

13.56MHz Reader/Writer

YHY632



参考手册

Version 3.2.1

Jul 2008

北京易火眼科技有限公司

1. 概述

YHY632 射频卡读写器是使用 PHILIPS 生产的 MF RC632 射频基站的非接触 IC 卡读写器，可以读写符合 ISO14443A、14443B 和 15693 标准的非接触 IC 卡，支持的卡片种类有：Mifare One S50、S70、UltraLight、Mifare Pro、以及符合 ISO14443 TYPE-A 的兼容卡、AT88RF020、SR176、SR1X4K 及符合 ISO14443B-4 协议的 CPU 卡、I.CODE SLI、Tag_it 等。

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。

2、适用范围

- a、小区门禁、巡更、会员 VIP 卡系统；
- b、公司考勤，出入管理，身份识别；
- c、公交 IC 卡，医疗以及防伪；
- d、一卡通，停车场管理；
- e、公司、校园食堂饭卡等；
- f、水、电、煤气表、自动售货机、电梯、饮水机、电话机等计费系统或身份识别系统应用。

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、在线升级功能

此设备可以通过实现在线 ISP 升级功能。

5、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	60	100	120	mA
工作温度	-20	25	+60	°C
存储温度	-35	25	+85	°C

- 频率：13.56MHz
- 读卡距离：0—6CM
- 串口波特率：4800 ~ 115200 BPS ，默认19200bps
- 尺寸：95×70×18（mm）
- 重量：60g

6、操作

YHY632读卡器采用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



系统加电后先自检，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，知道通过主控制器关闭它。接着系统开始进入工作状态，可以接收主控制器的指令。

7. SDK的使用

双击rfidxray.exe进入DEMO。

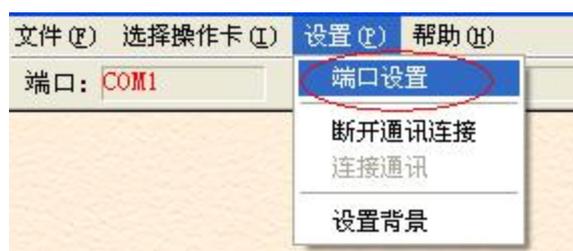
在DEMO界面里，每次点击按键所调用的库函数会出现在屏幕下部，并在屏幕右上角提示执行结果

7.1 端口设置

首先选择正确的串口号，点击【连接】，使读卡器和PC建立起通讯

点击【读取】产品信息键，可查看读卡器的具体的型号及所支持的卡片

片



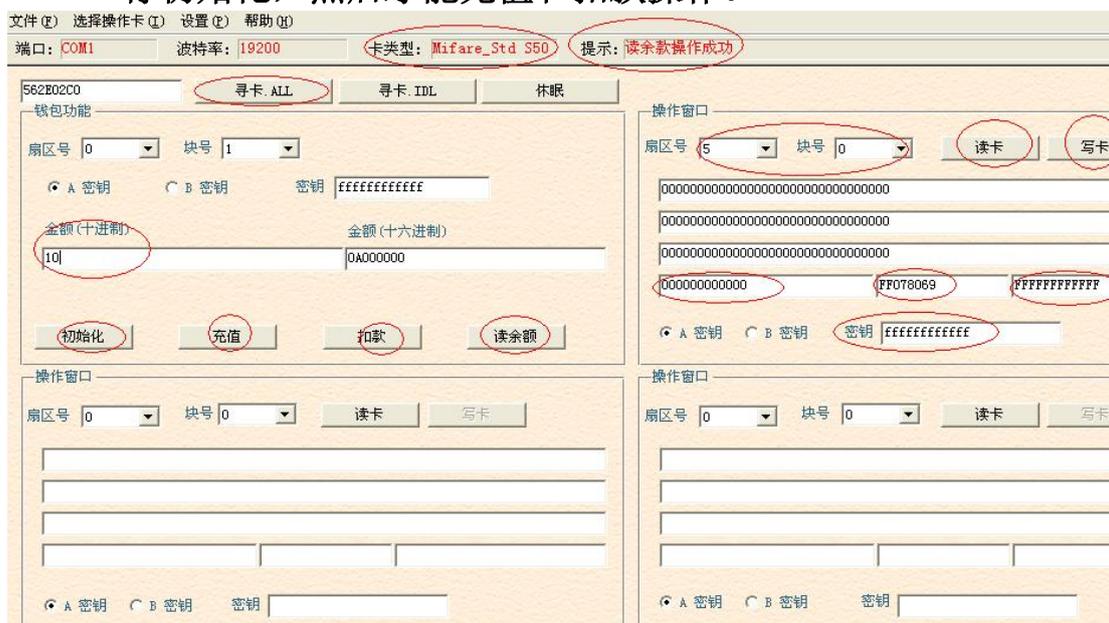
卡操作举例（请参考相应卡的 datasheet）:

mifare 卡操作

出厂密码: **FFFFFFFFFFFF**



- Step1:** 先选择 s50 卡进行操作。
- Step2:** 在天线感应区放上卡，然后点击“寻卡”，此时可以读到 4 个字节的 ID 号。
- Step3:** 读卡，选择某一个扇区，输入密码，开始空白卡用默认密码。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，输入密码，然后就可以写数据了。对于块 3，因为是密码块，开始没有熟练之前请不要写它，防止忘记密码后这个扇区作废！修改密码的操作就是写块 3 的操作。详情请参考 datasheet。
- Step5:** 钱包操作，每次对于空白卡，操作之前需要对钱包进行初始化，然后才能充值和扣款操作。



下面是不同卡的操作界面：

7.2 SHC1102

点击【寻卡】键，可得到卡片的序列号

输入正确的密钥，可对卡片进行读写操作



7.3 UltraLight

点击【寻卡】键，可得到卡片的序列号

选择相应的地址，可对卡片进行读写操作



7.4 Mifare_Std S50

点击【寻卡】键，可得到卡片的序列号

输入正确的密钥，可对卡片进行读、写、充值、扣款操作

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



7.5 Mifare_Std S70

点击【寻卡】键，可得到卡片的序列号

输入正确的密钥，可对卡片进行读、写、充值、扣款操作



7.6 Mifare_ProX

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



7.7 TRH1064

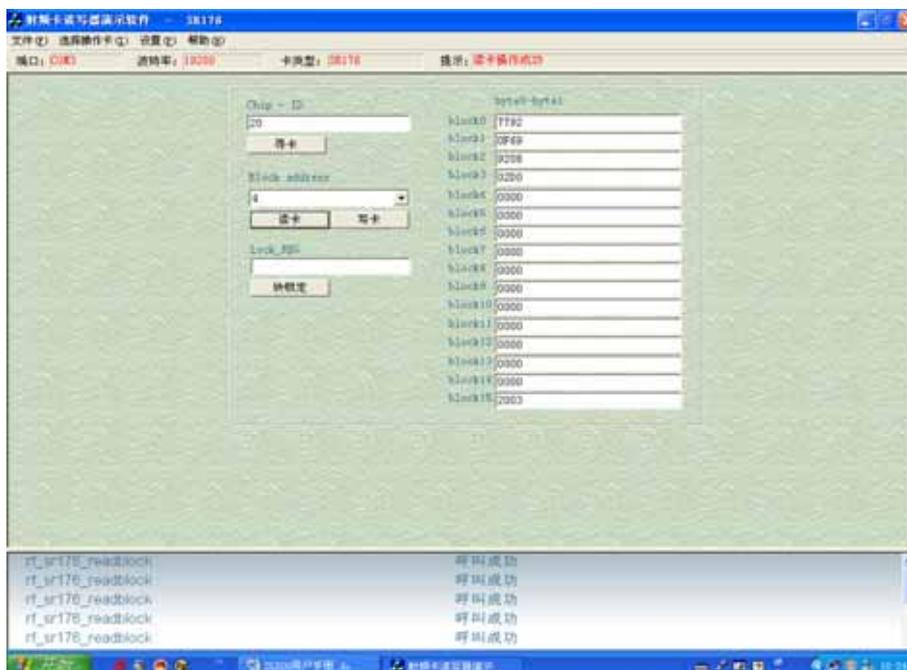
点击【寻卡】键，可得到卡片的序列号，此后可对卡片进行读、写和认证操作



7.8 SR176

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



7.9 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



7.10 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、

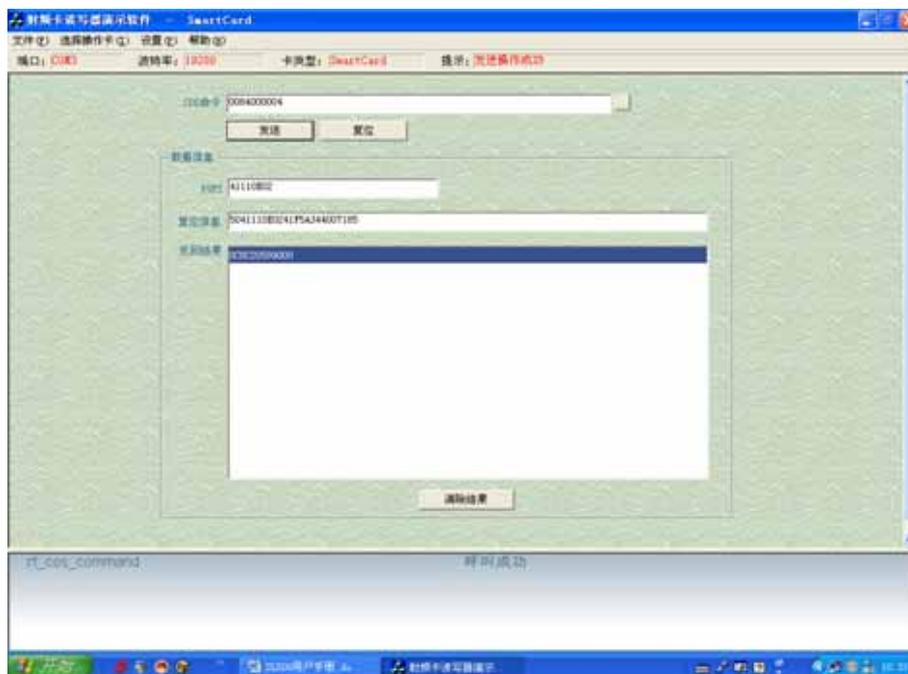
写、签名和块锁定操作



7.11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



7.12 Test_TypeB

在这个界面里，读卡器不做任何协议层的处理，只是按ISO14443B方式和

卡片建立一个数据传输通道，对应的库函数是int WINAPI rf_transceive



7.13 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，可对该卡进行读写操作



7.14 Tag_IT

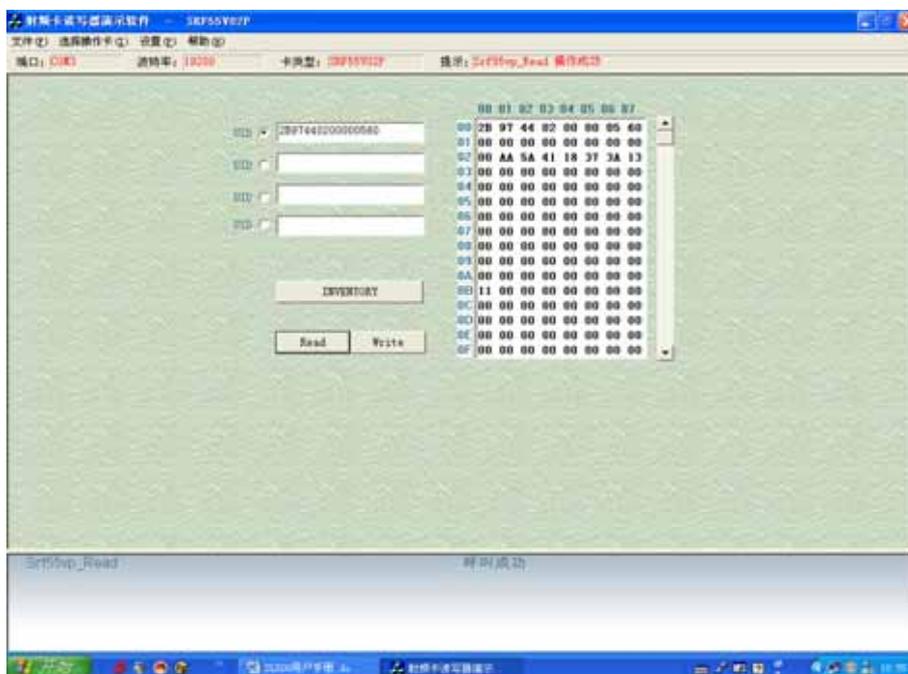
点击【INVENTORY】键，可得到卡片的序列号，最多可操作4张卡片

根据UID选定某一个卡，可对该卡进行读写操作



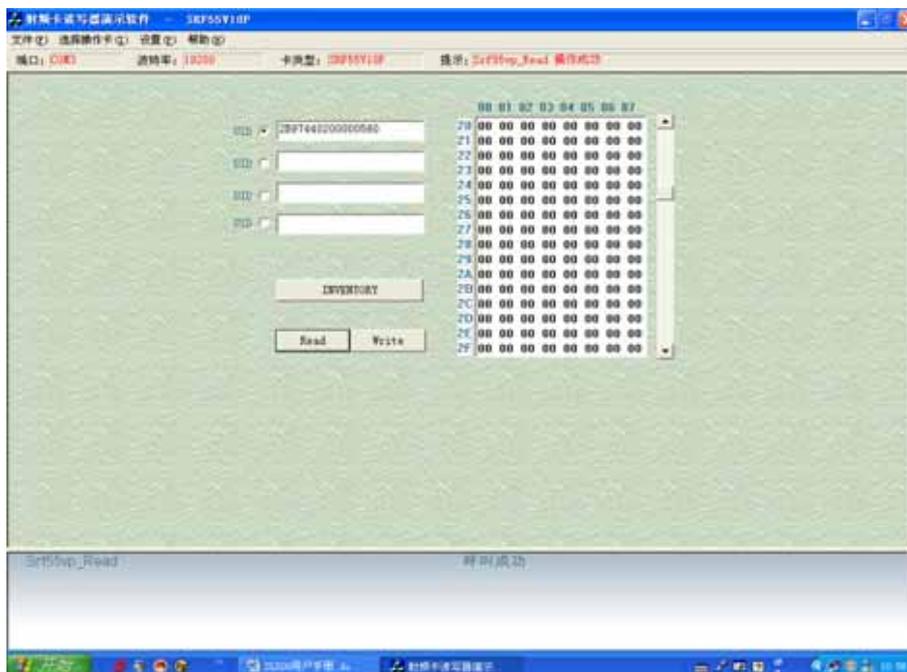
7.14 SRF55V02P

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



7.15 SRF55V10P

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

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 = 's': 设置为ST卡方式

type = 'I': 设置为ISO15693方式

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

Mifare_DESFire:

1. int WINAPI rf_DESFire_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

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

SHC1102:

1. int WINAPI rf_request
- SCH1102卡无防冲撞指令，寻卡成功后即处于激活状态

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] 密码验证模式
 block: [IN] 要验证密码的绝对块号
 pKey: [IN] 密钥内容, 6 字节

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

8.3.3 Mifare_DESFire

8.3.3.1 INT WINAPI RF_DESFIRE_RST

功能: 寻感应区内DESFire卡并复位

原型: int WINAPI rf_DESFire_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
pData: [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 = 7字节CSN + 复位信息内容

8.3.3.2 INT WINAPI RF_COS_COMMAND

功能: 向符合DESFire卡发送指令

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] 命令内容

cmdLen: [IN] 命令长度

pDate: [OUT] 卡片返回的数据

pMsgLg: [OUT] 返回数据长度

返回: 成功则返回0

8.3.4 Mifare_ProX

8.3.7.1 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

8.3.7.2 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

8.3.7.3 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

8.3.5 SHC1102

8.3.5.1 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

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: 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

8.4 ISO14443B函数

8.7.1 THR1064

8.7.1.1 INT WINAPI RF_TYPEB_RST

功能: 激活THR1064卡

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,8字节卡片序列号 + 4字节应用数据

pMsgLg: [OUT] 返回数据的长度

返回: 成功则返回0

8.7.1.2 INT WINAPI RF_THR1064_READ

功能: 读取THR1064卡一页数据

原型: int WINAPI rf_thr1064_read(unsigned short icdev,
 unsigned char page,
 unsigned char *pData,
 unsigned char *pMsgLen)

参数: icdev: [IN] 通讯设备标识符

page: [IN] 页地址, (0~3)

pDate: [OUT] 返回的数据

pMsgLen: [OUT] 返回数据的长度

返回: 成功返回0

8.4.1.3 INT WINAPI RF_THR1064_WRITE

功能: 写入THR1064卡一页数据

返回: 成功返回0

8.4.2.3 INT WINAPI RF_AT020_COUNT

功能: AT88RF020卡计数

原型: int WINAPI rf_at020_count(unsigned short icdev, unsigned char *pData)

参数: icdev: [IN] 通讯设备标识符

pDate: [IN] 签名, 6字节

返回: 成功返回0

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

功能: 读取AT88RF020卡一页数据

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

参数: icdev: [IN] 通讯设备标识符

page: [IN] 页地址, (0~31)

pDate: [OUT] 返回的数据

pMsgLen: [OUT] 返回数据的长度

返回: 成功返回0

8.4.2.5 INT WINAPI RF_AT020_WRITE

功能: 写入AT88RF020卡一页数据

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

功能: AT88RF020卡LOCK操作

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: SRIX4K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

8.4.3.9 INT WINAPI RF_ST_COMPLETION

功能: 命令ST卡进入DESACTIVED状态

原型: int WINAPI rf_st_completion(unsigned short icdev)

参数: icdev: [IN] 通讯设备标识符

返回: 成功则返回0

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 *pRxData,
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

pRxData: [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)

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Option_flag = 0

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Option_flag = 0

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=Address_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

Address_flag = 1, 只有UID符合的卡执行该命令

设Option_flag = 0

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=Address_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

Address_flag = 1, 只有UID符合的卡执行该命令

设Option_flag = 0

8.5.9 INT WINAPI ISO15693_WRITE

功能: ISO15693_Write

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

Address_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

Address_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

DSFID: [IN] 要写入的DSFID

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

Adres_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=Adres_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

Adres_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

8.6 Infineon电子标签专用函数

8.8.1 INT WINAPI SRF55VP_READ

功能: SRF55XXP读PAGE

原型: int WINAPI Srf55vp_Read(unsigned short icdev,
unsigned char *pUID,
unsigned char page,
unsigned char *pData,
unsigned char *pLen)

参数: icdev: [IN] 通讯设备标识符

pUID: [IN] UID 8字节

page: [IN] 地址

pData: [OUT] 返回的数据

unsigned char *pRdata,
unsigned char *pLen)

参数: icdev: [IN] 通讯设备标识符
pUID: [IN] UID 8字节
page: [IN] 地址
pWdata: [IN] 要写入的数据, 8字节
pWdata: [OUT] 返回的数据
pLen: [OUT] 返回数据的长度

返回: 成功则返回0

【注意事项】

1、操作卡片之前请多参考相应的卡片资料,涉及密码部分请谨慎操作,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。

申明: 请按照使用说明小心操作,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!

FILE END
