

YHY632F非接触式IC卡读写器

参考手册

北京易火眼科技有限公司



超轻、超薄、超流线型，黑色的机身，典雅端庄；

属性：公开

1.1 简介

YHY632 系列射频卡读写器是使用 NXP 生产的 CLRC632 系列射频基站的非接触 IC 卡读写器，可以读写符合 ISO14443A、ISO14443B 和 15693 标准的非接触 IC 卡，读写器和射频卡之间的数据传输采用了先进的加密算法，支持卡和设备双向验证功能；全面的通信错误自动侦测功能。防冲突，可同时处理多张射频卡。具有完整的读写卡功能，可以读写卡的块，钱包充值、扣款等。

产品型号和名称:

产品型号: 产品名称:

YHY632FU 13.56MHz 多协议非接触式 IC 卡读写器 USB 接口

YHY632FR 13.56MHz 多协议非接触式 IC 卡读写器 RS232 接口

1.2 支持的卡

- ▲.ISO 14443 A: Mifare_One(S50&S70)/UltraLight/Mifare_ProX
- ▲.ISO 14443B: AT88RF020/SR176/SRIX4K
- ▲.ISO 15693:Tag_it (2k) HF-1/I.CODE SLI 等

1.3、 产品特性

- 非接触式智能卡的读和写
- 可以感应 RFID 标签并用声光提示
- 射频频率: 13.56 MHz.
- 典型读写卡时间 <35ms
- 通信接口: RS232 或 USB, 波特率最高: 115200 bps
- 工作电压: DC 5V
- 红、绿 LED 显示 (可软件控制)
- 蜂鸣器状态报警提示功能

2、 应用

- 1、公司考勤，出入管理，身份识别；
- 2、医疗以及防伪，智能卡工厂；
- 3、一卡通、储值卡、网吧、会所等会员认证等。

3、与主控制器通讯

此设备和主控制器的通信是通过 USB 口实现的，主要是在主控制器的控制下来进行工作的。也就是说，此控制器所有的动作都来自于主控制器。由主控制器来发送显示、操作读卡器的指令，前端控制器在收到这些指令后，完成相应的功能，并给出处理的结果。

4、技术指标

电气特性

Parameter	Min	Typ	Max	Units
工作电源	4. 5	5. 0	5. 5	V
工作电流	30	80	120	mA
工作温度	0	25	+50	℃
存储温度	-10	25	+75	℃

- 频率：13.56MHz
- 读卡距离：0—9CM
- 波特率：9600 ~ 115200 bps ，默认115200
- 尺寸：95 ×70× 18 (mm)
- 重 量：82g (含USB线)
152g (含RS232线)

6、操作

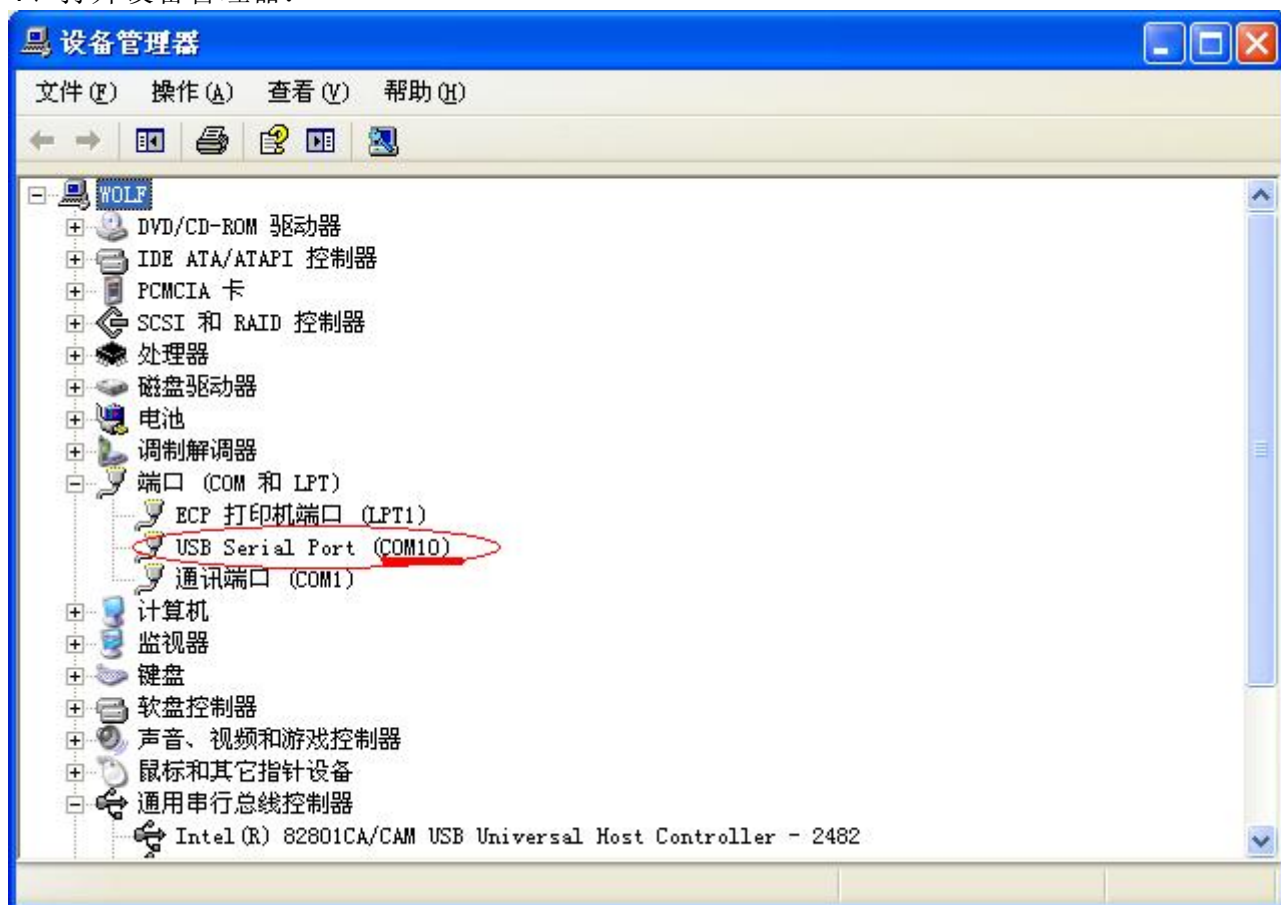
YHY632读写器采用USB Bridge to COM方式，将YHY632连接到PC的USB接口，安装驱动之后，虚拟出一个COM口。 可在“设备管理器”中查看虚拟出的COM号，如下图



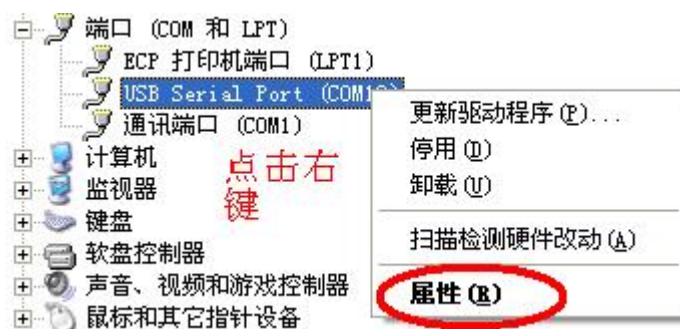
读写器加电后先自检，自检通过后红灯和绿灯会先后闪一下，然后蜂鸣器响2声，红灯（电源）会一直亮着，通过主控制器可以关闭它，绿灯也可以通过发送指令开关；接着系统开始进入工作状态，可以接收主控制器的指令。

