



北京圆志科信 读写模块

应用手册



RW202AX-D

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本资料内容。

0.2 更改历史记录

版本	描述	日期
V1.0	PDF 版本第一版发布	2008.4.10
V1.1	为方便客户, 增加读卡器发出命令和返回结果举例	2008.8.1
V1.2	增加动态哭函数: 向读写卡器下载 Mifare One 卡密钥 用已下载的密钥验证 MifareOne 卡密钥 同时公司地址发生变更	2008.10.9
V1.3	将漏掉的设置设备地址命令补上	2008.12.11
V1.4	rf_init_com(); rf_init_device_number(); rf_get_device_number(); 此三个函数在动态库中增加了“设备标识号”	2009.1.5
V1.5	按产品分类将说明书进行分类: 说明书分别分为 RW202AX.RW202BX.RW202CX.RW202EX,RW202PX	2009.7.30
V1.6	将漏掉的 rf_ClosePort()函数补上	2012.10.08
V1.7	RW202AX 与 RW202BX 功能合并成 1 个型号	2012.12.12
V1.8	增加主动输出功能	2019.12.10

目 录

0.1 声明	2
0.2 更改历史记录	3
1. 概述	7
1.1 RW202AX IC 卡读写器简介:	7
1.2 产品型号及之间的区别:	7
1.3 技术指标	7
1.4 PSAM 的卡安装.....	7
1.5 上电状态	7
2. 硬件连接	8
2.1 RS232 串口的连接 (适用于 RW202AA)	8
2.2 RS485 的连接 (适用于 RW202AB)	8
2.3 USB 接口的连接 (适用于 RW202AC)	8
3. 通过模块操作卡片步骤简述	9
3.1 通过模块操作 Mifare One 卡步骤简述:	9
3.2 通过模块操作 ISO14443 TYPE A CPU 卡步骤简述:	9
4. 与电脑连接 (可调用圆志 DLL 动态库)	9
4.1 库函数说明.....	10
通用函数功能: 获取动态库版本号	10
通用函数功能: DES 算法加密函数.....	10
通用函数功能: DES 算法解密算法函数.....	10
通用函数功能: 初始化串口	10
通用函数功能: 关闭串口	10
通用函数功能: 蜂鸣器控制	11
通用函数功能: 控制指示灯	11
非接触卡类型选择专用功能: 设置读写卡器非接触工作方式.....	11
非接触卡通用函数功能: 设置读写卡器天线状态.....	11
M1 卡/Ultralight 专用功能: 寻 TYPE_A 卡.....	11

M1 卡/Ultralight 专用功能：命令已激活的 TYPE_A 卡进入 HALT 状态.....	12
M1 卡/Ultralight 专用功能：读取 Ultralight 卡 4 页或者 MifareOne 卡一块数据.....	12
Ultralight 专用功能：向 Ultra light 卡中写入一块数据.....	12
M1 卡专用功能：防冲突.....	12
M1 卡专用功能：选卡.....	12
M1 卡专用功能：验证 MifareOne 卡密钥.....	13
M1 卡专用功能：写入 MifareOne 卡一块数据.....	13
M1 卡专用功能：将 Mifare One 卡某一扇区初始化为钱包.....	13
M1 卡专用功能：读取 Mifare One 卡钱包值.....	13
M1 卡专用功能： Mifare One 卡扣款.....	13
M1 卡专用功能： Mifare One 卡充值.....	14
M1 卡专用功能： Mifare One 卡数据回传.....	14
M1 卡专用功能： Mifare One 卡数据传送.....	14
TYPE A CPU 卡专用功能：寻感应区内符合 ISO14443 TYPE_A 标准的 CPU 卡并复位.....	14
TYPE A CPU 卡专用功能：向符合 ISO14443-4 标准的非接触 CPU 卡发送 COS 命令.....	14
5. 底层数据通讯协议：	15
5.1 RS232 协议.....	15
5.2 命令列表.....	16
5.2.1 设置模块工作在 ISO14443 TYPE A 模式：	16
5.2.2 Mifare one/Ultralight 卡寻卡：	17
5.2.3 Ultralight 卡选卡：	18
5.2.4 Ultralight 卡读卡：	19
5.2.5 Ultralight 卡写卡：	19
5.2.6 Mifare one 卡防冲突：	20
5.2.7 Mifare one 卡选卡：	21
5.2.8 Mifare one 卡密钥验证：	22
5.2.9 Mifare one 卡读块：	23
5.2.10 Mifare one 卡写块：	24
5.2.11 Mifare one 卡初始化钱包：	24
5.2.12 Mifare one 卡读钱包：	25

5.2.13 Mifare one 卡充值：	26
5.2.14 Mifare one 卡扣款：	27
5.2.15 Mifare one 卡钱包备份步骤 1：	28
5.2.16 Mifare one 卡钱包备份步骤 2：	29
5.2.17 Mifare one/Ultralight 卡休眠：	30
5.2.18 ISO14443 TYPE A CPU 卡复位：	30
5.2.19 ISO14443 TYPE A CPU 卡发送 COS 指令：	31
5.2.20 设置波特率：	32
5.2.21 设置模块天线状态：	33
5.2.22 控制 LED 引脚状态：	34
5.2.23 设置蜂鸣器时间：	35
6. 数据发送接收举例：	36
6.1 读卡器通用命令发送接收举例：	36
6.2 Ultralight 卡发送接收举例：	36
6.3 M1 卡发送接收举例：	37
6.4 ISO14443 TYPE A CPU 卡发送接收举例：	39
7. 主动输出功能	40
7.1 UART 协议：	40
附录 1： 指令汇总表.....	40

