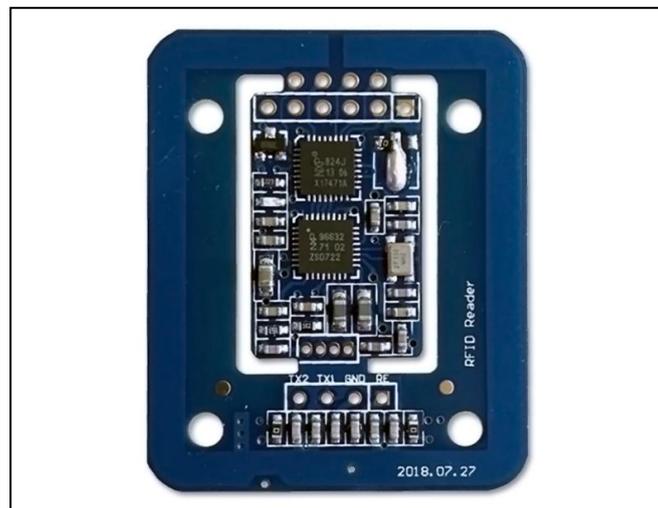




北京圆志科信 读写卡模块

应用手册



M104FHT-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	6
1.1 M104FHT-X 系列读写模块:	6
1.2 产品型号及之间的区别:	6
1.3 主要特点:	6
2.硬件描述:	7
2.1 电气特性:	7
2.2 管脚说明:	7
2.3 结构尺寸:	8
3.数据通讯协议:	9
3.1 UART 协议	9
4.不同类型卡片操作流程:	10
4.1 读写模块初始化:	10
4.2 ISO15693 卡操作步骤:	11
5.读卡器支持函数和指令汇总	11
5.1 系统函数.....	11
5.1.1 int WINAPI rf init com(初始化串口)	11
5.1.2 int WINAPI rf init type(初始化工作模式).....	12
5.1.3 int WINAPI rf antenna sta(设置天线状态).....	13
5.1.4 int WINAPI rf light(设置指示灯).....	14
5.1.5 int WINAPI rf get model(读取硬件版本号)	15
5.1.6 int WINAPI rf get snr(读取序列号)	15
5.1.7 int WINAPI rf init device number(初始设备号).....	16
5.1.8 int WINAPI rf get device number(获得设备 ID 号).....	17
5.2 ISO15693 指令集.....	18

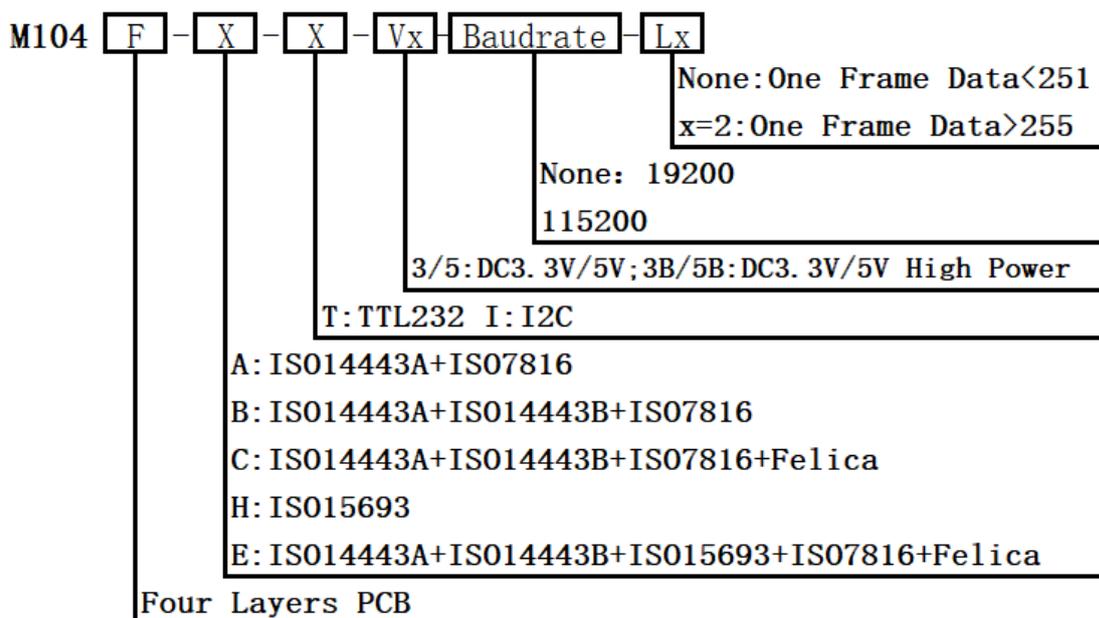
5.2.1 int WINAPI ISO15693 Inventory(寻卡).....	18
5.2.2 int WINAPI ISO15693 Get System Information(获得系统信息)	18
5.2.3 int WINAPI ISO15693 Read(读块)	19
5.2.4 int WINAPI ISO15693 Write(写块).....	20
5.2.5 int WINAPI ISO15693_Write_Multi_Block(写连续多块).....	21
5.2.6 int WINAPI ISO15693 Get Block Security(获得块锁定信息)	22
5.2.7 int WINAPI ISO15693 Lock Block(锁块).....	23
5.2.8 int WINAPI ISO15693 Write AFI(写 AFI)	24
5.2.9 int WINAPI ISO15693 Lock AFI(锁定 AFI)	25
5.2.10 int WINAPI ISO15693 Write DSFID(写 DSFID).....	26
5.2.11 int WINAPI ISO15693 Lock DSFID(锁 DSFID)	27
5.2.12 int WINAPI ISO15693 SET EAS ICODE(设置 ICODE EAS).....	27
5.2.13 int WINAPI ISO15693 EAS ALARM ICODE(I CODE EAS 报警)	28
5.2.14 int WINAPI ISO15693 RESET EAS ICODE(复位 ICODE EAS).....	29
5.2.15 int WINAPI ISO15693 LOCK EAS ICODE(锁定 ICODE EAS).....	30
5.2.16 int WINAPI ISO15693 Select(选卡).....	31
5.2.17 int WINAPI ISO15693 Inventorys(寻多张卡).....	32
5.2.18 int WINAPI ISO15693 Reset To Ready(复位)	32
5.2.19 int WINAPI ISO15693 Stay Quiet(进入静默).....	33
6.底层通讯举例 (默认 1 字节长度举例)	34
6.1 系统函数:	34
6.2 设置成 ISO15693 卡的模式(本模块可省略):	34
6.3 I CODE SL2 卡发送接收举例:	34
附录 1:硬件连接参考.....	35

1.概述

1.1 M104FHT-X 系列读写模块:

M104FHX 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 ISO15693 标准的卡片, 即 ISO15693 NXP I Code/TI Tag it。

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
M104FHTV3	UART DC3.3V 小功率	ISO15693: I CODE SL2, Tag it; 及其兼容卡片;	M104FHX I2C 参阅另 一份手册
M104FHTV5	UART DC5V 小功率		
M104FHTV3B	UART DC3.3V 大功率		
M104FHTV5B	UART DC5V 大功率		

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ISO15693: I CODE SL2, Tag it及其兼容卡片;
- 超小体积, 一体43.5x35.5mm;
分体: 25*16mm;
- 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:

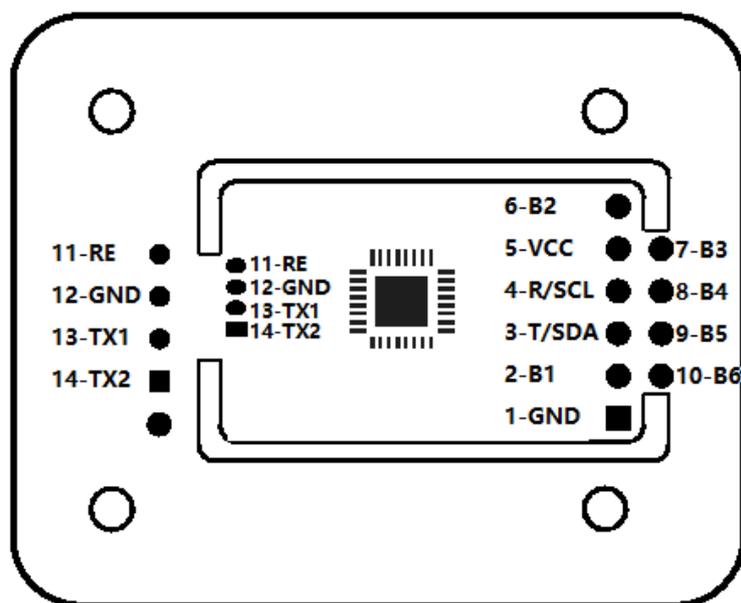
典型工作电源: DC3.3V/DC5V;

读卡电流: 大功率平均约 70 mA, 小功率平均约 40mA ;

读卡距离: 与卡片型号有关;

工作温度: -20-+70°C。

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

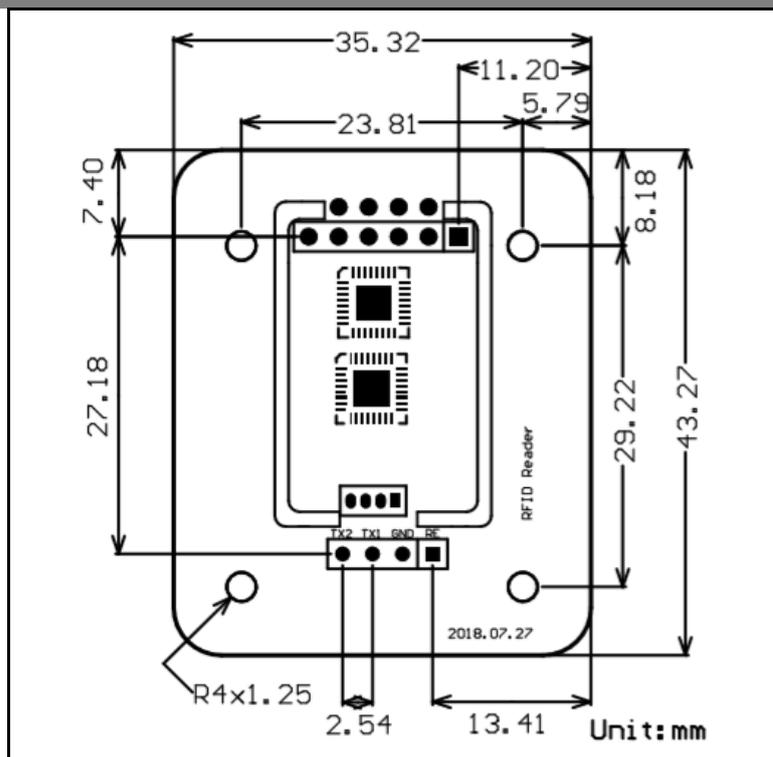
管脚	符号	描述
1	GND	电源地
2	B1	备用
3	TXD/SCL	UART 串口发送端
4	RXD/SDA	UART 串口接收端
5	VCC	电源 3.3V/5V
6	B2	备用
7	B3	备用(上电瞬间不能为低电平)
8	B4	备用
9	B5	备用
10	B6	备用
11	RE	天线数据接收管脚(天线分体时使用)
12	GND	天线地(天线分体时使用)
13	TX1	天线 1 发送管脚(天线分体时使用)
14	TX2	天线 2 发送管脚(天线分体时使用)

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；

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 0x0001~0xFFFF;

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（不包含插入的标识符 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:

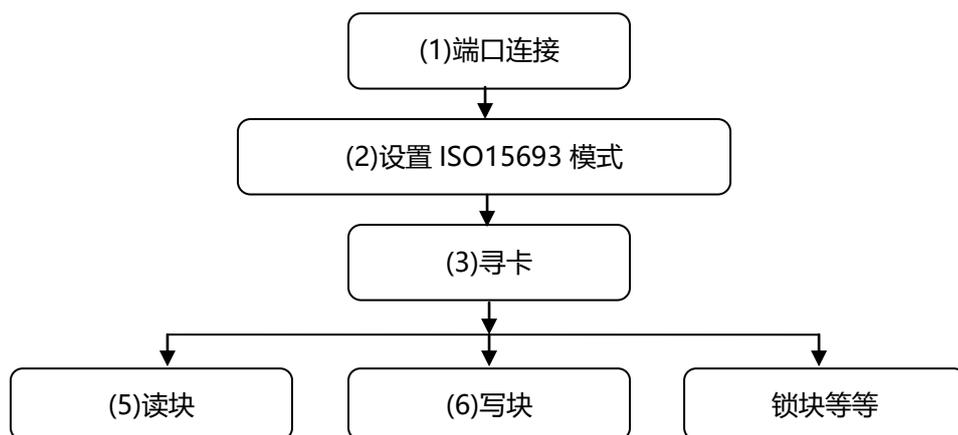
4.1 读写模块初始化:

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



本型号模块可省略

4.2 ISO15693 卡操作步骤:



5.读卡器支持函数和指令汇总

5.1 系统函数

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 ISO15693 指令集

5.2.1 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	校验	
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

5.2.2 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	0xXX	0x03

				节 UID		
--	--	--	--	-------	--	--

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 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

5.2.3 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；

bit3-bit7 备用

1 字节块号：要写入的绝对块号；

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	0x75	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	0x75	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 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

5.2.5 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；

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

5.2.8 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

5.2.9 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

5.2.10 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	校验	
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

5.2.11 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

5.2.12 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	0xXX	0x03

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

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

5.2.13 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

5.2.14 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

5.2.15 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

5.2.16 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

5.2.17 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 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	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:】

5.2.18 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	0xXX	0x03

				UID		
--	--	--	--	-----	--	--

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

5.2.19 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

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 设置成 ISO15693 卡的模式(本模块可省略):

【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

6.3 I CODE SL2 卡发送接收举例:

I CODE SL2 Inentory 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:

