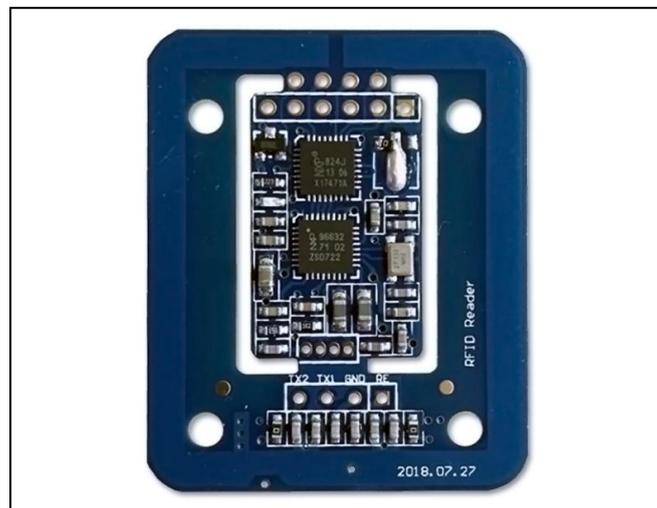




北京圆志科信 读写卡模块

应用手册



M104FET-X

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资料内容。

0.2 更改历史记录

版本	描述	日期
V1.0	第一版发布	2018.10.18
V1.1	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
V1.2	修订硬件 11-14 符号描述的错误	2019.04.08

目 录

0.1 声明	2
0.2 更改历史记录	3
1.概述	9
1.1 M104FET-X 系列读写模块:	9
1.2 产品型号及之间的区别:	9
1.3 主要特点:	10
2.硬件描述:	11
2.1 电气特性:	11
2.2 管脚说明:	11
2.3 结构尺寸:	12
3.数据通讯协议:	13
3.1 UART 协议	13
4.不同类型卡片操作流程:	14
4.1 读写模块初始化:	14
4.2 Mifare OneS50/S70 卡操作步骤:	14
4.3 Ultralight/Ntag 卡操作步骤:	15
4.4 Desfire EV1 卡操作步骤:	15
4.5 ISO14443 TYPE A/TYP E B CPU 卡操作步骤:	15
4.6 SRI512/SRI4K 卡操作步骤:	16
4.7 二代证卡操作步骤:	16
4.8 Felica 卡操作步骤:	16
4.9 ISO7816 CPU 卡/PSAM 卡操作步骤:	16
5.读卡器支持函数和指令汇总	17
5.1 系统函数指令.....	17
5.1.1 int WINAPI rf init com(初始化串口)	17
5.1.2 int WINAPI rf init type(初始化工作模式).....	18

5.1.3 int WINAPI rf antenna sta(设置天线状态).....	18
5.1.4 int WINAPI rf light(设置指示灯).....	20
5.1.5 int WINAPI rf get model(读取硬件版本号)	20
5.1.6 int WINAPI rf get snr(读取序列号)	21
5.1.7 int WINAPI rf init device number(初始设备号)	22
5.1.8 int WINAPI rf get device number(获得设备 ID 号)	23
5.2 Mifare One 卡指令	23
5.2.1 int WINAPI rf request(寻卡)	23
5.2.2 int WINAPI rf anticoll(防冲突)	24
5.2.3 int WINAPI rf select(选卡)	25
5.2.4 int WINAPI rf M1 authentication2(外部认证 2).....	26
5.2.5 int WINAPI rf M1 read(读块).mht.....	27
5.2.6 int WINAPI rf M1 read sector(读连续 3 块).mht.....	28
5.2.7 int WINAPI rf M1 write(写块).....	29
5.2.8 int WINAPI rf M1 write sector(写连续 3 块).....	30
5.2.9 int WINAPI rf M1 initval(初始化钱包)	31
5.2.10 int WINAPI rf M1 increment(充值).....	32
5.2.11 int WINAPI rf M1 decrement(扣款).....	32
5.2.12 int WINAPI rf M1 readval(读余额)	33
5.2.13 int WINAPI rf M1 restore(数据回传).....	34
5.2.14 int WINAPI rf M1 transfer(数据传送).....	35
5.2.15 int WINAPI rf halt(卡片休眠)	36
5.2.16 int WINAPI rf_download_key2(下载密钥 2)	37
5.2.17 int WINAPI rf M1 authentication3(外部认证 3).....	37
5.3 Ultralight 卡指令	38
5.3.1 int WINAPI rf request(寻卡)	38
5.3.2 int WINAPI int rf ul select(Ultralight/Ntag 选卡)	39
5.3.3 int WINAPI rf M1 read(读块).....	40
5.3.4 int WINAPI int rf ul write(Ultralight/Ntag 写块)	41
5.3.5 int WINAPI rf halt(卡片休眠).....	41
5.4 ISO14443A T=CL CPU 卡指令	42
5.4.1 int WINAPI rf typea rst(TypeACPU 卡复位)	42

5.4.2 int WINAPI rf cos command(发送 COS 命令)	43
5.4.3 int WINAPI rf_ISO14443_PPS(PPS 升频)	44
5.4.4 int WINAPI rf_cl_deselect(退出激活状态)	45
5.5 Ntag 系列指令	46
5.5.1 int WINAPI rf request(寻卡)	46
5.5.2 int WINAPI int rf ul select(Ultralight/Ntag 选卡)	47
5.5.3 int WINAPI rf M1 read(读块)	48
5.5.4 int WINAPI int rf ul write(Ultralight/Ntag 写块)	48
5.5.5 int WINAPI rf Ntag Fast Read(快速读取)	49
5.5.6 int WINAPI rf Ntag Get Version(得到版本号)	50
5.5.7 int WINAPI rf Ntag Password Auth(密码认证)	51
5.5.8 int WINAPI rf Ntag Read CNT(读取计数值)	51
5.5.9 int WINAPI rf Ntag Read Signature(读取签名)	52
5.5.10 int WINAPI rf NFC Select Sector(NFC 类型 2 标签选择扇区)	53
5.5.11 int WINAPI rf halt(卡片休眠)	54
5.6 ISO7816-4 SAM/CPU 卡指令	54
5.6.1 int WINAPI rf sam rst 2(SAM 卡复位新指令)	54
5.6.2 int WINAPI rf sam cos 2(SAM 卡 COS 发送新指令)	55
5.7 ISO14443B CPU 卡指令	56
5.7.1 int WINAPI rf atqb(复位)	56
5.7.2 int WINAPI rf cos command(发送 COS 命令)	57
5.7.3 int WINAPI rf cl deselect(退出激活状态)	58
5.7.4 int WINAPI rf_hltb(卡休眠)	58
5.8 SRI512-SRI4K 卡指令	59
5.8.1 int WINAPI rf st select(ST 卡选卡)	59
5.8.2 int WINAPI int rf sr1x4k getuid(获得 UID)	59
5.8.3 int WINAPI int rf sr1x4k readblock(读块)	60
5.8.4 int WINAPI int rf sr1x4k writeblock(写块)	61
5.8.5 int WINAPI int rf sr1512 writelockreg(SRI512 块锁定)	61
5.8.6 int WINAPI int rf sr1x4k writelockreg(SRI4K 块锁定)	62
5.8.7 int WINAPI rf st completion(退出激活)	63
5.9 二代证卡指令	64
5.9.1 int WINAPI rf get namecard uid(读取二代证 UID)	64
5.10 ISO15693 卡指令	64
5.10.1 int WINAPI ISO15693 Inventory(寻卡)	64

5.10.2 int WINAPI ISO15693 Get System Information(获得系统信息)	65
5.10.3 int WINAPI ISO15693 Read(读块)	66
5.10.4 int WINAPI ISO15693 Write(写块).....	67
5.10.5 int WINAPI ISO15693_Write_Multi_Block(写连续多块)	68
5.10.6 int WINAPI ISO15693 Get Block Security(获得块锁定信息)	69
5.10.7 int WINAPI ISO15693 Lock Block(锁块).....	70
5.10.8 int WINAPI ISO15693 Write AFI(写 AFI)	71
5.10.9 int WINAPI ISO15693 Lock AFI(锁定 AFI)	71
5.10.10 int WINAPI ISO15693 Write DSFID(写 DSFID).....	72
5.10.11 int WINAPI ISO15693 Lock DSFID(锁 DSFID)	73
5.10.12 int WINAPI ISO15693 SET EAS ICODE(设置 ICODE EAS).....	74
5.10.13 int WINAPI ISO15693 EAS ALARM ICODE(I CODE EAS 报警).....	75
5.10.14 int WINAPI ISO15693 RESET EAS ICODE(复位 ICODE EAS)	76
5.10.15 int WINAPI ISO15693 LOCK EAS ICODE(锁定 ICODE EAS)	76
5.10.16 int WINAPI ISO15693 Select(选卡).....	77
5.10.17 int WINAPI ISO15693 Inventorys(寻多张卡)	78
5.10.18 int WINAPI ISO15693 Reset To Ready(复位)	79
5.10.19 int WINAPI ISO15693 Stay Quiet(进入静默).....	80
5.11 Felica 卡指令.....	80
5.11.1 int WINAPI rf_felica_get_uid(读取 FelicaUID).....	80
6.底层通讯举例 (默认 1 字节长度举例)	81
6.1 系统函数:	81
6.2 Mifare One:	81
6.3 ISO14443 TYPE A T=CL CPU 卡:	83
6.4 Ntag213:	83
6.5 PSAM:	84
6.6 Desfire EV0/EV1:	85
6.7 设置成 ISO14443 TYPE B 的模式:	85

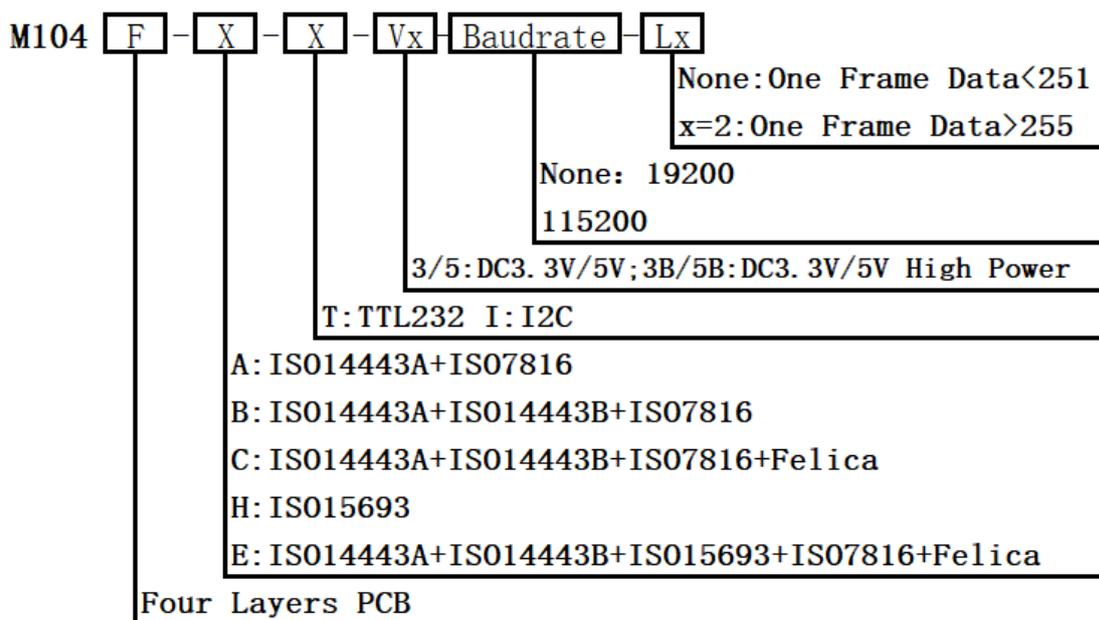
6.8 ISO14443 TYPE B CPU 卡发送接收举例:	85
6.9 设置成 ST 卡的模式:	86
6.10 SR176 卡发送接收举例:	86
6.11 设置成 ISO15693 卡的模式:	86
6.12 I CODE SL2 卡发送接收举例:	86
6.13 Felica 卡发送接收举例:	87
6.14 读取二代身份证 UID 发送接收举例:	87
附录 1:硬件连接参考	87

1.概述

1.1 M104FET-X 系列读写模块:

M104FEX 读写模块采用 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 UART 接口发送命令就可以实现对卡片完全的操作。该系列读写模块可读写 ISO14443A, ISO14443B, ISO15693, ISO7816 标准的卡片, 即 ISO14443-A Mifare One S50/S70/Ultralight/Ntag203/Ntag213/Ntag215/ntag216/FM1208Desfire/ISO14443-4 T=CL CPU 卡, ISO14443-B SRI512/SRI4K/二代证(只读卡号)/ ISO14443-4 T=CL CPU 卡, ISO15693 NXP I Code/TI Tag it, ISO7816-4 PSAM9600/PSAM38400 T=0/T=1 卡片以及满足 ISO7816-4 CPU 卡, **同时卡片具有检测 M1 卡是否为复制卡(只针对卡片, 不针对卡内内容)。**

1.2 产品型号及之间的区别:



型号	主要区别		
	供电	支持卡类型	备注
M104FETV3	UART DC3.3V 小功率	ISO14443-A: Mifare One S50,S70,UltraLight, FM11RF08, Ntag203, Ntag213,Ntag215, Ntag216,FM1208,CPU 卡(T=CL), DesfireEV1 D21,D41.D81,etc;	M104FEX I2C 参阅 另一份手 册
M104FETV5	UART DC5V 小 功率		

M104FETV3B	UART DC3.3V 大功率	ISO14443-B: SRI512,SRIX4K,CPU 卡(T=CL),二代身份证 (只读卡号) ;
M104FETV5B	UART DC5V 大 功率	ISO7816: PSAM9600 (T=0/T=1) ,PSAM38400 (T=0/T=1) ; ISO15693: I CODE SL2,Tag it; 其它协议: Felica (只读卡号) ; 及其兼容卡片;
<p>说明: (1) 波特率在上面型号后面增加波特率尾缀, 比如: M104FETV3-19200。</p> <p>(2) 模块默认长度字为 1 字节, 即每帧数据不能超过 255 字节, 如果每帧数据长度大于 255 字节, 则需要在波特率之后再增加尾缀 L2, 此应用主要针对部分特殊的 CPU 卡指令, 比如: M104FETV3-19200-L2, 对于新客户建议直接选用 2 字节长度模块。</p>		

1.3 主要特点:

- 采用 NXP高度集成芯片;
- 内嵌32位处理器;
- 4层板设计, 更有利于电磁兼容;
- 支持 ISO14443-A:

Mifare One S50,S70,Ultra Light,FM11RF08, ,Ntag203,
Ntag213,Ntag215, Ntag216,FM1208,CPU 卡(T=CL),
DesfireEV1 D21,D41.D81,etc;

ISO14443-B:

SRI512,SRIX4K,CPU 卡(T=CL),二代身份证 (只读卡号);

ISO7816:

PSAM9600 (T=0/T=1) ,PSAM38400 (T=0/T=1);

ISO15693: I CODE SL2,Tag it;

其它协议: Felica (只读卡号);

及其兼容卡片;

- 超小体积, 一体43.5x35.5mm;

分体: 25*16mm (**分体后PSAM卡不能再被使用, 引脚被破坏**);

- 读卡距离: 与不同型号卡片有关;
- 简单的命令集可完成对卡片的全部操作;

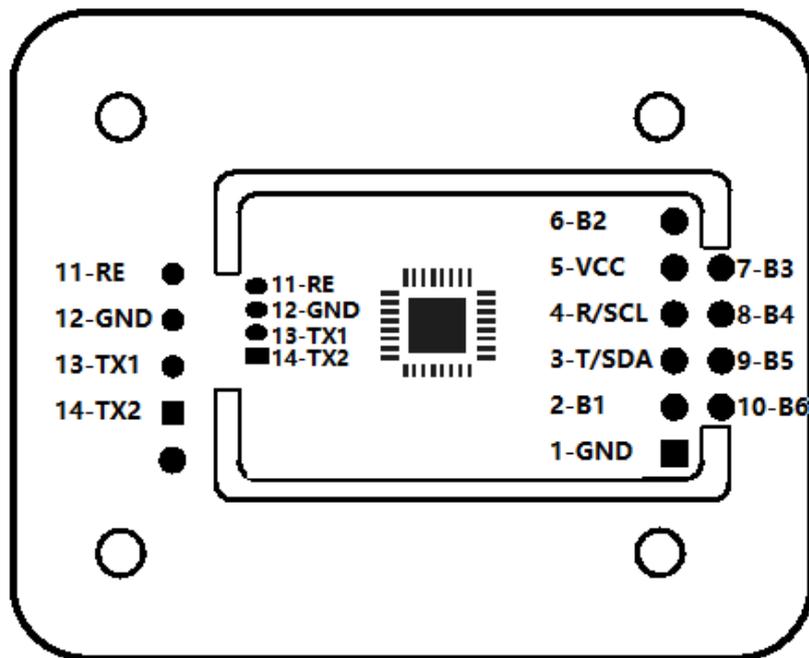
- 接口：UART (TTL232) ；
- 默认通讯速率：19.2Kbps，可通过指令调整为9.6K/14.4K/28.8K/38.4K/57.6K/115.2K；
- 可提供 C51函数库(例程)及windows操作系统下的例程函数库供二次开发；
- 基于模块的扩展功能很强可根据**用户要求修改软件定制**个性化模块,不用改变线路板；
- 自带看门狗；
- 配以KF904可立即连接电脑，通过电脑DEMO软件测试；

2.硬件描述：

2.1 电气特性：

典型工作电源： DC3.3V/DC5V；
读卡电流： 平均约 70 mA ；
读卡距离： 与卡片型号有关；
工作温度： -20-+70℃。

2.2 管脚说明：



特殊说明：2018.07.27 版本 PCB 的丝印 11-14 引脚（2.54 间距插针）顺序有误，以上图为准，在天线分体时要特别注意。

图 1：管脚示意图

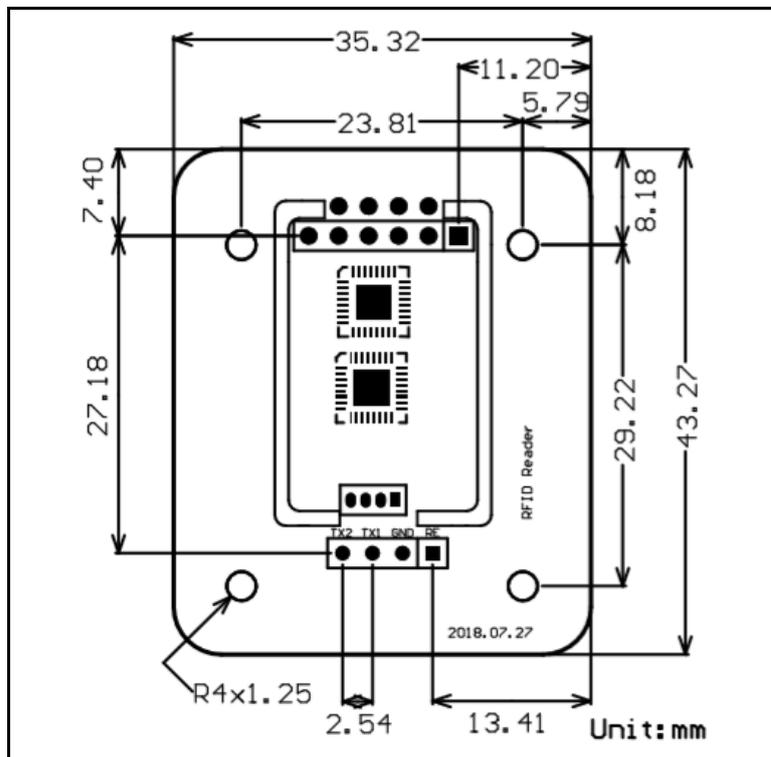
管脚	符号	描述
1	GND	电源地
2	B1	RS485 DIR控制引脚
3	TXD/SCL	UART 串口发送端
4	RXD/SDA	UART 串口接收端
5	VCC	电源 3.3V/5V
6	B2	SAM 卡时钟引脚
7	B3	2号 SAM 卡数据引脚(上电瞬间不能为低电平)
8	B4	1号 SAM 卡数据引脚
9	B5	1号 SAM 卡复位引脚
10	B6	2号 SAM 卡复位引脚
11	RE	天线数据接收管脚(天线分体时使用)
12	GND	天线地(天线分体时使用)
13	TX1	天线 1 发送管脚(天线分体时使用)
14	TX2	天线 2 发送管脚(天线分体时使用)

2.3 结构尺寸:

模块尺寸: 天线一体:43.5x35.5mm, 管脚间距: 2.54mm;

分体: 25*16mm (分体后 PSAM 卡不能再被使用, 引脚被破坏);

结构尺寸图:



3.数据通讯协议:

3.1 UART 协议

✚ UART 接口一帧的数据格式为 1 个起始位, 8 个数据位, 无奇偶校验位, 1 个停止位。

✚ 波特率: 19200。

✚ **发送数据封包格式:**

数据包帧头 02	数据包内容	数据包帧尾 03
----------	-------	----------

注:0x02、0x03 被使用为起始字符、结束字符, 0x10 被使用为 0x02,0x03 的辨识字符。

因此在通讯的传输数据之中(起始字符 0x02, 至结束字符 0x03 之中)的 0x02、0x03、0x10 字符之前, 皆必须补插入 0x10 做为数据辨识之用。例如起始字符 0x02, 至结束字符 0x03 之中有一原始数据为 0x020310, 补插入辨识字符之后, 将变更为 0x100210031010。

数据包内容:

模块地址	长度字	命令字	数据域	校验字
------	-----	-----	-----	-----

模块地址: 对于单独使用的模块来说固定为 0x0000;

对网络版模块来说为 0x0001~0xFFFE;

0xFFFF 为广播。

长度字: 指明从长度字到校验字的字节数

命令字: 本条命令的含义

数据域: 该条命令的内容,此项可以为空

校验字: 从模块地址到数据域最后一字节(不包含插入的辨识符 0x10)的逐字节累加值, 保留最后一字节。

✚ **返回数据封包格式: 同发送数据封包格式相同**

数据包内容:

模块地址	长度字	接收到的命令字	执行结果	数据域	校验字
------	-----	---------	------	-----	-----

模块地址: 对与单独使用的模块来说固定为 0x0000;

对网络版模块来说为本身的地址;

长度字: 指明从长度字到数据域最后一字节的字节数

命令字: 本条命令的含义

执行结果: 0x00 执行正确

0x01---0xFF 执行错误

数据域: 该条命令的内容,返回执行状态和命令内容

校验字： 从模块地址到数据域最后一字节（不包含插入的标识符 0x10）的逐字节累加值，保留最后一字节。

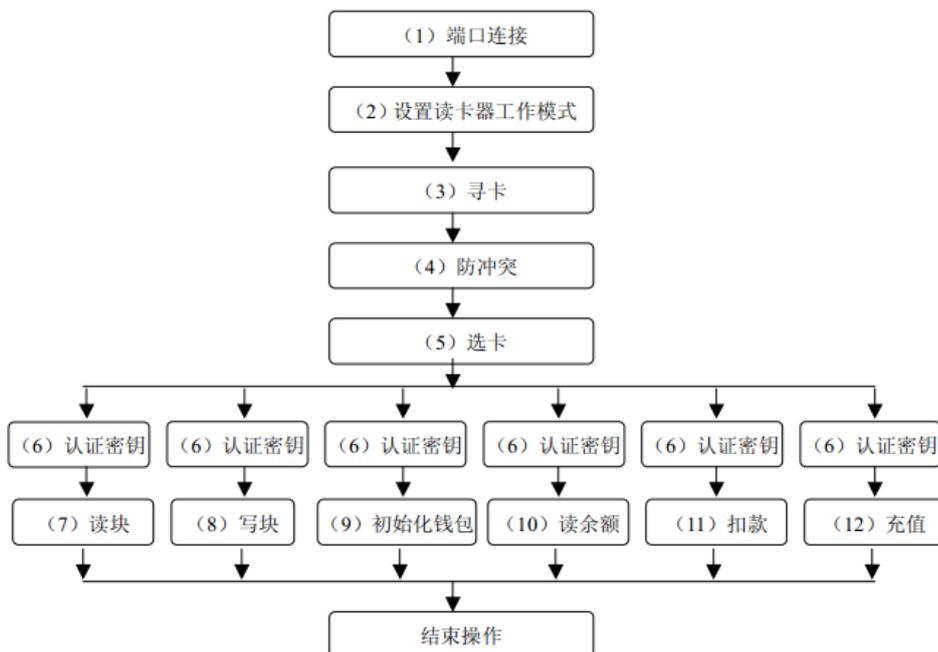
4.不同类型卡片操作流程:

4.1 读写模块初始化:

操作不同型号的卡片之前需要切换到相应协议：ISO14443A/ISO14443B/ISO15693



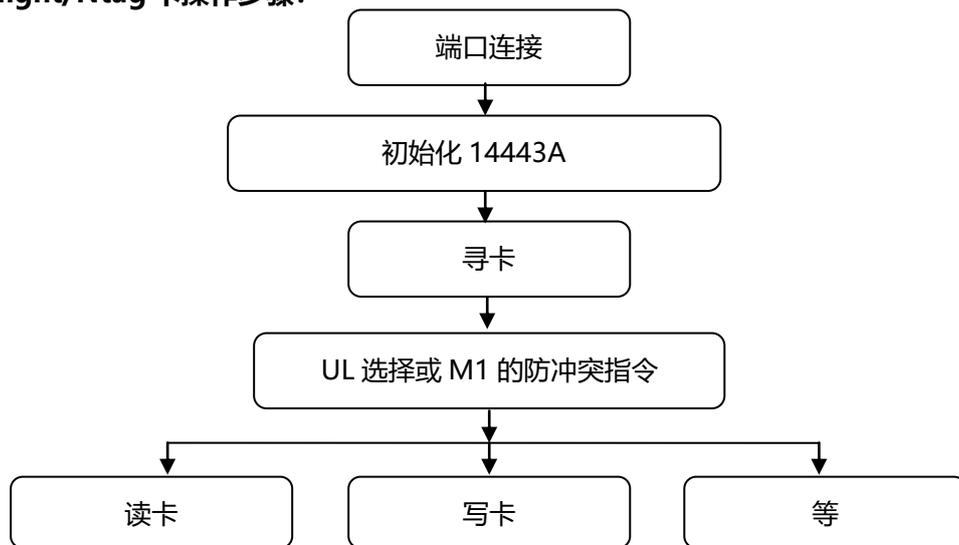
4.2 Mifare OneS50/S70 卡操作步骤:



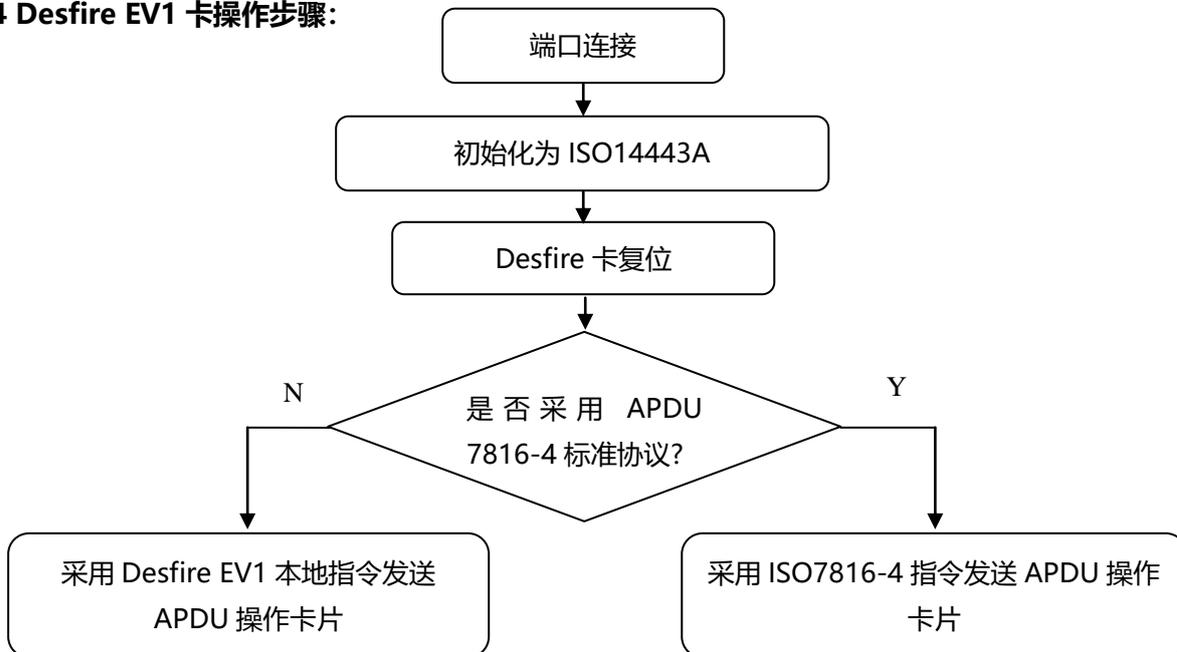
注意事项:

- (1) 寻卡，防冲突，选卡成功之后才可以进行块的读写以及钱包功能等操作；
- (2) 在进行块的读写，钱包等相关操作之前还需要进行密钥认证，只有通过才可以进行相应操作；
- (3) 想将某块作为钱包功能时，第一次必须用初始化钱包指令将该块进行初始化；
- (4) 在做钱包备份时，必须在同一扇区内进行操作；
- (5) 对于本读卡器,可以忽略第 5 步选卡操作,此条仅是为了兼容以往读卡器；

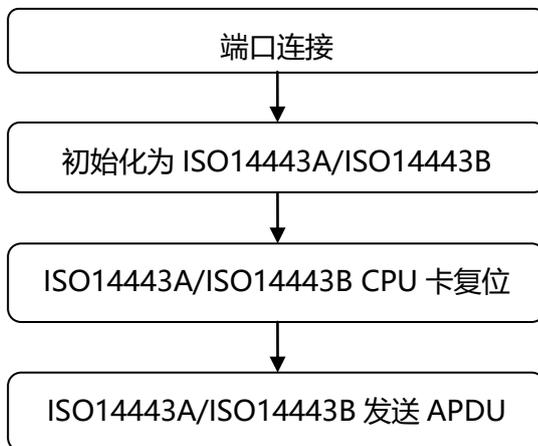
4.3 Ultralight/Ntag 卡操作步骤:

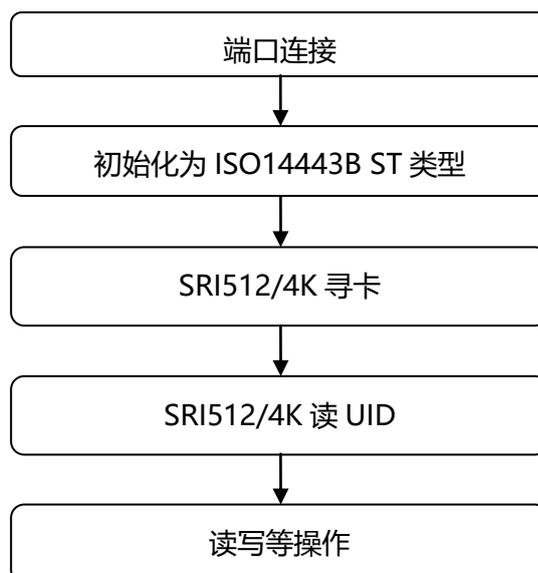
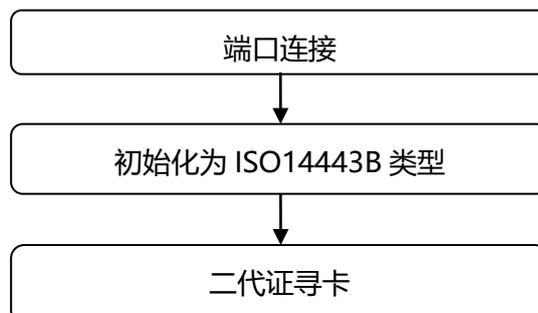
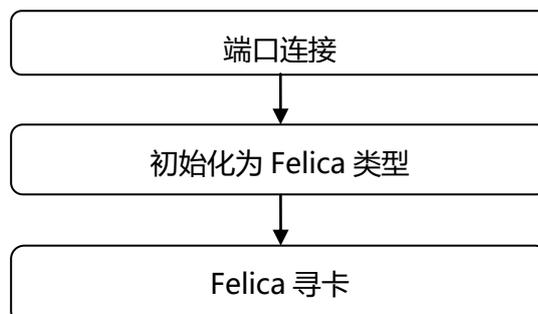
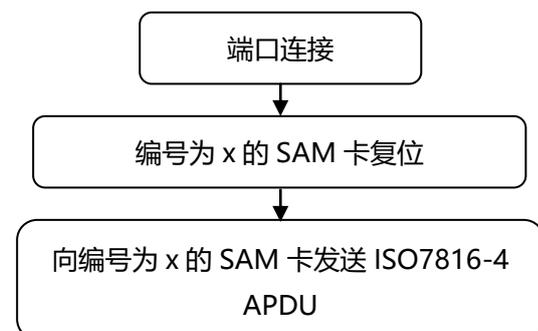


4.4 Desfire EV1 卡操作步骤:



4.5 ISO14443 TYPE A/TYPER B CPU 卡操作步骤:



4.6 SRI512/SRI4K 卡操作步骤:**4.7 二代证卡操作步骤:****4.8 Felica 卡操作步骤:****4.9 ISO7816 CPU 卡/PSAM 卡操作步骤:**

5.读卡器支持函数和指令汇总

模块操作指令请参阅相应 CHM 指令集: "M104FET-X-CN.chm"

注: 如果通过 PC 机调用 DLL 动态库, 长度字为 2 字节或者 1 字节的 DLL 动态库不同, 请特别注意。

5.1 系统函数指令

5.1.1 int WINAPI rf init com(初始化串口)

功能描述: 用于设置模块内部的通讯波特率, 模块默认上电为 19200 波特率, 外部 MCU 可以不调用此条指令直接进行通讯。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x15	1 字节 Baud	0xXX	0x03

发送数据 Baud: =0x01 表示设置模块波特率为 9600;
 =0x02 表示设置模块波特率为 14400;
 =0x03 表示设置模块波特率为 19200;
 =0x04 表示设置模块波特率为 28800;
 =0x05 表示设置模块波特率为 38400;
 =0x06 表示设置模块波特率为 57600;
 =0x07 表示设置模块波特率为 115200;

注: 模块如果不进行初始化, 默认波特率为 19200 或 115200, 初始化不会被保存, 模块掉电后会恢复成 19200 或 115200;

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x15	0x00	空	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x15	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例 (将模块波特率初始化为 19200):

【发送数据:】 02 00 00 04 15 10 03 1C 03

【接收数据:】 02 00 00 10 03 15 00 18 03

5.1.2 int WINAPI rf init type(初始化工作模式)

功能描述：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。

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x3A	1 字节 type	0xXX	0x03

说明：只支持单一协议的读卡器此函数可忽略

type = 'A': 设置为 TYPE_A 方式，对应 ASC 码为"0x41"

type = 'B': 设置为 TYPE_B 方式(Type B CPU 卡等)，对应 ASC 码为"0x42"

type = 's': 设置为 ST 卡方式 (Type B SR176,SRI512,SRI4K 等)，对应 ASC 码为"0x73"

type = 'r': 设置为 Atmel 卡方式 (Type B At88rf020 等)，对应 ASC 码为"0x72"

type = '1': 设置为 ISO15693 方式，对应 ASC 码为"0x31"

type = 'C': 设置为 Felica 方式，对应 ASC 码为"0x43"

正确返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x3A	0x00	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x3A	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 04 3A 41 7F 03

【接收数据:】 02 00 00 10 03 3A 00 3D 03

5.1.3 int WINAPI rf antenna sta(设置天线状态)

功能描述：设置读写卡器天线状态。

发送数据序列：

帧头	发送数据包内容	帧尾
----	---------	----

	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x04	0x05	1 字节 Model	0xXX	0x03
0x02	0x00,0x00	0x06	0x05	1 字节 Model+2 字节超时调节值 cDelay0, cDelay1	0xXX	0x03

长度字=0x04 时(下面功能与具体型号有关):

1 字节 Model: bit0 =0 关闭天线; =1 开启天线;
bit1=0 关闭自动寻卡; =1 打开自动寻卡;
bit2=0 无操作; =1 读卡芯片硬件复位成 ISO14443A 模式;
bit3-7: 无意义;

长度字>0x04 时(下面功能与具体型号有关): 数据域为

1 字节 Model: bit0 =0 关闭天线; =1 开启天线;
bit1=0 关闭自动寻卡; =1 打开自动寻卡;
bit2=0 无操作; =1 读卡芯片硬件复位成 ISO14443A 模式;
bit3-7: 无意义;

cDelay0: 命令执行时间超时时间高字节;

cDelay1: 命令执行时间超时时间低字节;

命令执行时间总超时时间=cDelay0*256+cDelay1 (毫秒)

正确返回数据序列:

帧头	正确返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x05	0x00	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x05	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 04 05 00 09 03

【接收数据:】 02 00 00 10 03 05 00 08 03

5.1.4 int WINAPI rf light(设置指示灯)

功能描述：点亮熄灭指示灯。

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x04	0x6A	1 字节控制信息	0xXX	0x03

注：数据部分为 1 字节控制信息：

数据=0x01 LED2 引脚输出高电平，如果外接 LED 指示灯，则灯熄灭；

数据=0x02 LED2 引脚输出低电平，如果外接 LED 指示灯，则灯点亮；

正确返回数据序列：

帧头	错误返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x10	0x03	0x6A	0x00	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

错误返回数据序列：

帧头	错误返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x10	0x03	0x6A	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 04 6A 01 6F 03

【接收数据:】 02 00 00 10 03 6A 00 6D 03

【发送数据:】 02 00 00 04 6A 10 02 70 03

【接收数据:】 02 00 00 10 03 6A 00 6D 03

5.1.5 int WINAPI rf get model(读取硬件版本号)

功能描述：读取读写卡器硬件版本号。

发送数据序列：

帧头	发送数据包内容					帧尾	
	模块地址	插入	长度	命令	发送数据		校验
0x02	0x00,0x00	0x10	0x03	0x16	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x16	0x00	2 字节版本号	0xXX	0x03

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x16	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 10 03 16 19 03

【接收数据:】 02 00 00 05 16 00 01 01 1D 03

5.1.6 int WINAPI rf get snr(读取序列号)

功能描述：读取读写卡器产品序列号。

发送数据序列：

帧头	发送数据包内容						帧尾
	模块地址	插入	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x10	0x03	0x17	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x17	0x00	N 字节返回序列号	0xXX	0x03

错误返回数据序列：

帧头	错误返回数据包内容						帧尾

	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x17	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 10 03 17 1A 03

【接收数据:】 02 00 00 0B 17 00 16 10 03 24 14 55 40 01 01 0A 03

5.1.7 int WINAPI rf_init_device_number(初始设备号)

功能描述：设定设备标识号。

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x05	0x13	2 字节 16 进制数据	0xXX	0x03

注：发送数据为 2 字节的 16 进制数；

模块即使设置了地址，当采用广播方式地址 0000 发送指令时，模块仍能正确返回数据，不过此时的返回的地址为读卡器的设定地址，不再是 0000；

正确返回数据序列：

帧头	正确返回数据包内容					帧尾	
	模块地址	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0xXX	0x13	0x00	空	0xXX	0x03

错误返回数据序列：

帧头	错误返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x10	0x03	0x13	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 05 13 12 34 5E 03

【接收数据:】 02 12 34 10 03 13 00 5C 03

5.1.8 int WINAPI rf get device number(获得设备 ID 号)

功能描述：得到设备标识号。

发送数据序列：

帧头	发送数据包内容						帧尾
	模块地址	插入	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x10	0x03	0x14	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x05	0x14	0x00	2 字节设备地址	0xXX	0x03

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x14	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 10 03 14 17 03

【接收数据:】 02 00 00 05 14 00 00 00 19 03

5.2 Mifare One 卡指令

5.2.1 int WINAPI rf request(寻卡)