如果USB模拟的串口号超过8个时可以参考下面的系列图示操作：

1、打开设备管理器：



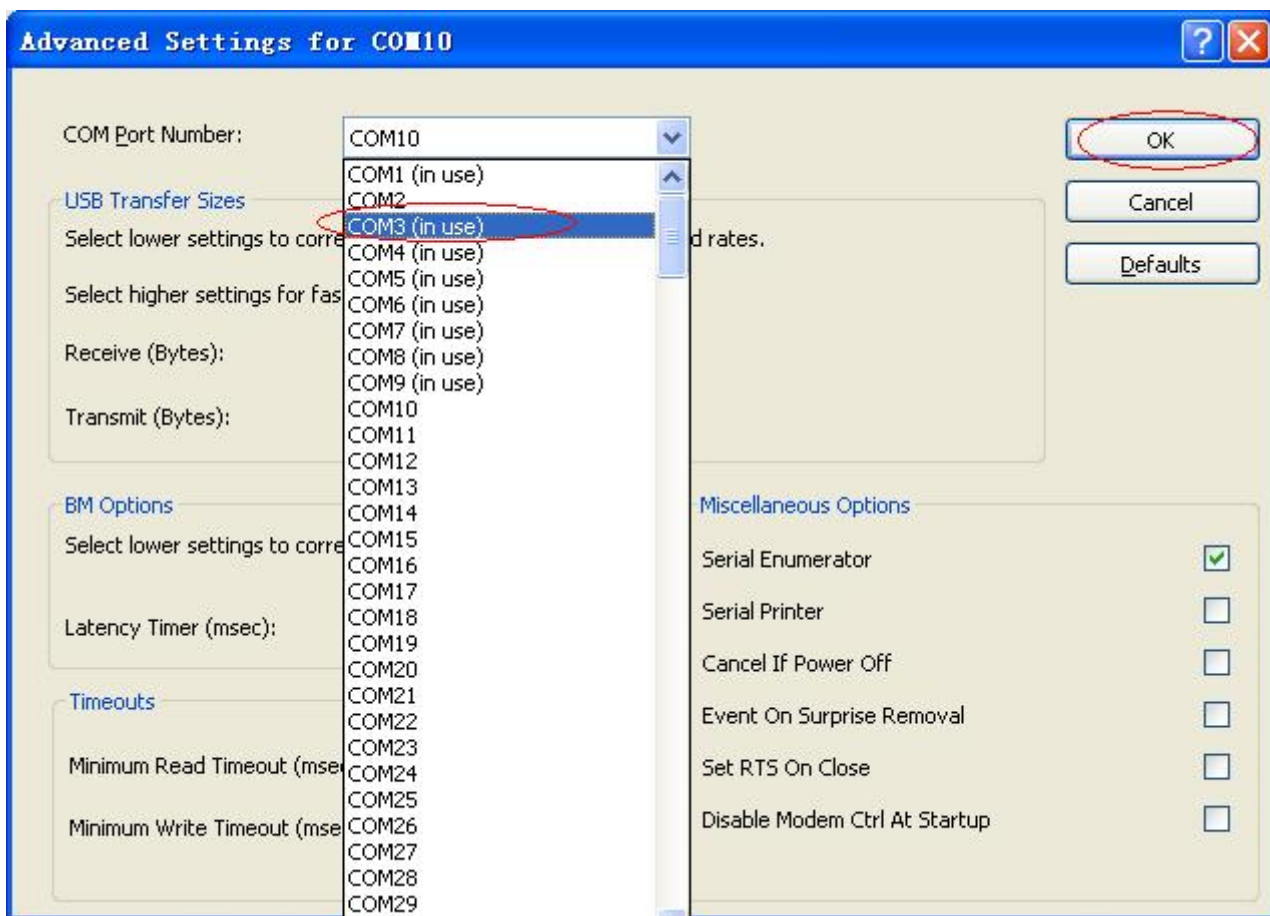
2、如下图，点击右键后选择属性：



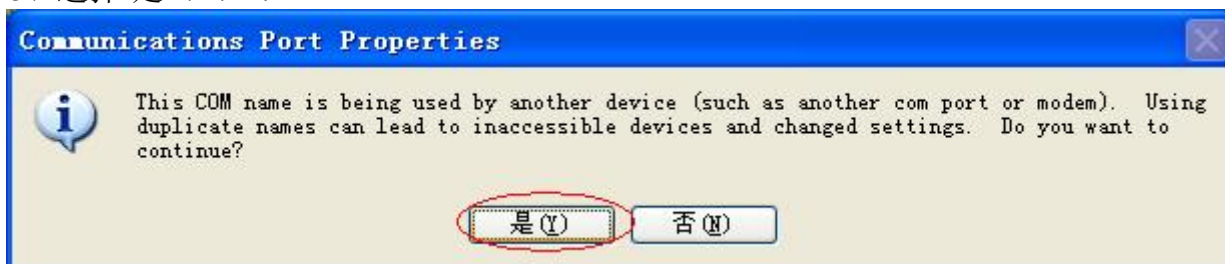
3、选择“Port Settings” 点击“Advanced”:



4、选择“COM3”或 其它串口，不要和现有的冲突就行:



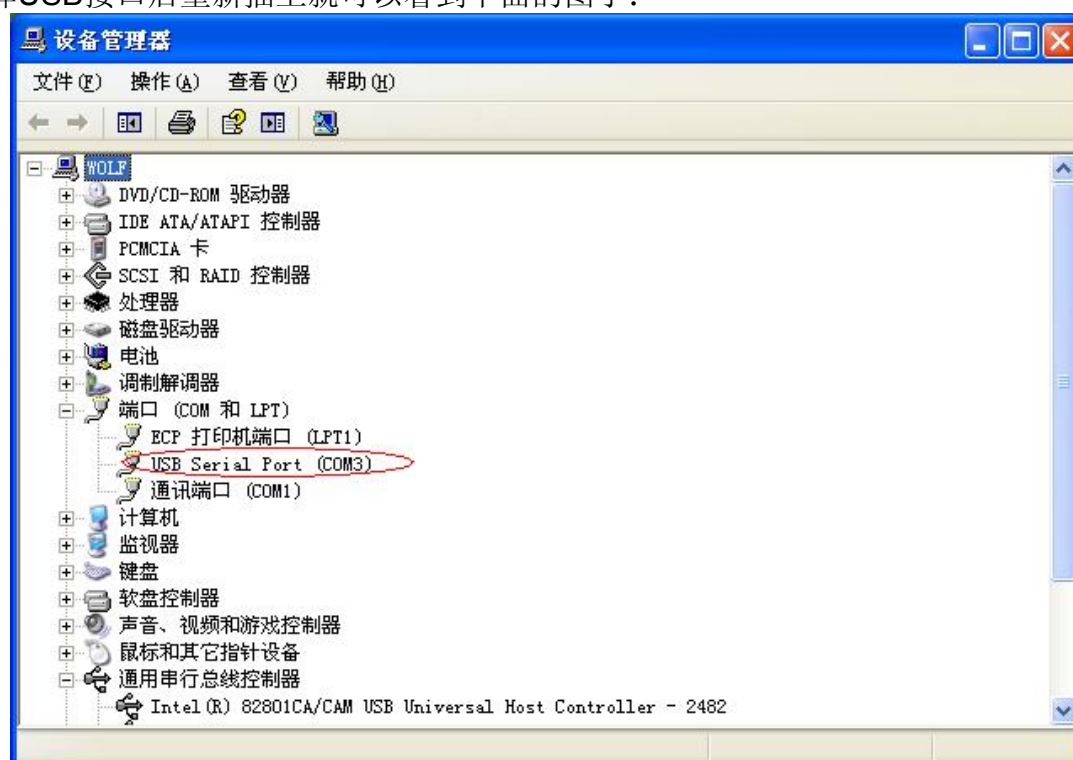
5、选择“是（Y）”：



6、如下图点击“确定”：



7、拔掉USB接口后重新插上就可以看到下面的图了：



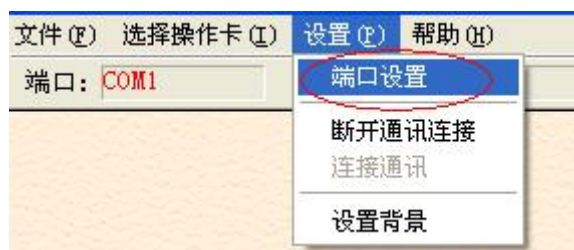
7. SDK的使用

双击rfidxray.exe进入DEMO。

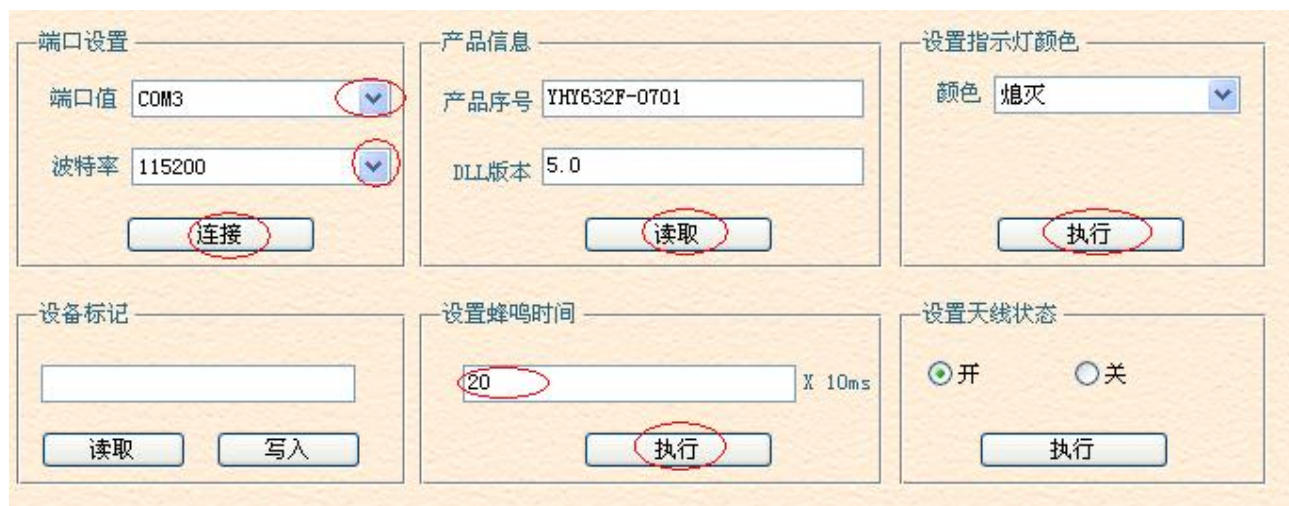
在DEMO界面里，每次点击按键所调用的库函数会出现在屏幕下部，并在屏幕右上角提示执行结果

7.1 端口设置

首先选择正确的串口号，波特率建议选择115200，然后点击【连接】，使读卡器和PC建立起通讯，如果连接失败，可以试着拔插一下USB接口后重新连接。



点击【读取】产品信息键，可查看读卡器的具体的型号。



7.2 卡操作举例

友情提示：操作某种卡之前请先花几分钟时间仔细阅读相应卡的datasheet，卡的结构，卡的操作规程，在卡的手册都有详细的描述。

7.2.1 ISO14443A/MIFARE类卡

7.2.1.1 Mifare S50 卡操作

A/B密钥原始密钥（12个F）：**FFFFFFFFFFFF**，A密钥是不可见的，一律以0填充读到的数据，所以您在demo见到密钥A处显示的是0时不要奇怪，改成其它密钥后一样读出来还是显示0；B密钥开始启用后同理，另外B密钥的空间还可以作为数据存储用。控制字开始是FF078069，具有读写权限。

注：NXP原始芯片密钥的出厂设置是，密钥A有全部权限，密钥B没有；
任何权限，可通过修改控制字来改变密钥属性。

Step1: 先选择 S50 卡进行操作。



Step2: 在天线感应区（感应距离和卡的大小有关）放上卡，然后点击“寻卡”，此时可以读到 4 个字节的 ID 号。

注意，卡不要和桌面垂直，卡在读卡器垂直方向区域感应效果最好，离开中心区域电磁场会减弱，其它卡同理。

Step3: 读卡，选择某一个扇区，输入密钥，开始空白卡用默认密钥。然后点击“读卡”，就可以读到整个扇区的数据了。

Step4: 写卡，选择某个块，输入密钥，然后就可以写数据了。每次只能同时写 16 个字节，不多也不少，哪怕您只需要写进去一个字节，也必须把其余 15 个字节填上。对于块 3，因为是密钥块，开始没有熟练之前请不要写它，防止忘记密钥后这个扇区作废！修改密钥的操作就是写块 3 的操作。

比如要把扇区 2 的 A 密钥从 **FFFFFFFFFFFF** 改为 123456789088，那么选择扇区 2，块号 3，然后在密钥 A 处修改为 123456789088 即可，其余可以先不变，然后点击“写卡”系统会提示成功与否。

详情请参考 s50 卡的 datasheet 和我们提供的 AB 密钥相关文档，计算不同扇区和 AB 密钥控制字可以参考我们提供的自动计算软件。

DC149433

寻卡 ALL

寻卡 IDL

休眠

钱包功能

扇区号 1 块号 1

☒ A 密钥
 ☐ B 密钥
 密钥 EEEEEEEEEEE

金额(十进制) 88
 金额(十六进制) 58000000

初始化

充值

扣款

读余额

操作窗口

扇区号 3 块号 0

读卡 写卡

00112233445566778899aabbddeeff00 写数据
 00000000000000000000000000000000
 00000000000000000000000000000000
 00000000000000000000000000000000
 00000000000000000000000000000000
 FF078069 FFFFFFFF
 A密钥 控制字 B密钥
☒ A 密钥 ☐ B 密钥 密钥 EEEEEEEEEEE 输入密钥

操作窗口

扇区号 0 块号 0

读卡 写卡

DC149433F986200648E859065002807
 00000000000000000000000000000000 读数据
 00000000000000000000000000000000
 00000000000000000000000000000000
 FF078069 FFFFFFFF
☒ A 密钥 ☐ B 密钥 密钥 EEEEEEEEEEE

操作窗口

扇区号 15 块号 0

读卡 写卡

00000000000000000000000000000000
 00000000000000000000000000000000
 00000000000000000000000000000000
 00000000000000000000000000000000
 00000000000000000000000000000000
 FF078069 FFFFFFFF
☒ A 密钥 ☐ B 密钥 密钥 EEEEEEEEEEE

S70卡的容量是S50卡的4倍，有4k字节容量；前32个扇区和s50的结构是一样的，但是后面8个扇区每个包含有16个块，这点和S50是不同的，当有大量数据需要保存的时候可以选择这种形式。其它操作方式和原理同S50卡。





7.2.1.3 Mifare UltraLight 卡操作

点击【寻卡】，可得到卡片的7字节序列号ID， 选择相应的地址，可对卡片进行读写操作。UL卡是一种低容量的卡，没有密钥认证，每页是4个字节。适合用于公园门票或地铁单程票应用，或者那些安全性要求不高的系统使用。

SN0.....SN6		PAGE0	045E8052
045E80E1BF0280		PAGE1	E1BF0280
寻卡.ALL	寻卡.IDL	PAGE2	DC48 0000
休眠		PAGE3	00000000
页地址		PAGE4	FFFFFFFF
2		PAGE5	00000000
读卡		PAGE6	00000000
写卡		PAGE7	00000000
		PAGE8	00000000
		PAGE9	00000000
		PAGE10	00000000
		PAGE11	00000000
		PAGE12	00000000
		PAGE13	00000000
		PAGE14	00000000
		PAGE15	00000000

7.2.1.4 Mifare_ProX卡操作

点击【复位】键，可得到卡片的序列号和复位信息

输入COS指令，点击【发送】键对卡片进行操作，具体请参考相应的卡资料。



7.2.2 ISO14443B类卡

7.2.2.1 AT88RF020

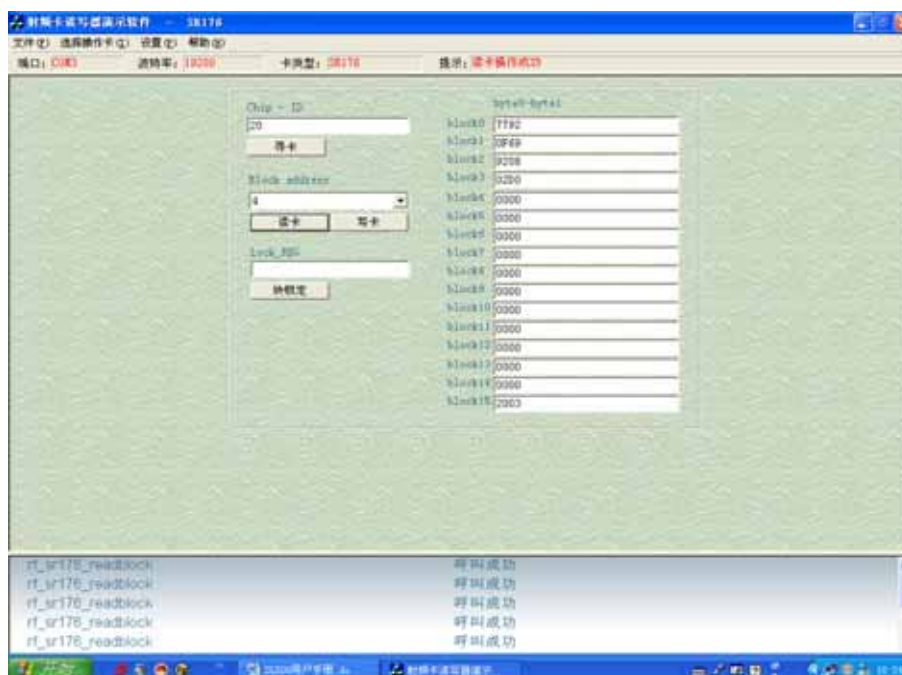


点击【寻卡】键，可得到卡片的序列号，进行密钥验证后可对卡片进行读、写、签名和块锁定操作。原始密钥是8个字节00。



7.2.2.2 SR176卡

点击【寻卡】键，可得到卡片的ID号，此后可对卡片进行读、写和块锁定操作。



7.2.2.2 SR1X4K

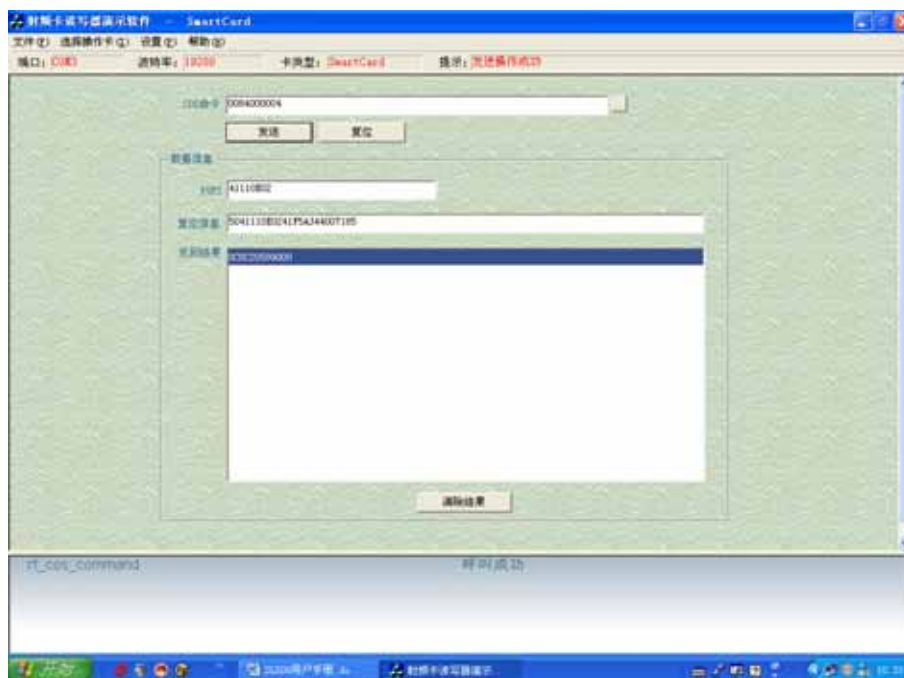
点击【寻卡】键，可得到卡片的ID号，点击【读取UID】得到卡片的UID，此后可对卡片进行读、写和块锁定操作。



7.2.2.4 符合ISO14443B-4协议的CPU卡

点击【复位】键，可得到卡片的序列号和复位信息

输入COS指令，点击【发送】键对卡片进行操作



7.2.3 ISO15693 类卡

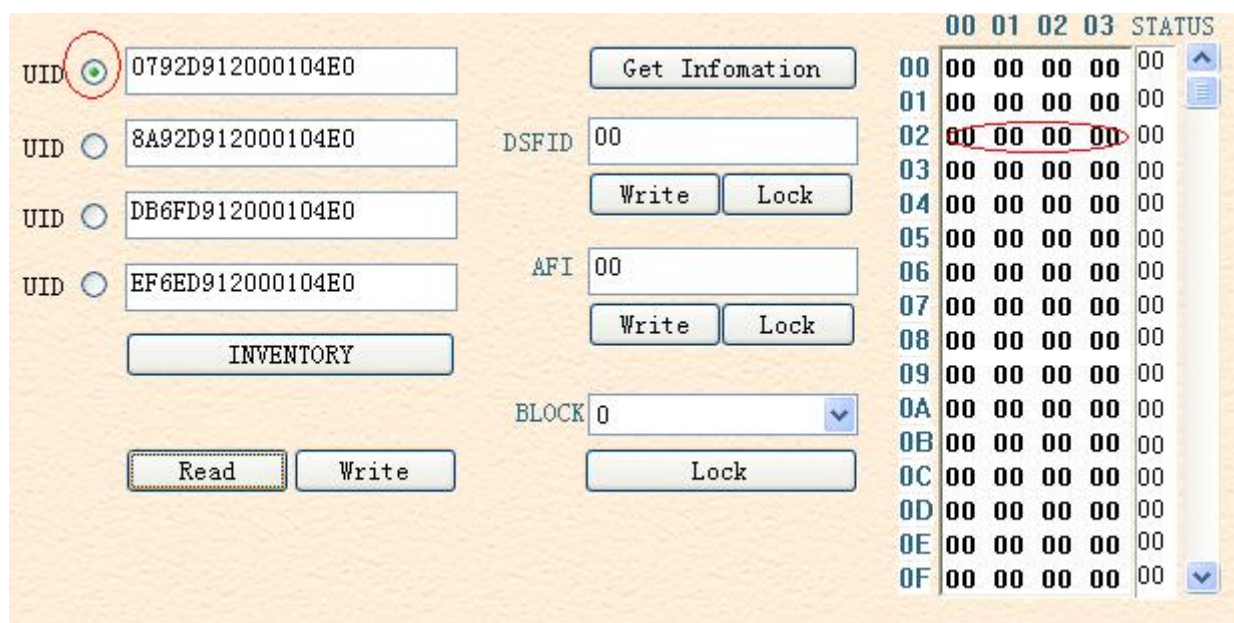
7.2.3.1 I.CODE SLI



点击【INVENTORY】键，可得到卡片的序列号，最多可操作4张卡片。



根据UID选定某一个卡，然后执行“Get Information”获得卡的信息，然后可以可对该卡进行读写操作了：



	00	01	02	03	STATUS
00	00	00	00	00	00
01	00	00	00	00	00
02	00	00	00	00	00
03	00	00	00	00	00
04	00	00	00	00	00
05	00	00	00	00	00
06	00	00	00	00	00
07	00	00	00	00	00
08	00	00	00	00	00
09	00	00	00	00	00
0A	00	00	00	00	00
0B	00	00	00	00	00
0C	00	00	00	00	00
0D	00	00	00	00	00
0E	00	00	00	00	00
0F	00	00	00	00	00

7.2.3.1 Tag_IT



点击【INVENTORY】键，可得到卡片的序列号，最多可操作4张卡片。根据UID选定某一个卡，可对该卡进行读写操作。



8. 库函数说明

系统函数为所有型号读卡器共有。

读卡器是否支持其它函数由具体型号决定。

8.1 系统函数

8.1.1 INT WINAPI LIB_VER

功能：获取动态库版本号

原型：int WINAPI lib_ver(unsigned int *pVer)

参数：pVer: [OUT] 动态库版本号

返回：成功返回0

8.1.2 INT WINAPI RF_INIT_COM

功能：初始化串口

原型：int WINAPI rf_init_com (int port,long baud)

参数: port: [IN] 串口号

baud: [IN] 为通讯波特率4800 ~ 115200 , 默认为19200bps

返回: 成功返回0

8.1.3 INT WINAPI RF_CLOSEPORT

功能: 关闭Com端口

原型: int WINAPI rf_ClosePort()

返回: 成功返回0

8.1.4 INT WINAPI RF_GET_MODEL

功能: 读取读写卡器型号及产品批号

原型: int WINAPI rf_get_model(unsigned short icdev,
 unsigned char *pVersion,
 unsigned char *pLen)

参数: icdev: [IN] 通讯设备标识符

pVersion: [OUT] 返回的信息

pLen: [OUT] 返回信息的长度

返回: 成功返回0

说明: 见“型号说明”部分

8.1.5 INT WINAPI RF_INIT_DEVICE_NUMBER

功能: 指定设备标识

原型: int WINAPI rf_init_device_number(unsigned short icdev)

参数: icdev: [IN] 通讯设备标识符

返回: 成功返回0

说明: 读卡器只响应设备标识符与本身相符或
设备标识符等于0x0000的指令。

8.1.6 INT WINAPI RF_GET_DEVICE_NUMBER

功能: 读取设备标识

原型: int WINAPI rf_get_device_number(unsigned short *plcdev)

参数: plcdev: [OUT] 返回通讯设备标识符

返回：成功返回0

说明：如果一个串口同时联有两台以上读卡器，不可使用该命令

8.1.7 INT WINAPI RF_INIT_TYPE

功能：设置读写卡器非接触工作方式

原型：int WINAPI rf_init_type(unsigned short icdev, unsigned char type)

参数：icdev: [IN] 通讯设备标识符

type: [IN] 读写卡器工作方式

返回：成功返回0

说明：只支持单一协议的读卡器此函数无效

type = 'A': 设置为TYPE_A方式

type = 'B': 设置为TYPE_B方式

type = 'r': 设置为AT88RF020卡方式

type = '1': 设置为ISO15693方式（123的1）

8.1.8 INT WINAPI RF_ANTENNA_STA

功能：设置读写卡器天线状态

原型：int WINAPI rf_antenna_sta(unsigned short icdev, unsigned char model)

参数：icdev: [IN] 通讯设备标识符

model: [IN] 天线状态

返回：成功返回0

说明：model = 0: 关闭天线

model = 1: 开启天线

8.1.9 INT WINAPI RF_LIGHT

功能：设置指示灯颜色

原型：int WINAPI rf_light(unsigned short icdev, unsigned char color)

参数：icdev: [IN] 通讯设备标识符

color: [IN] 0 = 熄灭

1 = 红色

2 = 绿色

3 = 黄色

返回：成功返回0

说明：部分型号读卡器无此功能

8.1.10 INT WINAPI RF_BEEP

功能：蜂鸣

原型：int WINAPI rf_bEEP(unsigned short icdev, unsigned char msec)

参数：icdev: [IN] 通讯设备标识符

msec: [IN] 蜂鸣时限，单位是10毫秒

返回：成功返回0

8.3 ISO14443A 函数

8.3.1 UltraLight

8.3.1.1 INT WINAPI RF_REQUEST

功能：寻TYPE_A卡

原型：int WINAPI rf_request(unsigned short icdev,
 unsigned char model,
 unsigned short *pTagType)

参数：icdev: [IN] 通讯设备标识符

model: [IN] 寻卡模式

pTagType: [OUT] 返回卡类型值

返回：成功返回0

说明：mode = 0x26: 寻未进入休眠状态的卡

mode = 0x52: 寻所有状态的卡

8.3.1.2 INT WINAPI INT RF_UL_SELECT

功能：ultra light卡防冲撞并激活

原型：int WINAPI int_rf_ul_select (unsigned short icdev,
 unsigned char *pSnr,
 unsigned char *pLen)

参数：icdev: [IN] 通讯设备标识符

pSnr: [OUT] 返回卡序列号

pLen: [OUT] 返回序列号的长度

返回: 成功返回0

8.3.1.3 INT WINAPI RF_M1_READ

功能: 读取MifareOne卡一块数据

原型: int WINAPI rf_m1_read (unsigned short icdev,
 unsigned char block,
 unsigned char *pData,
 unsigned char *pLen)

参数: icdev: [IN] 通讯设备标识符

block: [IN] M1卡绝对块号

pData: [OUT] 返回的数据

pLen: [OUT] 返回数据的长度

返回: 成功返回0

说明: 此函数也适用于UltraLight卡, UltraLight卡每个 page 4字节,
每次调用此函数返回4个连续的page的数据

8.3.1.4 INT WINAPI INT RF_UL_WRITE

功能: 向ultra light卡中写入一块数据

原型: int WINAPI int rf_ul_write (unsigned short icdev,
 unsigned char page,
 unsigned char *pData)

参数: icdev: [IN] 通讯设备标识符

page: [IN] UltraLight卡页地址 (0 ~ 0x0F)

pData: [IN] 写入的数据, 4字节

返回: 成功返回0

8.3.1.5 INT WINAPI RF_HALT

功能: 命令已激活的TYPE_A卡进入HALT状态

原型: int WINAPI rf_halt(unsigned short icdev)

参数: icdev: [IN] 通讯设备标识符

返回: 成功返回0

说明: 卡片接收到此命令后退出激活状态

8.3.2 Mifare_Std

8.3.2.1 INT WINAPI RF_REQUEST

功能：寻TYPE_A卡

原型：int WINAPI rf_request(unsigned short icdev,
 unsigned char model,
 unsigned short *pTagType)

参数：icdev: [IN] 通讯设备标识符

 model: [IN] 寻卡模式

 pTagType: [OUT] 返回卡类型值

返回：成功返回0

说明：mode = 0x26: 寻未进入休眠状态的卡

 mode = 0x52: 寻所有状态的卡

pTagType: 0x4400 = ultra_light

 0x0400 = Mifare_One(S50)

 0x0200 = Mifare_One(S70)

 0x4403 = Mifare_DESFire

 0x0800 = Mifare_Pro

 0x0403 = Mifare_ProX

 0x0033 = SHC1102

要对感应区内的ISO14443A卡进行读写操作应顺序调用以下库函数

MifareOne:

1. int WINAPI rf_request

2. int WINAPI rf_anticoll

3. int WINAPI rf_select

此后卡片处于激活状态

UltraLight:

1. int WINAPI rf_request

2. int WINAPI int rf_ul_select

此函数集合了防冲撞和激活指令，此后卡片处于激活状态

MifarePro(X):

1. int WINAPI rf_typea_rst

此函数集合了寻卡、防冲撞、激活及复位指令，此后卡片处于激活状态

8.3.2.2 INT WINAPI RF_ANTICOLL

功能: TYPE_A卡防冲撞

原型: int WINAPI rf_anticoll(unsigned short icdev,

unsigned char bcnt,

unsigned char *pSnr,

unsigned char *pLen)

参数: icdev: [IN] 通讯设备标识符

bcnt: [IN] 取值4

pSnr: [OUT] 返回的卡序列号

pLen: [OUT] 返回序列号的长度

返回: 成功返回0

8.3.2.3 INT WINAPI RF_SELECT

功能: 激活TYPE_A卡

原型: int WINAPI rf_select(unsigned short icdev,

unsigned char *pSnr,

unsigned char snrLen,

unsigned char *pSize)

参数: icdev: [IN] 通讯设备标识符

pSnr: [IN] 卡序列号

snrLen: [IN] 卡序列号长度

pSize: [OUT] 返回卡容量

返回: 成功返回0

说明: 卡片接收到该命令后进入激活状态，在一个感应区域内的同一时刻

只可有一张TYPE_A卡处于激活状态

8.3.2.4 INT WINAPI RF_M1_AUTHENTICATION2

功能：验证Mifare_Std卡密钥

原型：int WINAPI rf_m1_authentication2(unsigned short icdev,
unsigned char model,
unsigned char block,
unsigned char *pKey)

参数：icdev: [IN] 通讯设备标识符

Model: [IN] 密钥验证模式

block: [IN] 要验证密钥的绝对块号

pKey: [IN] 密钥内容，6 字节

返回：成功返回0

说明：model = 0x60: 验证A密钥

model = 0x61: 验证B密钥

8.3.2.5 INT WINAPI RF_M1_READ

功能：读取MifareOne卡一块数据

原型：int WINAPI rf_m1_read(unsigned short icdev,
unsigned char block,
unsigned char *pData,
unsigned char *pLen)

参数：icdev: [IN] 通讯设备标识符

block: [IN] M1卡绝对块号

pData: [OUT] 返回的数据

pLen: [OUT] 返回数据的长度

返回：成功返回0

说明：此函数也适用于ultra_light卡，ultra_light卡每个 page 4字节，
每次调用此函数返回4个连续的page的数据

8.3.2.6 INT WINAPI RF_M1_WRITE

功能：写入MifareOne卡一块数据

原型：int WINAPI rf_m1_write(unsigned short icdev,
unsigned char block,
unsigned char *pData)

参数: icdev: [IN] 通讯设备标识符
block: [IN] M1卡绝对块号
pData: [IN] 写入的数据, 16 字节
返回: 成功返回0

8.3.2.7 INT WINAPI RF_M1_INITVAL

功能: 将Mifare One 卡某一块初始化为钱包

原型: int WINAPI rf_m1_initval(unsigned short icdev,
unsigned char block,
long value)

参数: icdev: [IN] 通讯设备标识符
block: [IN] M1卡绝对块号
pValue: [IN] 初始金额, 16进制, 低字节在前
返回: 成功返回0

8.3.2.8 INT WINAPI RF_M1_READVAL

功能: 读取Mifare One卡钱包值

原型: int WINAPI rf_m1_readval(unsigned short icdev,
unsigned char block,
long *pValue)

参数: icdev: [IN] 通讯设备标识符
block: [IN] M1卡绝对块号
pValue: [OUT] 返回的值, 16进制, 低字节在前
返回: 成功返回0

8.3.2.9 INT WINAPI RF_M1_INCREMENT

功能: Mifare One卡充值

原型: int WINAPI rf_m1_increment (unsigned short icdev,
unsigned char block,
long value)

参数: icdev: [IN] 通讯设备标识符
block: [IN] M1卡绝对块号
value: [IN] 要增加的值, 16进制, 低字节在前

返回：成功返回0

8.3.2.10 INT WINAPI RF_M1_DECREMENT

功能：Mifare One卡扣款

原型：int WINAPI rf_M1_decrement (unsigned short icdev,
unsigned char block,
long value)

参数：icdev: [IN] 通讯设备标识符
block: [IN] M1卡绝对块号
value: [IN] 要扣的值，16进制，低字节在前

返回：成功返回0

8.3.2.11 INT WINAPI RF_M1_RESTORE

功能：Mifare One卡数据回传

原型：int WINAPI rf_M1_restore (unsigned short icdev, unsigned char block)

参数：icdev: [IN] 通讯设备标识符
block: [IN] M1卡绝对块号

返回：成功返回0

说明：用此函数将指定的块数据传入卡的buffer，然后可用rf_M1transfer()函数将buffer 中数据再传送到另一块中去，源地址和目的地址须在同一扇区内

8.3.2.12 INT WINAPI RF_M1_TRANSFER

功能：将M1卡BUFFER中的块数据传入指定的块

原型：int WINAPI rf_M1_transfer (unsigned short icdev, unsigned char block)

参数：icdev: [IN] 通讯设备标识符
block: [IN] M1卡绝对块号

返回：成功返回0

说明：该函数仅在increment、decrement和restore 命令之后调用

8.3.2.13 INT WINAPI RF_HALT

功能：命令已激活的TYPE_A卡进入HALT状态

原型：int WINAPI rf_halt(unsigned short icdev)

参数: icdev: [IN] 通讯设备标识符

返回: 成功返回0

说明: 卡片接收到此命令后退出激活状态

8.3.4 Mifare_ProX

8.3.4.1 INT WINAPI RF_TYPE_RST

功能: 寻感应区内符合ISO14443A标准的CPU卡并复位

原型: int WINAPI rf_typea_rst(unsigned short icdev,
 unsigned char model,
 unsigned char *pData,
 unsigned char *pMsgLg)

参数: icdev: [IN] 通讯设备标识符

model: [IN] 寻卡方式

pDate: [OUT] 返回的数据

pMsgLg: [OUT] 返回数据的长度

返回: 成功返回0

说明: mode = 0x26: 寻未进入休眠状态的卡

mode = 0x52: 寻所有状态的卡

pDate: 4字节CSN + 复位信息内容

8.3.4.2 INT WINAPI RF_COS_COMMAND

功能: 向符合ISO14443-4标准的非接触CPU卡发送COS 命令

原型: int WINAPI rf_cos_command(unsigned short icdev,
 unsigned char *pCommand,
 unsigned char cmdLen,
 unsigned char *pData,
 unsigned char *pMsgLg)

功能: 向符合ISO14443-4标准的CPU卡发送COS命令

参数: icdev: [IN] 通讯设备标识符

pCommand: [IN] cos 命令

cmdLen: [IN] cos 命令长度

pDate: [OUT] 卡片返回的数据, 含SW1、SW2

pMsgLg: [OUT] 返回数据长度

返回: 成功则返回0

8.3.4.3 INT WINAPI RF_CL_DESELECT

功能: 命令符合ISO14443协议的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

返回: 成功返回0

8.4 ISO14443B函数

8.4.2 AT88RF020

8.4.2.1 INT WINAPI RF_TYPEB_RST

功能: 寻感应区内符合ISO14443B标准的卡并建立SLOT

原型: int WINAPI rf_atqb(unsigned short icdev,
 unsigned char model,
 unsigned char *pData,
 unsigned char *pMsgLg)

参数: icdev: [IN] 通讯设备标识符

model: [IN] 寻卡方式0 = REQB,1=WUPB

pDate: [OUT] 卡片返回的数据

pMsgLg: [OUT] 返回数据的长度

返回: 成功则返回0

8.4.2.2 INT WINAPI RF_AT020_CHECK

功能: 验证AT88RF020卡密钥

原型: int WINAPI rf_at020_check(unsigned short icdev, unsigned char *pKey)

参数: icdev: [IN] 通讯设备标识符

pKey: [IN] 密钥, 8 字节

返回: 成功返回0

8.4.2.3 INT WINAPI RF_AT020_COUNT



说明: AT88RF020卡有签名计数功能, 调用此函数后, 6字节签名信息写入到卡片PAGE2的低6字节, 同时该页的高两个字节加1

8.4.2.4 INT WINAPI RF AT020 READ

```
原型: int WINAPI rf_at020_read (unsigned short icdev,  
                                unsigned char   page,  
                                unsigned char   *pData,  
                                unsigned char   *pMsgLen)
```

返回：成功返回0

8.4.2.5 INT WINAPI RF AT020 WRITE

```
原型: int WINAPI rf_at020_write(unsigned short icdev,
                                unsigned char  page,
                                unsigned char  *pData)
```

参数: icdev: [IN] 通讯设备标识符
page: [IN] 页地址, (0~31)
pDate: [IN] 要写入的数据, 8 字节
返回: 成功返回0

8.4.2.6 INT WINAPI RF AT020 LOCK

原型: int WINAPI rf_at020_lock(unsigned short icdev, unsigned char *pData)

参数: icdev: [IN] 通讯设备标识符

pDate: [IN] 数据, 4字节

返回: 成功返回0

8.4.2.7 INT WINAPI RF_ AT020_ DESELECT

功能: 命令AT88RF020卡退出激活状态

原型: int WINAPI rf_at020_deselect(unsigned short icdev)

参数: icdev: [IN] 通讯设备标识符

返回: 成功返回0

说明: 卡片接收此命令后退出激活状态, 不再接收其它指令直到再次进入激活状态

8.4.3 SR176SRIX4K

8.4.3.1 INT WINAPI RF_ST_SELECT

功能: 锁定一张ST卡(SR176/SRIX4K)

原型: int WINAPI rf_st_select(unsigned short icdev, unsigned char *pChip_ID)

参数: icdev: [IN] 通讯设备标识符

pChip_ID: [IN] 返回卡ID号, 一字节

返回: 成功则返回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。

8.4.3.2 INT WINAPI INT_RF_SR176_READBLOCK

功能: 读SR176卡1块数据

原型: int WINAPI int_rf_sr176_readblock(unsigned short icdev,
unsigned char block,
unsigned char *pData,
unsigned char *pLen)

参数: icdev: [IN] 通讯设备标识符

block: [IN] 块地址

pData: [OUT] 读出数据

pLen: [OUT] 读出数据的长度

返回: 成功则返回0

8.4.3.3 INT WINAPI INT_RF_SR176_WRITEBLOCK

功能：写SR176卡1块数据

原型：int WINAPI int rf_sr176_writeblock(unsigned short icdev,
unsigned char block,
unsigned char *pData)

参数：icdev: [IN] 通讯设备标识符

block: [IN] 块地址

pData: [IN] 要写入的数据，2字节

返回：成功则返回0

8.4.3.4 INT WINAPI INT_RF_SR176_PROTECTBLOCK

功能：SR176卡块锁定

原型：int WINAPI int rf_sr176_protectblock(unsigned short icdev, unsigned char lockreg)

参数：icdev: [IN] 通讯设备标识符

lockreg: [IN] LOCKREG

返回：成功则返回0

说明：SR176有16个块，lockreg每1位控制2个块

lockreg	BLOCK	bit_setting	
b7	14 & 15	0:Write Enable	1:Block set as ROM
b6	12 & 13	0:Write Enable	1:Block set as ROM
b5	10 & 11	0:Write Enable	1:Block set as ROM
b4	8 & 9	0:Write Enable	1:Block set as ROM
b3	6 & 7	0:Write Enable	1:Block set as ROM
b2	4 & 5	0:Write Enable	1:Block set as ROM
b1	2 & 3	0:Write Enable	1:Block set as ROM
b0	0 & 1	0:Write Enable	1:Block set as ROM

8.4.3.5 INT WINAPI INT_RF_SRIX4K_GETUID

功能：获取SRIX4K卡UID

原型：int WINAPI int rf_srix4k_getuid(unsigned short icdev,
unsigned char *pUid,
unsigned char *pLen)

参数：icdev: [IN] 通讯设备标识符

pUid: [OUT] 返回的UID

pLen: [OUT] 返回的UID长度

返回: 成功则返回0

8.4.3.6 INT WINAPI INT_RF_SRIX4K_READBLOCK

功能: 读SRIX4K卡1块数据

原型: int WINAPI int rf_srix4k_readblock(unsigned short icdev,
unsigned char block,
unsigned char *pData,
unsigned char *pLen)

参数: icdev: [IN] 通讯设备标识符

block: [IN] 块地址

pData: [OUT] 读出的数据

pLen: [OUT] 读出数据的长度

返回: 成功则返回0

8.4.3.7 INT WINAPI INT_RF_SRIX4K_WRITEBLOCK

功能: 写SRIX4K卡1块数据

原型: int WINAPI int rf_srix4k_writeblock(unsigned short icdev,
unsigned char block,
unsigned char *pData)

参数: icdev: [IN] 通讯设备标识符

block: [IN] 块地址

pData: [IN] 要写入的数据, 4字节

返回: 成功则返回0

8.4.3.8 INT WINAPI INT_RF_SRIX4K_PROTECTBLOCK

功能: SRIX4K卡块锁定

原型: int WINAPI int rf_srix4k_protectblock(unsigned short icdev, unsigned char lockreg)

参数: icdev: [IN] 通讯设备标识符

Lockreg: [IN] LOCKREG

返回: 成功则返回0

说明: SR1X4K卡7 - 15块可以被写保护,被写保护后的块将不再能够写入

lockreg	BLOCK	bit_setting	
b7	15	1:Write Enable	0:Block set as ROM
b6	14	1:Write Enable	0:Block set as ROM
b5	13	1:Write Enable	0:Block set as ROM
b4	12	1:Write Enable	0:Block set as ROM
b3	11	1:Write Enable	0:Block set as ROM
b2	10	1:Write Enable	0:Block set as ROM
b1	9	1:Write Enable	0:Block set as ROM
b0	7 & 8	1:Write Enable	0:Block set as ROM

8.4.4 TYPE_B SmartCard

8.4.4.1 INT WINAPI RF_ TYPEB_RST

功能: 寻感应区内符合ISO14443B标准的CPU卡并复位

原型: int WINAPI rf_atqb(unsigned short icdev,
 unsigned char model,
 unsigned char *pData,
 unsigned char *pMsgLg)

参数: icdev: [IN] 通讯设备标识符

 model: [IN] 寻卡方式0 = REQB,1=WUPB

 pDate: [OUT] 卡片返回的数据

 pMsgLg: [OUT] 返回数据的长度

返回: 成功则返回0

8.4.4.2 INT WINAPI RF_ COS_COMMAND

功能: 向符合ISO14443-4标准的非接触CPU卡发送COS 命令

原型: int WINAPI rf_cos_command(unsigned short icdev,
 unsigned char *pCommand,
 unsigned char cmdLen,
 unsigned char *pData,
 unsigned char *pMsgLg)

功能: 向符合ISO14443-4标准的CPU卡发送COS 命令

参数: icdev: [IN] 通讯设备标识符

pCommand: [IN] cos 命令

cmdLen: [IN] cos 命令长度

pDate: [OUT] 卡片返回的数据，含SW1、SW2

pMsgLg: [OUT] 返回数据长度

返回：成功则返回0

8.4.4.3 INT WINAPI RF_CL_DESELECT

功能：命令符合ISO14443协议的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

返回：成功返回0

8.4.4.4 INT WINAPI RF_TRANSCEIVE

功能：向符合ISO14443B标准的卡发送指令并接收返回数据

原型：int WINAPI rf_transceive (unsigned short icdev,

unsigned char crcon,

unsigned char *pTxData,

unsigned char sendLen,

unsigned char *pRxDate,

unsigned char *pMsgLg)

参数：icdev: [IN] 通讯设备标识符

Crcon: [IN] 非0表示发送接收CRC，CRC由读卡器内部计算和验证

等于0表示不发送不接收CRC

pTxData: [IN] 发送数据

sendLen: [IN] 发送数据长度

pRxDate: [OUT] 卡片返回的数据

pMsgLg: [OUT] 返回数据长度

8.5 ISO15693函数

8.5.1 INT WINAPI ISO15693_INVENTORY

功能：ISO15693_Inventory (单张卡)

原型：int WINAPI ISO15693_Inventory(unsigned short icdev,

unsigned char *pData,
unsigned char *pLen)

参数: icdev: [IN] 通讯设备标识符

pData: [OUT] 返回的数据, 1字节DSFID+8字节UID

pLen: [OUT] pdata长度

返回: 成功则返回0

8.5.2 INT WINAPI ISO15693_INVENTORYS

功能: ISO15693_Inventory (多张卡) , 扫描时间长, 有多个时才采用。

原型: int WINAPI ISO15693_Inventorys(unsigned short icdev,
unsigned char *pData,
unsigned char *pLen)

参数: icdev: [IN] 通讯设备标识符

pData: [OUT] 返回的数据, 每9个字节为一组, 每组结构为: 1字节DSFID + 8
字节UID

pLen: [OUT] 返回数据的长度

返回: 成功则返回0

8.5.3 INT WINAPI ISO15693_GET_SYSTEM_INFORMATION

功能: ISO15693_Get_System_Information

原型: int WINAPI ISO15693_Get_System_Information(unsigned short icdev,
unsigned char model,
unsigned char *pUID,
unsigned char *pData,
unsigned char *pLen)

参数: icdev: [IN] 通讯设备标识符

model: [IN] bit0=Select_flag,bit1=Addres_flag,bit2=Option_flag

pUID: [IN] UID 8字节

pData: [OUT] 返回的数据

pLen: [OUT] 返回数据的长度

返回: 成功则返回0

说明: Select_flag = 1, 只有处于SELECT状态的卡执行该命令

Addres_flag = 1, 只有UID符合的卡执行该命令

设ICODE SLI卡Option_flag = 0 , TI TAGIT卡Option_flag = 1。

8.5.4 INT WINAPI ISO15693_SELECT

功能: ISO15693_Select

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

参数: icdev: [IN] 通讯设备标识符

pUID: [IN] UID 8字节

返回: 成功则返回0

8.5.5 INT WINAPI ISO15693_RESET_TO_READY

功能: ISO15693_Reset_To_Ready

原型: int WINAPI ISO15693_Reset_To_Ready(unsigned short icdev,
unsigned char model,
unsigned char *pUID)

参数: icdev: [IN] 通讯设备标识符

model: [IN] bit0=Select_flag,bit1=Addres_flag,bit2=Option_flag

pUID: [IN] UID 8字节

返回: 成功则返回0

说明: Select_flag = 1, 只有处于SELECT状态的卡执行该命令

Addres_flag = 1, 只有UID符合的卡执行该命令

设ICODE SLI卡Option_flag = 0 , TI TAGIT卡Option_flag = 1。

8.5.6 INT WINAPI ISO15693_STAY_QUIET

功能: ISO15693_Stay_Quiet

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

参数: icdev: [IN] 通讯设备标识符

pUID: [IN] UID 8字节

返回: 成功则返回0

8.5.7 INT WINAPI ISO15693_GET_BLOCK_SECURITY

功能: ISO15693_Get_Block_Security

原型: int WINAPI ISO15693_Get_Block_Security(unsigned short icdev,

unsigned char model,
unsigned char *pUID,
unsigned char block,
unsigned char number,
unsigned char *pData,
unsigned char *pLen)

参数: icdev: [IN] 通讯设备标识符

model: [IN] bit0=Select_flag,bit1=Addres_flag,bit2=Option_flag

pUID: [IN] UID 8字节

block: [IN] 块号

number: [IN] 要读取的块数, < 0x40

pData: [OUT] 返回的数据

pLen: [OUT] 返回数据的长度

返回: 成功则返回0

说明: Select_flag = 1, 只有处于SELECT状态的卡执行该命令

Addres_flag = 1, 只有UID符合的卡执行该命令

设ICODE SLI卡Option_flag = 0, TI TAGIT卡Option_flag = 1。

8.5.8 INT WINAPI ISO15693_READ

功能: ISO15693_Read

原型: int WINAPI ISO15693_Read(unsigned short icdev,

unsigned char model,
unsigned char *pUID,
unsigned char block,
unsigned char number,
unsigned char *pData,
unsigned char *pLen);

参数: icdev: [IN] 通讯设备标识符

model: [IN] bit0=Select_flag,bit1=Addres_flag,bit2=Option_flag

pUID: [IN] UID 8字节

block: [IN] 块号

number: [IN] 要读取的块数, < 0x40

pData: [OUT] 返回的数据

pLen: [OUT] 返回数据的长度

返回: 成功则返回0

说明: Select_flag = 1, 只有处于SELECT状态的卡执行该命令

Addres_flag = 1, 只有UID符合的卡执行该命令

设ICODE SLI卡Option_flag = 0, TI TAGIT卡Option_flag = 1。

8.5.9 INT WINAPI ISO15693_WRITE

功能: ISO15693_Write

原型: int WINAPI ISO15693_Write(unsigned short icdev,

unsigned char model,
unsigned char *pUID,
unsigned char block,
unsigned char *pData);

参数: icdev: [IN] 通讯设备标识符

model: [IN] bit0=Select_flag,bit1=Addres_flag,bit2=Option_flag

pUID: [IN] UID 8字节

block: [IN] 块号

pData: [IN] 要写入的数据, 4字节

返回: 成功则返回0

说明: Select_flag = 1, 只有处于SELECT状态的卡执行该命令

Addres_flag = 1, 只有UID符合的卡执行该命令

如操作的是TI卡片设Option_flag = 1, 如操作的是I.CODE SLI卡片设Option_flag = 0

8.5.10 INT WINAPI ISO15693_LOCK_BLOCK

功能: ISO15693_Lock_Block

原型: int WINAPI ISO15693_Lock_Block(unsigned short icdev,

unsigned char model,
unsigned char *pUID,
unsigned char block)

参数: icdev: [IN] 通讯设备标识符

model: [IN] bit0=Select_flag,bit1=Addres_flag,bit2=Option_flag

pUID: [IN] UID 8字节

block: [IN] 块号

返回：成功则返回0

说明：Select_flag = 1，只有处于SELECT状态的卡执行该命令

Addres_flag = 1，只有UID符合的卡执行该命令

如操作的是TI卡片设Option_flag = 1，如操作的是I.CODE SLI卡片设Option_flag = 0

8.5.11 INT WINAPI ISO15693_WRITE_AFI

功能：ISO15693_Write_AFI

原型：int WINAPI ISO15693_Write_AFI(unsigned short icdev,
 unsigned char model,
 unsigned char *pUID,
 unsigned char AFI)

参数：icdev: [IN] 通讯设备标识符

model: [IN] bit0=Select_flag,bit1=Addres_flag,bit2=Option_flag

pUID: [IN] UID 8字节

AFI: [IN] 要写入的AFI

返回：成功则返回0

说明：Select_flag = 1，只有处于SELECT状态的卡执行该命令

Addres_flag = 1，只有UID符合的卡执行该命令

如操作的是TI卡片设Option_flag = 1，如操作的是I.CODE SLI卡片设Option_flag = 0

8.5.12 INT WINAPI ISO15693_LOCK_AFI

功能：ISO15693_Lock_AFI

原型：int WINAPI ISO15693_Lock_AFI(unsigned short icdev,
 unsigned char model,
 unsigned char *pUID)

参数：icdev: [IN] 通讯设备标识符

model: [IN] bit0=Select_flag,bit1=Addres_flag,bit2=Option_flag

pUID: [IN] UID 8字节

返回：成功则返回0

说明：Select_flag = 1，只有处于SELECT状态的卡执行该命令

Addres_flag = 1，只有UID符合的卡执行该命令

如操作的是TI卡片设Option_flag = 1，如操作的是I.CODE SLI卡片设Option_flag = 0

8.5.13 INT WINAPI ISO15693_WRITE_DSFD

功能: ISO15693_Write_DSFD

原型: int WINAPI ISO15693_Write_DSFD(unsigned short icdev,
unsigned char model,
unsigned char *UID,
unsigned char DSFD)

参数: icdev: [IN] 通讯设备标识符

model: [IN] bit0=Select_flag,bit1=Addres_flag,bit2=Option_flag

pUID: [IN] UID 8字节

DSFD: [IN] 要写入的DSFD

返回: 成功则返回0

说明: Select_flag = 1, 只有处于SELECT状态的卡执行该命令

Addres_flag = 1, 只有UID符合的卡执行该命令

如操作的是TI卡片设Option_flag = 1, 如操作的是I.CODE SLI卡片设Option_flag = 0

8.5.14 INT WINAPI ISO15693_LOCK_DSFD

功能: ISO15693_Lock_DSFD

原型: int WINAPI ISO15693_Lock_DSFD(unsigned short icdev,
unsigned char model,
unsigned char *pUID)

参数: icdev: [IN] 通讯设备标识符

model: [IN] bit0=Select_flag,bit1=Addres_flag,bit2=Option_flag

pUID: [IN] UID 8字节

返回: 成功则返回0

说明: Select_flag = 1, 只有处于SELECT状态的卡执行该命令

Addres_flag = 1, 只有UID符合的卡执行该命令

如操作的是TI卡片设Option_flag = 1, 如操作的是I.CODE SLI卡片设Option_flag = 0

故障及解决方法:

序号	故障现象	解决方法
1	开机时蜂鸣器不响	如果红色LED不亮看看是不是没有加上电,插其它USB端口试试
2	开机时红色LED不亮	如果蜂鸣器响则是LED故障

3	开机时蜂鸣器响个不停	拔掉USB接口后重新再插上
4	读不到卡	先检查电源是否同上，如果是，检查读卡器下面是不是金属，如果是则移开或垫高2CM以上。否则需要判断是不是对应的卡或者卡已经损坏。
5	能读到卡的ID号却读不了其它数据	如果是带密钥的卡，比如mifare卡，要考虑是不是密码不对，如果密码正确则有可能对应区损坏。
6	同时读几张卡读不了	如果是带有防冲突的协议的卡，则分开一些距离试一试。

如果这些故障无法排除，请尽快联系厂家。

【注意事项】

1、操作卡片之前请多参考相应的卡片资料，涉及密钥部分请谨慎操作，在写卡期间一定不能移开卡，否则容易丢失数据，我们对因违反操作规程产生的问题概不负责。

2、如果读卡器附近有大量金属等物可能会影响读写距离，请注意保持读卡器距离金属面1cm 以上。如遇特殊情况请及时联系我们的技术人员。

3、请放在儿童不能接触到的地方以免引起可能的伤害。

4、这个设备不具备防水防砸功能，所以请保护。

5、请按照使用说明小心操作，请不要私自打开设备外壳，一旦私自开启后果自负。

6、接口器件裸露时请注意静电的防护和漏电危险，我们不对因为非法操作引起的器件损坏和伤害负责。

如果有疑问请及时和我们取得联系。

非常感谢您使用本产品！欢迎给您给我们多提意见和建议，我们将不断的改进我们的产品和服务！

谢谢！

联系方式：

北京易火眼科技有限公司

地址：北京市海淀区清缘西里6号楼605室

Tel: 010-59870151;010-59754725

FAX: 010-59754725

E-mail: info@ehuoyan.com

网站: www.ehuoyan.com
