2 D21, D41, D81, T=CL CPU 卡, ISO14443-B SRI512, SRI4K, T=CL CPU 卡, 二代证(只读 UID), ISO15693 I Code SL2, Tag it, Felica(只读 UID), ISO7816-4 SAM, SIM 卡及其兼容卡片。

1.2 产品型号及之间的区别



型号	主要区别	备注
RW210EA	RS232 接口, 带 1 个 SAM 卡座	
RW210EB	RS485 接口, 带 1 个 SAM 卡座	
RW210EC	USB 接口 (CP2102), 带 1 个 SAM 卡座	需安装驱动
RW210ECN	USB 接口 (CH340), 带 1 个 SAM 卡座	需安装驱动
RW210EH	USB (HID) 接口, 带 1 个 SAM 卡座	免驱动
RW210ET	TTL232 接口, 带 1 个 SAM 卡座	

说明：以上型号均为标准默认型号，特殊需增加尾缀。

- (1) 如果出厂默认 115200 波特率时，需增加尾缀 115200。
- (2) RW210Ex 系列默认带 1 个 SAM 卡座，需支持 2 个时请特殊说明，型号尾缀增加“-2SAM”。
- (3) 模块默认长度字为 1 字节，即每帧数据不能超过 251 字节，如果每帧数据长度大于 255 字节，则需要在型号之后再增加尾缀 L2，此应用主要针对部分特殊的 CPU 卡指令。

1.3 主要特点

- 全新一代读卡器，内嵌 NXP 32 位 ARM 处理器，处理速度更快
- 采用 NXP 高度集成读写卡芯片射频基站，性能更优，更稳定
- 壳体外观专业美学设计，更美观
- 工作频率：13.56MHZ
- 支持 ISO14443A / MIFARE 标准/ISO14443B/ISO15693/ISO7816 国家标准
- 可支持 IC 卡：
 - ISO14443A: Mifare 1K/4K, FM11RF08, Ultralight, Ntag203, Nta213, Ntag215, Ntag216, 符合 T=CL 协议的 CPU 卡 (ISO14443A) FM1208, Desfire EV1/EV2 D21, D41, D81
 - ISO14443B: SRI512, SRI4K, 符合 T=CL 协议的 CPU 卡 (ISO14443B) ,二代证 (只读 UID 卡)
 - ISO15693: I Code SL2, Tag it
 - ISO17816-4: SAM/SIM9600(T=0 & T=1), SAM/SIM38400(T=0 & T=1), SAM/SIM115200(T=0 & T=1)
- 其他协议：Felica 卡及其兼容卡片；
- 可支持 SAM 卡，SIM 卡操作，可选装 1-2 个 PSAM 卡座或者 1-2 个 SOP8 ESAM 芯片或者 1-2 个 DIP8 ESAM 芯片
- 内嵌两个可控指示灯
- 内嵌蜂鸣器
- 自带看门狗，防止死机
- 上电默认自动寻卡，有卡靠近，绿色指示灯点亮
- 对外接口：RS232/RS485/USB/TTL232 接口可选
- 用户不必关心射频基站的复杂控制方法，只需通过简单的命令集可以实现对卡的完全操作
- 为了便于开发者的应用，我们提供了 PCDMO 测试软件，以及配套的各种开发环境例程：比如适合 Windows 的 java, javascript, VC, VB, C++ Builder, DELPHI, Power Builder, C#,

CVI 等,同时还有适合 Android, Linux 等例程

- 基于读卡器的扩展功能很强可**根据用户要求修改软件定制**个性化功能,不用改变硬件
- 丰富、完善的接口函数,可方便地应用于充电桩,交通一卡通,医疗,门禁考勤、汽车电子,感应锁,密集柜等智能卡管理应用系统

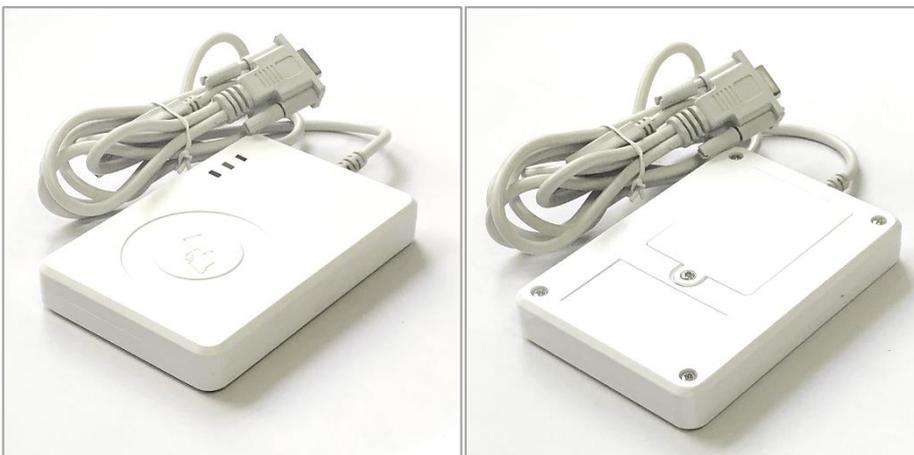
1.4 技术指标

- 工作频率: 13.56MHz
- 支持的标准: ISO14443A/Mifare/ISO14443B/ISO15693/ISO7816
- 支持 IC 卡:
ISO14443A: Mifare 1K/4K, FM11RF08, Ultralight, Ntag203, Ntag213, Ntag215, Ntag216, 符合 T=CL 协议的 CPU 卡 (ISO14443A) FM1208, Desfire EV1/EV2 D21, D41, D81
ISO14443B: SRI512, SRI4K, 符合 T=CL 协议的 CPU 卡 (ISO14443B) ,二代证 (只读 UID 卡)
ISO15693: I Code SL2, Tag it
ISO17816-4: SAM/SIM9600(T=0 & T=1), SAM/SIM38400(T=0 & T=1), SAM/SIM115200(T=0 & T=1)
其他协议: Felica 卡
及其兼容卡片;
- 供电电压: DC5V \pm 10%
- 接口: RS232/RS485/USB/TTL232 接口可选
- 通讯速率: 默认 19.2Kbp 或者 115200
- 最大功耗: 150mA
- 读卡距离: 40-60mm (依据标准 M1 卡测试所得)
- 工作温度: -20°C ~ +70°C
- 外形尺寸: 125mm*85mm*19mm
- 重量: 130~160g (内部 USB 线约 45g, RS232 线约 75g)

2. 硬件连接

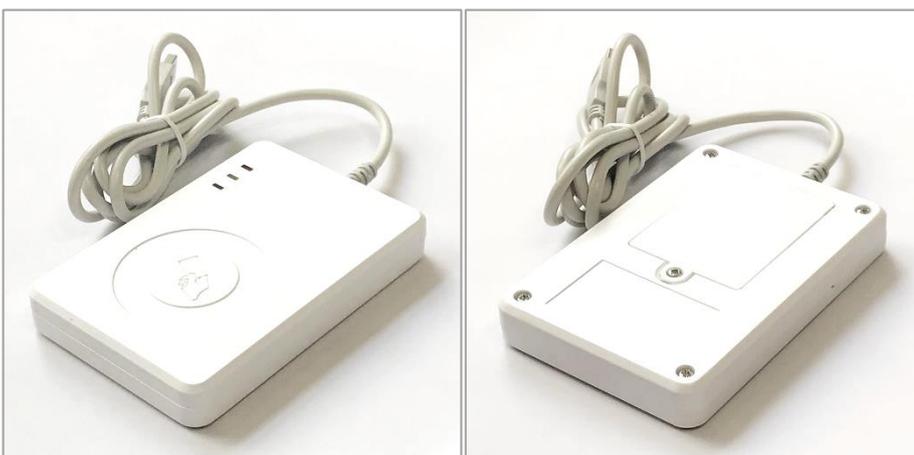
2.1 读卡器外观

2.1.1 RS232 串口读卡器

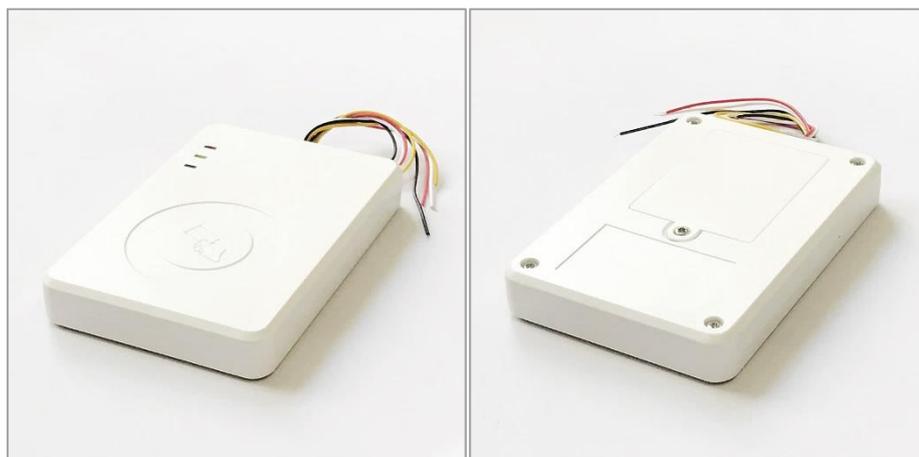


注：USB 插头仅用于取电，不用于通讯。

2.1.2 USB 读卡器



2.1.3 RS485 读卡器/TTL232 读卡器



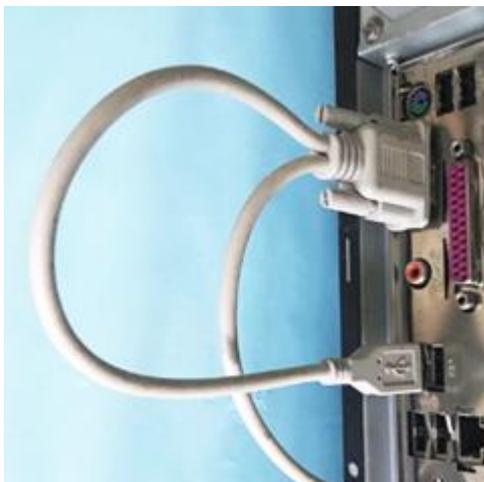
2.1.4 外壳尺寸图



2.2 读卡器硬件连接

2.2.1 RS232 串口连接

适用读卡器型号：RW210EA



如图方式，USB 为取电用途，不具有通讯功能，插入 USB 口取电 9 针串口头为通讯功能，插入主机 9 针串口即可

2.2.2 RS485 连接

适用读卡器型号：RW210EB



如图读卡器出线 4 根：

线色	含义
红色	电源正 5V
黑色	电源地
黄色	RS485-A
白色	RS485-B

2.2.3 TTL232 连接

适用读卡器型号：RW210ET

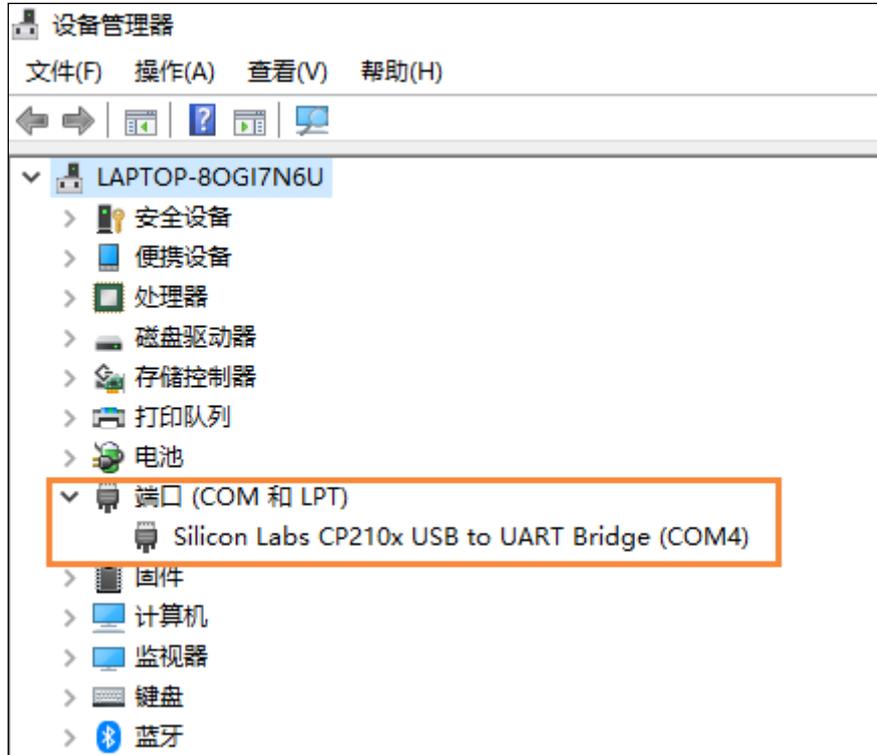


如图读卡器出线 4 根：

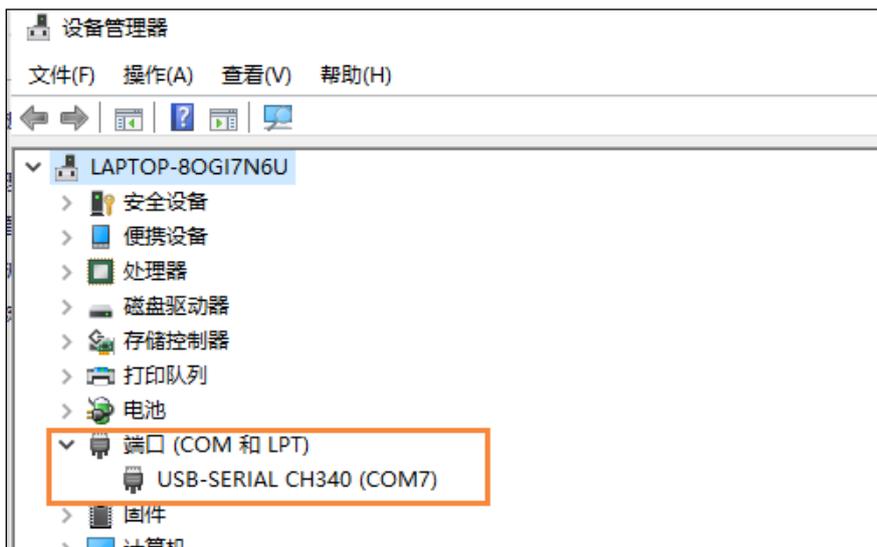
线色	含义
红色	电源正 5V
黑色	电源地
黄色	TTL232: TXD
白色	TTL232: RXD

2.2.4 USB 非免驱读卡器连接

(1)RW210EC 需要安装驱动程序 (CP210x), 以模拟出一个串口, 驱动可从附带资料中找到, 驱动安装成功之后, 在设备管理器中会显示出虚拟的串口, 并且没有黄色叹号表示驱动安装成功, 如下图:

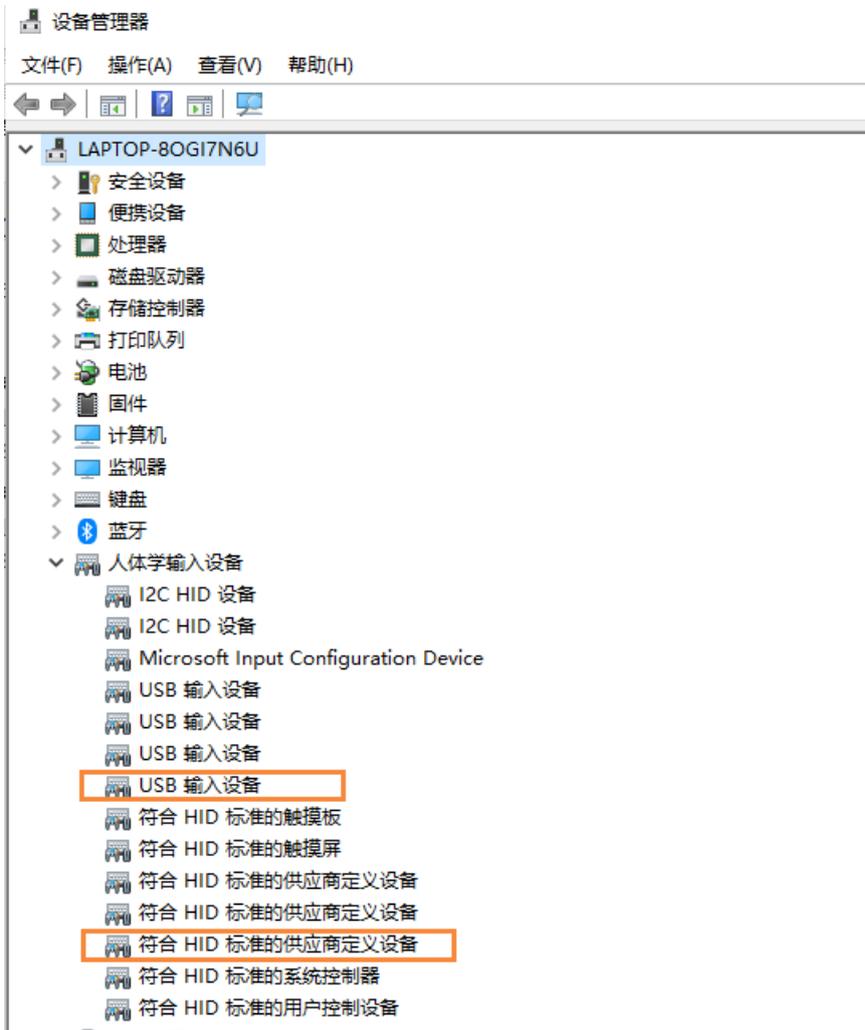


(2)RW210ECN 需要安装驱动程序 (CH340), 以模拟出一个串口, 驱动可从附带资料中找到, 驱动安装成功之后, 在设备管理器中会显示出虚拟的串口, 并且没有黄色叹号表示驱动安装成功, 如下图



2.2.5 USB 免驱读卡器连接

RW210EH 为 USB HID 接口，系统会自动安装驱动程序，无需用户单独安装，系统自动安装成功之后，会在设备管理器中增加 USB 描述符，如下图（不同的 Windows 版本显示可能不同）：



备注：硬件连接成功之后，可通过我公司附带的 PC_DEMO 测试软件进行常规功能测试，以协助您进行二次开发，具体演示软件说明，请参阅附带的 PC_DEMO 说明。

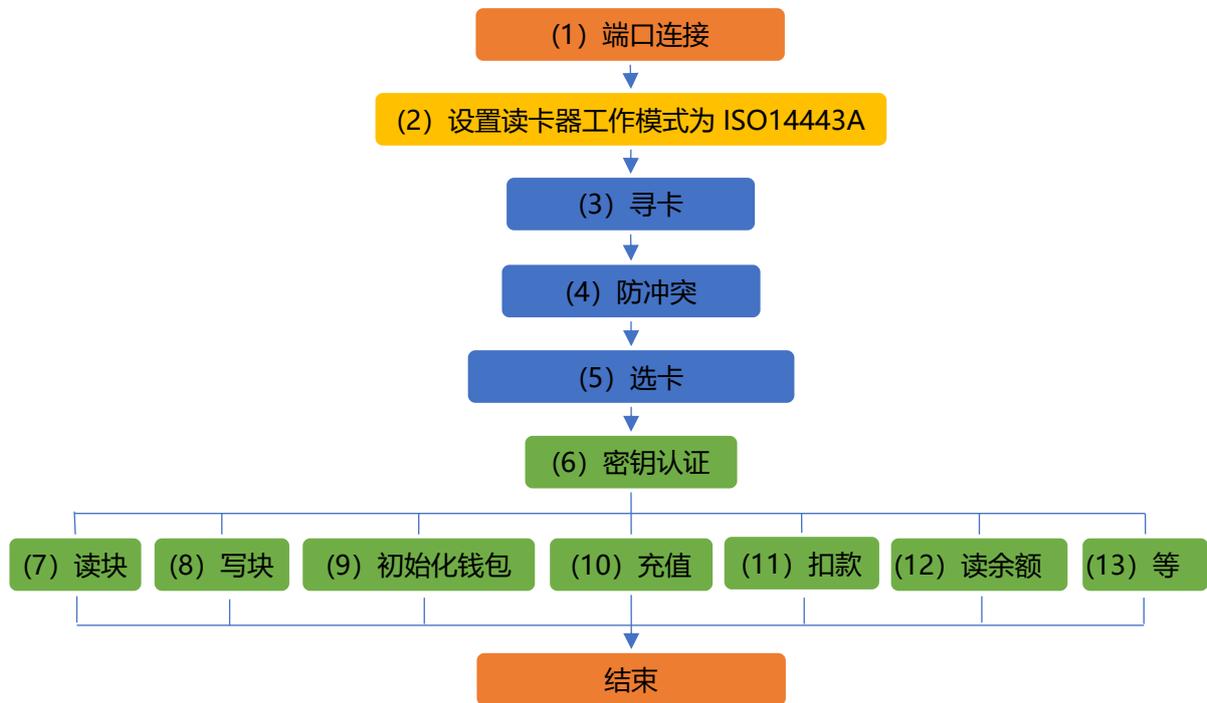
2.3 上电状态

读卡器上电后黄绿发光二极管闪烁几次后熄灭，红发光二极管(电源指示灯)长亮。读卡器默认上电有自动检测卡功能，当卡片靠近读卡器时，绿色指示灯会亮起，该功能在主机发送命令之后自动取消。