Mifare one/Ultralight/Ntag 卡寻卡：

功能描述：用于 Mifare one/Ultralight/Ntag 卡的寻卡，返回卡片类型：

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x04	0x46	1 字节寻卡 mode	0xXX	0x03

注：数据部分为 1 字节寻卡模式 mode；

mode= "0x26" 寻卡，功能同参数 0x52；

mode= "0x52" 寻天线范围内的所有状态的卡；

对于 M1 卡，可以通过将 mode 字节的 bit7 设置为 1 (bit6-bit0 为 0x26 或者 0x52 不变)，进行检测是否寻到的卡为复制卡，如果为复制卡会返回失败，不进行卡片的进一步操作，此检测只针对复制卡，不针对正常卡中的数据异动。

举例发送：

mode= "0xA6" 寻卡，功能同参数 0xD2；

mode= "0xD2" 寻天线范围内的所有状态的卡；

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x05	0x46	0x00	2 字节卡类型	0xXX	0x03

返回 2 字节卡类型：

返回数据=0x04 0x00 表示 Mifare one S50 卡；

返回数据=0x02 0x00 表示 Mifare one S70 卡；

返回数据=0x44 0x00 表示 Ultralight 卡；

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x46	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 04 46 52 9C 03

【接收数据:】 02 00 00 05 46 00 04 00 4F 03 （通过返回数据可判断为 S50 卡）

5.2.2 int WINAPI rf anticoll(防冲突)

功能描述：用于 Mifare one 卡的防冲突指令，返回卡片唯一序列号，注该指令发送之前必须先发送寻卡指令，此模块不需要再发送选卡命令，为了兼容老的模块发送选卡命令，会全部返回成功，同时，此模块，此条指令也可以作为 Ultralight 与 Ntag 标签的防冲突使用

发送数据序列：

帧头	发送数据包内容	帧尾
----	---------	----

	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x04	0x47	0x04	0x4F	0x03

注：数据部分为 1 字节卡序列号字节数；

发送数据=“0x04”，只是为了兼容老的模块，“0x04”无含义；

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x47	0x00	4/7/10 字节 卡号	0xXX	0x03

返回 4 或者 7 或者 10 字节卡序列号

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x47	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据：】 02 00 00 04 47 04 4F 03

【接收数据：】 02 00 00 07 47 00 42 0B C2 08 65 03 （返回卡号：42 0B C2 08）

5.2.3 int WINAPI rf select(选卡)

功能描述：用于 Mifare one 卡的选卡操作，此指令的目的是选定一张卡，然后就可以实现对这张选定卡的读写等操作了，此模块已无效，仅是为了兼容以往的产品，发送选卡指令中任意卡号的数据都会返回正确，新设计开发客户可忽略此指令。

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x07	0x48	4 字节卡号	0xXX	0x03

注：数据部分为 4 字节卡序列号字节数；

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	

0x02	0xXX,0xXX	0x04	0x48	0x00	1 字节返回值	0xXX	0x03
------	-----------	------	------	------	---------	------	------

注：返回数据为卡容量：

数据 0x08 表示 Mifare one S50;

数据 0x20 表示 Mifare one S70;

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x48	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 07 48 42 0B C2 08 66 03

【接收数据:】 02 00 00 04 48 00 08 54 03 （根据卡容量 0x08，可判断为 S50 卡）

5.2.4 int WINAPI rf M1 authentication2(外部认证 2)

功能描述：用于 Mifare one 卡的读写卡钱包操作之前的密钥验证，只有密钥正确之后才可以对卡进行读写或钱包操作。

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x0B	0x4A	1 字节密钥模式+1 字节绝对块+6 字节密钥	0xXX	0x03

注：发送数据部分；

1 字节密钥模式=0x60 表示采用 A 密钥认证；

1 字节密钥模式=0x61 表示采用 B 密钥认证；

1 字节绝对块号=0x00~0x3F 之间的任意块号 (Mifare One S50，共计 64 块)；

1 字节绝对块号=0x00~0xFF 之间的任意块号 (Mifare One S70，共计 256 块)；

6 字节密钥为要操作块所在扇区的密钥；

注：新出厂的卡片默认密钥模式为 A 密钥，6 字节密钥为 "FFFFFFFFFFFF"；

正确返回数据序列：

帧头	正确返回数据包内容	帧尾
----	-----------	----

	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x4A	0x00	空	0x4D	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x4A	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据：】 02 00 00 0B 4A 60 00 FF FF FF FF FF FF AF 03 (验证 A 密钥，0 块，密码：“FFFFFFFFFFFF”)

【接收数据：】 02 00 00 10 03 4A 00 4D 03

5.2.5 int WINAPI rf M1 read(读块).mht

功能描述：用于 Mifare one/Ultralight/Ntag 卡的读块内容操作，注 M1 卡在读块前需进行密钥验证。

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x04	0x4B	1 字节块号	0xXX	0x03

注：发送数据部分；

1 字节绝对块号=0x00~0x3F 之间的任意块号 (Mifare One S50, 共计 64 块)；

1 字节绝对块号=0x00~0xFF 之间的任意块号 (Mifare One S70, 共计 256 块)；

对于 Ultralight/Ntag 则为读取连续 4 页的数据，故 1 字节块号为连续 4 页的起始页号；

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x13	0x4B	0x00	16 字节内容	0xXX	0x03

注：返回数据为该块 16 字节内容；

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x4B	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 04 4B 00 4F 03 (读 0 块数据。)

【接收数据:】 02 00 00 13 4B 00 42 0B C2 08 83 08 04 00 62 63 64 65 66 67 68 69 30 03

(0 块数据: 42 0B C2 08 83 08 04 00 62 63 64 65 66 67 68 69)

5.2.6 int WINAPI rf M1 read sector(读连续 3 块).mht

功能描述：用于 Mifare one 卡的读块内容操作，注在读块前需进行密钥验证，读取连续 3 块，要求 3 个块为同一扇区。

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x04	0x22	1 字节起始块号	0xXX	0x03

注：发送数据部分；

1 字节绝对块号=0x00~0x3F 之间的任意块号 (Mifare One S50, 共计 64 块)；

1 字节绝对块号=0x00~0xFF 之间的任意块号 (Mifare One S70, 共计 256 块)；

说明：要求起始块号+2 之后为同一扇区

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x22	0x00	48 字节内容	0xXX	0x03

注：返回数据为该块 48 字节内容；

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x22	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

5.2.10 int WINAPI rf M1 increment(充值)

功能描述：用于 Mifare one 卡的钱包充值操作，注在钱包充值前需进行密钥验证。

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x08	0x50	1 字节绝对块号 + 4 字节 16 进制要增加的金额值 (低字节在前)	0xXX	0x03

注：发送数据部分；

1 字节绝对块号=0x00~0x3F 之间的任意块号 (Mifare One S50, 共计 64 块)；

1 字节绝对块号=0x00~0xFF 之间的任意块号 (Mifare One S70, 共计 256 块)；

4 字节 16 进制增加值：为充值的金额，低字节在前；

密钥块 (每个扇区的最后一块) 不能进行充值操作，否则会造成该扇区的失效，具体使用注意事项请参阅卡片说明书。

正确返回数据序列：

帧头	正确返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x50	0x00	空	0x53	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x50	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 08 50 01 64 00 00 00 BD 03 (01 块充值 16 进制 "64")

【接收数据:】 02 00 00 10 03 50 00 53 03

5.2.11 int WINAPI rf M1 decrement(扣款)

功能描述：用于 Mifare one 卡的钱包扣款操作，注在钱包扣款前需进行密钥验证。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x08	0x4F	1 字节绝对块号 + 4 字节 16 进制要减少的金额值 (低字节在前)	0xXX	0x03

注: 发送数据部分;

1 字节绝对块号=0x00~0x3F 之间的任意块号 (Mifare One S50, 共计 64 块);

1 字节绝对块号=0x00~0xFF 之间的任意块号 (Mifare One S70, 共计 256 块);

4 字节 16 进制增加值: 为扣款的金额, 低字节在前;

密钥块 (每个扇区的最后一块) 不能进行扣款操作, 否则会造成该扇区的失效, 具体使用注意事项请参阅卡片说明书。

正确返回数据序列:

帧头	正确返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x10	0x03	0x4F	0x00	空	0x52	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

错误返回数据序列:

帧头	错误返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x10	0x03	0x4F	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 08 4F 01 32 00 00 00 8A 03 (01 块扣款 16 进制 "32")

【接收数据:】 02 00 00 10 03 4F 00 52 03

5.2.12 int WINAPI rf M1 readval(读余额)

功能描述: 用于 Mifare one 卡的钱包余额读取操作, 注在钱包读取前需进行密钥验证。

发送数据序列:

帧头	发送数据包内容	帧尾
----	---------	----

	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x04	0x4E	1 字节绝对块号	0xXX	0x03

注：发送数据部分；

1 字节绝对块号=0x00~0x3F 之间的任意块号 (Mifare One S50, 共计 64 块)；

1 字节绝对块号=0x00~0xFF 之间的任意块号 (Mifare One S70, 共计 256 块)；

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x07	0x4E	0x00	4 字节 16 进制 钱包值	0xXX	0x03

注：返回数据为 4 字节 16 进制钱包值，低字节在前；

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x4E	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 04 4E 01 53 03

【接收数据:】 02 00 00 07 4E 00 96 00 00 00 EB 03 (读出金额 16 进制 "96")

5.2.13 int WINAPI rf M1 restore(数据回传)

功能描述：用于将 Mifare one 卡的指定块的数据内容回传至卡的 Buffer，然后通过数据备份步骤 2 将数据备份到同一扇区的另一指定块，注在此操作前需进行密钥验证，同时这两步操作需要在同一扇区内操作。

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x04	0x51	1 字节绝对块号	0xXX	0x03

注：发送数据部分；

1 字节绝对块号=0x00~0x3F 之间的任意块号 (Mifare One S50, 共计 64 块)；

1 字节绝对块号=0x00~0xFF 之间的任意块号 (Mifare One S70, 共计 256 块)；

注：要备份的块必须为钱包格式。

正确返回数据序列:

帧头	正确返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x51	0x00	空	0x54	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x51	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【Write 数据:】 02 00 00 04 51 01 56 03 (将 01 块内容传入卡的 BUFFER 区)

【Read 数据:】 02 00 00 10 03 51 00 54 03

5.2.14 int WINAPI rf M1 transfer(数据传送)

功能描述: 用于将 Mifare one 卡通过数据备份步骤 1 传入 Buffer 中的数据值备份到同一扇区的指定块, 注在此操作前需进行 Mifare one 卡数据备份步骤 1 的操作。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x04	0x52	1 字节绝对块号	0xXX	0x03

注: 发送数据部分;

1 字节绝对块号=0x00~0x3F 之间的任意块号 (Mifare One S50, 共计 64 块);

1 字节绝对块号=0x00~0xFF 之间的任意块号 (Mifare One S70, 共计 256 块);

正确返回数据序列:

帧头	正确返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x52	0x00	空	0x55	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x52	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【Write 数据:】 02 00 00 04 52 10 02 58 03 (将 BUFFER 区中的内容传入 02 块中进行备份)

【Read 数据:】 02 00 00 10 03 52 00 55 03

5.2.15 int WINAPI rf halt(卡片休眠)

功能描述: 用于将 Mifare one/Ultralight 卡休眠操作。

发送数据序列:

帧头	发送数据包内容						帧尾
	模块地址	插入	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x10	0x03	0x29	空	0x2C	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

正确返回数据序列:

帧头	正确返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0x00,0x00	0x10	0x03	0x29	0x00	空	0x2C	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0x00,0x00	0x10	0x03	0x29	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 10 03 29 2C 03

【接收数据:】 02 00 00 10 03 29 00 2C 03

5.2.16 int WINAPI rf_download_key2(下载密钥 2)

功能描述: 用于向 Mifare one 模块下载预存密钥, 与 “rf_M1_authentication3”配合使用
(适用 RW210Ex/RW202Ex/RW208Ex/M209Ex/M102Ex/M104Ex 的模块的读卡器)

发送数据序列:

帧头	发送数据包内容						帧尾
	模块地址	插入	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x10	0x10	0x83	1 字节扇区编号 (0-39) + 6 字节 A 密钥+6 字节 B 密钥	0x4F	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x83	0x00	空	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x10	0x03	0x83	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 10 10 83 00 FF 87 03

【接收数据:】 02 00 00 10 03 83 00 86 03

5.2.17 int WINAPI rf_M1_authentication3(外部认证 3)

功能描述: 用已下载的密钥对 MifareOne 卡进行认证, 与 rf_download_key2 函数配对使用
(适用 RW210Ex/RW202Ex/RW208Ex/M209Ex/M102Ex/M104Ex)

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x84	1 字节密钥类型 + 1 字节扇区号 (0-39)	0xXX	0x03

注: Mode: =0x60 认证 A 密钥;
=0x61 认证 B 密钥;

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0x00,0x00	0xXX	0x84	0x00	空	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0x00,0x00	0x10	0x03	0x84	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 05 84 60 00 E9 03

【接收数据:】 02 00 00 10 03 84 00 87 03

5.3 Ultralight 卡指令

5.3.1 int WINAPI rf request(寻卡)

Mifare one/Ultralight/Ntag 卡寻卡:

功能描述: 用于 Mifare one/Ultralight/Ntag 卡的寻卡, 返回卡片类型:

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x04	0x46	1 字节寻卡 mode	0xXX	0x03

注: 数据部分为 1 字节寻卡模式 mode;

mode= "0x26" 寻卡, 功能同参数 0x52;

mode= "0x52" 寻天线范围内的所有状态的卡;

