

13.56MHz Reader/Writer

SL500

参考手册

Version 2.1

Dec 2006

StrongLink

13.56MHz Reader/Writer

SL500

参考手册

Version 2.1
Dec 2006
StrongLink

目录

1. 概述.....	5
2. 型号分类和说明.....	6
3. 与PC的连接.....	6
3.1 SL500-RS232.....	6
3.2 SL500-USB.....	7
4. SDK的安装和使用.....	8
5. DEMO界面.....	8
5.2 SHC1102.....	9
5.3 UltraLight.....	9
5.4 Mifare Std S50.....	10
5.5 Mifare Std S70.....	10
5.6 Mifare ProX.....	11
5.7 TRH1064.....	11
5.8 SR176.....	12
5.9 SR1X4K.....	12

5.10 AT88RF020	13
5.11符合ISO14443B-4协议的CPU卡	13
5.12 Test TypeB	14
5.13 I.CODE SLI	14
5.14 Tag IT	15
5.14 SRF55V02P	15
5.15 SRF55V10P	16
6. 库函数说明	17
6.1 系统函数	17
6.1.1 INT WINAPI LIB VER	17
6.1.2 INT WINAPI RF INIT COM	17
6.1.3 INT WINAPI RF CLOSEPORT	17
6.1.4 INT WINAPI RF GET MODEL	17
6.1.5 INT WINAPI RF INIT DEVICE NUMBER	17
6.1.6 INT WINAPI RF GET DEVICE NUMBER	18
6.1.7 INT WINAPI RF INIT TYPE	18
6.1.8 INT WINAPI RF ANTENNA STA	18
6.1.9 INT WINAPI RF LIGHT	18
6.1.10 INT WINAPI RF BEEP	18
6.2 加密函数	19
6.2.1 INT WINAPI DES ENCRYPT	19
6.2.2 INT WINAPI DES DECRYPT	19
6.3 ISO14443A 函数	19
6.3.1 UltraLight	19
6.3.1.1 INT WINAPI RF REQUEST	19
6.3.1.2 INT WINAPI INT RF UL SELECT	20
6.3.1.3 INT WINAPI RF M1 READ	20
6.3.1.4 INT WINAPI INT RF UL WRITE	20
6.3.1.5 INT WINAPI RF HALT	20
6.3.2 Mifare Std	21
6.3.2.1 INT WINAPI RF REQUEST	21
6.3.2.2 INT WINAPI RF ANTICOLL	22
6.3.2.3 INT WINAPI RF SELECT	22
6.3.2.4 INT WINAPI RF M1 AUTHENTICATION2	22
6.3.2.5 INT WINAPI RF M1 READ	23
6.3.2.6 INT WINAPI RF M1 WRITE	23
6.3.2.7 INT WINAPI RF M1 INITVAL	23
6.3.2.8 INT WINAPI RF M1 READVAL	23
6.3.2.9 INT WINAPI RF M1 INCREMENT	24
6.3.2.10 INT WINAPI RF M1 DECREMENT	24
6.3.2.11 INT WINAPI RF M1 RESTORE	24
6.3.2.12 INT WINAPI RF M1 TRANSFER	24
6.3.2.13 INT WINAPI RF HALT	24
6.3.3 Mifare DESFire	25
6.3.3.1 INT WINAPI RF DESFIRE RST	25
6.3.3.2 INT WINAPI RF COS COMMAND	25
6.3.4 Mifare ProX	25
6.3.4.1 INT WINAPI RF TYPE RST	25
6.3.4.2 INT WINAPI RF COS COMMAND	26
6.3.4.3 INT WINAPI RF CL DESELECT	26
6.3.5 SHC1102	26
6.3.5.1 INT WINAPI RF REQUEST	26
6.3.5.2 INT WINAPI RF SHC1102 AUTH	26
6.3.5.3 INT WINAPI RF SHC1102 READ	27
6.3.5.4 INT WINAPI RF SHC1102 WRITE	27
6.3.5.5 INT WINAPI RF HALT	27
6.4 ISO14443B函数	27

6.4.1 THR1064	27
6.4.1.1 INT WINAPI RF TYPEB RST	27
6.4.1.2 INT WINAPI RF THR1064 READ	28
6.4.1.3 INT WINAPI RF THR1064 WRITE	28
6.4.1.4 INT WINAPI RF THR1064 CHECK	28
6.4.2 AT88RF020	28
6.4.2.1 INT WINAPI RF TYPEB RST	28
6.4.2.2 INT WINAPI RF AT020 CHECK	29
6.4.2.3 INT WINAPI RF AT020 COUNT	29
6.4.2.4 INT WINAPI RF AT020 READ	29
6.4.2.5 INT WINAPI RF AT020 WRITE	29
6.4.2.6 INT WINAPI RF AT020 LOCK	30
6.4.2.7 INT WINAPI RF AT020 DESELECT	30
6.4.3 SR176SR1X4K	30
6.4.3.1 INT WINAPI RF ST SELECT	30
6.4.3.2 INT WINAPI INT RF SR176 READBLOCK	30
6.4.3.3 INT WINAPI INT RF SR176 WRITEBLOCK	30
6.4.3.4 INT WINAPI INT RF SR176 PROTECTBLOCK	31
6.4.3.5 INT WINAPI INT RF SR1X4K GETUID	31
6.4.3.6 INT WINAPI INT RF SR1X4K READBLOCK	31
6.4.3.7 INT WINAPI INT RF SR1X4K WRITEBLOCK	31
6.4.3.8 INT WINAPI INT RF SR1X4K PROTECTBLOCK	32
6.4.3.9 INT WINAPI RF ST COMPLETION	32
6.4.4 TYPE B SmartCard	32
6.4.4.1 INT WINAPI RF TYPEB RST	32
6.4.4.2 INT WINAPI RF COS COMMAND	33
6.4.4.3 INT WINAPI RF CL DESELECT	33
6.4.4.4 INT WINAPI RF TRANSCEIVE	33
6.5 ISO15693函数	34
6.5.1 INT WINAPI ISO15693 INVENTORY	34
6.5.2 INT WINAPI ISO15693 INVENTORYS	34
6.5.3 INT WINAPI ISO15693 GET SYSTEM INFORMATION	34
6.5.4 INT WINAPI ISO15693 SELECT	34
6.5.5 INT WINAPI ISO15693 RESET TO READY	35
6.5.6 INT WINAPI ISO15693 STAY QUIET	35
6.5.7 INT WINAPI ISO15693 GET BLOCK SECURITY	35
6.5.8 INT WINAPI ISO15693 READ	36
6.5.9 INT WINAPI ISO15693 WRITE	36
6.5.10 INT WINAPI ISO15693 LOCK BLOCK	36
6.5.11 INT WINAPI ISO15693 WRITE AFI	37
6.5.12 INT WINAPI ISO15693 LOCK AFI	37
6.5.13 INT WINAPI ISO15693 WRITE DSFID	37
6.5.14 INT WINAPI ISO15693 LOCK DSFID	38
6.6 Infineon电子标签专用函数	38
6.6.1 INT WINAPI SRF55VP READ	38
6.6.2 INT WINAPI SRF55VP WRITEBYTE	38
6.6.3 INT WINAPI SRF55VP WRITE	39
6.6.4 INT WINAPI SRF55VP WRITE REREAD	40

1. 概述

- ◇ 接口：RS232 or USB
- ◇ 工作电压：DC4.5V – DC5.5V

- ◇ 工作电流: 150MA
- ◇ 尺寸: 110 x 81 x 26 mm
- ◇ 工作频率: 13.56MHz
- ◇ 支持协议: ISO14443A ISO1443B ISO15693
- ◇ 兼容系统: Windows 32
- ◇ 工作温度: -20℃ - +50℃
- ◇ 存储温度: -25℃ - +60℃
- ◇ 重量: 100g

2. 型号分类和说明

SL500系列读卡器支持ISO14443A, ISO14443B和ISO15693协议。根据读卡器型号的最后一位字母, 分为以下几款, 如下表所示

	SL500L	SL500A	SL500D	SL500E	SL500F
ISO14443A	√	√		√	√
ISO14443B				√	√
ISO15693			√		√

注: SL500L和SL500A的差别

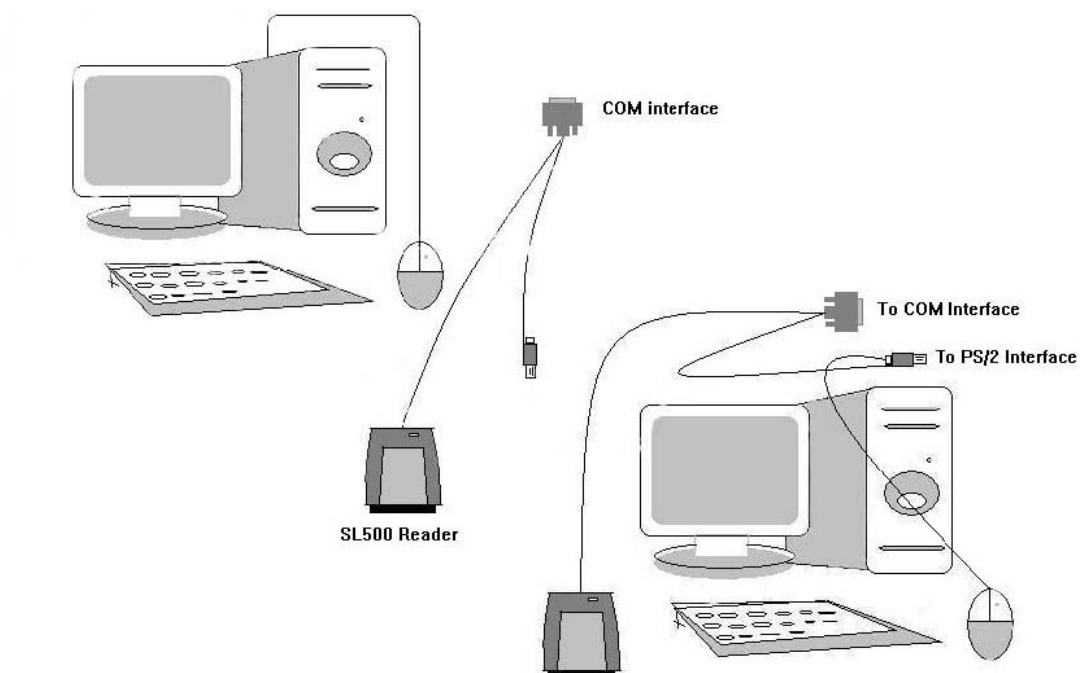
SL500L支持的卡片: Mifare_1K、Mifare_4K、Mifare_UltraLight

SL500A支持的卡片: Mifare_1K、Mifare_4K、Mifare_UltraLight、Mifare_ProX

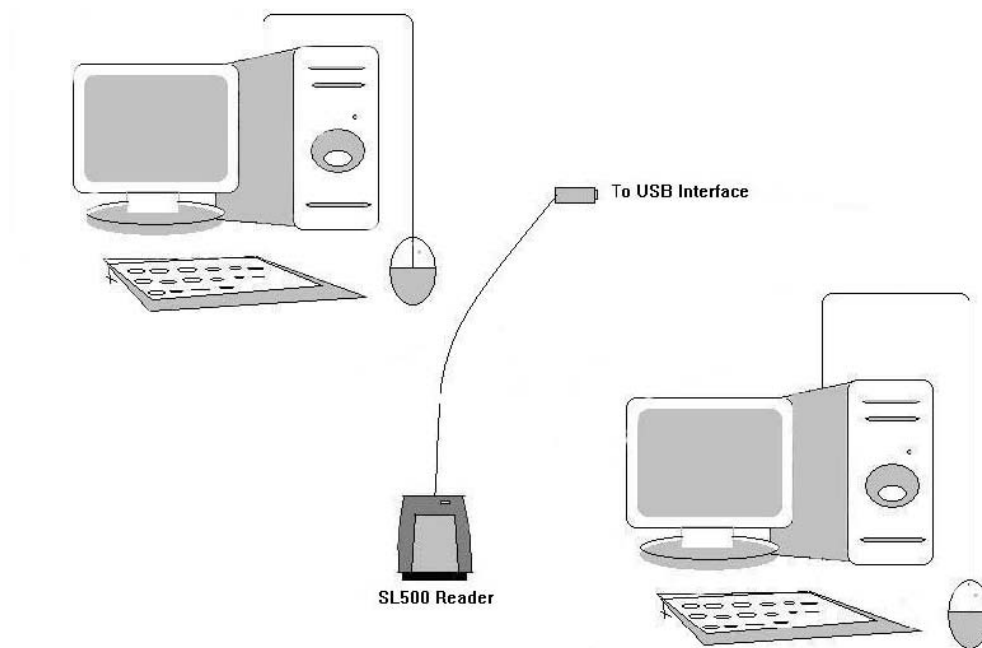
3. 与PC的连接

3.1 SL500-RS232

RS232接口读卡器从PS/2口取电

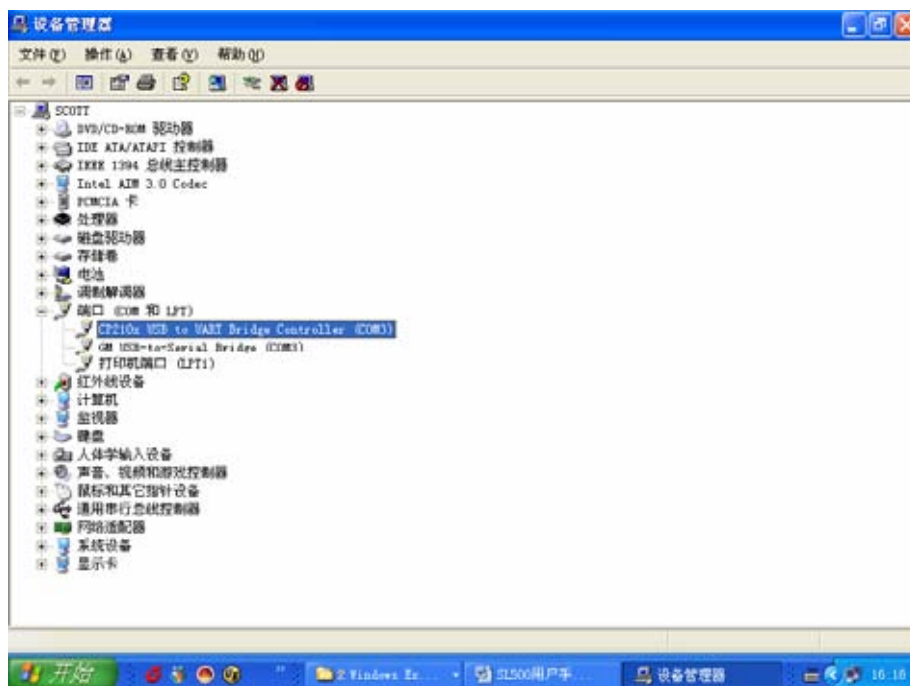


3.2 SL500-USB



SL500 - USB读卡器采用USB Bridge to COM方式，将SL500连接到PC的USB接口，安装驱动之后，虚拟出一个COM，此后的操作和SL500-RS232完全一样。

可在“设备管理器”中查看虚拟出的COM号，如下图



4. SDK的安装和使用

执行InstDemo.exe可将读卡器DEMO软件和动态库安装到PC机上，并在桌面上生成相应图标

默认的安装目录是C:\RFREADER，包含以下内容

C:\RFREADER\Sample	各种高级语言调用库函数的例子源代码
C:\RFREADER\UsuDriver	USB接口驱动
C:\RFREADER\ICTransfer.exe	DEMO软件
C:\RFREADER\MasterRD.dll	应用接口动态库
C:\RFREADER\MasterCOM.dll	串口管理动态库，被MasterRD.dll调用
C:\RFREADER\SL_Paper.dll	DEMO软件背景库
C:\RFREADER\UserManual.pdf	本手册
C:\RFREADER\RFHELP.chm	动态库说明，CHM格式
C:\RFREADER\AppConfig.ini	DEMO软件配置文件

5. DEMO界面

双击桌面上的读卡器图标进入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

5.1 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



5.2 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



5.3 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



5.4 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



5.5 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



5.6 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



5.7 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



5.8 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



5.9 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



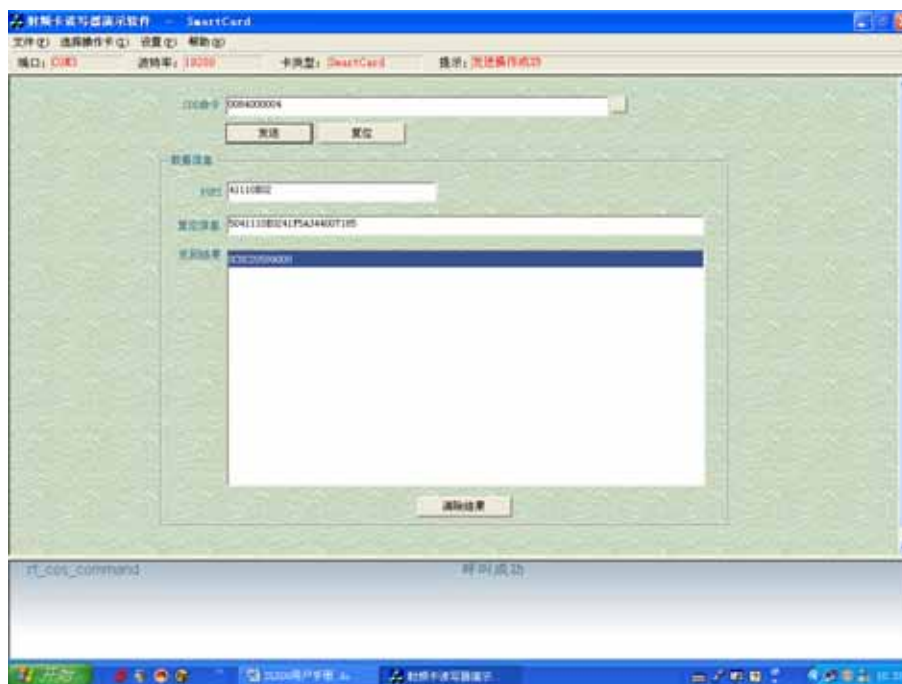
5.10 AT88RF020

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



5.11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



5.12 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



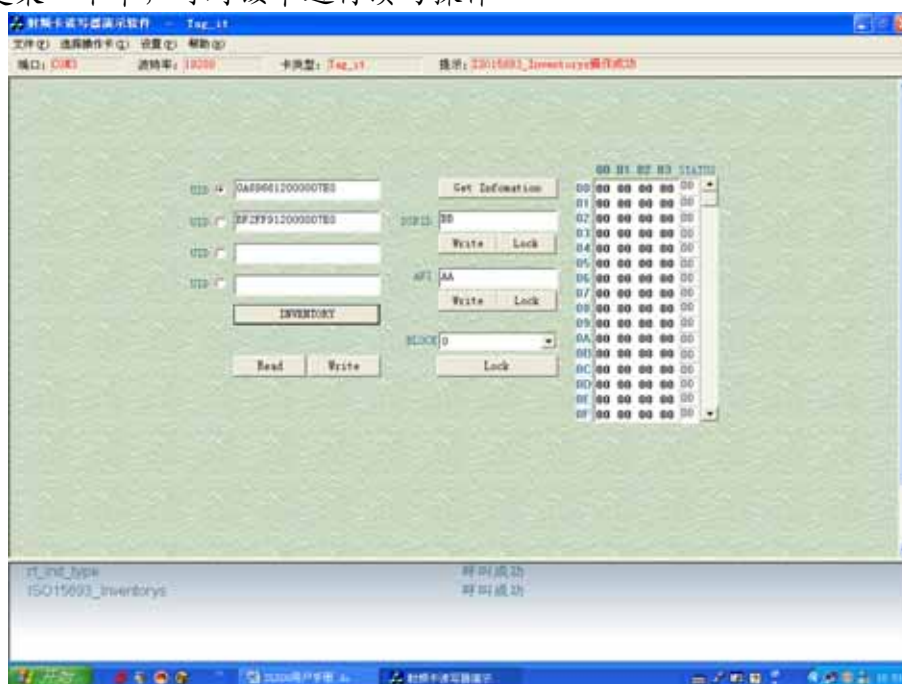
5.13 I.CODE SLI

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



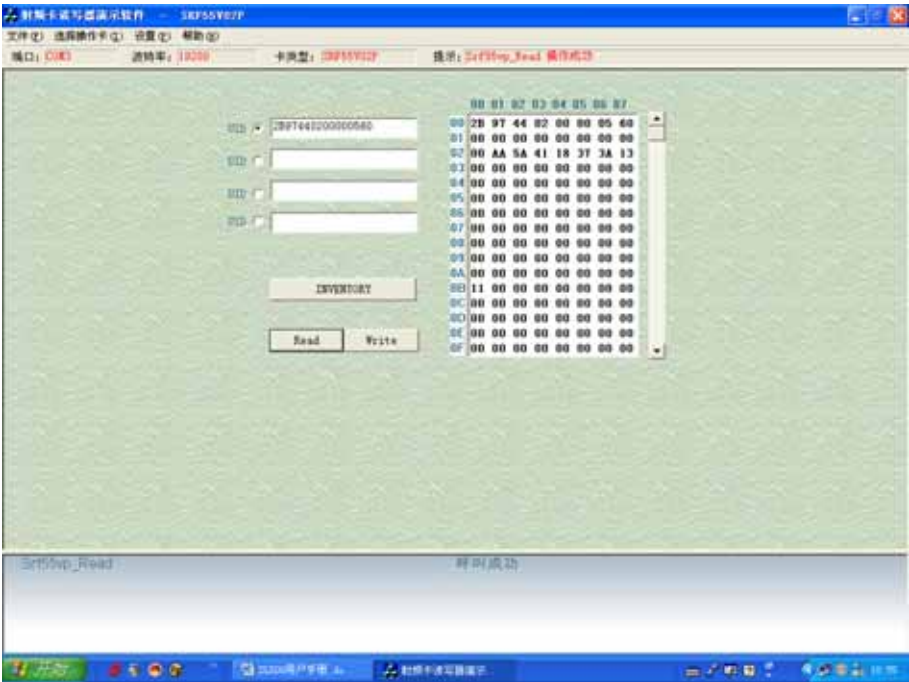
5.14 Tag_IT

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



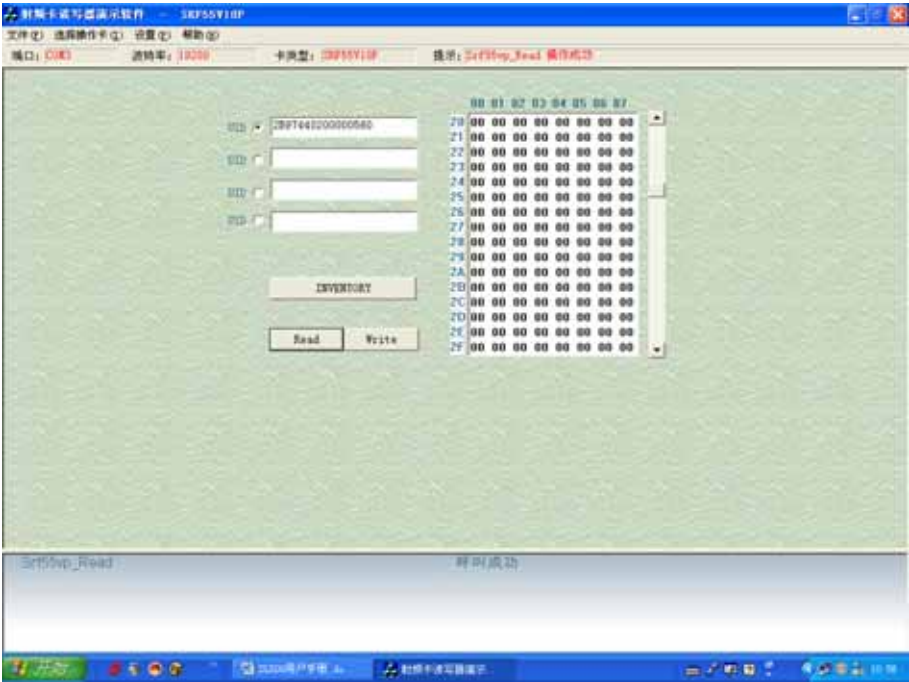
5.14 SRF55V02P

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



5.15 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



6. 库函数说明

系统函数和加密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

6.1 系统函数

6.1.1 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

6.1.2 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

返回: 成功返回0

6.1.3 INT WINAPI RF_CLOSEPORT

功能: 关闭Com端口

原型: int WINAPI rf_ClosePort()

返回: 成功返回0

6.1.4 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

6.1.5 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: SL系列读卡器支持组网功能, 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。

6.1.6 INT WINAPI RF_GET_DEVICE_NUMBER

功能: 读取设备标识

原型: int WINAPI rf_get_device_number(unsigned short *pIcdev)

参数: pIcdev: [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

6.1.7 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 = 's': 设置为ST卡方式

type = 'l': 设置为ISO15693方式

6.1.8 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

6.1.9 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: SL300L型读卡器无此功能

6.1.10 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

6.2 加密函数

6.2.1 INT WINAPI DES_ENCRYPT

功能: DES算法加密函数

原型: int WINAPI des_encrypt (unsigned char *pSzOut,
 unsigned char *pSzIn,
 unsigned int inlen,
 unsigned char *pKey,
 unsigned int keylen)

参数: pSzOut: [OUT] 返回的DES加密结果, 长度等于明文长度

pSzIn: [IN] 明文

inlen: [IN] 明文长度,8字节的整数倍

pKey: [IN] 密钥

keylen: [IN] 密钥长度,如果大于8字节, 是3des,如果小于等于8字节单des.

不足

补零

返回：成功返回0

6.2.2 INT WINAPI DES_DECRYPT

功能：DES算法解密算法函数

原型：int WINAPI des_decrypt (unsigned char *pSzOut,
 unsigned char *pSzIn,
 unsigned int inlen,
 unsigned char *pKey,
 unsigned int keylen)

参数：pSzOut: [OUT] 返回的DES解密结果，长度等于密文长度

pSzIn: [IN] 密文

inlen: [IN] 密文长度,8字节的整数倍

pKey: [IN] 密钥

keylen: [IN] 密钥长度,如果大于8字节，是3des,如果小于等于8字节单des.

不足

补零

返回：成功返回0

6.3 ISO14443A 函数

6.3.1 UltraLight

6.3.1.1 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

6.3.1.2 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

6.3.1.3 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

6.3.1.4 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

6.3.1.5 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

6.3.2 Mifare_Std

6.3.2.1 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 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

6.3.2.2 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

6.3.2.3 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

6.3.2.4 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

返回: 成功返回0

说明: model = 0x60: 验证A密钥
model = 0x61: 验证B密钥

6.3.2.5 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

6.3.2.6 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

6.3.2.7 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

6.3.2.8 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

6.3.2.9 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

6.3.2.10 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

6.3.2.11 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

6.3.2.12 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

6.3.2.13 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

6.3.3 Mifare_DESFire

6.3.3.1 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

6.3.3.2 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

6.3.4 Mifare_ProX

6.3.4.1 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

6.3.4.2 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

6.3.4.3 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

6.3.5 SHC1102

6.3.5.1 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

6.3.5.2 INT WINAPI RF_SHC1102_AUTH

功能: SHC1102卡验证密码

原型: int WINAPI rf_Shc1102_Auth(unsigned short icdev, unsigned char *pPassword)

参数: icdev: [IN] 通讯设备标识符

pPassword: [IN] 密码, 4字节

返回: 成功返回0

说明: SCH1102卡认证操作只可在以下两种情况下进行

1. 寻卡 » 认证

2. 寻卡 » 读0-8BLOCK » 认证

其它时刻进行认证返回失败信息

6.3.5.3 INT WINAPI RF_SHC1102_READ

功能: 读取SHC1102卡一块数据

原型: int WINAPI rf_Shc1102_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] SHC1102卡块号(0x00 - 0x0F)

pData: [OUT] 返回的数据

pLen: [OUT] 返回数据的长度

返回: 成功返回0

6.3.5.4 INT WINAPI RF_SHC1102_WRITE

功能: 写SHC1102卡一块数据

原型: int WINAPI rf_Shc1102_Write(unsigned short icdev,
 unsigned char block,
 unsigned char *pData)

参数: icdev: [IN] 通讯设备标识符

block: [IN] SHC1102卡块号(0x00 - 0x0F)

pData: [IN] 要写入的数据4字节

返回: 成功返回0

6.3.5.5 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: 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

6.4 ISO14443B函数

6.4.1 THR1064

6.4.1.1 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

6.4.1.2 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

6.4.1.3 INT WINAPI RF_THR1064_WRITE

功能: 写入THR1064卡一页数据

原型: int WINAPI rf_thr1064_write(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: [IN] 要写入的数据

pMsgLen: [OUT] 写入数据的长度

返回: 成功返回0

6.4.1.4 INT WINAPI RF_THR1064_CHECK

功能: 验证THR1064卡密码

原型: int WINAPI rf_thr1064_check(unsigned short icdev, unsigned char *pKey)

参数: icdev: [IN] 通讯设备标识符

pKey: [IN] 密码, 8 字节

返回: 成功返回0

6.4.2 AT88RF020

6.4.2.1 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

6.4.2.2 INT WINAPI RF_AT020_CHECK

功能：验证AT88RF020卡密码

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] 密码，8 字节

返回：成功返回0

6.4.2.3 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

6.4.2.4 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

6.4.2.5 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

6.4.2.6 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

6.4.2.7 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

6.4.3 SR176SR1X4K

6.4.3.1 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。

6.4.3.2 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

6.4.3.3 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

6.4.3.4 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

6.4.3.5 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

6.4.3.6 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

6.4.3.7 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

6.4.3.8 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

6.4.3.9 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

6.4.4 TYPE_B SmartCard

6.4.4.1 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

6.4.4.2 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

6.4.4.3 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

6.4.4.4 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

6.5 ISO15693函数

6.5.1 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

6.5.2 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

6.5.3 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=Address_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
 Address_flag = 1, 只有UID符合的卡执行该命令
 设Option_flag = 0

6.5.4 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

6.5.5 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=Address_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

设Option_flag = 0

6.5.6 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

6.5.7 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

6.5.8 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Option_flag = 0

6.5.9 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

6.5.10 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

6.5.11 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

6.5.12 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

6.5.13 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

6.5.14 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

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

6.6 Infineon电子标签专用函数

6.6.1 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

pLen: [OUT] 返回数据的长度

返回: 成功则返回0

6.6.2 INT WINAPI SRF55VP_WRITEBYTE

功能: SRF55XXP写1BYTE

原型: int WINAPI Srf55vp_WriteByte(unsigned short icdev,
 unsigned char *pUID,
 unsigned char page,
 unsigned char byteaddr,
 unsigned char data)

参数: icdev: [IN] 通讯设备标识符

pUID: [IN] UID 8字节

page: [IN] 地址

byteaddr: [IN] 要写字节在PAGE中的偏移地址, 0 - 8

data: [IN] 要写入的数据

返回: 成功则返回0

6.6.3 INT WINAPI SRF55VP_WRITE

功能: SRF55XXP写PAGE

原型: int WINAPI Srf55vp_Write(unsigned short icdev,
 unsigned char *pUID,
 unsigned char page,
 unsigned char *pData)

参数: icdev: [IN] 通讯设备标识符

pUID: [IN] UID 8字节

page: [IN] 地址

pData: [IN] 要写入的数据, 8字节

返回: 成功则返回0

6.6.4 INT WINAPI SRF55VP_WRITE_REREAD

功能: SRF55XXP写PAGE并返回写入后该PAGE的实际数据

原型: int WINAPI Srf55vp_Write_Reread(unsigned short icdev,

```
unsigned char *pUID,  
unsigned char page,  
unsigned char *pWdata,  
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