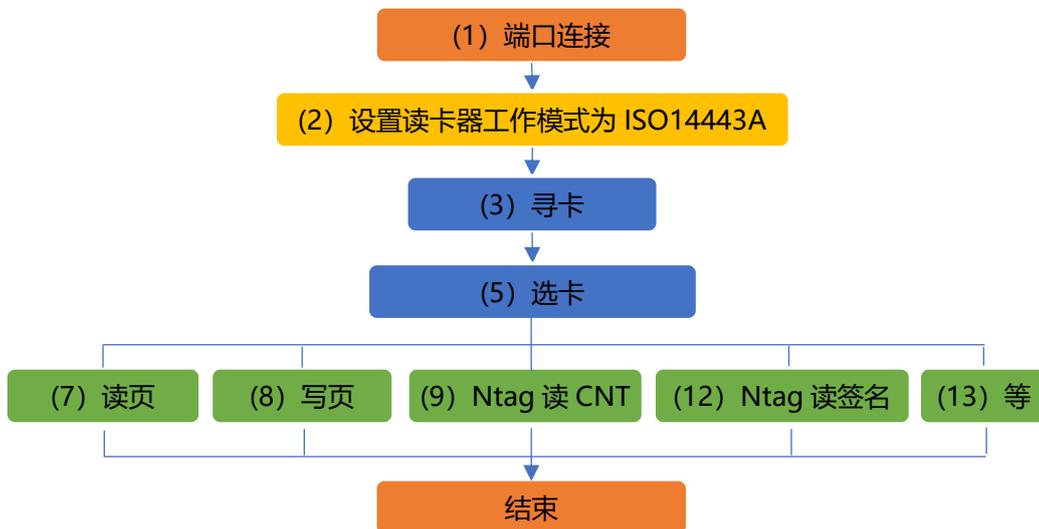
注：读卡器也可以实现上电默认自动寻卡，并主动输出卡号功能，如有需求请联系圆志。

3. 通过读卡器操作卡片步骤简述

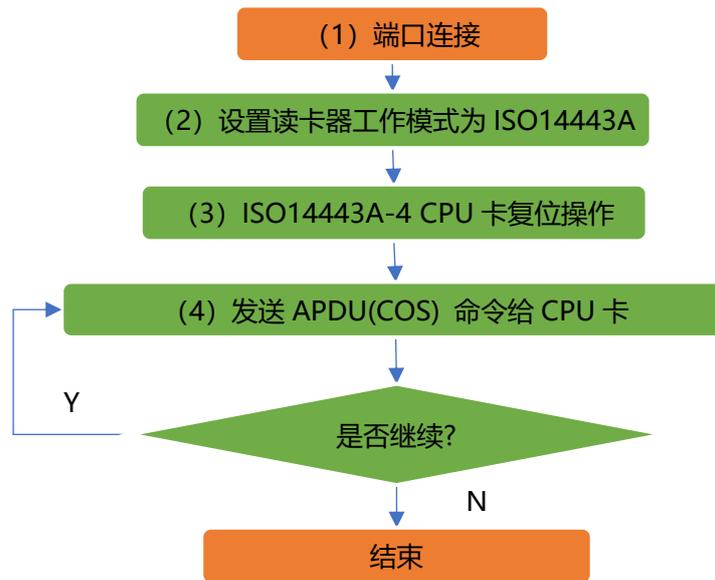
3.1 MifareOne 卡操作流程



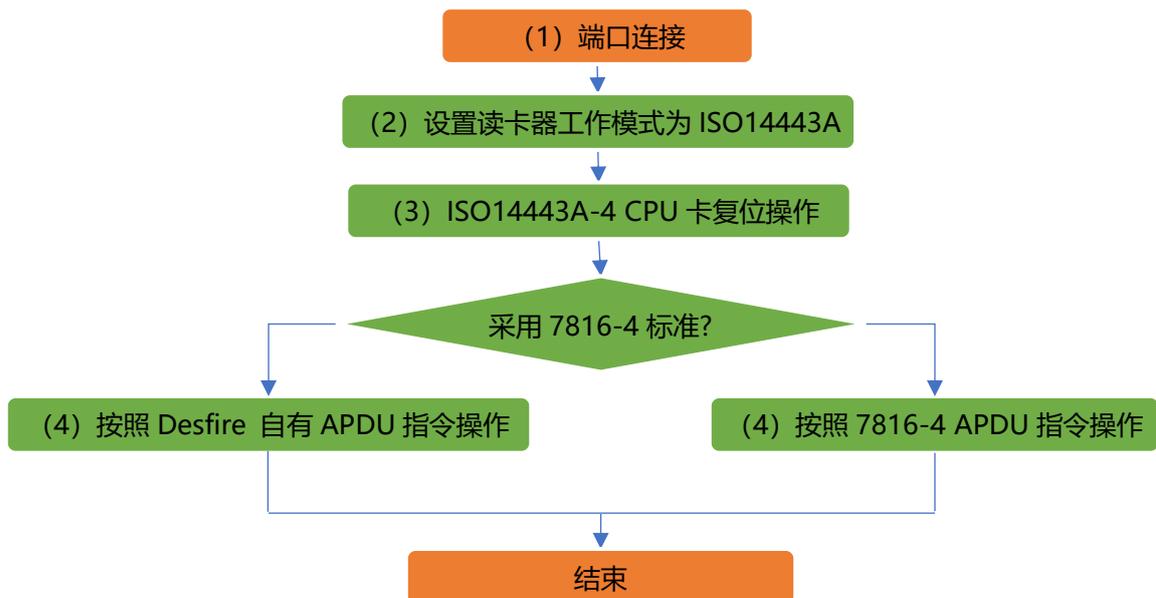
3.2 Mifare Ultralight / Ntag 卡操作流程



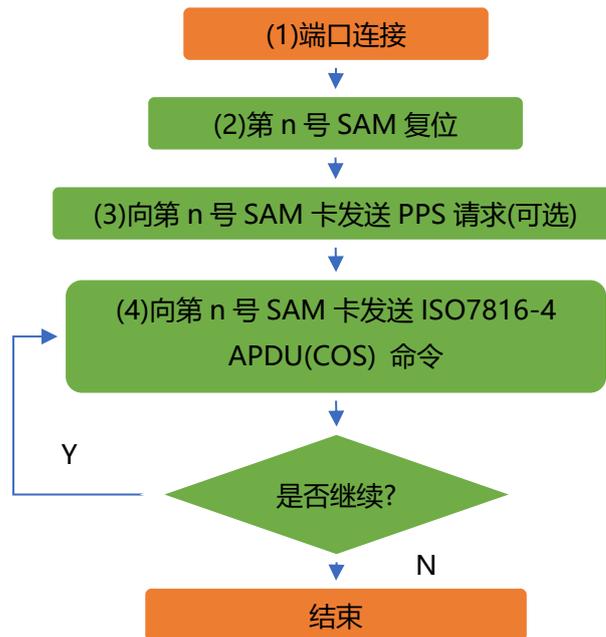
3.3 ISO14443 TYPE A T=CL CPU 卡操作流程



3.4 Mifare Desfire EV0/EV1/EV2 卡操作流程

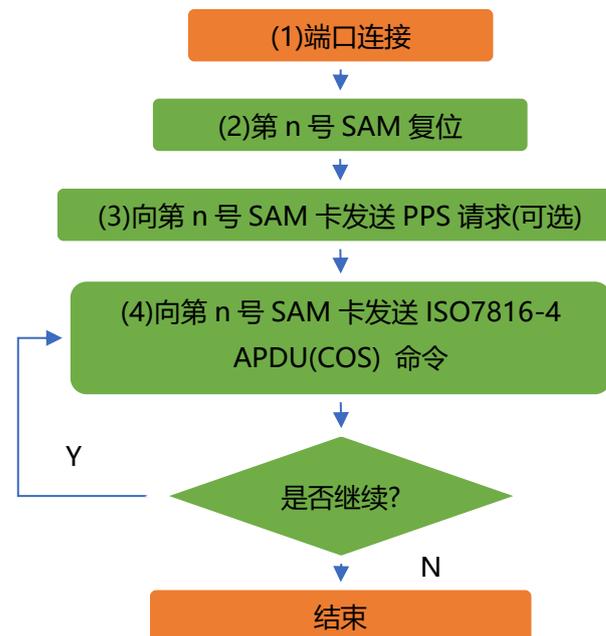


3.5 ISO7816-4 卡操作流程 (可支持多个卡)



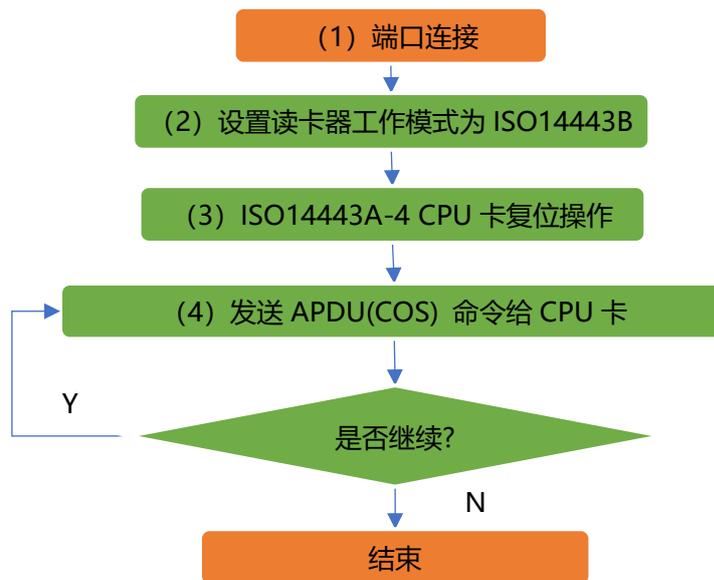
新的用户建议采用此操作流程，可以扩展对多个 PSAM 卡片的支持。

3.6 ISO7816-4 卡操作流程 (只支持 1 个卡)

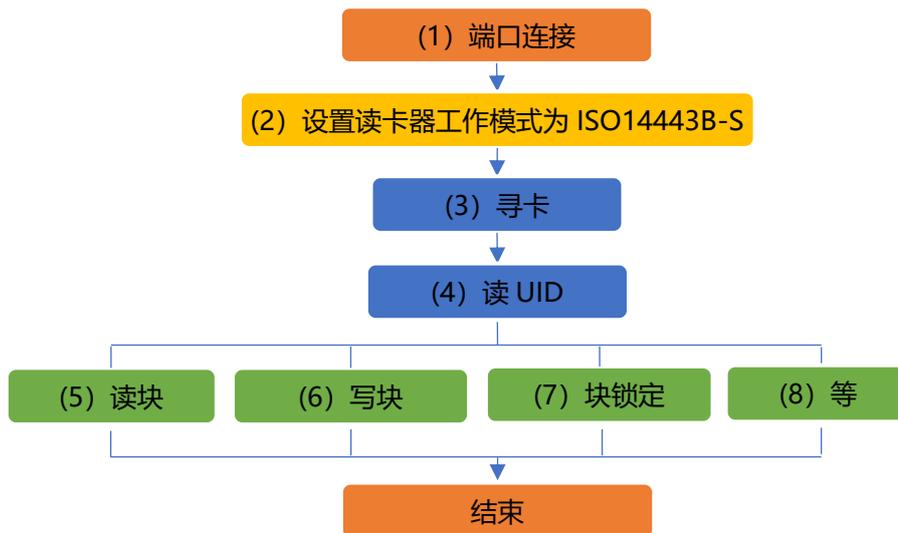


此操作流程是为了兼容老的型号读卡器，只能支持一个 PSAM 卡，对应读卡器内部的 1 号卡座位置。

3.7 ISO14443 TYPE B T=CL CPU 卡操作流程



3.8 ISO14443 TYPE B ST 卡操作流程



3.9 ISO14443 TYPE B 二代证卡操作流程



3.10 Felica 卡操作流程



3.11 ISO15693 卡操作流程



4. 动态库函数说明

4.1 DLL 库应用说明

对于 RW210EA,RW210EB,RW210EC,RW210ECN,RW210ET:

(1) 32 位库函数用于 32 位系统时，需将该对应动态库 MasterRDnew.dll 和 MasterCom.dll 复制到 C:\Windows\system32 文件夹下。

(2) 32 位库函数用于 64 位系统时，需将该对应动态库 MasterRDnew.dll 和 MasterCom.dll 复制到 C:\Windows\SysWOW64 文件夹下。

(3) 读卡器默认例程均为 32 位库，需要 64 位库函数，可以联系圆志索取。

(4) 长度字为 2 字节或者 1 字节的 DLL 动态库不同，读卡器也不同，默认全为 1 字节长度。

对于 RW210EH:

(1) 32 位库函数用于 32 位系统时，需将该对应动态库 MasterRDHID.dll 和 MasterHID.dll 复制到 C:\Windows\system32 文件夹下。

(2) 32 位库函数用于 64 位系统时，需将该对应动态库 MasterRDHID.dll 和 MasterHID.dll 复制到 C:\Windows\SysWOW64 文件夹下。

(3) 读卡器默认例程均为 32 位库，需要 64 位库函数，可以联系圆志索取。

对于 RW210EH:

原型: int WINAPI rf_init_com (unsigned short icdev, int nVID, int nPID)

参数: icdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

nVID: [IN]6790 (0x1A86)

nPID: [IN]57360 (0xE010)

返回: 成功返回 0

失败返回非 0

备注: 对应 dll 库为 MasterRDHID.dll 和 MasterHID.dll

通用函数功能: 关闭串口

功能: 关闭连接端口

原型: int WINAPI rf_ClosePort ()

参数: 空

返回: 成功返回 0

失败返回非 0

通用函数功能: 设置读写卡器非接触工作方式

功能: 设置读写卡器非接触工作方式, 此条适用于支持多协议的读写器

原型: int WINAPI rf_init_type (unsigned short wlcdev, unsigned char byType)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byType: [IN]读写卡器工作模式

=0x41 ('A') 设置为 ISO14443A 模式 (MF1,Ultralight,Ntag,TypeA CPU 等);

=0x42 ('B') 设置为 ISO14443B 模式 (Type B CPU 卡, 二代证等);

=0x43 ('C') 设置为 Felica 方式 (Felica);

=0x73 ('s') 设置为 ST 卡方式 (Type B SR176,SRI512,SRI4K 等);

=0x72 ('r') 设置为 Atmel 卡方式 (Type B At88rf020 等);

=0x31 ('1') 设置为 ISO15693 方式 (I Code SL2,Tag it 等);

返回: 成功返回 0

失败返回非 0

通用函数功能: 设置读写卡器天线状态

功能: 设置读写卡器天线状态

原型: int WINAPI rf_antenna_sta (unsigned short wlcdev, unsigned char byMode)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byMode: [IN]1 字节控制信息