对于 M1 卡, 可以通过将 mode 字节的 bit7 设置为 1 (bit6-bit0 为 0x26 或者 0x52 不变), 进行检测是否寻到的卡为复制卡, 如果为复制卡会返回失败, 不进行卡片的进一步操作, 此检测只针对复制卡, 不针对正常卡中的数据异动。

举例发送:

mode= "0xA6" 寻卡, 功能同参数 0xD2;

mode= "0xD2" 寻天线范围内的所有状态的卡;

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	

0x02	0xXX,0xXX	0x05	0x46	0x00	2 字节卡类型	0xXX	0x03
------	-----------	------	------	------	---------	------	------

返回 2 字节卡类型:

返回数据=0x04 0x00 表示 Mifare one S50 卡;

返回数据=0x02 0x00 表示 Mifare one S70 卡;

返回数据=0x44 0x00 表示 Ultralight 卡;

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x46	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 04 46 52 9C 03

【接收数据:】 02 00 00 05 46 00 04 00 4F 03 (通过返回数据可判断为 S50 卡)

5.3.2 int WINAPI int rf ul select(Ultralight/Ntag 选卡)

功能描述: Ultra light/Ultralight C/NtagXX 卡防冲撞并激活, 对于一些新模块功能同 0x47 指令。

发送数据序列:

帧头	发送数据包内容						帧尾
	模块地址	插入	长度	命令	发送数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x33	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x07	0x33	0x00	7 字节卡号	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x33	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据：】 02 00 00 10 03 33 36 03

【接收数据：】 02 00 00 0A 33 00 04 6E F0 BA E1 22 80 DC 03

5.3.3 int WINAPI rf M1 read(读块)

功能描述：用于 Mifare one/Ultralight/Ntag 卡的读块内容操作，注 M1 卡在读块前需进行密钥验证。

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x04	0x4B	1 字节块号	0xXX	0x03

注：发送数据部分；

1 字节绝对块号=0x00~0x3F 之间的任意块号 (Mifare One S50, 共计 64 块)；

1 字节绝对块号=0x00~0xFF 之间的任意块号 (Mifare One S70, 共计 256 块)；

对于 Ultralight/Ntag 则为读取连续 4 页的数据，故 1 字节块号为连续 4 页的起始页号；

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x13	0x4B	0x00	16 字节内容	0xXX	0x03

注：返回数据为该块 16 字节内容；

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x4B	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据：】 02 00 00 04 4B 00 4F 03 (读 0 块数据。)

【接收数据：】 02 00 00 13 4B 00 42 0B C2 08 83 08 04 00 62 63 64 65 66 67 68 69 30 03

(0 块数据： 42 0B C2 08 83 08 04 00 62 63 64 65 66 67 68 69)

5.3.4 int WINAPI int rf ul write(Ultralight/Ntag 写块)

功能描述：向 Ultralight/Ultralight C/Ntag 卡中写入一页数据。

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x08	0x35	1 字节页号+4 字节写入数据	0xXX	0x03

正确返回数据序列：

帧头	正确返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x10	0x03	0x35	0x00	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

错误返回数据序列：

帧头	错误返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x10	0x03	0x35	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据：】 02 00 00 08 35 04 11 11 11 11 85 03

【接收数据：】 02 00 00 10 03 35 00 38 03

5.3.5 int WINAPI rf halt(卡片休眠)

功能描述：用于将 Mifare one/Ultralight 卡休眠操作。

发送数据序列：

帧头	发送数据包内容					帧尾	
	模块地址	插入	长度	命令	发送数据		校验
0x02	0x00,0x00	0x10	0x03	0x29	空	0x2C	0x03

白色部分为正常的指令序列，因为在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03，根据通讯协议需在其前面增加 0x10，即插入的部分；

正确返回数据序列:

帧头	正确返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0x00,0x00	0x10	0x03	0x29	0x00	空	0x2C	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0x00,0x00	0x10	0x03	0x29	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 10 03 29 2C 03

【接收数据:】 02 00 00 10 03 29 00 2C 03

5.4 ISO14443A T=CL CPU 卡指令

5.4.1 int WINAPI rf typea rst(TypeACPU 卡复位)

功能描述: 用于进行满足 ISO14443-4 TYPE A CPU 卡(FM1208/Desfire 等 CPU 卡)复位操作, 正确返回卡片复位信息。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x04	0x53	0x26 或者	0x7D 或者	0x03
				0x52	0xA9	

注: 发送数据=0x26 寻卡功能同 0x52;

发送数据=0x52 表示寻天线范围内的所有卡;

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0XXX	0x53	0x00	4/7 字节 CSN+返回信息	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x53	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

FM1208 卡, 返回 4 字节 UID+16 字节复位信息:

【发送数据:】 02 00 00 04 53 52 A9 03

【接收数据:】 02 FF FF 17 53 00 93 42 7A 0A 10 10 78 80 A0 10 02 20 90 00 00 00 00 00 93 42
7A 0A 74 03

Desfire 卡, 返回 7 字节 UID+6 字节复位信息:

【发送数据:】 02 00 00 04 53 52 A9 03

【接收数据:】 02 FF FF 10 10 53 00 04 2A 5D 7A CE 22 80 06 75 77 81 10 02 80 CB 03

5.4.2 int WINAPI rf cos command(发送 COS 命令)

功能描述: 用于发送 ISO14443-4 TYPE A CPU 卡(FM1208/Desfire 等 CPU 卡)COS 指令。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x54	COS 指令	0xXX	0x03

注: 具体 COS 指令请参阅 TYPE A CPU 卡数据手册;

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x54	0x00	针对 COS 指令的返回数据	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	

0x02	0xXX,0xXX	0x10	0x03	0x54	非零	空	0xXX	0x03
------	-----------	------	------	------	----	---	------	------

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

FM1208 取随机数：

【发送数据：】 02 00 00 08 54 00 84 00 00 04 E4 03

【接收数据：】 02 00 00 09 54 00 7B A3 5F 28 90 00 92 03

Desfire EV0/EV1 ISO7816-4 标准协议：

Desfire EV1 取随机数：

【发送数据：】 02 00 00 08 54 00 84 00 00 08 E8 03

【接收数据：】 02 FF FF 0D 54 00 DF F3 16 2B 30 D6 A5 B1 90 00 5E 03

Desfire EV1 选择目录：

【发送数据：】 02 00 00 0F 54 00 A4 04 00 07 D2 76 00 00 85 01 00 E0 03

【接收数据：】 02 FF FF 05 54 00 90 00 E7 03

Desfire EV0/EV1 本地 APDU：

Desfire EV1 选择 AID：

【发送数据：】 02 00 00 0C 54 90 5A 00 00 10 03 00 00 00 00 4D 03

【接收数据：】 02 FF FF 05 54 00 91 00 E8 03

DesfireEV1 读取版本号：

【发送数据：】 02 00 00 08 54 90 60 00 00 00 4C 03

【接收数据：】 02 FF FF 0C 54 00 04 01 01 01 00 18 05 91 AF C2 03

【发送数据：】 02 00 00 08 54 90 AF 00 00 00 9B 03

【接收数据：】 02 FF FF 0C 54 00 04 01 01 01 10 03 18 05 91 AF C5 03

【发送数据：】 02 00 00 08 54 90 AF 00 00 00 9B 03

【接收数据：】 02 FF FF 13 54 00 04 2A 5D 7A CE 22 80 BA 14 91 91 70 22 10 10 91 00 FD 03

5.4.3 int WINAPI rf_ISO14443_PPS(PPS 升频)

功能描述：用于进行满足 ISO14443-4 TYPE A CPU 卡 PPS 请求操作。PPS 请求必须在复位指令执行后，发送第一条 COS 指令之前进行。

注：此功能只有软件版本 8 之后才有。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x05	0xC0	1 字节 cCID+1 字节 cSetting	0xXX	0x03

注: cCID:固定为 0;

cSetting: =0: 106K; =1: 212K; =2: 424K; =3: 848K;

PPS 请求必须在复位指令执行后, 发送第一条 COS 指令之前进行.

正确返回数据序列:

帧头	正确返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0xC0	0x00	空	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0xC0	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】

【接收数据:】

5.4.4 int WINAPI rf_cl_deselect(退出激活状态)

功能描述: 用于 ISO14443 TYPE B CPU 卡取消选择指令, 指令同 TYPE A CPU 卡:

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x6B	空	0xXX	0x03

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0x00,0x00	0x07	0x6B	0x00	空	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0x00,0x00	0x10	0x03	0x6B	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】

【接收数据:】

5.5 Ntag 系列指令

5.5.1 int WINAPI rf request(寻卡)

Mifare one/Ultralight/Ntag 卡寻卡:

功能描述: 用于 Mifare one/Ultralight/Ntag 卡的寻卡, 返回卡片类型:

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x04	0x46	1 字节寻卡 mode	0xXX	0x03

注: 数据部分为 1 字节寻卡模式 mode;

mode= "0x26" 寻卡, 功能同参数 0x52;

mode= "0x52" 寻天线范围内的所有状态的卡;

对于 M1 卡, 可以通过将 mode 字节的 bit7 设置为 1 (bit6-bit0 为 0x26 或者 0x52 不变), 进行检测是否寻到的卡为复制卡, 如果为复制卡会返回失败, 不进行卡片的进一步操作, 此检测只针对复制卡, 不针对正常卡中的数据异动。

举例发送:

mode= "0xA6" 寻卡, 功能同参数 0xD2;

mode= "0xD2" 寻天线范围内的所有状态的卡;

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x05	0x46	0x00	2 字节卡类型	0xXX	0x03

返回 2 字节卡类型:

返回数据=0x04 0x00 表示 Mifare one S50 卡;

返回数据=0x02 0x00 表示 Mifare one S70 卡;

返回数据=0x44 0x00 表示 Ultralight 卡;

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x46	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 04 46 52 9C 03

【接收数据:】 02 00 00 05 46 00 04 00 4F 03 (通过返回数据可判断为 S50 卡)

5.5.2 int WINAPI int rf ul select(Ultralight/Ntag 选卡)

功能描述: Ultra light/Ultralight C/NtagXX 卡防冲撞并激活, 对于一些新模块功能同 0x47 指令。

发送数据序列:

帧头	发送数据包内容						帧尾
	模块地址	插入	长度	命令	发送数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x33	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x07	0x33	0x00	7 字节卡号	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x33	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 10 03 33 36 03

【接收数据:】 02 00 00 0A 33 00 04 6E F0 BA E1 22 80 DC 03

5.5.3 int WINAPI rf M1 read(读块)

功能描述: 用于 Mifare one/Ultralight/Ntag 卡的读块内容操作, 注在读块前需进行密钥验证。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x04	0x4B	1 字节块号	0xXX	0x03

注: 发送数据部分;

1 字节绝对块号=0x00~0x3F 之间的任意块号 (Mifare One S50, 共计 64 块);

1 字节绝对块号=0x00~0xFF 之间的任意块号 (Mifare One S70, 共计 256 块);

对于 Ultralight/Ntag 则为读取连续 4 页的数据, 故 1 字节块号为连续 4 页的起始页号;

正确返回数据序列:

帧头	正确返回数据包内容					帧尾	
	模块地址	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x13	0x4B	0x00	16 字节内容	0xXX	0x03

注: 返回数据为该块 16 字节内容;

错误返回数据序列:

帧头	错误返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x10	0x03	0x4B	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 04 4B 00 4F 03 (读 0 块数据。)

【接收数据:】 02 00 00 13 4B 00 42 0B C2 08 83 08 04 00 62 63 64 65 66 67 68 69 30 03

(0 块数据: 42 0B C2 08 83 08 04 00 62 63 64 65 66 67 68 69)

5.5.4 int WINAPI int rf ul write(Ultralight/Ntag 写块)

功能描述: 向 Ultralight/Ultralight C/Ntag 卡中写入一页数据。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x08	0x35	1 字节页号+4 字节 写入数据	0xXX	0x03

正确返回数据序列:

帧头	正确返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x10	0x03	0x35	0x00	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

错误返回数据序列:

帧头	错误返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x10	0x03	0x35	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 08 35 04 11 11 11 11 85 03

【接收数据:】 02 00 00 10 03 35 00 38 03

5.5.5 int WINAPI rf Ntag Fast Read(快速读取)

功能描述: Ntag213 系列快速读取。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x05	0x88	1 字节 cStartAdr +1 字节 cEndAdr, 一次 最多读取 15 块	0xXX	0x03

注: cStartAdr: 起始页号

cEndAdr: 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

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x88	0x00	(终止页-起始页+1) *4 字节数据	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x88	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 05 88 00 0E 9B 03

【接收数据:】 02 00 00 3F 88 00 04 4A 3C FA 72 9A 3D 80 55 48 00 00 E1 10 10 12 00 10 03 1B D1 10 02 16 53 70 91 00 00 00 00 22 22 22 22 11 22 33 44 00 00 00 00 11 11 11 11 00 00 00 00 00 00 00 00 00 00 00 00 85 03

5.5.6 int WINAPI rf Ntag Get Version(得到版本号)

功能描述: Ntag213 系列获取版本号。

发送数据序列:

帧头	发送数据包内容						帧尾
	模块地址	插入	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x10	0x03	0x87	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x87	0x00	8 字节版本号	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	

0x02	0xXX,0xXX	0x10	0x03	0x87	非零	空	0xXX	0x03
------	-----------	------	------	------	----	---	------	------

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 10 03 87 8A 03

【接收数据:】 02 00 00 0B 87 00 00 04 04 10 02 01 00 0F 10 03 AF 03

5.5.7 int WINAPI rf Ntag Password Auth(密码认证)

功能描述：Ntag213 系列密钥认证。

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x07	0x8A	4 字节密钥	0xXX	0x03

正确返回数据序列：

帧头	正确返回数据包内容					帧尾	
	模块地址	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0xXX	0x8A	0x00	2 字节 PACK	0xXX	0x03

错误返回数据序列：

帧头	错误返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x10	0x03	0x8A	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 07 8A FF FF FF FF 8D 03

【接收数据:】 02 00 00 05 8A 00 12 34 D5 03

5.5.8 int WINAPI rf Ntag Read CNT(读取计数值)

功能描述：Ntag213 系列读取计数值。

发送数据序列：

帧头	发送数据包内容	帧尾
----	---------	----

	模块地址	插入	长度	命令	发送数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x89	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x07	0x89	0x00	3 字节 CNT 计数	0xXX	0x03

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x89	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据：】 02 00 00 10 03 89 8C 03

【接收数据：】 02 00 00 06 89 00 14 00 00 A3 03 03

5.5.9 int WINAPI rf Ntag Read Signature(读取签名)

功能描述：Ntag213 系列读取签名。

发送数据序列：

帧头	发送数据包内容						帧尾
	模块地址	插入	长度	命令	发送数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x8B	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x8B	0x00	32 字节数字签名	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x8B	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 10 03 8B 8E 03

【接收数据:】 02 00 00 23 8B 00 D1 38 F2 2C 7C C3 AD FE D0 50 AC D4 5A 10 03 98 C8 22 AD
21 BC 75 BA 3A 1E C2 7C 60 46 A5 CC 67 19 24 03

5.5.10 int WINAPI rf NFC Select Sector(NFC 类型 2 标签选择扇区)

功能描述: NFC 类型 2 标签选择扇区

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x86	1 字节 cSector	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

cSector: [IN] 输入扇区编号(00-FEh);

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x07	0x86	0x00	空	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x86	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】

【接收数据:】

5.5.11 int WINAPI rf halt(卡片休眠)

功能描述: 用于将 Mifare one/Ultralight 卡休眠操作。

发送数据序列:

帧头	发送数据包内容						帧尾
	模块地址	插入	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x10	0x03	0x29	空	0x2C	0x03

白色部分为正常的指令序列, 因为在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03, 根据通讯协议需在其前面增加 0x10, 即插入的部分;

正确返回数据序列:

帧头	正确返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0x00,0x00	0x10	0x03	0x29	0x00	空	0x2C	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

错误返回数据序列:

帧头	错误返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0x00,0x00	0x10	0x03	0x29	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 10 03 29 2C 03

【接收数据:】 02 00 00 10 03 29 00 2C 03

5.6 ISO7816-4 SAM/CPU 卡指令

5.6.1 int WINAPI rf sam rst 2(SAM 卡复位新指令)

功能描述: 对超过 1 个 SAM 卡执行复位操作 (读卡器需支持多张 SAM 卡)

发送数据序列:

帧头	发送数据包内容	帧尾
----	---------	----

	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x19	1 字节 Mode	0xXX	0x03

注: Mode:

bit7-4: SAM 卡编号 b0000-b0101,

bit3-2 固定为 b00,

bit2-1: =b00 复位 9600 的卡片,

=b01 复位 38400 的卡,

=b10 复位 115200 的卡

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x19	0x00	复位信息	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x1A	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例 (对编号为 1 的 SAM 卡发送指令):

【发送数据:】 02 00 00 04 19 00 1D 03

【接收数据:】 02 00 00 16 19 00 3B 7D 94 00 00 4C 31 76 68 10 03 4C 4B 12 10 02 16 4F 95 2D
00 AB 03

5.6.2 int WINAPI rf sam cos 2(SAM 卡 COS 发送新指令)

功能描述: 向超过 1 个的 SAM 卡发送 COS 命令 (读卡器需支持多张 SAM 卡)

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x1A	1 字节 SAM 卡 编号 (0-1) +n 字节 COS	0xXX	0x03

				指令		
--	--	--	--	----	--	--

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x1A	0x00	COS 的执行结果	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x1A	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例 (对编号为 1 的 SAM 卡发送指令):

【发送数据:】 02 00 00 09 1A 01 00 84 00 00 08 B0 03

【接收数据:】 02 00 00 0D 1A 00 06 06 BC 11 57 6A F1 58 90 00 9A 03

5.7 ISO14443B CPU 卡指令

5.7.1 int WINAPI rf atqb(复位)

功能描述: ISO14443 TYPE B 卡片复位, 一些新读卡器支持 PPS 功能

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x04	0x3B	1 字节 Mode	0x3F	0x03

注: 1 字节 Mode: bit0~bit3=0000 复位未休眠的卡
 bit0~bit3=0001 复位所有的卡
 bit4~bit7=0000 106K
 bit4~bit7=0001 212K (PPS)
 bit4~bit7=0010 424K (PPS)
 bit4~bit7=0011 848K (PPS)

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	

0x02	0xXX,0xXX	0x07	0x3B	0x00	返回复位信息	0xXX	0x03
------	-----------	------	------	------	--------	------	------

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x3B	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 04 3B 00 3F 03

【接收数据:】 02 12 34 0F 3B 00 50 28 F4 14 3A 14 00 00 00 F7 71 85 4B 03

5.7.2 int WINAPI rf cos command(发送 COS 命令)

功能描述: 用于发送 ISO14443-4 TYPE B CPU 卡 COS 指令, 指令同 TYPE A CPU 卡。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x54	COS 指令	0xXX	0x03

注: 具体 COS 指令请参阅 TYPE A CPU 卡数据手册;

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x54	0x00	针对 COS 指令的返回数据	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x54	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 08 54 00 84 00 00 04 E4 03 (取 4 字节随机数指令)

【接收数据:】 02 00 00 09 54 00 7B A3 5F 28 90 00 92 03 (4 字节随机数 7B A3 5F 28, 90 00 表示执行正确)

5.7.3 int WINAPI rf cl deselect(退出激活状态)

功能描述：用于 ISO14443 TYPE B CPU 卡取消选择指令，指令同 TYPE A CPU 卡：

发送数据序列：

帧头	发送数据包内容						帧尾
	模块地址	插入	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x10	0x03	0x6B	空	0xXX	0x03

白色部分为正常的指令序列，因为在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03，根据通讯协议需在其前面增加 0x10，即插入的部分；

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x07	0x6B	0x00	空	0xXX	0x03

错误返回数据序列：

帧头	错误返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x10	0x03	0x6B	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】

【接收数据:】

5.7.4 int WINAPI rf_hltb(卡休眠)

功能描述：ISO14443 TYPE B 卡片 Halt

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x3D	4 字节 PUPI	0xXX	0x03

正确返回数据序列：

帧头	正确返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x10	0x03	0x3D	0x00	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x3D	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】

【接收数据:】

5.8 SRI512-SRI4K 卡指令

5.8.1 int WINAPI rf st select(ST 卡选卡)

功能描述：ISO14443 TYPE B SR176/SRI512/SRIX4K 卡选卡：

发送数据序列：

帧头	发送数据包内容						帧尾
	模块地址	插入	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x10	0x03	0x60	空	0x63	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x04	0x60	0x00	返回 Chip_ID	0xXX	0x03

注：Chip_ID 为伪 ID 号，有可能同一张卡每次返回的 Chip_ID 也不同；

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x60	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 10 03 60 63 03

【接收数据:】 02 00 00 06 60 00 20 00 FF 85 03

5.8.2 int WINAPI int rf srix4k getuid(获得 UID)

功能描述：ISO14443 TYPE B SRI512/SRIX4K 卡读取 UID：

发送数据序列：

帧头	发送数据包内容	帧尾

	模块地址	插入	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x10	0x03	0x68	空	0x6B	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x04	0x68	0x00	返回 8 字节 UID	0xXX	0x03

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x68	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 10 03 68 6B 03

【接收数据:】 02 00 00 0B 68 00 D0 10 02 1A 1F 2D 99 28 0E 7A 03

5.8.3 int WINAPI int rf srix4k readblock(读块)

功能描述：ISO14443 TYPE B SRI512/SRIX4K 卡读块：

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x04	0x65	1 字节块号	0xXX	0x03

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x07	0x65	0x00	返回 4 字节数据	0xXX	0x03

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x65	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例(读取块 0):

【发送数据:】 02 00 00 04 65 00 69 03

【接收数据:】 02 00 00 07 65 00 12 34 56 78 80 03

5.8.4 int WINAPI int rf srix4k writeblock(写块)

功能描述: ISO14443 TYPE B SRI512,SRIX4K 写块。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x08	0x66	1 字节块号+4 字节写入数据	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

正确返回数据序列:

帧头	正确返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x66	0x00	空	0x64	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x66	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例 (向块 4 写入数据: 00000000):

【发送数据:】 02 00 00 08 66 04 00 00 00 00 72 03

【接收数据:】 02 00 00 10 03 66 00 69 03

5.8.5 int WINAPI int rf sri512 writelockreg(SRI512 块锁定)

功能描述: ISO14443 TYPE B SRI512 卡块锁定:

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x6C	2 字节块锁定信息 +1 字节 00+1 字节 Chip_ID	0xXX	0x03

注: 2 字节块锁定信息 (如果控制位设置成 0 表示不可以再读写, 对应的块只能读出不能再对其修改, 每个位只能从 1 变 0。需要注意的一点, 当 2 字节块锁定信息修改后, 需要发送选卡指令,

锁块才会有效):

bit 0: 控制第 0 块;

bit 1: 控制第 1 块;

以此类推

bit 15: 控制第 15 块;

1 字节 Chip_ID: 可选的 Chip_ID

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x07	0x6C	0x00	空	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x6C	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】

【接收数据:】

5.8.6 int WINAPI int rf srix4k writelockreg(SRI4K 块锁定)

功能描述: ISO14443 TYPE B SRIX4K 卡块锁定:

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x04	0x69	1 字节块锁定	0xXX	0x03

注: 1 字节块锁定信息 (如果控制位设置成 0 表示不可以再读写, 对应的块只能读出不能再对其修改, 每个位只能从 1 变 0。需要注意的一点, 当 1 字节块锁定信息修改后, 需要发送选卡指令, 锁块才会有效):

bit 0: 控制第 7, 8 块;

bit 1: 控制第 9 块;

bit 2: 控制第 10 块;

bit 3: 控制第 11 块;

bit 4: 控制第 12 块;

bit 5: 控制第 13 块;

bit 6: 控制第 14 块;

bit 7: 控制第 15 块;

正确返回数据序列:

帧头	正确返回数据包内容						帧尾

	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x07	0x69	0x00	空	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x69	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】

【接收数据:】

5.8.7 int WINAPI rf st completion(退出激活)

功能描述: 用于 SR176/SRI512/SRIX4K 卡取消选择。

发送数据序列:

帧头	发送数据包内容						帧尾
	模块地址	插入	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x10	0x03	0x61	空	0x64	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x04	0x61	0x00	空	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x61	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 10 03 61 64 03

【接收数据:】 02 00 00 10 03 61 00 64 03

5.9 二代证卡指令

5.9.1 int WINAPI rf get namecard uid(读取二代证 UID)

功能描述：ISO14443 TYPE B 得到中国二代证 UID 号：

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x04	0x82	1 字节 mode 0x00	0x86	0x03

正确返回数据序列：

帧头	正确返回数据包内容					帧尾	
	模块地址	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x0B	0x82	0x00	8 字节 UID	0xXX	0x03

错误返回数据序列：

帧头	错误返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x10	0x03	0x82	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 04 82 00 86 03

【接收数据:】 02 00 00 0B 82 00 30 95 75 05 40 01 83 B8 48 03

5.10 ISO15693 卡指令

5.10.1 int WINAPI ISO15693 Inventory(寻卡)

功能描述：用于 ISO15693 卡 Inventory 寻卡操作。

发送数据序列：

帧头	发送数据包内容					帧尾	
	模块地址	插入	长度	命令	发送数据		校验
0x02	0x00,0x00	0x10	0x03	0x70	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

正确返回数据序列：

帧头	正确返回数据包内容					帧尾
	模块地址	长度	命令	执行结果	返回数据	

0x02	0xXX,0xXX	0xXX	0x70	0x00	1 字节 DSFID+8 字 节 UID	0xXX	0x03
------	-----------	------	------	------	----------------------------	------	------

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x70	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 10 03 70 73 03

【接收数据:】 02 00 00 0C 70 00 00 20 C1 AB 0F 00 01 04 E0 FC 03

5.10.2 int WINAPI ISO15693 Get System Information(获得系统信息)

功能描述: 用于 ISO15693 卡获取系统信息。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x0C	0x7B	1 字节模式+8 字 节 UID	0xXX	0x03

注: 此处的 1 字节模式与 8 字节 UID 不起作用, 数据可以任意, 但是要带入;

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x7B	0x00	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 字节卡片容 量 (如果存在) +1 字节 厂商信息 (如果存在)	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容						帧尾
----	-----------	--	--	--	--	--	----

	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x7B	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 0C 7B 00 00 00 00 00 00 00 00 00 87 03

【接收数据:】 02 00 00 11 7B 00 0F 4F 45 B6 E6 12 81 07 E0 00 00 3F 10 03 8B 12 03

5.10.3 int WINAPI ISO15693 Read(读块)

功能描述：用于 ISO15693 卡读取卡片中指定块的数据。

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x74	1 字节模式选择+8 字节 UID+1 字节起始块号+1 字节要读取的块数 (块数要<0x10)	0xXX	0x03