1. 概述

1.1 RW202AX IC 卡读写器简介:

RW202AX 读写器采用 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, 只需通过简单的接口发送命令或操作函数就可以实现对卡片完全的操作。该系列读写模块支持 ISO14443-A Ultralight, Mifare One S50,S70, FM11RF08,FM1208 及其兼容卡片。

1.2 产品型号及之间的区别:

型号	主要区别	备注
RW202AA	RS232 接口, 支持 Ultralight, Mifare One S50,S70, FM11RF08, FM1208	
RW202AB	RS485 接口, 支持 Ultralight, Mifare One S50,S70, FM11RF08, FM1208	
RW202AC	USB 接口, 支持 Ultralight, Mifare One S50,S70, FM11RF08, FM1208	
RW202AT-D	TTL 串口, 支持 Ultralight, Mifare One S50,S70, FM11RF08, FM1208+主动输出功能	

1.3 技术指标

- 工作频率: 13.56MHz
- 支持的标准: ISO14443A/Mifare
- 可读卡型: Mifare 1K/4K, FM11RF08, Ultra Light, 符合 T=CL 协议的 CPU 卡 (ISO14443A)
- 供电电压: DC5V ± 10%
- 接口: RS232, RS485 或者 USB
- 通讯速率: 19.2Kbps
- 最大功耗: 100mA
- 读卡距离: 40-60mm
- 工作温度: -10°C ~ +70°C
- 外形尺寸: 120 * 84 * 25 (mm)

1.4 PSAM 的卡安装

此型号不支持 PSAM 卡, 如果需要请选择 RW202DX 系列。

1.5 上电状态

读卡器上电后的默认波特率为 19200, 黄绿发光二极管闪烁几次后熄灭, 内部红发光二极管长亮:

2. 硬件连接

2.1 RS232 串口的连接 (适用于 RW202AA)

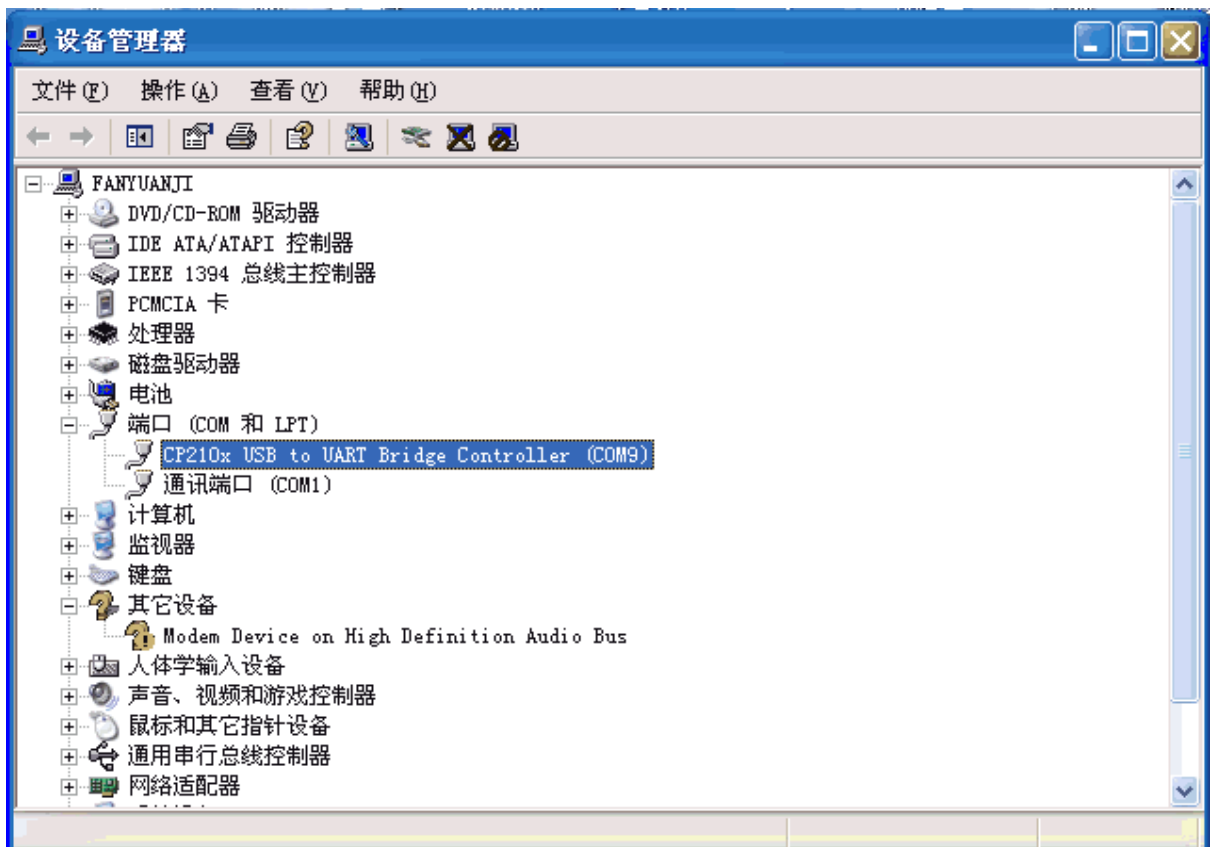
9 针 D 型串口座插到计算机的 COM 口, 同时将读卡器接口线上附带的 USB 取电口插到计算机 USB 口上进行取电就可以了;

2.2 RS485 的连接 (适用于 RW202AB)

读卡器有 4 根出线, 白色为 RS485 的 A, 灰色为 485 的 B, 红色为电源 5V, 黑色为地线(电源的负极), 按对应线序进行硬件连线, 确保连接无误后即可。

2.3 USB 接口的连接 (适用于 RW202AC)

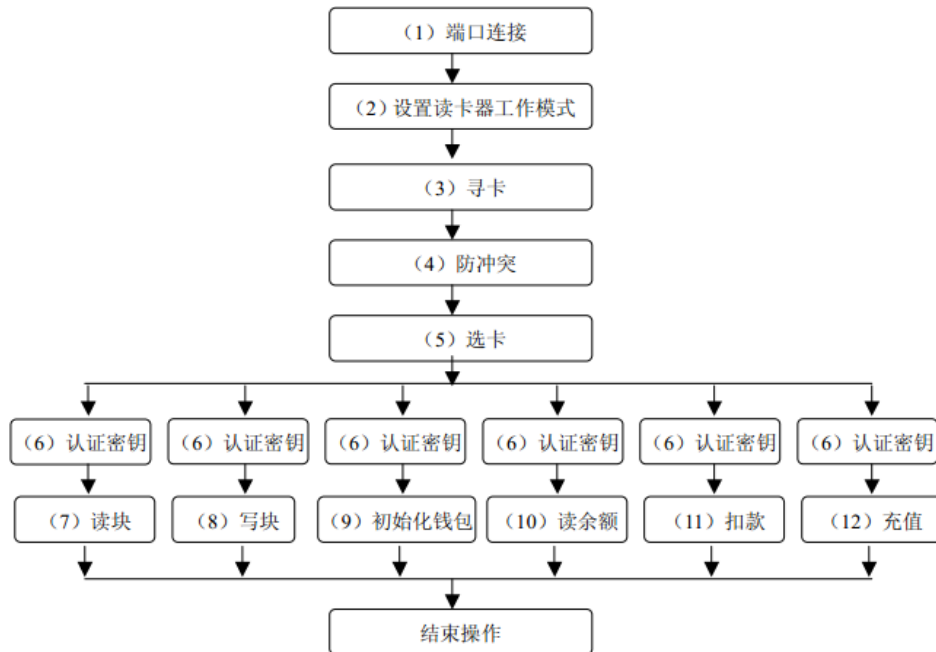
此种硬件连接比较简单, USB 口直接插到电脑的 USB 口上就可以了, 需要注意的是此 USB 口需要安装驱动程序, 以模拟出一个串口, 驱动可从附带资料中找到, 驱动安装成功之后, 在设备管理器中会显示出虚拟的串口, 比如下图为 COM9, 至此, 硬件连接成功。



备注: 硬件连接成功之后, 可通过我公司附带的 PC_DEMO 测试软件进行常规功能测试, 以协助您进行二次开发, 具体演示软件说明, 请参阅附带的 PC_DEMO 说明。

3. 通过模块操作卡片步骤简述

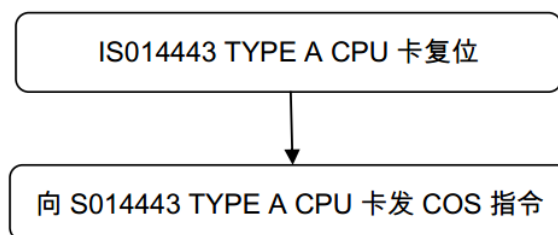
3.1 通过模块操作 Mifare One 卡步骤简述:



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;

3.2 通过模块操作 ISO14443 TYPE A CPU 卡步骤简述:



4. 与电脑连接 (可调用圆志 DLL 动态库)

注: 在调用下面 DLL 函数之前, 对于 Win32 系统需将该对应动态库 MasterRDnew.dll 和 MasterCom.dll 复制到 C:\Windows\system32 文件夹下; 而对于 Win64 系统需要将该对应动态库 MasterRDnew.dll 和 MasterCom.dll 复制到 C:\Windows\SysWOW64 文件夹下.

4.1 库函数说明

通用函数功能：获取动态库版本号

原型：int WINAPI lib_ver(unsigned int *nVer)

参数：*nVer：2 字节动态库版本号

返回：成功返回 0

通用函数功能：DES 算法加密函数

原型：int (WINAPI* des_encrypt)(unsigned char *szOut, unsigned char *szIn, unsigned int inlen, unsigned char *key, unsigned int keylen);

参数：szOut：输出的 DES 值，长度等于明文长度

szIn：明文

inlen：明文长度,8 字节的整数倍

key：密钥

keylen：密钥长度,如果大于 8 字节，是 3des,如果小于等于 8 字节单 des.不足补零

返回：成功返回 0

通用函数功能：DES 算法解密算法函数

原型：int (WINAPI* des_decrypt)(unsigned char *szOut, unsigned char *szIn, unsigned int inlen, unsigned char *key, unsigned int keylen);

参数：szOut：输出的 DES 值，长度等于密文长度

szIn：密文

inlen：密文长度,8 字节的整数倍

key：密钥

keylen：密钥长度,如果大于 8 字节，是 3des,如果小于等于 8 字节单 des.不足补零

返回：成功返回 0

通用函数功能：初始化串口

原型：int WINAPI rf_init_com (unsigned short icdev,int port,long baud)

参数：icdev：通讯设备标识符，0-65536

port：串口号，取值为 1~9

baud：为通讯波特率 9600~115200 (9600, 14400, 19200, 28800, 38400, 57600, 115200)

读卡器上电默认波特率为 19200

返回：成功返回 0

通用函数功能：关闭串口

原型：int WINAPI rf_ClosePort ()

参数：空

返回：成功返回 0

通用函数功能：蜂鸣器控制

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

参数：icdev： 通讯设备标识符， 0-65536

msec： 蜂鸣时限， 单位是 10 毫秒

返回：成功返回 0

通用函数功能：控制指示灯

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

参数：icdev： 通讯设备标识符， 0-65536

color:

0 = 熄灭黄灯

1 = 熄灭绿灯

2 = 点亮绿灯

3 = 点亮黄灯

非接触卡类型选择专用功能：设置读写卡器非接触工作方式

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

参数：icdev： 通讯设备标识符， 0-65536

type： 读写卡器工作方式

返回：成功返回 0

说明：type='A'： 设置为 TYPE_A 方式

非接触卡通用函数功能：设置读写卡器天线状态

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

参数：icdev： 通讯设备标识符， 0-65536

model： 天线状态

返回：成功返回 0

说明：model=0： 关闭天线 model=1： 开启天线

M1 卡/Ultralight 专用功能：寻 TYPE_A 卡

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

参数：icdev： 通讯设备标识符， 0-65536

model： 寻卡模式

TagType： 返回卡类型值

返回：成功返回 0

说明：mode=0x26： 寻未进入休眠状态的卡

mode=0x52： 寻所有状态的卡

M1 卡/Ultralight 专用功能：命令已激活的 TYPE_A 卡进入 HALT 状态

原型：int WINAPI rf_halt(unsigned short icdev)

参数：icdev： 通讯设备标识符， 0-65536

返回：成功返回 0

M1 卡/Ultralight 专用功能：读取 Ultralight 卡 4 页或者 MifareOne 卡一块数据

原型：int WINAPI rf_M1_read (unsigned short icdev, unsigned char block, unsigned char *pData,unsigned char *pLen)

参数：icdev： 通讯设备标识符， 0-65536

block： M1 卡绝对块号

pData： 读出数据

pLen： 读出数据的长度

返回：成功返回 0

Ultralight 专用功能：向 Ultra light 卡中写入一块数据

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

参数：icdev： [IN] 通讯设备标识符

page： [IN] ultra light 卡页地址 (0 ~ 0x0f)

pData： [IN] 写入的数据， 4 字节

返回：成功返回 0

M1 卡专用功能：防冲突

原型：int WINAPI rf_anticoll(word icdev, unsigned char bcnt, unsigned char *pSnr, unsigned char *pLen)

参数：icdev： 通讯设备标识符.(0-65536)

bcnt： 值=4

pSnr： 返回卡号

pLen： 返回卡号长度

返回：成功返回 0

M1 卡专用功能：选卡

原型：int WINAPI rf_select(word icdev, unsigned char *pSnr, unsigned char snrLen, unsigned char *pSize)

参数：icdev： 通讯设备标识符. (0-65536)

pSnr： 卡号

snrLen： 卡号长度

pSize： 卡的类型

返回： 成功返回 0

说明： 当选卡指令执行完毕后，卡片进入到激活状态，同时在同样的天线区域，仅有一张卡片处

于激活状态。

M1 卡专用功能：验证 MifareOne 卡密钥

原型：int WINAPI rf_M1_authentication2(unsigned short icdev, unsigned char model, unsigned char block, unsigned char *key)

参数：icdev： 通讯设备标识符， 0-65536

model： 密码验证模式

block： 要验证密码的绝对块号

key： 密钥内容， 6 字节

返回：成功返回 0

说明：model=0x60： 验证 A 密钥

model=0x61： 验证 B 密钥

M1 卡专用功能：写入 MifareOne 卡一块数据

原型：int WINAPI rf_M1_write (unsigned short icdev, unsigned char block, unsigned char *data)

参数：icdev： 通讯设备标识符， 0-65536

block： M1 卡绝对块号

data： 写入的数据， 16 字节

返回：成功返回 0

M1 卡专用功能：将 Mifare One 卡某一扇区初始化为钱包

原型：int WINAPI rf_M1_initval (unsigned short icdev, unsigned char block, long value)

参数：icdev： 通讯设备标识符， 0-65536

block： M1 卡绝对块号

value： 初始值， 16 进制， 低字节在前

返回：成功返回 0

M1 卡专用功能：读取 Mifare One 卡钱包值

原型：int WINAPI rf_M1_readval(WORD icdev, unsigned char block, long* pValue)

参数：icdev： 通讯设备标识符， 0-65536

block： M1 卡绝对块号

pValue： 返回的值， 16 进制， 低字节在前

返回：成功返回 0

M1 卡专用功能： Mifare One 卡扣款

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

参数：icdev： 通讯设备标识符， 0-65536

block： M1 卡绝对块号

value： 要扣的值， 16 进制， 低字节在前

返回：成功返回 0

M1 卡专用功能： Mifare One 卡充值

原型：int WINAPI rf_M1_increment (unsigned short icdev, unsigned char block, long value)

参数：icdev： 通讯设备标识符， 0-65536

block： M1 卡绝对块号

value： 要增加的值， 16 进制， 低字节在前

返回：成功返回 0

M1 卡专用功能： Mifare One 卡数据回传

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

参数：icdev： 通讯设备标识符， 0-65536

block： M1 卡绝对块号

返回：成功返回 0

说明：用此函数将指定的块内容传入卡的 buffer，然后可用 rf_M1transfer()函数将 buffer 中数据再传送到另一块中去

M1 卡专用功能： Mifare One 卡数据传送

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

参数：icdev： 通讯设备标识符， 0-65536

block： M1 卡绝对块号

返回：成功返回 0

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

TYPE A CPU 卡专用功能： 寻感应区内符合 ISO14443 TYPE_A 标准的 CPU 卡并复位

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

参数：icdev： [IN] 通讯设备标识符

model： [IN] 寻卡方式

pDate： [OUT]返回的数据

pMsgLg： [OUT]返回数据的长度

返回：成功返回 0

说明：mode = 0x26： 寻未进入休眠状态的卡

mode = 0x52： 寻所有状态的卡

pDate： 4 字节 CSN + 复位信息内容

TYPE A CPU 卡专用功能： 向符合 ISO14443-4 标准的非接触 CPU 卡发送 COS 命令

原型：int WINAPI rf_cos_command(word icdev, unsigned char *pCommand, unsigned char cmdLen, unsigned char *pData, unsigned char *pMsg)

参数：icdev： [IN] 通讯设备标识符

pCommand: [IN] cos 命令
 cmdLen: [IN] cos 命令长度
 pDate: [OUT]卡片返回的数据, 含 SW1、SW2
 pMsgLg: [OUT]返回数据长度

返回: 成功则返回 0

5. 底层数据通讯协议:

5.1 RS232 协议

- RS232 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的逐字节累加值 (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的逐字节累加值（最后一字节）。

5.2 命令列表

5.2.1 设置模块工作在 ISO14443 TYPE A 模式：

功能描述：用于设置模块工作于 ISO14443 TYPE A 工作模式：

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	0x3A	0x41	0x7F	0x03

注：数据部分为 1 字节模式控制字；

发送数据=“A”表示使模块工作于 ISO14443 TYPE A 模式，对应 ASCII 码为 0x41；

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	0x3A	0x00	空	0x3D	0x03

注：黄色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	0x3A	非零	无	XXXX	0x03

注：黄色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

5.2.2 Mifare one/Ultralight 卡寻卡：

功能描述：用于 Mifare one/Ultralight 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	0x26 或者	0x70 或者	0x03
				0x52	0x9C	

注：数据部分为 1 字节寻卡模式；

发送数据 = "0x26" 寻未进入睡眠状态的卡；

发送数据 = "0x52" 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	0x00,0x00	0x05	0x46	0x00	0x04 0x00 或者	0x4F 或者	0x03
					0x10	0x02 0x00	

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；

注：黄色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	0x46	非零	空	XXXX	0x03

注：黄色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）

5.2.3 Ultralight 卡选卡：

功能描述：用于选择 UltraLight 卡片，此条指令需要在 0x46 寻卡指令之后操作。

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	0x33	空	0x36	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，即黄色的部分；

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	0x00,0x00	XXXX	0x33	0x00	7 字节返回的卡号	XXXX	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	0x33	非零	空	XXXX	0x03

注：黄色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

5.2.4 Ultralight 卡读卡:

功能描述: 用于 Ultralight 卡读写操作, 此指令与 mifare one 读块相同, 此指令读出的为连续 4 页的内容。

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	XXXX	0x4B	1 字节的起始页的页号	XXXX	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	XXXX	0x4B	0x00	返回从起始页开始的连续 4 页内容	XXXX	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	0x4B	非零	空	XXXX	0x03

注: 黄色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 (此例读出从页 0 到页 3 的连续 4 页的内容):

【发送数据:】 02 00 00 04 4B 00 4F 03

【接收数据:】 02 00 00 13 4B 00 04 6E F0 12 BA E1 22 80 F9 48 00 00 00 00 00 50 03

5.2.5 Ultralight 卡写卡:

功能描述: 用于选择 UltraLight 卡片, 此条指令需要在 0x46 寻卡指令之后操作。

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	0x08	0x35	1 字节页号+4 字节写入数据	0x36	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	0x00,0x00	0x10	0x03	0x35	0x00	空	0x38	0x03

注: 黄色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	0x35	非零	空	XXXX	0x03

注: 黄色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

5.2.6 Mifare one 卡防冲突:

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**, 并且如果需要对卡进行读写等操作时, 在该条指令之后还要发送**选卡指令**:

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	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" Mifare S50,S70,FM11RF08 卡序列号为 4 字节, 故数据为 0x04;

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	0x47	0x00	4 字节卡号	XXXX	0x03
------	-----------	------	------	------	--------	------	------

返回 4 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	0x00,0x00	0x10	0x03	0x47	非零	空	XXXX	0x03

注: 黄色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)

5.2.7 Mifare one 卡选卡:

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。

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	XXXX	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	0x00,0x00	0x04	0x48	0x00	0x08 或者	XXXX	0x03
					0x20		

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	0x00,0x00	0x10	0x03	0x48	非零	空	XXXX	0x03

注：黄色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）

5.2.8 Mifare one 卡密钥验证：

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	XXXX	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“FFFFFFFF”；

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	0x4A	0x00	空	0x4D	0x03

注：黄色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	0x4A	非零	空	XXXX	0x03
------	-----------	------	------	------	----	---	------	------

注：黄色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

5.2.9 Mifare one 卡读块：

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**。

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	XXXX	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	0x00,0x00	0x13	0x4B	0x00	16 字节内容	XXXX	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	0x00,0x00	0x10	0x03	0x4B	非零	空	XXXX	0x03

注：黄色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)

5.2.10 Mifare one 卡写块:

功能描述: 用于 Mifare one 卡的写块内容操作, 注在写块前需进行**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	0x14	0x4C	1 字节绝对块号+16 字节写入内容	XXXX	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);

16 字节要写入的内容;

对于密钥块 (每个扇区的最后一块) 的写操作一定要谨慎, 否则有可能造成该扇区的失效, 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	0x00,0x00	0x10	0x03	0x4C	0x00	空	0x4F	0x03

注: 黄色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	0x4C	非零	空	XXXX	0x03

注: 黄色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 14 4C 01 11 71 03
(将第一块写成 "11111111111111111111111111111111"))

【接收数据:】 02 00 00 10 03 4C 00 4F 03

5.2.11 Mifare one 卡初始化钱包:

功能描述: 用于 Mifare one 卡的钱包初始化操作, 注在钱包初始化前需进行**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	0x4D	1 字节绝对块号+4 字节初始化金额	XXXX	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 字节初始化金额；4 字节 16 进制初始化金额，低字节在前；

密钥块（每个扇区的最后一块）不能初始化为钱包，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	0x00,0x00	0x10	0x03	0x4D	0x00	空	0x50	0x03

注：黄色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	0x4D	非零	空	XXXX	0x03

注：黄色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 4D 01 64 00 00 00 BA 03 (01 块初始化为钱包, 金额 16 进制 "64")

【接收数据:】02 00 00 10 03 4D 00 50 03

5.2.12 Mifare one 卡读钱包：

功能描述：用于 Mifare one 卡的钱包读取操作，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	

0x02	0x00,0x00	0x04	0x4E	1 字节绝对块号	XXXX	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	0x00,0x00	0x07	0x4E	0x00	4 字节 16 进制 钱包值	XXXX	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	0x00,0x00	0x10	0x03	0x4E	非零	空	XXXX	0x03

注：黄色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 Mifare one 卡充值：

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)	XXXX	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	0x00,0x00	0x10	0x03	0x50	0x00	空	0x53	0x03

注：黄色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	0x50	非零	空	XXXX	0x03

注：黄色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

5.2.14 Mifare one 卡扣款：

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)	XXXX	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	0x00,0x00	0x10	0x03	0x4F	0x00	空	0x52	0x03

注：黄色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	0x4F	非零	空	XXXX	0x03

注：黄色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

5.2.15 Mifare one 卡钱包备份步骤 1：

功能描述：用于将 Mifare one 卡的指定块的钱包内容回传至卡的 Buffer，然后通过钱包备份步骤 2 将钱包值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	XXXX	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	0x00,0x00	0x10	0x03	0x51	0x00	空	0x54	0x03

注：黄色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	0x51	非零	空	XXXX	0x03

注：黄色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

5.2.16 Mifare one 卡钱包备份步骤 2：

功能描述：用于将 Mifare one 卡通过钱包备份步骤 1 传入 Buffer 中的钱包值备份到**同一扇区**的指定块，注在此操作前需进行 **Mifare one 卡钱包备份步骤 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	XXXX	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	0x00,0x00	0x10	0x03	0x52	0x00	空	0x55	0x03

注：黄色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	0x52	非零	空	XXXX	0x03

注：黄色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

5.2.17 Mifare one/Ultralight 卡休眠:

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，即黄色的部分；

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：黄色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	XXXX	0x03

注：黄色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

5.2.18 ISO14443 TYPE A CPU 卡复位:

功能描述: 用于进行满足 ISO14443 TYPE A 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 表示寻天线范围内的未休眠卡；

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	0x00,0x00	XXXX	0x53	0x00	4 字节 CSN+返回 信息	XXXX	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	0x53	非零	空	XXXX	0x03

注：黄色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 53 52 A9 03

【接收数据：】 02 00 00 0F 53 00 16 61 1B 82 10 10 78 80 90 10 02 20 90 00 C0 03

5.2.19 ISO14443 TYPE A CPU 卡发送 COS 指令：

功能描述：用于发送 ISO14443 TYPE A 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	XXXX	0x54	COS 指令	XXXX	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	0x00,0x00	XXXX	0x54	0x00	针对 COS 指令的返回数据	XXXX	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	0x54	非零	空	XXXX	0x03

注: 黄色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)

5.2.20 设置波特率:

功能描述: 用于设置模块内部的通讯波特率:

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	插入	发送数据	校验	
0x02	0x00,0x00	0x04	0x15	0x10	0x03	0x1C	0x03

发送数据=0x01 表示设置模块波特率为 9600;

发送数据=0x02 表示设置模块波特率为 14400;

发送数据=0x03 表示设置模块波特率为 19200;

发送数据=0x04 表示设置模块波特率为 28800;

发送数据=0x05 表示设置模块波特率为 38400;

发送数据=0x06 表示设置模块波特率为 57600;

发送数据=0x07 表示设置模块波特率为 115200;

特别注意: 模块默认的波特率是 19200, 模块不会保存此设置, 当重新上电后, 模块又恢复到默认的 19200, 故当更改波特率后, 因中间掉电有可能造成通讯故障;

注：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，即黄色的部分；

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	0x15	0x00	空	0x18	0x03

注：黄色部分 0x10 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	0x15	非零	空	XXXX	0x03

注：黄色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 15 10 03 1C 03

【接收数据:】 02 00 00 10 03 15 00 18 03

5.2.21 设置模块天线状态：

功能描述：用于设置模块的天线工作状态：

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	0x00 或者	0x09 或者	0x03
				0x01	0x0A	

注：发送数据 bit0=0 关闭天线；
 发送数据 bit0=1 开启天线；
 发送数据 bit1=0 关闭自动寻卡；
 发送数据 bit1=1 开启自动寻卡；

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	插入	长度	命令	执行结果	返回数据	校验	

0x02	0x00,0x00	0x10	0x03	0x05	0x00	空	0x08	0x03
------	-----------	------	------	------	------	---	------	------

注：黄色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	0x05	非零	空	XXXX	0x03

注：黄色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

5.2.22 控制 LED 引脚状态：

功能描述：用于控制模块 LED 引脚的状态：

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	数据	校验	
0x02	0x00,0x00	0x04	0x6A	0x00 或者	0x6E 或者	0x03
				0x10	0x03	

注：数据部分为 1 字节 LED 引脚状态控制：

数据=0x00 黄灯熄灭；

数据=0x01 绿灯熄灭；

数据=0x02 绿灯点亮；

数据=0x03 黄灯点亮；

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，即黄色的部分；

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	0x6A	0x00	空	0x6D	0x03

注：黄色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	0x6A	非零	空	XXXX	0x03

注：黄色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 00 6E 03

【接收数据:】 02 00 00 10 03 6A 00 6D 03

5.2.23 设置蜂鸣器时间：

功能描述：用于设置蜂鸣器时间：

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	数据	校验	
0x02	0x00,0x00	0x04	0x1D	0x00~0xFF	0x21 或者	0x03
				0x10	0x03	

注：数据部分为 1 字节蜂鸣时间，范围 0x00~0xFF，单位为 ms

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，即黄色的部分；

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	0x1D	0x00	空	0x20	0x03

注：黄色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	0x1D	非零	空	XXXX	0x03

注：黄色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 1D 10 03 71 03

【接收数据:】 02 00 00 10 03 1D 00 21 03

6. 数据发送接收举例:

6.1 读卡器通用命令发送接收举例:

端口连接并成功:

【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

控制蜂鸣器声音长短:

【发送数据:】 02 00 00 04 1D 50 71 03

【接收数据:】 02 00 00 10 03 1D 00 20 03

点亮黄色 LED 灯:

【发送数据:】 02 00 00 04 6A 10 03 71 03

【接收数据:】 02 00 00 10 03 6A 00 6D 03

熄灭黄色 LED 灯:

【发送数据:】 02 00 00 04 6A 00 6E 03

【接收数据:】 02 00 00 10 03 6A 00 6D 03

6.2 Ultralight 卡发送接收举例:

Ultralight 卡寻卡并成功

【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 41 7F 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

【发送数据:】 02 00 00 04 46 52 9C 03

【接收数据:】 02 00 00 05 46 00 44 00 8F 03

【发送数据:】 02 00 00 10 03 33 36 03

【接收数据:】 02 00 00 0A 33 00 04 DB CF 5 E3 25 80 C4 03

读卡并成功

【发送数据:】 02 00 00 04 4B 00 4F 03

【接收数据:】 02 00 00 13 4B 00 04 DB CF 98 51 E3 25 80 17 48 00 00 00 91 53 E5 A5 03

【发送数据:】 02 00 00 04 4B 04 53 03

【接收数据:】 02 00 00 13 4B 00 04 DB CF 98 51 E3 25 80 17 48 00 00 00 91 53 E5 A5 03

【发送数据:】 02 00 00 04 4B 08 57 03

【接收数据:】 02 00 00 13 4B 00 04 DB CF 98 51 E3 25 80 17 48 00 00 00 91 53 E5 A5 03

【发送数据:】 02 00 00 04 4B 0C 5B 03

【接收数据:】 02 00 00 13 4B 00 04 DB CF 98 51 E3 25 80 17 48 00 00 00 91 53 E5 A5 03

写页, 将页 4 全写成 1:

【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

Ultralight 卡休眠

【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

6.3 M1 卡发送接收举例:

Mifare S50 卡寻卡并成功

【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 41 7F 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

【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

【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

【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

读扇区 0 并成功

【发送数据:】 02 00 00 0B 4A 60 00 FF FF FF FF FF AF 03

【接收数据:】 02 00 00 10 03 4A 00 4D 03

【发送数据:】 02 00 00 04 4B 00 4F 03

【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

【发送数据:】 02 00 00 04 4B 01 50 03

【接收数据:】 02 00 00 13 4B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 5E 03

【发送数据:】 02 00 00 04 4B 10 02 51 03

【接收数据:】 02 00 00 13 4B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 5E 03

【发送数据:】 02 00 00 04 4B 10 03 52 03

【接收数据:】 02 00 00 13 4B 00 00 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF 47 03

写扇区 0 块 1, 将块 1 全写成 1:

【发送数据:】 02 00 00 0B 4A 60 01 FF FF FF FF FF B0 03

【接收数据:】 02 00 00 10 03 4A 00 4D 03

【发送数据:】 02 00 00 14 4C 01 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 71 03

【接收数据:】 02 00 00 10 03 4C 00 4F 03

将扇区 0 块 1 初始化为钱包, 初始值为 100:

【发送数据:】 02 00 00 0B 4A 60 01 FF FF FF FF FF B0 03

【接收数据:】 02 00 00 10 03 4A 00 4D 03

【发送数据:】 02 00 00 08 4D 01 64 00 00 00 BA 03

【接收数据:】 02 00 00 10 03 4D 00 50 03

将扇区 0 块 1 充值 100:

【发送数据:】 02 00 00 0B 4A 60 01 FF FF FF FF FF FF B0 03

【接收数据:】 02 00 00 10 03 4A 00 4D 03

【发送数据:】 02 00 00 08 50 01 64 00 00 00 BD 03

【接收数据:】 02 00 00 10 03 50 00 53 03

将扇区 0 块 1 扣款 50:

【发送数据:】 02 00 00 0B 4A 60 01 FF FF FF FF FF FF B0 03

【接收数据:】 02 00 00 10 03 4A 00 4D 03

【发送数据:】 02 00 00 08 4F 01 32 00 00 00 8A 03

【接收数据:】 02 00 00 10 03 4F 00 52 03

读扇区 0 块 1 余额为 150:

【发送数据:】 02 00 00 0B 4A 60 01 FF FF FF FF FF FF B0 03

【接收数据:】 02 00 00 10 03 4A 00 4D 03

【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

Mifare one 卡休眠

【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

6.4 ISO14443 TYPE A CPU 卡发送接收举例:

FM1208 卡复位并成功:

【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 41 7F 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

【发送数据:】 02 00 00 04 53 52 A9 03

【接收数据:】 02 00 00 0F 53 00 16 61 1B 82 10 10 78 80 90 10 02 20 90 00 C0 03

FM1208 发送 COS 指令(0084000004)并成功返回:

【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

7. 主动输出功能

7.1 UART 协议:

上电默认主动输出卡号。

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。**

波特率可选择:

输出波特率 : 19200,8, N,1;

输出格式: 0xAA 0x55 + 0x04/0x07/0x0A(卡号长度)+具体卡号+校验 (从长度字到卡号最后

1 字节异或校验)。

蜂鸣器: 上电默认有卡响一声。

此功能只适用于 RW210AA-D 与 RW210AC-D。

特别说明: 读卡器同样具有读写卡片功能, 收到上位机指令之后, 会关闭主动输出卡号功能, 读写卡片完成之后, 如果需要主动输出功能, 需要通过设置天线指令进行打开主动输出功能。

附录 1: 指令汇总表

序号	指令	描述	执行正确返回	执行错误返回
1	0x3A	设置模块工作在 ISO14443 TYPE A 状态	0x00	非 0
2	0x46	Mifare one/Ultralight 寻卡	0x00	非 0
3	0x33	Ultralight 选卡	0x00	非 0
4	0x4B	Ultralight 读块	0x00	非 0
5	0x35	Ultralight 写块	0x00	非 0
6	0x47	Mifare one 防冲突	0x00	非 0
7	0x48	Mifare one 选卡	0x00	非 0
8	0x4A	Mifare one 密钥验证	0x00	非 0
9	0x4B	Mifare one 读块	0x00	非 0
10	0x4C	Mifare one 写块	0x00	非 0

序号	指令	描述	执行正确返回	执行错误返回
11	0x4D	Mifare one 初始化钱包	0x00	非 0
12	0x4E	Mifare one 读钱包	0x00	非 0
13	0x50	Mifare one 钱包充值	0x00	非 0
14	0x4F	Mifare one 钱包扣款	0x00	非 0
15	0x51	Mifare one 钱包备份步骤 1	0x00	非 0
16	0x52	Mifare one 钱包备份步骤 2	0x00	非 0
17	0x29	Mifare one/Ultralight 卡休眠	0x00	非 0
18	0x53	ISO14443 TYPE A CPU 卡复位	0x00	非 0
19	0x54	ISO14443 TYPE A CPU 卡发送 COS 指令	0x00	非 0
20	0x15	设置波特率	0x00	非 0
21	0x05	设置模块天线状态	0x00	非 0
22	0x6A	控制 LED 引脚状态	0x00	非 0
23	0x1D	设置蜂鸣时间	0x00	非 0