bit0=0: 关闭天线; bit0=1: 开启天线

bit1=0: 关闭自动寻卡; bit1=1: 开启自动寻卡

bit2=0: 无任何操作; bit2=1: 复位读卡芯片

bit3-7: 备用, 固定为 0

返回: 成功返回 0

失败返回非 0

说明: bit1 和 bit2 功能需要读卡器支持才可以

通用函数功能: 设置读写卡器天线状态新指令

功能: 设置读写卡器天线状态, 此函数需 DLL 版本大于等于 V28。

原型: int WINAPI rf_antenna_sta_new (unsigned short wlcdev,
unsigned char *pSet, unsigned char bySetLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

当 bySetLg=1 时:

pSet[0]: [IN]1 字节控制信息

bit0=0: 关闭天线; bit0=1: 开启天线

bit1=0: 关闭自动寻卡; bit1=1: 开启自动寻卡

bit2=0: 无任何操作; bit2=1: 复位读卡芯片

bit3-7: 备用, 固定为 0

当 bySetLg=3 时:

pSet[0]: [IN]1 字节控制信息

bit0=0: 关闭天线; bit0=1: 开启天线

bit1=0: 关闭自动寻卡; bit1=1: 开启自动寻卡

bit2=0: 无任何操作; bit2=1: 复位读卡芯片

bit3-7: 备用, 固定为 0

pSet[1]: [IN]读卡器读卡超时时间设定高字节, 单位 ms

pSet[2]: [IN]读卡器读卡超时时间设定低字节, 单位 ms

总超时时间 ms= pSet[1]*256 + pSet[2]

返回: 成功返回 0

失败返回非 0

说明: bit1 和 bit2 功能和 bySetLg=3 功能需要读卡器支持才可以, 此函数 dll 需要 V28 及之后版本。

通用函数功能: 蜂鸣器控制

功能: 蜂鸣器控制

原型: int WINAPI rf_beep (unsigned short wlcdev, unsigned char byMsec)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byMsec: [IN]1 字节蜂鸣时限 0x00~0xFF, 单位是 10 毫秒

返回: 成功返回 0

失败返回非 0

通用函数功能：控制指示灯

功能：控制指示灯亮灭

原型：int WINAPI rf_light (unsigned short wlcdev, unsigned char byColor)

参数：wlcdev: [IN]通讯设备标识符，等于 0 或与读卡器设备号一致

byColor: [IN]1 字节控制信息

= 0x00 熄灭黄灯，对应硬件控制引脚输出高电平

= 0x01 熄灭绿灯，对应硬件控制引脚输出高电平

= 0x02 点亮绿灯，对应硬件控制引脚输出低电平

= 0x03 点亮黄灯，对应硬件控制引脚输出低电平

返回：成功返回 0

失败返回非 0

通用函数功能：读取读写卡器硬件版本号

功能：读取读写卡器硬件版本号

原型：int WINAPI rf_get_model (unsigned short wlcdev, unsigned char *pVer)

参数：wlcdev: [IN]通讯设备标识符，等于 0 或与读卡器设备号一致

pVer: [OUT]返回的信息

返回：成功返回 0

失败返回非 0

通用函数功能：读取读写卡器产品序列号

功能：读取读写卡器产品序列号

原型：int WINAPI rf_get_sn (unsigned short wlcdev, unsigned char *pSnr, int *pSnrLg)

参数：wlcdev: [IN]通讯设备标识符，等于 0 或与读卡器设备号一致

pSnr: [OUT]返回的产品序列号

pSnrLg: [OUT]返回长度

返回：成功返回 0

失败返回非 0

通用函数功能：设置读写卡器设备号

功能：设置读写卡器设备号，**该功能需要读卡器支持才可以**

原型：int WINAPI rf_init_device_number (unsigned short wlcdev,
unsigned short wlcdevNew)

参数：wlcdev: [IN]通讯设备标识符，等于 0 或与读卡器设备号一致

wlcdevNew: [IN]新的通讯设备标识符，0-65536

返回：成功返回 0

失败返回非 0

通用函数功能：获取读写卡器设备号

功能：读取读写卡器设备号

原型：int WINAPI rf_get_device_number (unsigned short wlcdev, unsigned short *wlcdevCur)

参数：wlcdev: [IN]通讯设备标识符，等于 0 或与读卡器设备号一致

wlcdevCur: [OUT]当前通讯设备标识符，0-65536

返回：成功返回 0

失败返回非 0

4.3 MifareOne 卡函数指令**MF1 卡专用功能：寻 ISO14443A 卡**

功能：寻 ISO14443 TYPE_A 卡

原型：int WINAPI rf_request (unsigned short wlcdev, unsigned char byMode,
unsigned short *pTagType)

参数：wlcdev: [IN]通讯设备标识符，等于 0 或与读卡器设备号一致

byMode: [IN]寻卡模式

=0x26 寻天线内未休眠卡片

=0x52 寻天线内所有卡片

pTagType: [OUT]卡返回值

返回：成功返回 0

失败返回非 0

MF1 卡专用功能：ISO14443A 卡防冲突

功能：14443 TYPE_A 卡防冲撞

原型：int WINAPI rf_anticoll (unsigned short wlcdev, unsigned char byCnt,
unsigned char *pSnr, unsigned char *pSnrLg)

参数：wlcdev: [IN]通讯设备标识符，等于 0 或与读卡器设备号一致

byCnt: [IN]值=4

pSnr: [OUT]返回卡号

pSnrLg: [OUT]返回卡号长度

返回：成功返回 0

失败返回非 0

MF1 卡专用功能：ISO14443A 卡选卡

功能：激活 14443 TYPE_A 卡

原型：int WINAPI rf_select (unsigned short wlcdev, unsigned char *pSnr,
unsigned char bySnrLg, unsigned char *pSize)

参数：wlcdev: [IN]通讯设备标识符，等于 0 或与读卡器设备号一致

pSnr: [IN]卡号

bySnrLg: [IN]卡号长度
pSize: [OUT]卡返回值

返回: 成功返回 0

失败返回非 0

说明: 当选卡指令执行完毕后, 卡片进入到激活状态, 同时在同样的天线区域, 仅有一张卡片处于激活状态。

MF1 卡专用功能: 验证 MifareOne 卡密钥

功能: 验证 MifareOne 卡密钥

原型: int WINAPI rf_M1_authentication2(unsigned short wlcdev, unsigned char byMode,
unsigned char byBlockNo, unsigned char *pKey)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byMode: [IN]密码验证模式
=0x60: 验证 A 密钥
=0x61: 验证 B 密钥

byBlockNo: [IN]要验证密码的绝对块号

pKey: [IN]密钥内容, 6 字节, 卡片出厂默认密钥 6 字节 0xFF

返回: 成功返回 0

失败返回非 0

MF1 卡专用功能: 读取 MifareOne 卡一块数据

功能: 读取 MifareOne 卡一块数据

原型: int WINAPI rf_M1_read(unsigned short wlcdev, unsigned char byBlockNo,
unsigned char *pData, unsigned char *pDataLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byBlockNo: [IN]M1 卡绝对块号
pData: [OUT]读出数据
pDataLg: [OUT]读出数据的长度

返回: 成功返回 0

失败返回非 0

MF1 卡专用功能: 写入 MifareOne 卡一块数据

功能: 向 Mifare One 卡写入 16 字节数据

原型: int WINAPI rf_M1_write(unsigned short wlcdev,
unsigned char byBlockNo,
unsigned char *pData)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byBlockNo: [IN]M1 卡绝对块号
pData: [IN]写入的数据, 16 字节

返回: 成功返回 0
失败返回非 0

MF1 卡专用功能: 将 Mifare One 卡某一块初始化为钱包

功能: 将 Mifare One 卡某一块初始化为钱包

原型: int WINAPI rf_M1_initval (unsigned short wlcdev, unsigned char byBlockNo, long lValue)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byBlockNo: [IN]M1 卡绝对块号

lValue: [IN]初始值, 10 进制

返回: 成功返回 0
失败返回非 0

MF1 卡专用功能: 读取 Mifare One 卡钱包值

功能: 读取 Mifare One 卡钱包值

原型: int WINAPI rf_M1_readval (unsigned short wlcdev, unsigned char byBlockNo, long *pValue)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byBlockNo: [IN]M1 卡绝对块号

pValue: [OUT]返回的值, 10 进制

返回: 成功返回 0
失败返回非 0

MF1 卡专用功能: Mifare One 卡扣款

功能: Mifare One 卡扣款

原型: int WINAPI rf_M1_decrement (unsigned short wlcdev,
unsigned char byBlockNo,
long lValue)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byBlockNo: [IN]M1 卡绝对块号

lValue: [IN]要扣减的值, 10 进制

返回: 成功返回 0
失败返回非 0

MF1 卡专用功能: Mifare One 卡充值

功能: Mifare One 卡充值

原型: int WINAPI rf_M1_increment (unsigned short wlcdev,
unsigned char byBlockNo,
long lValue)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byBlockNo: [IN]M1 卡绝对块号

lValue: [IN]要增加的值, 10 进制

返回: 成功返回 0

失败返回非 0

MF1 卡专用功能: Mifare One 卡数据回传

功能: Mifare One 卡数据回传到内部 BUFFER 中

原型: int WINAPI rf_M1_restore (unsigned short wlcdev, unsigned char bySrcBlockNo)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

bySrcBlockNo: [IN] M1 卡源块号, 与目标块号应在同一扇区

返回: 成功返回 0

失败返回非 0

说明: 用此函数将指定的块内容传入卡的 buffer, 然后可用 rf_M1transfer()函数将 buffer 中数据再传送到另一块中去

MF1 卡专用功能: Mifare One 卡数据传送

功能: 将 Mifare One 卡 BUFFER 中的块数据传入指定的块

原型: int WINAPI rf_M1_transfer (unsigned short wlcdev, unsigned char byDestBlockNo)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byDestBlockNo: [IN] M1 卡目标块号, 与源块号应在同一扇区

返回: 成功返回 0

失败返回非 0

说明: 该函数仅在 restore 命令之后调用

MF1 卡专用功能: Mifare One 卡读取连续 3 块数据

功能: 读取 MifareOne 卡连续 3 块数据

原型: int WINAPI rf_M1_read_sector (unsigned short wlcdev, unsigned char byStartBlockNo, unsigned char *pData, unsigned char *pDataLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byStartBlockNo: [IN] M1 卡绝对起始块号, 要求起始块号+2 之后不能超过同一扇区

pData: [OUT]返回的 48 字节数据

pDataLg: [OUT]返回数据的长度

返回: 成功返回 0

失败返回非 0

注意: 此功能需要读卡器支持才可以

MF1 卡专用功能: Mifare One 卡写连续 3 块数据

功能: 向 MifareOne 卡写入 48 字节数据, 连续 3 块数据

原型: int WINAPI rf_M1_write_sector (unsigned short wlcdev,
unsigned char byStartBlockNo,
unsigned char *pData)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致
byStartBlockNo: [IN] M1 卡起始绝对块号, 块号必须为同扇区的首块
pData: [IN] 写入的数据, 48 字节

返回: 成功返回 0
失败返回非 0

注意: 此功能需要读卡器支持才可以

MF1 卡专用功能: 命令已激活的 ISO14443A 卡进入 HALT 状态

功能: 命令已激活的 TYPE_A 卡进入 HALT 状态

原型: int WINAPI rf_halt (unsigned short wlcdev)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

返回: 成功返回 0
失败返回非 0

4.4 Ultralight/Ntag 卡函数指令

UL /Ntag 专用功能: 寻 ISO14443A 卡

功能: 寻 ISO14443 TYPE_A 卡

原型: int WINAPI rf_request (unsigned short wlcdev, unsigned char byMode,
unsigned short * pTagType)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致
byMode: [IN]寻卡模式
=0x26 寻天线内未休眠卡片
=0x52 寻天线内所有卡片

pTagType: [OUT]卡返回值

返回: 成功返回 0
失败返回非 0

UL/Ntag 专用功能: UltraLight/Ntag 选卡

功能: Ultralight/Ntag 选卡

原型: int WINAPI int rf_ul_select (unsigned short wlcdev,
unsigned char *pSnr, unsigned char *pSnrLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致
pSnr: [OUT]返回卡序列号
pSnrLg: [OUT]返回序列号的长度

返回: 成功返回 0

失败返回非 0

UL/Ntag 专用功能：读取 Ultralight/Ntag 卡 4 页数据

功能：读取 Ultralight/Ntag 卡连续 4 页数据，函数同 MF1 卡读块命令

原型：int WINAPI rf_M1_read (unsigned short wlcdev, unsigned char byBlockNo,
unsigned char *pData, unsigned char *pDataLg)

参数：wlcdev: [IN]通讯设备标识符，等于 0 或与读卡器设备号一致

byBlockNo: [IN]Ultralight/Ntag 起始页号

pData: [OUT]读出数据，从起始页开始连续 4 页数据

pDataLg: [OUT]读出数据的长度

返回：成功返回 0

失败返回非 0

UL/Ntag 专用功能：向 Ultralight/Ntag 卡中写入一页数据

功能：向 Ultralight/Ntag 卡中写入一页数据

原型：int WINAPI int rf_ul_write (unsigned short wlcdev, unsigned char byPage,
unsigned char *pData)

参数：wlcdev: [IN]通讯设备标识符，等于 0 或与读卡器设备号一致

byPage: [IN]ultralight/Ntag 卡页地址

pData: [IN]写入的数据，4 字节

返回：成功返回 0

失败返回非 0

Ntag21x 专用功能：Ntag21x 系列快速读取

功能：Ntag21x 系列快速读取

原型：int WINAPI rf_Ntag_Fast_Read (unsigned short wlcdev,
unsigned char byStartAdr, unsigned char byEndAdr,
unsigned char *pData, unsigned char *pDataLg)

参数：wlcdev: [IN]通讯设备标识符，等于 0 或与读卡器设备号一致

byStartAdr: [IN]起始页号

byEndAdr: [IN]结束页号，结束页号要大于等于起始页号，最大 15 页，每次 15*4=60 字节

pData: [OUT]返回数据

pDataLg: [OUT]返回长度

返回：成功返回 0

失败返回非 0

Ntag21x 专用功能：Ntag21x 系列获取版本号

功能：Ntag21x 系列获取版本号

原型: int WINAPI rf_Ntag_Get_Version (unsigned short wlcdev,
unsigned char *pData, unsigned char *pDataLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

pData: [OUT]返回数据

pDataLg: [OUT]返回长度

返回: 成功返回 0

失败返回非 0

Ntag21x 专用功能: Ntag21x 系列密钥认证

功能: Ntag21x 系列密钥认证

原型: int WINAPI rf_Ntag_Password_Auth (unsigned short wlcdev,
unsigned char *pPWD,
unsigned char *pData, unsigned char *pDataLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

pPWD: [IN]密码认证 (输入 4 字节密码)

pData: [OUT]返回数据 (2 字节 PACK)

pDataLg: [OUT]返回长度

返回: 成功返回 0

失败返回非 0

Ntag21x 专用功能: Ntag21x 系列读取计数值

功能: Ntag21x 系列读取计数值

原型: int WINAPI rf_Ntag_Read_CNT (unsigned short wlcdev,
unsigned char *pData, unsigned char *pDataLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

pData: [OUT]返回数据 (3 字节计数)

pDataLg: [OUT]返回长度

返回: 成功返回 0

失败返回非 0

Ntag21x 专用功能: Ntag21x 系列读取签名

功能: Ntag21x 系列读取签名

原型: int WINAPI rf_Ntag_Read_Signature(unsigned short wlcdev,
unsigned char *pData, unsigned char *pDataLg)

参数: wlcdev: [IN]通讯设备标识符

pData: [OUT]返回数据

pDataLg: [OUT]返回长度

返回: 成功返回 0

失败返回非 0

NFC 卡专用功能: NFC 类型 2 标签选择扇区

功能: NFC 类型 2 标签选择扇区

原型: int WINAPI rf_NFC_Select_Sector (unsigned short wlcdev, unsigned char bySec)

参数: wlcdev: [IN]通讯设备标识符

bySec: [IN]输入扇区编号(00-FEh)

返回: 成功返回 0

失败返回非 0

UL/Ntag 专用功能: 命令已激活的 ISO14443A 卡进入 HALT 状态

功能: 命令已激活的 TYPE_A 卡进入 HALT 状态

原型: int WINAPI rf_halt (unsigned short wlcdev)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

返回: 成功返回 0

失败返回非 0

4.5 ISO14443A-4 CPU 卡函数指令

此为 CPU 卡通用函数, 适合满足 ISO14443A-4 标准的 CPU 卡, 比如:FM1208,Desfire 等

TYPE A CPU 卡专用功能: 符合 ISO14443A-4 标准的非接触 CPU 卡复位

功能: 寻感应区内符合 ISO14443A-4 标准的 CPU 卡并复位

原型: int WINAPI rf_typea_rst (unsigned short wlcdev, unsigned char byMode,
unsigned char *pData, unsigned char *pDataLg)

参数: wlcdev: [IN] 通讯设备标识符, 等于 0 或与读卡器设备号一致

byMode: [IN] 寻卡方式

=0x26 寻天线内未休眠卡片

=0x52 寻天线内所有卡片

pDate: [OUT] 返回 4/7/10 字节 CSN + 复位信息内容

pDataLg: [OUT] 返回数据的长度

返回: 成功返回 0

失败返回非 0

TYPE A CPU 卡专用功能: 向符合 ISO14443A-4 CPU 卡发送 COS 命令

功能: 向符合 ISO14443A-4 CPU 卡发送 COS 命令

原型: int WINAPI rf_cos_command (unsigned short wlcdev,
unsigned char *pCmd, unsigned int nCmdLg,
unsigned char *pData, unsigned int *pDataLg)

参数: wlcdev: [IN] 通讯设备标识符, 等于 0 或与读卡器设备号一致
 pCmd: [IN] COS 命令
 nCmdLg: [IN] COS 命令长度
 pDate: [OUT] 卡片返回的数据, 含 SW1、SW2
 pDataLg: [OUT] 返回数据长度
 返回: 成功则返回 0
 失败返回非 0

TYPE A CPU 卡专用功能: 向符合 ISO14443A-4 CPU 卡发送 PPS 请求

功能: 向符合 ISO14443A-4 CPU 卡发送 PPS 请求

注意: 此功能需要读卡器支持

原型: int WINAPI rf_ISO14443_PPS (unsigned short wlcdev,
 unsigned char byCID,
 unsigned char byMode)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致
 byCID: [IN]cCID=0
 byMode: [IN] byMode =0x00: 106K
 byMode =0x01: 212K
 byMode =0x02: 424K
 byMode =0x03: 848K

返回: 成功返回 0
 失败返回非 0

TYPE A CPU 卡专用功能: ISO14443A-4 CPU 卡退出激活状态

功能: ISO14443A-4 CPU 卡退出激活状态

原型: int WINAPI rf_cl_deselect (unsigned short wlcdev)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

返回: 成功返回 0
 失败返回非 0

4.6 ISO7816-4 SAM 卡函数指令 (支持多个卡)

新的用户建议采用此操作流程, 可以扩展对多个 PSAM 卡片的支持。

ISO7816-4 SAM/SIM 卡专用功能: 第 N 号卡复位(支持多卡)

功能: 第 N 号 ISO7816-4 CPU/SAM/SIM 卡复位

原型: int WINAPI rf_sam_rst_new (unsigned short wlcdev,
 unsigned char *pSet, unsigned char bySetLg,
 unsigned char *pData, unsigned char *pDataLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

pSet: [IN] bit1-bit0 波特率选择:

00: 表示复位 9600 的卡;

01: 表示复位 38400 的卡;

10 : 表示复位 115200 的卡;

bit3-bit2 改变选中卡状态:

00: 表示复位操作;

bit7-bit6-bit5-bit4 选择 SAM 卡:

0000: 表示 1 号卡;

0001: 表示 2 号卡;

0010: 表示 3 号卡;

0011: 表示 4 号卡;

0100: 表示 5 号卡;

bySetLg: [IN] 取值为 1

pDate: [OUT]返回的复位信息

pDataLg: [OUT]返回复位信息的长度

返回: 成功返回 0

失败返回非 0

ISO7816-4 SAM/SIM 卡专用功能: 向第 N 号卡发送 APDU(COS)指令(支持多卡)

功能: ISO7816-4 CPU/SAM/SIM 卡发送 APDU(COS)指令

原型: int WINAPI rf_sam_cos_new (unsigned short wlcdev, unsigned char bySAMNo,
 unsigned char *pCmd, unsigned int nCmdLg,
 unsigned char *pData, unsigned int *pDataLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

ySAMNo: [IN] SAM 卡编号

pCmd: [IN] COS 命令

nCmdLg: [IN] COS 命令长度

pDate: [OUT] 卡片返回的数据, 含 SW1、SW2

pDataLg: [OUT] 返回数据长度

返回: 成功返回 0

错误返回非 0

ISO7816-4 SAM/SIM 卡专用功能: 向第 N 号卡发送 PPS 请求(支持多卡)

功能: 向第 N 号 ISO7816-4 CPU/SAM/SIM 卡发送 PPS 请求指令

原型: int WINAPI rf_sam_rst_new (unsigned short wlcdev,
 unsigned char *pSet, unsigned char bySetLg,
 unsigned char *pData, unsigned char *pDataLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

当 bySetLg =1 时:

pSet [0]: [IN] bit1-bit0 波特率选择:

XX: 备用

bit3-bit2 改变选中卡状态:

10: 表示升频到 38400 (cSetLen=1)

bit7-bit6-bit5-bit4 选择 SAM 卡:

0000: 表示 1 号卡;

0001: 表示 2 号卡;

0010: 表示 3 号卡;

0011: 表示 4 号卡;

0100: 表示 5 号卡;

当 bySetLg =3 时:

pSet [0]: [IN] bit1-bit0 波特率选择:

XX: 备用

bit3-bit2 改变选中卡状态:

11: 表示升频到 38400 或者 115200 (cSetLen=3)

bit7-bit6-bit5-bit4 选择 SAM 卡:

0000: 表示 1 号卡;

0001: 表示 2 号卡;

0010: 表示 3 号卡;

0011: 表示 4 号卡;

0100: 表示 5 号卡;

pSet [1]: [IN] =0x10 (固定值 cSetLen=3);

pSet [2]: [IN] =0x13 PPS 到 38400 (cSetLen=3);

=0x18 PPS 到 115200 (cSetLen=3);

bySetLg: [IN] =1 or =3;

pDate: [OUT]返回的复位信息

pDataLg: [OUT]返回复位信息的长度

返回: 成功返回 0

失败返回非 0

4.7 ISO7816-4 SAM 卡函数指令 (只支持 1 个卡)

此操作流程是为了兼容老的型号读卡器, 只能支持一个 PSAM 卡, 对应读卡器内部的 1 号卡座位置

ISO7816-4 SAM/SIM 卡专用功能: 初始化卡片波特率 (只支持 1 个卡)

功能: 设置读写卡器 SAM 卡通讯波特率

原型: int WINAPI rf_init_sam (unsigned short wlcdev, unsigned char byBaud)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byBaud: [IN]SAM 卡波特率

=0x00: SAM 卡通讯波特率为 9600
 =0x01: SAM 卡通讯波特率为 38400
 =0x02: SAM 卡通讯波特率为 115200

返回: 成功返回 0
 错误返回非 0

ISO7816-4 SAM/SIM 卡专用功能: 复位卡片 (只支持 1 个卡)

功能: ISO7816-4 CPU/SAM/SIM 卡复位

原型: int WINAPI rf_sam_rst (unsigned short wlcdev,
 unsigned char *pData, unsigned char *pDataLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致
 pData: [OUT]返回的复位信息
 pDataLg: [OUT]返回复位信息的长度

返回: 成功返回 0
 失败返回非 0

ISO7816-4 SAM/SIM 卡专用功能: 发送 APDU(COS)指令 (只支持 1 个卡)

功能: ISO7816-4 CPU/SAM/SIM 卡发送 APDU(COS)指令

原型: int WINAPI rf_sam_cos (unsigned short wlcdev,
 unsigned char *pCmd, unsigned char byCmdLg,
 unsigned char *pData, unsigned char *pDataLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致
 pCmd: [IN]COS 命令
 byCmdLg: [IN]COS 命令长度
 pData: [OUT]卡片返回的数据, 含 SW1、SW2
 pDataLg: [OUT]返回数据长度

返回: 成功返回 0
 错误返回非 0

4.8 ISO14443B-4 CPU 卡指令

此为 CPU 卡通用函数, 适合满足 ISO14443B-4 标准的 CPU 卡

TYPE B CPU 卡专用功能: 符合 ISO14443B-4 CPU 卡复位

功能: 寻感应区内符合 ISO14443B-4 标准的 CPU 卡并复位

原型: int WINAPI rf_atqb(WORD wlcdev,unsigned char byMode,
 unsigned char *pData,unsigned char *pDataLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致
 byMode: [IN]寻卡方式

bit0~bit3=0000: 寻未进入休眠状态的卡
 bit0~bit3=0001: 寻所有状态的卡
 bit4~bit7=0000: 106K
 bit4~bit7=0001: 212K (PPS), 需读卡器支持
 bit4~bit7=0010: 424K (PPS), 需读卡器支持
 bit4~bit7=0011: 848K (PPS), 需读卡器支持

pDate: [OUT]返回的数据

pDataLg: [OUT]返回数据的长度

返回: 成功返回 0

失败返回非 0

TYPE B CPU 卡专用功能: 向符合 ISO14443B-4 CPU 卡发送 COS 命令

功能: 向符合 ISO14443B-4 标准的非接触 CPU 卡发送 COS 命令

原型: int WINAPI rf_cos_command (WORD wlcdev,

unsigned char *pCmd, unsigned int nCmdLg,

unsigned char *pData, unsigned int *pDataLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

pCmd: [IN]COS 命令

nCmdLg: [IN]COS 命令长度

pDate: [OUT]卡片返回的数据, 含 SW1、SW2

pDataLg: [OUT]返回数据长度

返回: 成功返回 0

失败返回非 0

TYPE B CPU 卡专用功能: 向符合 ISO14443B-4 CPU 卡退出激活状态

功能: ISO14443B-4 CPU 卡退出激活状态

原型: int WINAPI rf_cl_deselect (unsigned short icdev)

参数: icdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

返回: 成功返回 0

失败返回非 0

TYPE B CPU 卡专用功能: ISO14443 TYPE B 卡休眠

功能: ISO14443 TYPE B 卡休眠, 此函数需 DLL 版本大于等于 V28。

原型: int WINAPI rf_haltb (unsigned short wlcdev,

unsigned char *pPUPI, unsigned char byPUPILg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

pPUPI: [IN] 4 字节 PUPI

byPUPILg: [IN] PUPI 字节长度

返回: 成功返回 0

失败返回非 0

4.9 ISO14443B ST 卡指令

ST 卡专用功能: 寻卡

功能: 用于 ISO14443B SR176/SRI512/SRIX4K 卡的寻卡

原型: int WINAPI rf_st_select (unsigned short wlcdev, unsigned char *pChipID)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

pChipID: [OUT]返回卡 ID 号, 一字节

返回: 成功则返回 0

失败返回非 0

说明: 返回的 ID 号是随机的, 同一张卡每次返回值不一定相同

ST 卡只有锁定之后才能够对其进行其它操作。

ST 卡专用功能: ST 卡获取 UID

功能: 用于 ISO14443B SRI512/SRIX4K 卡的读取 UID 操作

原型: int WINAPI rf_srix4k_getuid (unsigned short wlcdev,
unsigned char *pUID, unsigned char *pUIdLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

pUID: [OUT]返回 8 字节 UID 卡号

pUIdLg: [OUT]返回 UID 长度

返回: 成功返回 0

失败返回非 0

ST 卡专用功能: ST 卡读块

功能: 用于 ISO14443B SRI512/SRIX4K 卡的读块操作

原型: int WINAPI rf_srix4k_readblock (unsigned short wlcdev, unsigned char byBlockNo,
unsigned char *pData, unsigned char *pDataLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byBlockNo: [IN]块地址

pData: [OUT]读出 4 字节数据

pDataLg: [OUT]读出数据的长度

返回: 成功则返回 0

失败返回非 0

ST 卡专用功能: ST 卡写块

功能: 用于 ISO14443B SRI512/SRIX4K 卡的写块操作

原型: int WINAPI rf_srix4k_writeblock (unsigned short wlcdev,

unsigned char byBlockNo,
unsigned char *pData)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byBlockNo: [IN]块地址

pData: [IN]要写入的数据, 4 字节

返回: 成功返回 0

失败返回非 0

ST 卡专用指令: SRI512 卡块锁定

功能: SRI512 卡块锁定

原型: int WINAPI rf_sri512_writelockreg (unsigned short wlcdev, unsigned char *pLockReg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

pLockReg: [IN]4 字节数据=2 字节 lockreg+1 字节 00+1 字节 0x00

返回: 成功返回 0

失败返回非 0

说明: (1) SRI512 卡 2 字节 lockreg

Lockreg	块号	位含义
b15	15	1:未锁定; 0:锁定
b14	14	1:未锁定; 0:锁定
...
...
b1	1	1:未锁定; 0:锁定
b0	0	1:未锁定; 0:锁定

(2) Lockreg 不可以被擦除, 每个位只能从 1 变 0, 不能从 0 变 1, 一旦写入, 对应的块就会被锁死, 只能读取, 不能再被写入, 为了使设置有效, 发送锁块命令之后, 需要再发送寻卡指令, 确保锁定有效。

ST 卡专用指令: SRI4K 卡块锁定

功能: SRI4K 卡块锁定

原型: int WINAPI rf_srix4k_writelockreg (unsigned short wlcdev, unsigned char byLockReg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byLockReg: [IN] 1 字节 lockreg

返回: 成功返回 0

失败返回非 0

说明: SRI4K 卡 1 字节 lockreg

Lockreg	块号	位含义
b7	15	1:未锁定; 0:锁定
b6	14	1:未锁定; 0:锁定

b5	13	1:未锁定; 0:锁定
b4	12	1:未锁定; 0:锁定
b3	11	1:未锁定; 0:锁定
b2	10	1:未锁定; 0:锁定
b1	9	1:未锁定; 0:锁定
b0	7 & 8	1:未锁定; 0:锁定

(2) lockreg 不可以被擦除, 每个位只能从 1 变 0, 不能从 0 变 1, 一旦写入, 对应的块就会被锁死, 只能读取, 不能再被写入, 为了使设置有效, 发送锁块命令之后, 需要再发送寻卡指令, 确保锁定有效。

ST 卡专用指令: ST 卡退出选择

功能: 用于 ISO14443B SR176/SRI512/SRIX4K 卡退出选择。

原型: int WINAPI rf_st_completion (unsigned short wlcdev)

参数: wlcdev: [IN] 通讯设备标识符, 等于 0 或与读卡器设备号一致

返回: 成功返回 0

失败返回非 0

4.10 二代证卡指令

二代证专用功能: 读取二代证的 UID 卡号

功能: 读取二代证 UID 卡号

原型: int WINAPI rf_get_namecard_uid (unsigned short wlcdev, unsigned char byMode, unsigned char *pData, int *pDataLg)

参数: wlcdev: [IN] 通讯设备标识符, 等于 0 或与读卡器设备号一致

byMode: [IN] Mode 固定为 0

pData: [OUT] 返回卡片 8 字节序列号

pDataLg: [OUT] 返回卡号长度

返回: 成功返回 0

失败返回非 0

4.11 Felica 卡指令

Felica 专用功能: 读取 Felica 的 UID 卡号

功能: 寻 Felica 卡 UID 卡号

原型: int WINAPI rf_felica_get_uid (unsigned short wlcdev, unsigned char *pUID, int *pUidLg)

参数: wlcdev: [IN] 通讯设备标识符, 等于 0 或与读卡器设备号一致

pUID: [OUT] 返回卡片 6 字节序列号

pUidLg: [OUT] 返回卡号长度

返回: 成功返回 0

失败返回非 0

4.12 ISO15693 卡指令

ISO15693 卡专用指令: ISO15693 卡寻卡

功能: 用于 ISO15693 卡的寻卡。

原型: int WINAPI ISO15693_Inventory (unsigned short wlcdev,
unsigned char *pData, unsigned char *pDataLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

pData: [OUT] 返回数据=1 字节 DSFID+8 字节 UID

pDataLg: [OUT] 返回数据长度

返回: 成功则返回 0

失败则返回非 0

ISO15693 卡专用指令: ISO15693 卡获取卡片系统信息

功能: 用于 ISO15693 卡片获取卡片系统信息。

原型: int WINAPI ISO15693_Get_System_Information (unsigned short wlcdev,
unsigned char byMode,
unsigned char *pUID,
unsigned char *pData,
unsigned char *pDataLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byMode: [IN] byMode 信息=0x00

pUID: [IN] UID 8 字节=0x00

pData: [OUT] 返回数据=1 字节信息标志+8 字节 UID+1 字节 DSFID (如果存在)
+1 字节 AFI (如果存在) +2 字节卡片信息 (如果存在) +1 字节厂商信息 (如果存在)

1 字节信息标志:

Bit0 = 0 (DSFID 不存在); =1 (DSFID 存在)

Bit1 = 0 (AFI 不存在); =1 (AFI 存在)

Bit2 = 0 (卡片信息不存在); =1 (卡片信息存在)

Bit3 = 0 (厂商信息不存在); =1 (厂商信息存在)

Bit7-Bit4: 备用

pDataLg: [OUT] 返回数据长度

返回: 成功则返回 0

失败则返回非 0

ISO15693 卡专用指令: ISO15693 卡读块

功能: 用于 ISO15693 卡读块操作。

原型: int WINAPI ISO15693_Read (unsigned short wlcdev,
 unsigned char byMode,
 unsigned char *pUID,
 unsigned char byBlockNo,
 unsigned char byNumber,
 unsigned char *pData,
 unsigned char *pDataLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byMode: [IN] byMode 信息
 bit0=1: 只有处于 SELECT 状态的卡执行该命令
 bit1=1: 只有 UID 相同的卡执行该命令
 bit2=0: 每块返回 4 字节数据
 =1: 每块返回 1 字节安全状态 + 4 字节数据

pUID: [IN] UID 8 字节

byBlockNo: [IN] 起始块号

byNumber: [IN] 要写入的块的个数, 取值 <=0x0F

pData: [OUT] (byNumber * 4) 或者 (byNumber * 5) 字节数据

pDataLg: [OUT] 返回数据长度

返回: 成功则返回 0

失败则返回非 0

ISO15693 卡专用指令: ISO15693 卡写块

功能: 用于 ISO15693 卡写块操作。

原型: int WINAPI ISO15693_Write (unsigned short wlcdev,
 unsigned char byMode,
 unsigned char *pUID,
 unsigned char byBlockNo,
 unsigned char *pData)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byMode: [IN] byMode 信息
 bit0=1: 只有处于 SELECT 状态的卡执行该命令
 bit1=1: 只有 UID 相同的卡执行该命令
 bit2=0: 操作 NXP 卡
 =1: 操作 TI 卡

pUID: [IN] UID 8 字节

byBlockNo: [IN] 要写入的块号

pData: [IN] 4 字节数据

返回: 成功则返回 0

失败则返回非 0

ISO15693 卡专用指令：ISO15693 卡写连续多块

功能：用于 ISO15693_写多个连续块操作。

原型：int WINAPI ISO15693_Write_Multi_Block (unsigned short wlcdev,
unsigned char byMode,
unsigned char *pUID,
unsigned char byStartBlockNo,
unsigned char byNumber,
unsigned char *pData)

参数： wlcdev: [IN]通讯设备标识符，等于 0 或与读卡器设备号一致
byMode: [IN] byMode 信息
bit0=1: 只有处于 SELECT 状态的卡执行该命令
bit1=1: 只有 UID 相同的卡执行该命令
bit2=0: 操作 NXP 卡
=1: 操作 TI 卡
pUID: [IN] UID 8 字节
byStartBlockNo: [IN] 起始块号
byNumber: [IN] 要写入的块的个数，取值 <0x0A
pData: [IN] (byNumber * 4) 字节数据

返回：成功则返回 0
失败则返回非 0

备注：此功能需要读卡器支持才可以

ISO15693 卡专用指令：ISO15693 卡锁块

功能：用于 ISO15693 卡锁块操作。

原型：int WINAPI ISO15693_Lock_Block (unsigned short wlcdev, unsigned char byMode,
unsigned char *pUID, unsigned char byBlockNo)

参数： wlcdev: [IN]通讯设备标识符，等于 0 或与读卡器设备号一致
byMode: [IN] byMode 信息
bit0=1: 只有处于 SELECT 状态的卡执行该命令
bit1=1: 只有 UID 相同的卡执行该命令
bit2=0: 操作 NXP 卡
=1: 操作 TI 卡

pUID: [IN] UID 8 字节
byBlockNo: [IN] 要锁定的块号
返回：成功则返回 0
失败则返回非 0

ISO15693 卡专用指令：获取 ISO15693 卡块锁定信息

功能：用于获取 ISO15693 卡块锁定信息。

原型: int WINAPI ISO15693_Get_Block_Security (unsigned short wlcdev,
unsigned char byMode,
unsigned char *pUID,
unsigned char byBlockNo,
unsigned char byNumber,
unsigned char *pData,
unsigned char *pDataLg)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byMode: [IN] byMode 信息

bit0=1: 只有处于 SELECT 状态的卡执行该命令

bit1=1: 只有 UID 相同的卡执行该命令

bit2=0: 操作 NXP 卡

=1: 操作 TI 卡

pUID: [IN] UID 8 字节

byBlockNo: [IN] 起始块号

byNumber: [IN] 要获取的块安全状态的个数, 取值<0x40

pData: [OUT] 返回的数据, 每个字节代表一个块状态, 等于 1 块被锁定, 等于 0 未锁定

pDataLg: [OUT] 返回数据的长度

返回: 成功则返回 0

失败则返回非 0

ISO15693 卡专用指令: ISO15693 卡写 AFI

功能: 用于 ISO15693 卡写 AFI 操作

原型: int WINAPI ISO15693_Write_AFI (unsigned short wlcdev, unsigned char byMode,
unsigned char *pUID, unsigned char byAFI)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致

byMode: [IN] byMode 信息

bit0=1: 只有处于 SELECT 状态的卡执行该命令

bit1=1: 只有 UID 相同的卡执行该命令

bit2=0: 操作 NXP 卡

=1: 操作 TI 卡

pUID: [IN] UID 8 字节

byAFI: [IN] 1 字节 AFI

返回: 成功则返回 0

失败则返回非 0

ISO15693 卡专用指令: ISO15693 卡锁定 AFI

功能: 用于 ISO15693 卡锁定 AFI 操作

原型: int WINAPI ISO15693_Lock_AFI(unsigned short wlcdev, unsigned char byMode,

unsigned char *pUID)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致
byMode: [IN] byMode 信息
bit0=1: 只有处于 SELECT 状态的卡执行该命令
bit1=1: 只有 UID 相同的卡执行该命令
bit2=0: 操作 NXP 卡
=1: 操作 TI 卡
pUID: [IN] UID 8 字节
返回: 成功则返回 0
失败则返回非 0

ISO15693 卡专用指令: ISO15693 卡写 DSFID

功能: 用于 ISO15693 卡写 DSFID 操作

原型: int WINAPI ISO15693_Write_DSFIID (unsigned short wlcdev, unsigned char byMode, unsigned char *pUID, unsigned char byDSFIID)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致
byMode: [IN] byMode 信息
bit0=1: 只有处于 SELECT 状态的卡执行该命令
bit1=1: 只有 UID 相同的卡执行该命令
bit2=0: 操作 NXP 卡
=1: 操作 TI 卡
pUID: [IN] UID 8 字节
byDSFIID: [IN] 1 字节 DSFIID
返回: 成功则返回 0
失败则返回非 0

ISO15693 卡专用指令: ISO15693 卡锁定 DSFIID

功能: 用于 ISO15693 卡锁定 DSFIID 操作

原型: int WINAPI ISO15693_Lock_DSFIID (unsigned short wlcdev, unsigned char byMode, unsigned char *pUID)

参数: wlcdev: [IN]通讯设备标识符, 等于 0 或与读卡器设备号一致
byMode: [IN] byMode 信息
bit0=1: 只有处于 SELECT 状态的卡执行该命令
bit1=1: 只有 UID 相同的卡执行该命令
bit2=0: 操作 NXP 卡
=1: 操作 TI 卡
pUID: [IN] UID 8 字节
返回: 成功则返回 0
失败则返回非 0

ISO15693 卡专用指令: ISO15693 卡选卡

功能: 用于 ISO15693 卡选卡操作

原型: int WINAPI ISO15693_Select (unsigned short wlcdev, unsigned char *pUID)

参数: wlcdev: [IN] 通讯设备标识符, 等于 0 或与读卡器设备号一致

pUID: [IN] UID 8 字节

返回: 成功则返回 0

失败则返回非 0

ISO15693 卡专用指令: ISO15693 卡复位

功能: 用于 ISO15693 卡复位操作

原型: int WINAPI ISO15693_Reset_To_Ready (unsigned short wlcdev,
unsigned char byMode,
unsigned char *pUID)

参数: wlcdev: [IN] 通讯设备标识符, 等于 0 或与读卡器设备号一致

byMode: [IN] byMode 信息

bit0=1: 只有处于 SELECT 状态的卡执行该命令

bit1=1: 只有 UID 相同的卡执行该命令

bit2=0: 操作 NXP 卡

=1: 操作 TI 卡

pUID: [IN] UID 8 字节

返回: 成功则返回 0

失败则返回非 0

ISO15693 卡专用指令: ISO15693 卡进入静默

功能: 用于 ISO15693 卡进入静默状态操作。

原型: int WINAPI ISO15693_Stay_Quiet (unsigned short wlcdev, unsigned char *pUID)

参数: wlcdev: [IN] 通讯设备标识符, 等于 0 或与读卡器设备号一致

pUID: [IN] UID 8 字节

返回: 成功则返回 0

失败则返回非 0

ISO15693 卡专用指令: ISO15693 卡寻多张卡

功能: ISO15693_Inventorys(多张卡)

原型: int WINAPI ISO15693_Inventorys (unsigned short wlcdev, unsigned char *pData,
unsigned char *pDataLg)

参数: wlcdev: [IN] 通讯设备标识符, 等于 0 或与读卡器设备号一致

pData: [OUT] 返回数据: 1 字节卡片发生冲突时隙的计数 + 1 字节卡片接收错误计数+

1 字节接收正确卡片计数 + ((1 字节 DSFID + 8 字节 UID) * 正确卡片计数) 组字节卡号

pDataLg: [OUT] 返回信息长度

返回: 成功则返回 0

失败则返回非 0

ISO15693-NXP 卡专用指令: ISO15693-NXP 卡设置 EAS

功能: 用于 ISO15693-NXP 卡设置 EAS 操作。

原型: int ISO15693_SET_EAS_ICODE (unsigned short wlcdev, unsigned char byMode, unsigned char *pUID)

参数: wlcdev: [IN] 通讯设备标识符, 等于 0 或与读卡器设备号一致

byMode: [IN] byMode 信息

bit0=1: 只有处于 SELECT 状态的卡执行该命令

bit1=1: 只有 UID 相同的卡执行该命令

bit2=0: 操作 NXP 卡

pUID: [IN] UID 8 字节

返回: 成功则返回 0

失败则返回非 0

ISO15693-NXP 卡专用指令: ISO15693-NXP 卡 EAS ALARM

功能: 用于 ISO15693-NXP 卡 EAS ALARM 操作。

原型: int ISO15693_EAS_ALARM_ICODE (unsigned short wlcdev, unsigned char byMode, unsigned char * pUID, unsigned char *pData, unsigned char *pDataLg);

参数: wlcdev: [IN] 通讯设备标识符, 等于 0 或与读卡器设备号一致

byMode: [IN] byMode 信息

bit0=1: 只有处于 SELECT 状态的卡执行该命令

bit1=1: 只有 UID 相同的卡执行该命令

bit2=0: 操作 NXP 卡

pUID: [IN] UID 8 字节

pData: [OUT] 32 字节 alarm 信息

pDataLg: [OUT] 返回信息长度

返回: 成功则返回 0

失败则返回非 0

ISO15693-NXP 卡专用指令：ISO15693-NXP 卡复位 EAS

功能：用于 ISO15693-NXP 卡复位 EAS 操作。

原型：int ISO15693_RESET_EAS_ICODE (unsigned short wlcdev, unsigned char byMode,
unsigned char *pUID);

参数：wlcdev: [IN]通讯设备标识符，等于 0 或与读卡器设备号一致

byMode: [IN] byMode 信息

bit0=1: 只有处于 SELECT 状态的卡执行该命令

bit1=1: 只有 UID 相同的卡执行该命令

bit2=0: 操作 NXP 卡

pUID: [IN] UID 8 字节

返回：成功则返回 0

失败则返回非 0

ISO15693-NXP 卡专用指令：ISO15693-NXP 卡锁定 EAS

功能：用于 ISO15693-NXP 卡锁定 EAS 操作。

原型：int ISO15693_LOCK_EAS_ICODE (unsigned short wlcdev, unsigned char byMode,
unsigned char *pUID)

参数：wlcdev: [IN]通讯设备标识符，等于 0 或与读卡器设备号一致

byMode: [IN] byMode 信息

bit0=1: 只有处于 SELECT 状态的卡执行该命令

bit1=1: 只有 UID 相同的卡执行该命令

bit2=0: 操作 NXP 卡

pUID: [IN] UID 8 字节

返回：成功则返回 0

失败则返回非 0