注：1 字节模式选择，其中各位含义如下：

bit0=1 表示只有被选中的卡才执行此操作；

bit1=1 表示只有 UID 与指令中 UID 相同的卡才执行此操作；

bit2=1 表示如果要操作 TI 的卡，此 bit2 设置为 1，如果要操作 NXP 的卡片，此 bit2 应设为 0；

bit3-bit7 备用

1 字节起始块号：要读取的绝对起始块号；

1 字节要读取的块数：从起始块号开始后面要连续读出的快的数量，此数要小于 0x10 块；

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x74	0x00	返回的数据	0xXX	0x03

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x74	非零	空	0xXX	0x03

发送与返回正确举例(向块 7 写入数据 11223344):

【发送数据:】 02 00 00 11 75 10 02 20 C1 AB 0F 00 01 04 E0 07 11 22 33 44 B9 03

【接收数据:】 02 00 00 10 03 75 00 78 03

5.10.5 int WINAPI ISO15693_Write_Multi_Block(写连续多块)

功能描述: 用于 ISO15693 卡向多个连续块写入数据。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x6F	1 字节模式选择+8 字节 UID+1 字节起始块号+1 字节要写入的块数 (小于等于 10) +要写入块数*4 字节的数据	0xXX	0x03

注: 1 字节模式选择, 其中各位含义如下:

bit0=1 表示只有被选中的卡才执行此操作;

bit1=1 表示只有 UID 与指令中 UID 相同的卡才执行此操作;

bit2=1 表示如果要操作 TI 的卡, 此 bit2 设置为 1, 如果要操作 NXP 的卡片, 此 bit2 应设为 0;

bit3-bit7 备用

1 字节起始块号: 要写入的连续块起始地址;

1 字节要写入的块数: 块数要小于等于 10;

写入的数据: 要写入的块数*4 为要写入的数据;

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x6F	0x00	空	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x6F	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例(向块 F 写入数据)

自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例(读取从 00 块开始的 28 块的卡片安全状态信息):

【发送数据:】 02 00 00 0E 7C 10 02 20 C1 AB 0F 00 01 04 E0 00 1C 28 03

【接收数据:】 02 00 00 1F 7C 00 01 01 01 00 9E 03

5.10.7 int WINAPI ISO15693 Lock Block(锁块)

功能描述：用于 ISO15693 卡将指定的块锁死，块锁死之后只能读出，不可以再进行修改。

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x76	1 字节模式选择+8 字节 UID+1 字节 块号	0xXX	0x03

注：1 字节模式选择，其中各位含义如下：

bit0=1 表示只有被选中的卡才执行此操作；

bit1=1 表示只有 UID 与指令中 UID 相同的卡才执行此操作；

bit2=1 表示如果要操作 TI 的卡，此 bit2 设置为 1，如果要操作 NXP 的卡片，此 bit2 应设为 0；

bit3-bit7 备用

1 字节块号：要锁定的块号；

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x76	0x00	空	0xXX	0x03

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x76	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例(将块 2 锁定):

【发送数据:】 02 00 00 0D 76 10 02 20 C1 AB 0F 00 01 04 E0 10 02 07 03

【接收数据:】 02 00 00 10 03 76 00 79 03

5.10.8 int WINAPI ISO15693 Write AFI(写 AFI)

功能描述: 用于 ISO15693 卡写 AFI。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x77	1 字节模式选择+8 字节 UID+1 字节要写入的 AFI	0xXX	0x03

注: 1 字节模式选择, 其中各位含义如下:

bit0=1 表示只有被选中的卡才执行此操作;

bit1=1 表示只有 UID 与指令中 UID 相同的卡才执行此操作;

bit2=1 表示如果要操作 TI 的卡, 此 bit2 设置为 1, 如果要操作 NXP 的卡片, 此 bit2 应设为 0;

bit3-bit7 备用

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x77	0x00	空	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x77	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例(将 AFI 写成 0x00):

【发送数据:】 02 00 00 0D 77 10 02 20 C1 AB 0F 00 01 04 E0 00 06 03

【接收数据:】 02 00 00 10 03 77 00 7A 03

5.10.9 int WINAPI ISO15693 Lock AFI(锁定 AFI)

功能描述: 用于 ISO15693 卡将 AFI 锁死, 锁死之后不能再对其修改。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x78	1 字节模式选择 +8 字节 UID	0xXX	0x03

注: 1 字节模式选择, 其中各位含义如下:

bit0=1 表示只有被选中的卡才执行此操作;

bit1=1 表示只有 UID 与指令中 UID 相同的卡才执行此操作;

bit2=1 表示如果要操作 TI 的卡, 此 bit2 设置为 1, 如果要操作 NXP 的卡片, 此 bit2 应
设为 0;

bit3-bit7 备用

正确返回数据序列:

帧头	正确返回数据包内容					帧尾	
	模块地址	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0xXX	0x78	0x00	空	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x10	0x03	0x78	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后
自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例(将 AFI 锁死):

【发送数据:】 02 00 00 0C 78 10 02 20 C1 AB 0F 00 01 04 E0 06 03

【接收数据:】 02 00 00 10 03 78 00 7B 03

5.10.10 int WINAPI ISO15693 Write DSFID(写 DSFID)

功能描述: 用于 ISO15693 卡写 DSFID。

发送数据序列:

帧头	发送数据包内容				帧尾	
	模块地址	长度	命令	发送数据		
0x02	0x00,0x00	0xXX	0x79	1 字节模式选择+8 字节 UID+1 字节要写入的 DSFID	0xXX	0x03

注：1 字节模式选择，其中各位含义如下：

bit0=1 表示只有被选中的卡才执行此操作；

bit1=1 表示只有 UID 与指令中 UID 相同的卡才执行此操作；

bit2=1 表示如果要操作 TI 的卡，此 bit2 设置为 1，如果要操作 NXP 的卡片，此 bit2 应设为 0；

bit3-bit7 备用

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x79	0x00	空	0xXX	0x03

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x79	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例(将 DSFID 写成 0x00)：

【发送数据:】 02 00 00 0D 79 10 02 20 C1 AB 0F 00 01 04 E0 00 08 03

【接收数据:】 02 00 00 10 03 79 00 7C 03

5.10.11 int WINAPI ISO15693 Lock DSFID(锁 DSFID)

功能描述：用于 ISO15693 卡将 DSFID 锁死，锁死之后不能再对其修改。

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x7A	1 字节模式选择+8 字节 UID	0xXX	0x03

注：1 字节模式选择，其中各位含义如下：

bit0=1 表示只有被选中的卡才执行此操作；

bit1=1 表示只有 UID 与指令中 UID 相同的卡才执行此操作；

bit2=1 表示如果要操作 TI 的卡，此 bit2 设置为 1，如果要操作 NXP 的卡片，此 bit2 应设为 0；

bit3-bit7 备用

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x7A	0x00	空	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x7A	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例(将 DSFID 锁死):

【发送数据:】 02 00 00 0C 7A 10 02 20 C1 AB 0F 00 01 04 E0 08 03

【接收数据:】 02 00 00 10 03 7A 00 7D 03

5.10.12 int WINAPI ISO15693 SET EAS ICODE(设置 ICODE EAS)

功能描述: 用于 NXP I CODE 卡 SET EAS。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x7D	1 字节模式选择+8 字节 UID	0xXX	0x03

注: 1 字节模式选择, 其中各位含义如下:

bit0=1 表示只有被选中的卡才执行此操作;

bit1=1 表示只有 UID 与指令中 UID 相同的卡才执行此操作;

bit2=1 表示如果要操作 TI 的卡, 此 bit2 设置为 1, 如果要操作 NXP 的卡片, 此 bit2 应设为 0;

bit3-bit7 备用

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x7D	0x00	空	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容						帧尾
----	-----------	--	--	--	--	--	----

	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x7D	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例(将 EAS 位置 1):

【发送数据:】 02 00 00 0C 7D 10 02 13 A5 40 58 00 01 04 E0 C0 03

【接收数据:】 02 00 00 10 03 7D 00 80 03

5.10.13 int WINAPI ISO15693 EAS ALARM ICODE(I CODE EAS 报警)

功能描述：NXP I CODE 卡 EAS ALARM，前提需要先将卡 SET EAS。

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x80	1 字节模式选择+8 字节 UID	0xXX	0x03

注：1 字节模式选择，其中各位含义如下：

bit0=1 表示只有被选中的卡才执行此操作；

bit1=1 表示只有 UID 与指令中 UID 相同的卡才执行此操作；

bit2=0 此处应设为 0，只有 I CODE 支持该功能；

bit3-bit7 备用

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x80	0x00	32 字节 Alarm 信息	0xXX	0x03

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x80	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 0C 80 10 02 5F EA 40 58 00 01 04 E0 54 03

【接收数据:】 02 00 00 23 80 00 2F B3 62 70 D5 A7 90 7F E8 B1 80 38 D2 81 49 76 82 DA 9A86
6F AF 8B B0 F1 9C D1 12 A5 72 37 EF C7 03

5.10.14 int WINAPI ISO15693 RESET EAS ICODE(复位 ICODE EAS)

功能描述: 用于 ISO15693 卡 NXP I CODE 卡 RESET EAS。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x7E	1 字节模式选择+8 字节 UID	0xXX	0x03

注: 1 字节模式选择, 其中各位含义如下:

bit0=1 表示只有被选中的卡才执行此操作;

bit1=1 表示只有 UID 与指令中 UID 相同的卡才执行此操作;

bit2=1 表示如果要操作 TI 的卡, 此 bit2 设置为 1, 如果要操作 NXP 的卡片, 此 bit2 应设为 0;

bit3-bit7 备用

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x7E	0x00	空	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x7E	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例(将 EAS 位复位为 0):

【发送数据:】 02 00 00 0C 7E 10 02 13 A5 40 58 00 01 04 E0 C1 03

【接收数据:】 02 00 00 10 03 7E 00 81 03

5.10.15 int WINAPI ISO15693 LOCK EAS ICODE(锁定 ICODE EAS)

功能描述: 用于 ISO15693 卡 NXP I CODE 卡 LOCK EAS。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x7F	1 字节模式选择+8 字节 UID	0xXX	0x03

注：1 字节模式选择，其中各位含义如下：

bit0=1 表示只有被选中的卡才执行此操作；

bit1=1 表示只有 UID 与指令中 UID 相同的卡才执行此操作；

bit2=1 表示如果要操作 TI 的卡，此 bit2 设置为 1，如果要操作 NXP 的卡片，此 bit2 应设为 0；

bit3-bit7 备用

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x7F	0x00	空	0xXX	0x03

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x7F	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例(将 ISO15693 卡 EAS 位当前状态锁定)：

【发送数据:】 02 00 00 0C 7F 10 02 13 A5 40 58 00 01 04 E0 C2 03

【接收数据:】 02 00 00 10 03 7F 00 82 03

5.10.16 int WINAPI ISO15693 Select(选卡)

功能描述：用于 ISO15693 卡选择卡。

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x72	1 字节模式选择+8 字节 UID	0xXX	0x03

注：1 字节模式选择，其中各位含义如下：

bit0=1 表示只有被选中的卡才执行此操作；

bit1=1 表示只有 UID 与指令中 UID 相同的卡才执行此操作;

bit2=1 表示如果要操作 TI 的卡, 此 bit2 设置为 1, 如果要操作 NXP 的卡片, 此 bit2 应设为 0;

bit3-bit7 备用

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x72	0x00	空	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x72	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】 02 00 00 0C 72 10 02 20 C1 AB 0F 00 01 04 E0 00 03

【接收数据:】 02 00 00 10 03 72 00 75 03

5.10.17 int WINAPI ISO15693 Inventorys(寻多张卡)

功能描述: 用于 ISO15693_Inventorys(多张卡)。

发送数据序列:

帧头	发送数据包内容						帧尾
	模块地址	插入	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x10	0x03	0x85	空	0xXX	0x03

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x85	0x00	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+每 9 个字节为一组, 每组结构为: 1 字节 DSFID + 8	0xXX	0x03

					字节 UID		
--	--	--	--	--	--------	--	--

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x85	非零	空	0xXX	0x03

注: 插入部分为模块在返回数据时, 在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的, 故在操作接收数据时需过滤掉;

发送与返回正确举例:

【发送数据:】

【接收数据:】

5.10.18 int WINAPI ISO15693 Reset To Ready(复位)

功能描述: 用于 ISO15693 卡复位到准备状态。

发送数据序列:

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x73	1 字节模式选择+8 字节 UID	0xXX	0x03

注: 1 字节模式选择, 其中各位含义如下:

bit0=1 表示只有被选中的卡才执行此操作;

bit1=1 表示只有 UID 与指令中 UID 相同的卡才执行此操作;

bit2=1 表示如果要操作 TI 的卡, 此 bit2 设置为 1, 如果要操作 NXP 的卡片, 此 bit2 应设为 0;

bit3-bit7 备用

正确返回数据序列:

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0xXX	0x73	0x00	空	0xXX	0x03

错误返回数据序列:

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	

0x02	0xXX,0xXX	0x10	0x03	0x73	非零	空	0xXX	0x03
------	-----------	------	------	------	----	---	------	------

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 0C 73 10 02 20 C1 AB 0F 00 01 04 E0 01 03

【接收数据:】 02 00 00 10 03 73 00 76 03

5.10.19 int WINAPI ISO15693 Stay Quiet(进入静默)

功能描述：用于 ISO15693 卡进行静默状态。

发送数据序列：

帧头	发送数据包内容					帧尾
	模块地址	长度	命令	发送数据	校验	
0x02	0x00,0x00	0xXX	0x71	8 字节 UID	0xXX	0x03

正确返回数据序列：

帧头	正确返回数据包内容					帧尾	
	模块地址	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0xXX	0x71	0x00	空	0xXX	0x03

错误返回数据序列：

帧头	错误返回数据包内容						帧尾	
	模块地址	插入	长度	命令	执行结果	返回数据		校验
0x02	0xXX,0xXX	0x10	0x03	0x71	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头 0x02 帧尾 0x03 之间出现了 0x02 或 0x10 或 0x03 后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据:】 02 00 00 0B 71 20 C1 AB 0F 00 01 04 E0 FC 03

【接收数据:】 02 00 00 10 03 71 00 74 03

5.11 Felica 卡指令

5.11.1 int WINAPI rf_felica_get_uid(读取 FelicaUID)

功能描述：读取 Felica 读取 UID 卡号。

发送数据序列：

帧头	发送数据包内容						帧尾
	模块地址	插入	长度	命令	发送数据	校验	
0x02	0x00,0x00	0x10	0x03	0x81	空	0x84	0x03

注：插入部分为模块在返回数据时，在帧头0x02帧尾0x03之间出现了0x02或0x10或0x03后自动增加的，故在操作接收数据时需过滤掉；

正确返回数据序列：

帧头	正确返回数据包内容						帧尾
	模块地址	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x09	0x81	0x00	6 字节卡号	0xXX	0x03

错误返回数据序列：

帧头	错误返回数据包内容							帧尾
	模块地址	插入	长度	命令	执行结果	返回数据	校验	
0x02	0xXX,0xXX	0x10	0x03	0x81	非零	空	0xXX	0x03

注：插入部分为模块在返回数据时，在帧头0x02帧尾0x03之间出现了0x02或0x10或0x03后自动增加的，故在操作接收数据时需过滤掉；

发送与返回正确举例：

【发送数据：】 02 00 00 10 03 81 84 03

【接收数据：】 02 00 00 09 81 00 00 62 92 CF 47 BD 51 03

6.底层通讯举例（默认 1 字节长度举例）

**注：2 字节长度是在 1 字节长度字前增加 1 字节长度字高字节，同时长度字增加 1，校验字也随之变化
比如：1 字节长度发送指令为 02 00 00 04 15 07 20 03，变为 2 字节长度发送指令后则为 02 00 00 00 05 15 07 21 03**

6.1 系统函数：

端口连接：

[Send data] 02 00 00 04 15 07 20 03

[Receive data] 02 FF FF 10 03 15 00 16 03

6.2 Mifare One:

初始化为 ISO14443A，本型号可忽略

[Send data]02 00 00 04 05 00 09 03

[Receive data] 02 00 00 10 03 05 00 08 03

[Send data]02 00 00 04 3A 41 7F 03

[Receive data] 02 00 00 10 03 3A 00 3D 03

[Send data]02 00 00 04 05 01 0A 03

[Receive data] 02 00 00 10 03 05 00 08 03

Mifare S50 寻卡/防冲突/选卡

[Send data]02 00 00 04 46 52 9C 03
[Receive data] 02 00 00 05 46 00 04 00 4F 03
[Send data]02 00 00 04 47 04 4F 03
[Receive data] 02 00 00 07 47 00 42 0B C2 08 65 03
[Send data]02 00 00 07 48 42 0B C2 08 66 03
[Receive data] 02 00 00 04 48 00 08 54 03

Mifare S50 读块

[Send data]02 00 00 0B 4A 60 00 FF FF FF FF FF AF 03
[Receive data] 02 00 00 10 03 4A 00 4D 03
[Send data]02 00 00 04 4B 00 4F 03
[Receive data] 02 00 00 13 4B 00 42 0B C2 08 83 08 04 00 62 63 64 65 66 67 68 69 30 03
[Send data]02 00 00 04 4B 01 50 03
[Receive data] 02 00 00 13 4B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 5E 03
[Send data]02 00 00 04 4B 10 02 51 03
[Receive data] 02 00 00 13 4B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 5E 03
[Send data]02 00 00 04 4B 10 03 52 03
[Receive data] 02 00 00 13 4B 00 00 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF 47 03

Mifare S50 写块

[Send data]02 00 00 0B 4A 60 01 FF FF FF FF FF B0 03
[Receive data] 02 00 00 10 03 4A 00 4D 03
[Send data]02 00 00 14 4C 01 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 71 03
[Receive data] 02 00 00 10 03 4C 00 4F 03

Mifare S50 初始化钱包

[Send data]02 00 00 0B 4A 60 01 FF FF FF FF FF B0 03
[Receive data] 02 00 00 10 03 4A 00 4D 03
[Send data]02 00 00 08 4D 01 64 00 00 00 BA 03
[Receive data] 02 00 00 10 03 4D 00 50 03

Mifare S50 充值

[Send data]02 00 00 0B 4A 60 01 FF FF FF FF FF B0 03
[Receive data] 02 00 00 10 03 4A 00 4D 03
[Send data]02 00 00 08 50 01 64 00 00 00 BD 03
[Receive data] 02 00 00 10 03 50 00 53 03

Mifare S50 扣款

[Send data]02 00 00 0B 4A 60 01 FF FF FF FF FF FF B0 03

[Receive data] 02 00 00 10 03 4A 00 4D 03

[Send data]02 00 00 08 4F 01 32 00 00 00 8A 03

[Receive data] 02 00 00 10 03 4F 00 52 03

Mifare S50 读余额

[Send data]02 00 00 0B 4A 60 01 FF FF FF FF FF FF B0 03

[Receive data] 02 00 00 10 03 4A 00 4D 03

[Send data]02 00 00 04 4E 01 53 03

[Receive data] 02 00 00 07 4E 00 96 00 00 00 EB 03

Mifare one 卡片休眠

[Send data]02 00 00 10 03 29 2C 03

[Receive data] 02 00 00 10 03 29 00 2C 03

6.3 ISO14443 TYPE A T=CL CPU 卡:

初始化为 ISO14443A, 本型号可忽略

[Send data]02 00 00 04 05 00 09 03

[Receive data] 02 00 00 10 03 05 00 08 03

[Send data]02 00 00 04 3A 41 7F 03

[Receive data] 02 00 00 10 03 3A 00 3D 03

[Send data]02 00 00 04 05 01 0A 03

[Receive data] 02 00 00 10 03 05 00 08 03

T=CL CPU 卡复位

[Send data]02 00 00 04 53 52 A9 03

[Receive data] 02 00 00 0F 53 00 16 61 1B 82 10 10 78 80 90 10 02 20 90 00 C0 03

向 T=CL CPU 卡发送 COS 命令

[Send data]02 00 00 08 54 00 84 00 00 04 E4 03

[Receive data] 02 00 00 09 54 00 7B A3 5F 28 90 00 92 03

6.4 Ntag213:

Ntag 寻卡

[Send data]02 00 00 04 46 52 9C 03

[Receive data] 02 00 00 05 46 00 44 00 8F 03

Ntag 选卡

[Send data]02 00 00 10 03 33 36 03

[Receive data] 02 00 00 0A 33 00 04 4A 3C 72 9A 3D 90 03

Ntag 获取版本号

[Send data]02 00 00 10 03 87 8A 03

[Receive data] 02 00 00 0B 87 00 00 04 04 10 02 01 00 0F 10 03 AF 03

Ntag 获取数字签名

[Send data]02 00 00 10 03 8B 8E 03

[Receive data] 02 00 00 23 8B 00 D1 38 F2 2C 7C C3 AD FE D0 50 AC D4 5A 10 03 98 C8 22 AD
21 BC 75 BA 3A 1E C2 7C 60 46 A5 CC 67 19 24 03

Ntag 获取 CNT

[Send data]02 00 00 10 03 89 8C 03

[Receive data] 02 00 00 06 89 00 06 00 00 95 03

Ntag 密钥认证, 需使能卡片密钥功能

[Send data]02 00 00 07 8A FF FF FF FF 8D 03

[Receive data] 02 00 00 05 8A 00 12 34 D5 03

Ntag 写页

[Send data]02 00 00 08 35 0A 11 11 11 11 8B 03

[Receive data] 02 00 00 10 03 35 00 38 03

Ntag 快速读取

[Send data]02 00 00 05 88 00 0E 9B 03

[Receive data] 02 00 00 3F 88 00 04 4A 3C FA 72 9A 3D 80 55 48 00 00 E1 10 10 12 00 01 10 03
A0 0C 34 10 03 00 FE 00 00 00 00 22 22 22 22 11 22 33 44 00 00 00 00 11 11 11 11 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 0F 03

Ntag 读页

[Send data]02 00 00 04 4B 00 4F 03

[Receive data] 02 00 00 13 4B 00 04 4A 3C FA 72 9A 3D 80 55 48 00 00 E1 10 10 12 00 4B 03

6.5 PSAM:**PSAM 复位**

[Send data]02 00 00 04 19 00 1D 03

[Receive data] 02 FF FF 16 19 00 3B 7D 94 00 00 4B 43 76 68 10 03 4C 4B 12 10 02 16 7C AA C9
00 98 03

向 SAM 卡发送 APDU 指令

[Send data]02 00 00 09 1A 00 00 84 00 00 08 AF 03

[Receive data] 02 FF FF 0D 1A 00 D0 E9 F1 B7 7D 9E DF A8 90 00 B8 03

6.6 Desfire EV0/EV1:

按照 ISO7816-4 向 Desfire EV0/EV1 发送 APDU:

Desfire 获取随机数 (ISO7816-4):

[Send data]02 00 00 08 54 00 84 00 00 08 E8 03

[Receive data] 02 FF FF 0D 54 00 DF F3 16 2B 30 D6 A5 B1 90 00 5E 03

Desfire 选择 MF(ISO7816-4):

[Send data]02 00 00 0F 54 00 A4 04 00 07 D2 76 00 00 85 01 00 E0 03

[Receive data] 02 FF FF 05 54 00 90 00 E7 03

按照 Desfire EV0/EV1 转有指令发送 APDU:

Desfire 选择 AID (专有 APDU):

[Send data]02 00 00 0C 54 90 5A 00 00 10 03 00 00 00 00 4D 03

[Receive data] 02 FF FF 05 54 00 91 00 E8 03

Desfire 获取版本号 (专有 APDU):

[Send data]02 00 00 08 54 90 60 00 00 00 4C 03

[Receive data] 02 FF FF 0C 54 00 04 01 01 01 00 18 05 91 AF C2 03

[Send data]02 00 00 08 54 90 AF 00 00 00 9B 03

[Receive data] 02 FF FF 0C 54 00 04 01 01 01 10 03 18 05 91 AF C5 03

[Send data]02 00 00 08 54 90 AF 00 00 00 9B 03

[Receive data] 02 FF FF 13 54 00 04 2A 5D 7A CE 22 80 BA 14 91 91 70 22 10 10 91 00 FD 03

6.7 设置成 ISO14443 TYPE B 的模式:

【发送数据:】 02 00 00 04 05 00 09 03

【接收数据:】 02 00 00 10 03 05 00 08 03

【发送数据:】 02 00 00 04 3A 42 80 03

【接收数据:】 02 00 00 10 03 3A 00 3D 03

【发送数据:】 02 00 00 04 05 01 0A 03

【接收数据:】 02 00 00 10 03 05 00 08 03

6.8 ISO14443 TYPE B CPU 卡发送接收举例:

ISO14443 TYPE B 卡复位并成功:

【发送数据:】 02 00 00 04 3B 00 3F 03

【接收数据:】 02 00 00 0F 3B 00 50 28 F4 13 0B 14 00 00 00 F7 71 85 D5 03

ISO14443 TYPE B CPU 卡发送 COS 指令:

【发送数据:】 02 00 00 08 54 00 84 00 00 08 E8 03

【接收数据:】 02 00 00 0D 54 00 63 B1 66 CD B4 8F 2A 1F 90 00 C4 03

6.9 设置成 ST 卡的模式:

【发送数据:】 02 00 00 04 05 00 09 03
【接收数据:】 02 00 00 10 03 05 00 08 03
【发送数据:】 02 00 00 04 3A 73 B1 03
【接收数据:】 02 00 00 10 03 3A 00 3D 03
【发送数据:】 02 00 00 04 05 01 0A 03
【接收数据:】 02 00 00 10 03 05 00 08 03

6.10 SR176 卡发送接收举例:

SR176 卡选卡:

【发送数据:】 02 00 00 10 03 60 63 03
【接收数据:】 02 00 00 06 60 00 20 B1 66 9D 03

SR176 卡读块:

【发送数据:】 02 00 00 04 62 00 66 03
【接收数据:】 02 00 00 05 62 00 C7 C6 F4 03

SR176 卡写块:

【发送数据:】 02 00 00 06 63 04 22 33 C2 03
【接收数据:】 02 00 00 10 03 63 00 66 03

6.11 设置成 ISO15693 卡的模式:

【发送数据:】 02 00 00 04 05 00 09 03
【接收数据:】 02 00 00 10 03 05 00 08 03
【发送数据:】 02 00 00 04 3A 31 6F 03
【接收数据:】 02 00 00 10 03 3A 00 3D 03
【发送数据:】 02 00 00 04 05 01 0A 03
【接收数据:】 02 00 00 10 03 05 00 08 03

6.12 I CODE SL2 卡发送接收举例:

I CODE SL2 Inventory 寻卡:

【发送数据:】 02 00 00 10 03 70 73 03
【接收数据:】 02 00 00 0C 70 00 00 20 C1 AB 0F 00 01 04 E0 FC 03

I CODE SL2 Stay quiet 保持静默:

【发送数据:】 02 00 00 0B 71 20 C1 AB 0F 00 01 04 E0 FC 03
【接收数据:】 02 00 00 10 03 71 00 74 03

I CODE SL2 卡读块:

