

NFC & MIFARE & ISO14443A&B & ISO15693 & ISO18000 & ISO7816 IC 卡读写模块

# JMY600 系列读写卡模块

---

## 通用技术手册

(Revision 6.20)

北京金木雨电子有限公司

2021/5/6



在使用本产品前请仔细阅读本说明书，如果有任何疑问，请联系我们，我们会给您详尽的解答



# 目录

文档更新记录.....	7
1 概述.....	8
2 功能.....	8
2.1 基本工作模式.....	8
2.2 自动寻卡.....	8
2.3 自动寻卡并输出卡序列号.....	9
2.4 空闲模式.....	9
2.5 关闭模式.....	9
2.6 EMV/PBOC 认证模式.....	9
2.7 默认功能配置.....	9
2.8 ISP 功能.....	10
2.9 读卡器安全认证功能.....	10
2.10 加密通讯功能.....	10
2.11 NFC 目标功能.....	10
2.12 NFC Tag 模拟模式.....	10
3 通讯协议.....	11
3.1 测试工具软件.....	11
3.2 JCP04 通讯协议.....	11
3.2.1 数据发送格式.....	11
3.2.2 数据返回格式.....	11
3.3 JCP05 通讯协议.....	11
3.3.1 数据包发送格式.....	11
3.3.2 数据包返回格式.....	12
3.4 数据包返回时间.....	12
4 通讯接口.....	13
4.1 UART 和 RS232C 接口.....	13
4.1.1 物理接口.....	13
4.1.2 通讯过程.....	13
4.2 IIC 接口.....	13
4.2.1 物理接口.....	13
4.2.2 数据链路.....	14
4.2.2.1 时钟和数据交换.....	14
4.2.2.2 开始条件 (Start Condition).....	14
4.2.2.3 结束条件 (Stop Condition).....	14
4.2.2.4 确认符 (ACK).....	14
4.2.2.5 总线状态.....	14
4.2.2.6 设备地址.....	15
4.2.2.7 写数据操作.....	15
4.2.2.8 读数据操作.....	15
4.2.3 通讯过程.....	15
4.2.4 通讯过程描述.....	16
4.3 SPI 接口.....	16



4.3.1	物理接口.....	16
4.3.1.1	总线工作方式.....	16
4.3.1.2	写数据操作.....	17
4.3.1.3	读数据操作.....	17
4.3.2	数据链路.....	18
4.3.2.1	SPI 指令.....	18
4.3.2.2	SPI 状态.....	18
4.3.3	通迅过程描述.....	18
4.4	USB 接口.....	19
5	应用层协议详解.....	20
5.1	说明.....	20
5.2	系统设置命令.....	21
5.2.1	复位模块配置至出厂状态.....	21
5.2.2	读产品信息.....	21
5.2.3	读 PCD 的产品信息（包含 UID）.....	22
5.2.4	模块工作模式设定.....	22
5.2.5	设置模块为空闲模式.....	23
5.2.6	设置 LED.....	24
5.2.7	设置蜂鸣器.....	24
5.2.8	读片内数据 FLASH.....	25
5.2.9	写片内数据 FLASH.....	25
5.2.10	读片外数据 FLASH.....	26
5.2.11	写片外数据 FLASH.....	26
5.2.12	设定 UART 通讯波特率.....	27
5.2.13	设定 UART 多机通讯地址.....	27
5.2.14	设定 IIC 通讯地址.....	28
5.2.15	设定多卡操作.....	28
5.2.16	设定 ISO15693 自动寻卡的 AFI 和 AFI 使能.....	29
5.2.17	设定自动寻卡时间间隔.....	29
5.2.18	设定开机时的默认自动寻卡状态.....	30
5.2.19	设定自动寻卡并输出卡片序列号开机默认状态.....	30
5.2.20	设定射频输出功率.....	31
5.2.21	设定读卡模式.....	32
5.2.22	设定当前读卡天线.....	32
5.3	读卡模块安全认证.....	33
5.3.1	取认证码.....	33
5.3.2	模块安全认证.....	33
5.3.3	修改模块认证密钥.....	34
5.3.4	加密通讯通道.....	34
5.4	ISO14443A/B CPU 卡命令.....	35
5.4.1	ISO14443 TYPE A 寻卡.....	35
5.4.2	ISO14443 TYPE A 寻全部卡.....	35
5.4.3	ISO14443-4 TYPE A 卡复位（RATS）.....	36
5.4.4	ISO14443-4 TYPE B 寻卡.....	36



5.4.5	按照 EMV 或 PBOC 的规范寻卡.....	37
5.4.6	发送 APDU 到 ISO14443-4 卡片 .....	38
5.4.7	ISO14443-4 TYPE B 卡休眠.....	38
5.5	MIFARE 1K/4K/mini 卡命令 .....	40
5.5.1	MIFARE 寻卡 .....	40
5.5.2	MIFARE 1K/4K 读数据块 .....	40
5.5.3	MIFARE 1K/4K 读多个数据块 .....	41
5.5.4	MIFARE 1K/4K 写数据块 .....	41
5.5.5	MIFARE 1K/4K 写多个数据块 .....	42
5.5.6	MIFARE 1K/4K 初始化钱包 .....	42
5.5.7	MIFARE 1K/4K 读钱包 .....	43
5.5.8	MIFARE 1K/4K 钱包充值 .....	44
5.5.9	MIFARE 1K/4K 钱包扣款 .....	44
5.5.10	MIFARE 1K/4K 备份钱包 .....	45
5.5.11	ISO14443A 卡休眠.....	45
5.5.12	下载 MIFARE 1K/4K 卡片密钥到模块中 .....	46
5.5.13	有关密钥标识.....	46
5.6	MIFARE Ultralight/Ultralight C/Ultralight EV1 卡命令.....	48
5.6.1	MIFARE Ultralight/Ultralight C/Ultralight EV1 寻卡.....	48
5.6.2	MIFARE Ultralight/Ultralight C/Ultralight EV1 读卡.....	48
5.6.3	MIFARE Ultralight/Ultralight C/Ultralight EV1 写卡.....	48
5.6.4	MIFARE UltraLightC 认证密钥.....	49
5.6.5	MIFARE UltraLightC 读取 Ek(RndB) .....	49
5.6.6	MIFARE UltraLight C 认证 Ek(RndA + RndB').....	50
5.6.7	Ultralight EV1 读卡片版本号 .....	51
5.6.8	Ultralight EV1 快速读取数据 .....	51
5.6.9	Ultralight EV1 读计数器 .....	52
5.6.10	Ultralight EV1 计数器递增.....	52
5.6.11	Ultralight EV1 认证密钥 .....	53
5.6.12	Ultralight EV1 读取签名 .....	53
5.6.13	Ultralight EV1 读取卡片计数器的 Tearing 事件 .....	53
5.6.14	Ultralight EV1 读取虚拟卡片信息 .....	54
5.7	MIFARE Plus 卡命令 .....	55
5.7.1	卡片准备命令.....	55
5.7.1.1	MIFARE Plus 寻卡 .....	55
5.7.1.2	MIFARE Plus 卡复位 .....	55
5.7.1.3	MIFARE Plus 寻卡并复位 .....	55
5.7.2	卡片初始化命令.....	55
5.7.2.1	MIFARE Plus Write Perso .....	55
5.7.2.2	MIFARE Plus Commit Perso .....	56
5.7.2.3	MIFARE Plus 切换到 Level2/3.....	56
5.7.3	卡片 LEVEL 3 应用层命令 .....	57
5.7.3.1	MIFARE Plus 授权数据块 .....	57
5.7.3.2	MIFARE Plus 读数据块.....	57



5.7.3.3	MIFARE Plus 写数据块 .....	58
5.7.3.4	MIFARE Plus 创建钱包 .....	58
5.7.3.5	MIFARE Plus 读钱包余额 .....	59
5.7.3.6	MIFARE Plus 钱包充值 .....	59
5.7.3.7	MIFARE Plus 钱包扣款 .....	60
5.7.3.8	MIFARE Plus 备份钱包 .....	60
5.7.3.9	MIFARE PlusFirst Authenticate.....	61
5.7.3.10	MIFARE PlusFollowing Authenticate .....	61
5.7.4	卡片返回的状态码 .....	63
5.8	DESfire 卡命令 .....	64
5.8.1	卡片准备命令 .....	64
5.8.1.1	DESfire 寻卡 .....	64
5.8.1.2	DESfire 卡复位.....	64
5.8.1.3	DESfire 寻卡并复位.....	64
5.8.1.4	DESFire Authenticate .....	64
5.8.1.5	DESFire Authenticate first step Get ekNo(RndB) .....	65
5.8.1.6	DESFire Authenticate second step get ekNo(RndA') .....	66
5.8.1.7	DESFire Select Application .....	66
5.8.2	卡片初始化命令 .....	67
5.8.2.1	DESfireFormat Card .....	67
5.8.2.2	DESFire Create Application.....	67
5.8.2.3	DESFire Change Key Settings .....	68
5.8.2.4	DESFire Get Key Settings .....	68
5.8.2.5	DESFire Change Key .....	69
5.8.2.6	DESFire Get Key Version .....	69
5.8.2.7	DESFire Delete Application.....	70
5.8.2.8	DESFire Get Version.....	70
5.8.2.9	DESFire Get Application IDs.....	71
5.8.2.10	DESFire Get File IDs .....	71
5.8.2.11	DESFire Get FileSettings.....	72
5.8.2.12	DESFire Change File Settings .....	72
5.8.2.13	DESFire Create Std Data File .....	73
5.8.2.14	DESFire Create Backup Data File .....	73
5.8.2.15	DESFire Create Value File .....	74
5.8.2.16	DESFire Create Linear Record File .....	75
5.8.2.17	DESFire Create Cyclic Record File .....	75
5.8.2.18	DESFire Delete File .....	76
5.8.3	卡片应用层命令 .....	76
5.8.3.1	DESFire Read Data.....	76
5.8.3.2	DESFire Write Data .....	77
5.8.3.3	DESFire Get Value.....	78
5.8.3.4	DESFire Credit.....	78
5.8.3.5	DESFire Debit.....	79
5.8.3.6	DESFire Limited Credit.....	79



5.8.3.7	DESFire Write Record .....	80
5.8.3.8	DESFire Read Record.....	81
5.8.3.9	DESFire Clear Record File .....	81
5.8.3.10	DESFire Commit Transaction.....	82
5.8.3.11	DESFire Abort Transaction.....	82
5.8.4	卡片返回的状态码.....	84
5.9	SR176 卡命令 .....	85
5.9.1	SR 系列卡单通道初始化.....	85
5.9.2	SR 系列卡选定卡.....	85
5.9.3	SR 系列卡操作结束 .....	85
5.9.4	SR176 读卡.....	86
5.9.5	SR176 写卡.....	86
5.9.6	SR176 块锁定.....	86
5.10	SRI512/1K/2K/4K 卡命令 .....	88
5.10.1	SR 系列卡单通道初始化.....	88
5.10.2	SRI 系列卡 16 通道初始化.....	88
5.10.3	SR 系列卡选定卡.....	88
5.10.4	SRI 系列卡回到初始化状态.....	88
5.10.5	SR 系列卡操作结束.....	89
5.10.6	SRI 系列卡读卡.....	89
5.10.7	SRI 系列卡写卡.....	89
5.10.8	SRI 系列卡块锁定.....	89
5.10.9	SRI 系列卡读 UID.....	90
5.10.10	SRIX 系列卡认证.....	90
5.11	SAM 或 CPU 卡命令.....	91
5.11.1	SAM 或 CPU 卡复位 .....	91
5.11.2	设定 SAM 或 CPU 卡复位后通讯速率.....	91
5.11.3	发送 APDU 到 SAM 或 CPU 卡.....	92
5.11.4	弹出 CPU 卡.....	92
5.12	ISO15693 电子标签命令.....	94
5.12.1	ISO15693 inventory.....	94
5.12.2	ISO15693 inventory 所有在天线区域内的卡 .....	94
5.12.3	ISO15693 stay quiet.....	95
5.12.4	ISO15693 get system information.....	95
5.12.5	ISO15693 reset to ready .....	96
5.12.6	ISO15693 read blocks.....	96
5.12.7	ISO15693 write blocks.....	97
5.12.8	ISO15693 lock block.....	97
5.12.9	ISO15693 write AFI.....	98
5.12.10	ISO15693 lock AFI .....	98
5.12.11	ISO15693 write DSFID.....	98
5.12.12	ISO15693 lock DSFID .....	99
5.12.13	ISO15693 get blocks security .....	99
5.13	I.CODE 1 电子标签命令.....	101



5.13.1	I.CODE1 寻卡.....	101
5.13.2	I.CODE 1 读卡.....	101
5.13.3	I.CODE 1 写卡.....	102
5.13.4	I.CODE 1 卡休眠.....	102
5.14	二代身份证命令.....	103
5.14.1	身份证寻卡并获取 UID.....	103
5.14.2	身份证寻卡.....	103
5.14.3	发送 APDU 到身份证卡片.....	104
5.15	NFC 功能命令.....	105
5.15.1	设备 NFC 工作模式设置.....	105
5.15.2	发起者端命令.....	105
5.15.2.1	发起者寻找目标设备命令.....	105
5.15.2.2	发起者与目标设备数据交换命令.....	106
5.15.3	目标端命令.....	106
5.15.3.1	目标缓存交换数据命令.....	106
5.15.3.2	目标读取状态及数据交换结果.....	107
5.16	NFC Tag 功能命令.....	108
5.16.1	综述.....	108
5.16.2	发起者端命令.....	108
5.16.2.1	发起者读取模拟 NFC Tag 数据.....	108
5.16.2.2	发起者写入模拟 NFC Tag 数据.....	108
5.16.3	目标端命令.....	109
5.16.3.1	目标读取模拟 NFC Tag 数据.....	109
5.16.3.2	目标写入模拟 NFC Tag 数据.....	109
5.16.3.3	目标写入模拟 NFC Tag UID.....	110
5.17	FeliCa 卡片命令.....	111
5.17.1	收发 FeliCa 卡片命令.....	111
5.18	ISO18000-3M3 命令.....	112
5.18.1	错误代码.....	112
5.18.2	ActivateCard.....	112
5.18.3	ActivateCards.....	112
5.18.4	Read.....	113
5.18.5	Write.....	114
5.18.6	KILL.....	115
5.18.7	Lock.....	115
5.18.8	Access.....	117
5.18.9	BlockWrite.....	118
5.18.10	BlockErase.....	119
5.18.11	Inventory.....	119
5.18.12	ReqRnCard.....	120



# 文档更新记录

版本	日期	改动内容
V4.41	2014 年 1 月 23 日	在 4.1.1 中增加“起始位：1 位”。
V4.45	2014 年 3 月 19 日	修正了文字书写中的错误。
V4.48	2014 年 4 月 2 日	增加了自动寻卡并输出卡号可选 JCP04 或 JCP05 通讯协议 增加了 JMY614 中弹出卡片的指令。
V4.49	2014 年 5 月 12 日	更改文档书写中的错误。
V4.51	2014 年 5 月 15 日	增加了对 MIFARE Ultralight EV1 的支持。
V4.53	2014 年 6 月 11 日	修正文字错误。
V4.54	2014 年 11 月 5 日	修正 5.2.9 和 5.2.11 的文字错误。
V4.60	2014 年 12 月 12 日	将 SR176 和 SRI 系列卡的读写卡功能改为多块操作，程序内部进行了处理，兼容之前的单块操作。
V4.61	2015 年 1 月 17 日	对 IIC 通讯部分的上拉电阻做了补充说明，并建议新用户使用我们的 IIC 例子程序和开发测试板。 建议新用户使用 TransPort 来了解通讯协议。
V5.00	2015 年 1 月 31 日	增加读卡器安全认证功能 章节 5.3。
V5.10	2016 年 3 月 18 日	增加密文命令传输方式 章节 5.3.4。
V5.20	2016 年 3 月 24 日	增加 NFC 功能 章节 5.15。
V5.21	2016 年 4 月 21 日	增加 NFC Tag 功能 章节 5.16。
V5.22	2016 年 5 月 23 日	在配置自动寻卡并输出 UID 时增加了 JCP02 和 JCP03 的协议。
V5.30	2016 年 9 月 30 日	增加 FeliCa 卡片功能支持。
V5.31	2018 年 4 月 16 日	复位模块到出厂状态后，模块增加复位操作。 Ultralight 卡片增加读多块功能。
V5.50	2018 年 11 月 28 日	增加了 SPI 通信方式章节。 优化了 IIC 通信协议章节的描述。 更改文档书写中的错误。
V5.52	2019 年 9 月 23 日	增加 ISO15693 多卡 Inventory 操作。 增加 ISO14443A 多卡寻卡操作。
V5.53	2019 年 10 月 8 日	修改 MIFARE PLUS 部分的命令格式
V6.00	2020 年 6 月 30 日	增加对 ISO18000-3M3 卡片的支持
V6.02	2020 年 7 月 15 日	修正格式错误
V6.10	2021 年 2 月 1 日	为支持非 PBOC 标准的旧式 CPU 卡修改了章节 5.4.3 和 5.4.4
V6.20	2021 年 5 月 6 日	增加了多天线设备的设定当前天线指令。





# 1 概述

本文详细介绍了 JMY600 系列读卡模块的工作模式和通讯协议，本手册的使用对象为使用 JMY600 系列 RFID 模块的程序员，我们也有通讯协议的例子代码，可以在随货的产品光盘上找到，也可以在金木雨的网站找到。如果在编写程序中依然有任何的问题，请随时联系我们的销售人员，或发送电子邮件到：[jinmuyu@vip.sina.com](mailto:jinmuyu@vip.sina.com) 我们会给您满意的答复。

# 2 功能

JMY600 系列读卡模块是精心设计的高性能产品，模块具有很多先进功能。

## 2.1 基本工作模式

模块的基本工作模式是上位机问答式，上位机发送一条指令给模块，模块执行后返回结果。在模块返回结果前，再次发送新的指令，模块是不接受的。因此，在上位机编写程序时，应等待模块的执行结果返回后再进行下一条指令的发送。

## 2.2 自动寻卡

本读卡模块支持对 ISO14443A 和 ISO15693 的自动寻卡，开启自动寻卡后，模块进行不断的寻卡，当寻到卡片后，在 ICC 引脚上出现低电平，此时，可以直接对卡片进行读写而无需事先寻卡。

上电默认的自动寻卡状态是通过 0x1D 命令设定的，这个设定是掉电保存的。上电后，可以通过命令（0x11）将自动寻卡功能临时开启或关闭，在模块重新上电后，将恢复为设定的默认状态。

如果读卡模式设定为 ISO15693 模式，那么自动寻卡功能就是针对 ISO15693 的卡片；如果工作模式设定为 ISO14443A 模式，那么自动寻卡就是针对 ISO14443A 的。在自动寻卡状态下，任何其他类型的卡片操作都会造成读卡模式的改变，从而造成错误的自动寻卡结果。

自动寻卡功能对 ISO15693 全功能支持。

自动寻卡功能对 MIFARE 1K/4K 和 Ultra Light 全功能支持。

自动寻卡功能在工作时可以寻到 ISO14443A CPU 卡，如果需要操作 CPU 卡，需要先发送 RATS 命令（命令代码 0x30），模块在一个正确的 RATS 命令后，会关闭自动寻卡功能，请在使用中注意。

自动寻卡只有当感应区内只有一张同类型卡片时才会有正确响应，多张卡片将没有响应。如果天线区域中有多张卡片，可能会造成数据错乱，因此，模块在开启自动寻卡后，模块的多卡操作功能就将被强制禁止，从而避免卡片数据错乱。



## 2.3 自动寻卡并输出卡序列号

本系列模块可以通过设定达到自动寻卡并把读到的卡号输出的功能，但此功能只能在使用 UART 或 RS232C 时使用，支持的卡片类型是 ISO14443 TYPE A 和 ISO15693 以及二代身份证。

卡号输出方式可以是单次输出，也可以是连续输出，直到卡片移开。

输出格式可以是 HEX 方式也可以是 ASCII 方式。ASCII 输出格式只输出 UID，HEX 输出格式按照 0x20 命令、0x5C 命令和 0x71 命令的返回格式。

开启此功能后不能进行卡片的命令交互方式的读写操作，不能通过命令去控制模块做其他的事情，如果要做，需要首先停止本功能。此时可以临时停止此功能，方法是通过“[模块工作模式设定](#)”命令。完成对模块的其他操作后，可以再次开启本功能。

此功能在 IIC、USB 或 SPI 通讯模式下无法工作，当模块收到 IIC、USB 或 SPI 接口的命令后，此功能被自动屏蔽。

## 2.4 空闲模式

这是为省电而设计的功能，模块在非 USB 通讯模式时，可以设定空闲模式，此时模块射频电场关闭，功耗降低到约 3mA。再次发送任何命令到模块即可唤醒模块，此时模块会开启射频电场，恢复对卡片的操作。请参照命令：[设置模块为空闲模式](#)

## 2.5 关闭模式

有些模块配有 CE 引脚，当 CE 引脚为低电平时，模块正常工作，当 CE 引脚为高电平时，模块被禁止，模块功耗降低到大约 30 到 100uA（取决于读卡芯片，RC522/RC523 的较低）。再次将模块的 CE 引脚拉低即可唤醒模块，唤醒时间大约为 50ms。

## 2.6 EMV/PBOC 认证模式

部分 JMY600 系列读卡模块内置了 PBOC 和 EMV 的认证工作模式，此模式支持 EMV 和 PBOC 的 LEVEL 1 的协议认证和电气特性认证，发送启动认证模式命令就可以开启。

我们有充分的能力可以设计出能满足认证要求的读卡器，如果您要将读卡器送检，请联系我们的销售人员或技术支持。

## 2.7 默认功能配置

本模块提供模块的默认操作方式设定功能，可以设定的参数有：自动寻卡开或关，自动输出卡序列号开或关，连续或单次输出卡号，HEX 或 ASCII 格式输出卡号，在自动寻卡并输出卡号时 ISO15693 模式中可设定的 AFI，自动寻卡（此功能不输出卡号，为读写卡而设计）时间间隔，多卡操作开或关，可设定的 UART 通讯地址和波特率，可设定的 IIC 地址，可设定的 RF 输出功率。设定命令的详细说明请参照：[系统设置命令](#)



## 2.8 ISP 功能

本模块可以通过 USB 或 UART/RS232C 进行升级内部软件，如果客户需要定制某些功能或已经发货的产品有程序瑕疵，可以通过此功能对模块进行程序更新，此功能可以提供很大方便。使用方法是，在 PC 上运行我们提供的升级软件，按照我们提供的方法，把我们提供的程序文件下载进去即可。

## 2.9 读卡器安全认证功能

我们提供了控制器与读卡模块之间的身份认证方法。这是一个读卡模块与控制器之间的双向认证，可以应用在一些对安全敏感的应用环境，通过此认证后，可以确保使用的是 JINMU YU 的正规产品。

**有关读卡模块认证的相关方法，请联系我公司，签订保密协议后即可获得。**

## 2.10 加密通讯功能

一般情况下读卡模块的通讯是明文传送的，但在某些数据敏感的应用环境，通讯中的数据需要加密，我们提供了一个加密传输的应用方法，应用中的敏感数据以密文的方式传送。

**有关读卡模块加密通讯的方法，请联系我公司，签订保密协议后即可获得。**

## 2.11 NFC 目标功能

有些读卡模块的 PCD 使用了支持 NFC 功能的芯片，如 PN512，那么这个模块是支持 NFC 目标功能的，同时也支持发起者或目标的主动通讯模式，可以同其他的支持主动通讯模式的 NFC 设备进行通讯。如果发起设备仅仅支持被动模式，那么被动模式是唯一可选的通讯模式。

## 2.12 NFC Tag 模拟模式

支持 NFC 目标功能的读卡模块可以设定模拟成一个 NFC Tag，这是一个 128 字节的 FLASH 数据存储空间，按照 Ultralight 卡片的模式组织存储空间，4 字节为一块，共计 32 块。

当模块设定为模拟 NFC Tag 时，这个空间中的数据可以通过通讯端口直接写入初始化内容，而不必通过 RF 界面由其他读卡模块初始化。

正常使用 NFC Tag 时，可以通过通用读卡器的 Ultralight 卡读功能进行读取，写功能则必须由我公司的读写模块中的专用指令进行写。



## 3 通讯协议

### 3.1 测试工具软件

对于新接触本协议的用户，建议在 PC 端使用我们提供的工具：“TransPort”来做实验，先了解一下协议的各部分数据组织方法。对于 RS232C 或 USB HID 接口的模块可以直接连接 PC 操作；对于 UART 或 IIC 接口的模块，我们提供有 MT500 测试板，可以把 PC 输出的 RS232C 信号转换为 UART 接口或 IIC 接口，并直接完成数据的交互，详细信息请咨询销售人员。

在编写程序时，建议参考我们的例子代码，尤其是 IIC 接口，只有时序正确才能达到最高的通讯速度，请用户务必注意。

### 3.2 JCP04 通讯协议

JCP04 通讯协议是早期广泛使用的协议，JCP05 通讯协议是在 JCP04 的基础上改进的，JMY600 系列模块同时支持这两种协议，在新开发的产品中推荐使用 JCP05 协议。

#### 3.2.1 数据发送格式

长度字	命令字	数据域	校验字
-----	-----	-----	-----

- 长度字：1 字节，指明从长度字到数据域最后一字节的字节数。
- 命令字：1 字节，通讯协议的应用层命令，详见[应用层协议](#)。
- 数据域：数据长度由命令字决定，长度为 0x00 至 0xFC 字节。
- 校验字：1 字节，从长度字到数据域最后一字节的逐字节异或（XOR）值

#### 3.2.2 数据返回格式

- 成功返回：

长度字	命令字	数据域	校验字
-----	-----	-----	-----

- 失败返回：

长度字	命令字取反	校验字
-----	-------	-----

失败返回的含义是指模块与卡片通讯失败。

### 3.3 JCP05 通讯协议

#### 3.3.1 数据包发送格式

- 主机发送：

长度字	通讯地址	命令字	数据域	校验字
-----	------	-----	-----	-----



- 长度字：2 字节，指明从长度字到数据域最后一字节的字节数，高字节在前，取值范围为：0x0004~0x01FE。
- 通讯地址：1 字节，UART 或 RS232C 多机通讯的设备地址，1 为设备出厂默认地址，0 为群呼地址
- 命令字：1 字节，通讯协议的应用层命令，详见[应用层协议](#)
- 数据域：数据长度由命令字决定，长度为 0 至 506 字节，根据所使用的处理器不同，部分型号会小于 506 字节。
- 校验字：1 字节，从长度字到数据域最后一字节的逐字节异或（XOR）值

### 3.3.2 数据包返回格式

- 成功返回：

长度字	通讯地址	命令字	数据域	校验字
-----	------	-----	-----	-----

- 失败返回：

长度字	通讯地址	命令字取反	校验字
-----	------	-------	-----

失败返回的含义是指模块与卡片通讯失败。

### 3.4 数据包返回时间

从机收到指令后开始执行指令，执行指令需要的时间一般在 100ms 以内，根据指令类型的不同，从机执行时间可以很长（可以有 60 秒，如某些 CPU 卡生成密钥对的指令），需要等待的时间由所操作的卡片以及指令类型决定，因此主机端需要根据指令不同来设定等待时间。

详细的指令时间可以使用 TransPort 软件进行测试，获得的结果可以作为编写实际应用程序的依据。



## 4 通讯接口

### 4.1 UART 和 RS232C 接口

#### 4.1.1 物理接口

UART 接口即异步串行通讯口，接口电平符合 TTL 电平规范。UART 中使用 TX 和 RX 两根引线以及 GND 进行通讯，使用时需要把主机端的 TX 连接到设备端的 RX，主机端的 RX 连接到设备端的 TX。

RS232C 即经过电平转换的 UART 接口，接口电平符合 RS232C 规范。RS232C 中使用 TXD 和 RXD 两根引线以及 GND 进行通讯，使用时需要把主机端的 TXD 连接到设备端的 RXD，主机端的 RXD 连接到设备端的 TXD。

UART 和 RS232C 通信协议采用字节为单位，接收和发送数据都是十六进制格式，通信参数如下：

波特率：19200 bps（默认），可更改为 115200bps、9600bps、38400bps 和 57600bps

起始位：1 位

数据位：8 位

校验位：无

停止位：1 位

流控制：无

#### 4.1.2 通讯过程

UART 和 RS232C 的通讯为异步通讯，主机端发送命令给从机端，从机收到指令后执行，然后发送执行结果给主机端。

在收到前一条指令的返回信息前，下发新的指令是无意义的。

### 4.2 IIC 接口

#### 4.2.1 物理接口

IIC 接口是两线同步串行通讯，接口电平符合 TTL 电平规范。IIC 接口的 2 根线为 SDA 和 SCL，SDA 为串行数据线，SCL 为串行时钟线，连接方法为主机端的 SDA 与设备端的 SDA 相连，主机端 SCL 与设备端 SCL 相连。IIC 接口在设备端配置为漏极开路输出，用户可以根据需要连接上拉电阻到电源，上拉电阻根据实际情况，可选 2.7K~10K，电源电压是 3V~5V。如果不连接上拉电阻，那么会影响通讯速率，甚至导致通讯不成功。

IIC 总线可以连接多达 128 个设备，模块的 IIC 默认地址是 0xA0，用户可以通过命令（代码：0x19）来修改这个设置，以达到在同一 IIC 总线上连接多个读卡模块的应用。

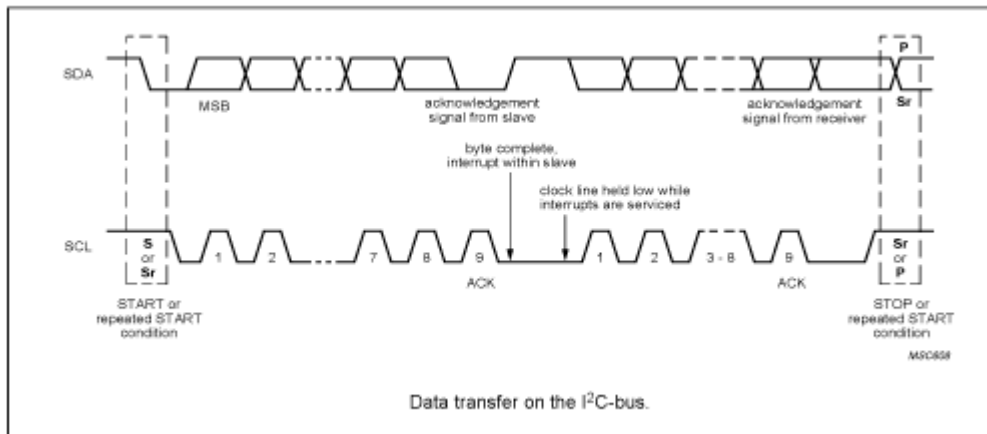


## 4.2.2 数据链路

以下为 IIC 通讯的图示，在如下的图示中，指示的是 JCP04 协议的数据包格式。

### 4.2.2.1 时钟和数据交换

通常情况下，SDA 引脚上的数据只在 SCL 低时才被更改，在 SCL 高时的数据更改，则在随后的定义条件会指示重新开始或停止。

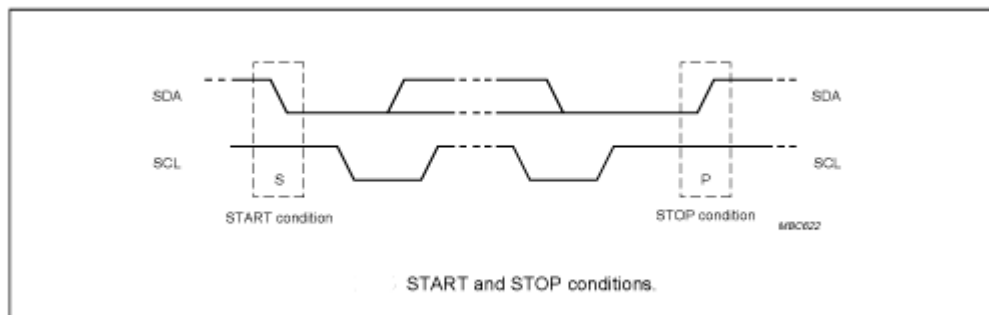


### 4.2.2.2 开始条件 (Start Condition)

在 SCL 高时的高到低的 SDA 过渡是一个开始状态，这必须先于其它任一命令。

### 4.2.2.3 结束条件 (Stop Condition)

一个在 SCL 高时的高到低的 SDA 过渡是一个结束状态。

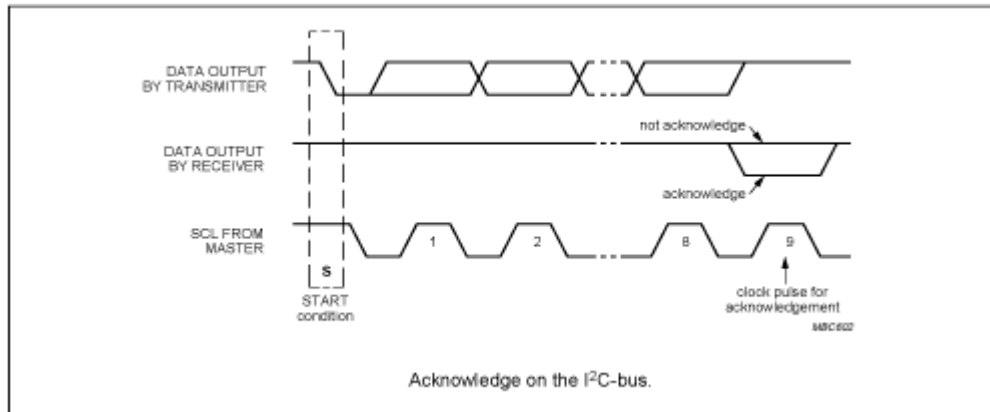


### 4.2.2.4 确认符 (ACK)

所有地址和数据字以 8 位字的形式在模块间连续（串口）传送。模块发送零，以确定它不忙，并确定它已收到每一个字。这发生在第九个时钟周期期间。

### 4.2.2.5 总线状态

模块收到命令后，那么不再接受 IIC 总线上的任何信息，直到当前命令执行结束后，才能再次响应 IIC 总线上的信息。



#### 4.2.2.6 设备地址

启动条件后，该模块需要一个 8 位的设备地址字，使该芯片能够进行读/写操作。

该设备地址字由七个地址位和 1 个操作选择位组成。

该模块地址的前 7 位是 1010000（0xA0 十六进制）

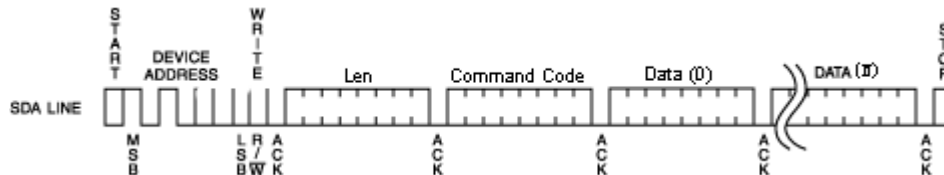
该设备地址的第八位是读/写操作选择位。如果该位是高位，则一次读操作被启动，如果该位是低位，则一次写操作被启动。



The first byte after the START procedure.

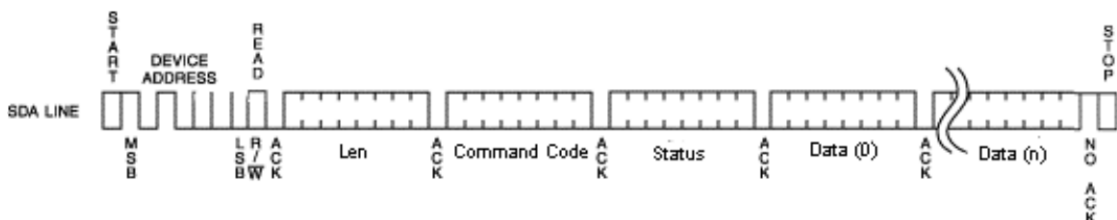
#### 4.2.2.7 写数据操作

上位机设备通过写操作发送命令给模块



#### 4.2.2.8 读数据操作

上位机设备使用读操作得到结果



### 4.2.3 通讯过程

该模块是 IIC 总线的的一个从设备，那么，上位机需要使用写命令将“命令数据包”写





入模块。写操作之后，模块开始执行刚刚写入的命令，上位机在此时向模块不断发出“读”命令。如果模块未作出回应，那么刚刚发出的指令尚未执行完毕。如果模块对一个读操作有回应，则刚才命令执行结束，此时上位机能够从模块上读取结果和/或数据。

## 4.2.4 通讯过程描述

例如：我们需要读取 MIFARE 卡的第一块的数据，以 JCP04 应用层协议为例，我们需要：

发送命令：0A210001FFFFFFFFFFFF2A

其中包含步骤：

- A. 写命令到模块中
  1. 发送起始条件
  2. 发送控制字节，在此为 0xA0，含义为：地址 0xA0+写控制 0x00
  3. 发送模块命令序列：0x0A210001FFFFFFFFFFFF
  4. 发送模块命令序列校验字节：0x2A
  5. 发送停止条件
- B. 发送读命令，如果模块无 ACK，表明模块正在工作，此时重复发送读命令
  1. 发送起始条件
  2. 发送控制字节，此处为 0xA1，含义为：地址 0xA0+读控制 0x01
  3. 如果模块无 ACK，返回到 B，重复，如果有 ACK，则到下一步 C
- C. 接收模块返回的数据
  1. 接收 1 字节并发送 ACK，如接收到的内容为 0x12，含义为本数据包有 0x12 字节有效数据
  2. 接收剩余的 17 字节（0x12-1=0x11），每接收到一字节都需要发送 ACK
  3. 接收校验字节，然后发送 NACK
  4. 发送停止条件
- D. 验证校验字节，如果正确，通讯过程成功

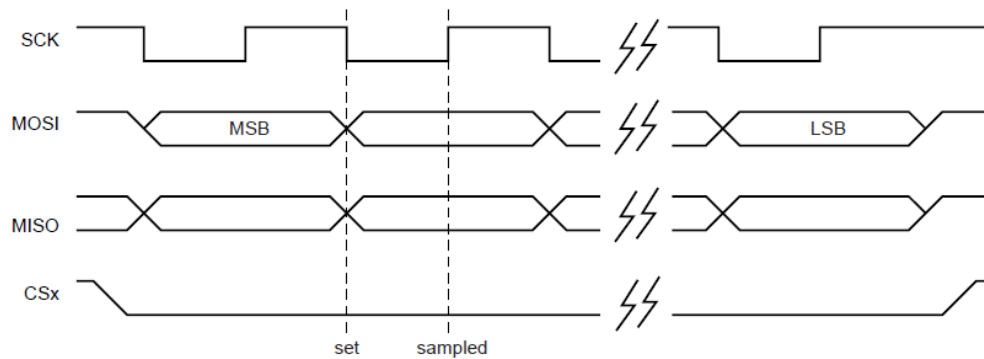
## 4.3 SPI 接口

### 4.3.1 物理接口

SPI 是串行外设接口（Serial Peripheral Interface）的缩写。SPI，是一种高速的，全双工，同步的通信总线，它以主从方式工作，这种模式通常有一个主设备和一个或多个从设备，需要至少 4 根线。MISO（SPI Bus Master Input/Slave Output）主设备数据输入，从设备数据输出。MOSI（SPI Bus Master Output/Slave Input）主设备数据输出，从设备数据输入。CLK 时钟信号，由主设备产生。CS（Chip select）从设备片选信号，由主设备控制。

#### 4.3.1.1 总线工作方式

标准 SPI 接口模式 3，即 SCK 空闲时为高电平，数据有效在上升沿



时钟极性: CPOL=1, SPI 总线空闲为高电平

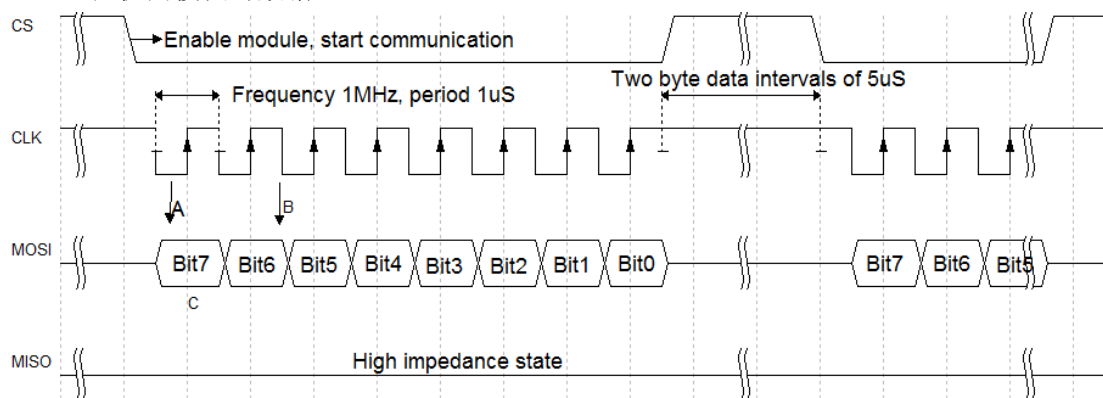
时钟相位: CPHA =1, 在 SCK 第二个跳变沿 (上升) 采样

CPHA=1、CPOL=1, MISO 引脚和 MOSI 引脚上的数据的 MSB 位必须与 SPSCK 的第一个边沿同步, 在 SPI 传输过程中, 在同步时钟信号周期开始时 (下降沿) 数据上线, 然后在同步时钟信号的上升沿时, SPI 的接收方捕捉位信号, 在时钟信号的一个周期结束时 (下降沿), 下一位数据信号上线, 再重复上述过程, 直到一个字节的 8 位信号传输结束。

**示例:** 使用 1MHz 的 SPI 时钟与模块通信, 一个时钟周期为 1uS, 在 CS 信号使能后, 即 CS 引脚由高电平跳变为低电平, 主机发起一次数据读写开始, 在时钟的每个 SCK 的下降沿, 送入 MOSI 引脚一位高电平 1 或低电平 0 信号, 高位在前, 随后在每个时钟周期 1/2uS 位置的时钟上升沿时, 将数据位发送出去或读取 MISO 引脚上的一位电平位信号, 在 8 个时钟上升沿后, 完成一个字节的读取和发送, 最后将 CLK 和 CS 信号置为 “1”, 由于接收到指令或数据后要有 MCU 的处理, 因此, 两个字节操作之间要有 5uS 的时间间隔。

#### 4.3.1.2 写数据操作

上位机向模块写数据

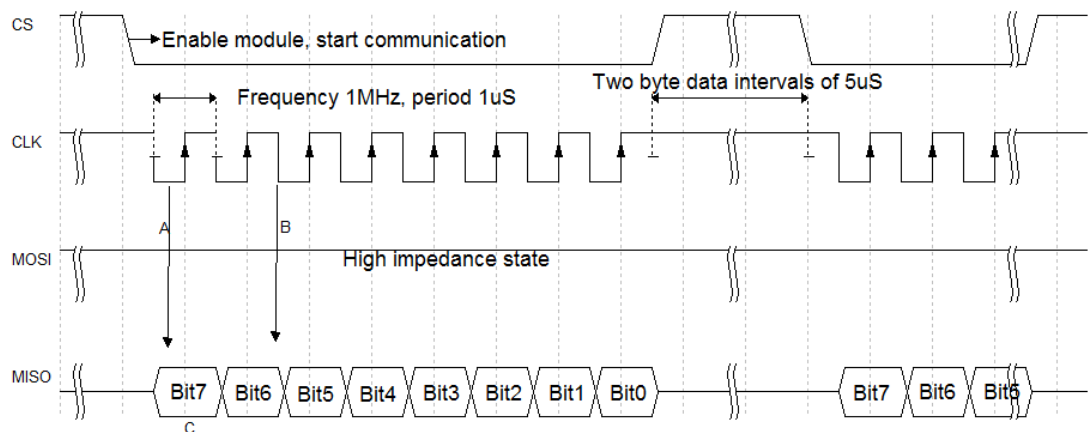


注:

- 时钟下降沿时, MOSI 数据位上线
- 时钟上升沿时, MOSI 发送出去一位数据 bit
- 字节的高位在前

#### 4.3.1.3 读数据操作

上位机读取模块应答的数据



注:

- A. 时钟下降沿时, 准备读取 MISO 数据线上的一位数据
- B. 时钟上升沿时, 读取 MISO 数据线上的一位数据 bit
- C. 字节的高位在前

## 4.3.2 数据链路

### 4.3.2.1 SPI 指令

- SPI 查询状态: 0x04 查询 SPI 设备状态, 根据返回代码选择后续命令。
- SPI 写数据: 0x10 向 Slave 设备发送数据。
- SPI 读数据: 0x20 从 Slave 设备读回数据。

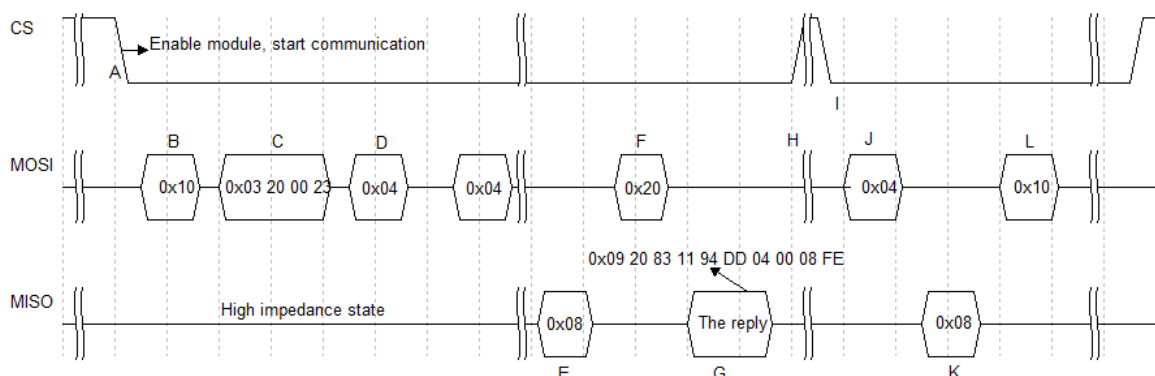
### 4.3.2.2 SPI 状态

SPI 查询命令返回的信息如下:

- 0x08 Slave 设备前次命令执行完毕, Master 设备可以读取执行结果, 或发送新的命令。
- 0x42 Slave 设备空闲, 可以接受 Master 设备下发的命令数据。

## 4.3.3 通讯过程描述

SPI 接口的 Master 设备如需操作 Slave 设备, 需要首先向 Slave 设备发送“查询”命令, 根据查询结果决定下一步的操作。



通讯过程:



- A. 发送数据前先将 CS 引脚拉低，使能模块，进入通信状态。
- B. 发送查询命令 0x04。
- C. 读取 SPI 数据线上的返回结果。
- D. 如果返回结果为 0x42 或 0x08，那么可以继续。如果不是，返回到“B”。
- E. 发送“写命令”，一个字节 0x10。
- F. “写命令”完成发送后，再发送符合 JCP04（或 JCP05）协议的数据包指令，例如寻卡指令：03 20 00 23
- G. 发送完成后，在用户设定的超时时间（根据不同操作，此超时时间可达 300 秒，但一般应用中，此时间不超过 1 秒）内，每间隔 1uS 发送一次发送一次查询命令，询问模块是否空闲，若返回结果为 0x08，那么继续。如果不是，返回“G”
- H. 若超过超时时间无应答，则退出等待，本次指令发送失败，将 CS 引脚置高，返回。
- I. 模块响应“成功应答”，一个字节 0x08，即模块已完成收到的数据处理，处理结果待发送
- J. 向模块发送“读指令”，一个字节 0x20
- K. 发送完“读指令”后，从 Slave 设备读取数据，这些数据是符合 JCP04（或 JCP05）协议的数据包，接收长度是 JCP04 协议包的第 1 个字节，或 JCP05 协议包的前 2 个字节。
- L. 数据包接收完成后，即完成一次数据包的发送和应答过程，将数据线状态恢复至默认电平状态即可。
- M. 将 CS 引脚电平置高，程序执行完成，返回。

## 4.4 USB 接口

USB 接口符合 USB2.0 HID 规范，连接方法为主机端与设备端同名相连。

在 Windows 操作系统中，不需要额外安装驱动程序，也就是俗称的：免驱。

在 Windows 系统中，读卡器使用 SetReport 和 GetReport 发送命令和获取操作结果。

在使用 USB 通讯接口时，通讯中发送的每个数据包长度被限制在 64 字节，这是 USB HID 在 Windows API 的限制，在其他操作系统下可以自由设定发送长度。接收时数据包的长度没有限制。



## 5 应用层协议详解

### 5.1 说明

本章的通讯协议部分，是 JCP04 和 JCP05 的全部内容，有些产品并不支持全部内容，比如有些模块没有配置 SAM 卡座，有些模块不支持 ISO14443B 或 ISO15693 的某些卡片，用户需要根据订货型号和产品说明书中的“卡片支持类型表”来确定可以支持哪些卡片的命令。对于不支持的命令部分，读卡模块会返回错误。

本章将详细解释通讯协议应用层的命令和数据部分构成，应用层协议只说明了命令和数据部分，整个数据包的构成格式可以是 JPC04 和 JCP05，但命令代码和数据部分完全相同。

我们以下列格式说明每条命令的构成：

帧头	命令	数据	校验
----	----	----	----

在这里，帧头是指 JCP04 或 JCP05 的帧头，解释如下：

JCP04 的帧头：1 字节长度信息，不包含校验字节在内的其他所有字节数

JCP05 的帧头：共 3 字节，2 字节长度信息（高字节在前）和 1 字节串行通讯地址，长度信息为不包含校验字节在内的所有其他字节数，串行通讯地址用于在 485 通讯时选择指定的设备进行通讯。

校验：从帧头开始到数据部最后一个字节的异或（XOR）运算结果。

举例说明，我们以如下的命令来分别说明。

帧头	0x11	模式	校验
----	------	----	----

这是控制模块工作状态的命令，我们现在要通过这个命令关闭天线，因此，“模式”赋值为 0x00，那么，下发的命令就是：

JCP04： 0x03 11 00 12；其中，0x03 为帧头，除去校验字节，长度为 3，因此取值 0x03；0x11 为命令；0x00 为参数，意为关闭自动寻卡，关闭天线；0x12 为校验字节，前面的 3 个字节的异或结果为 0x12；

JCP05： 0x00 05 00 11 00 14；其中，0x00 05 00 为帧头，0x00 05 为长度，0x00 为串行通讯地址；0x11 为命令；0x00 为参数，意为关闭自动寻卡，关闭天线；0x14 为校验字节，前面的 5 个字节的异或结果为 0x14；

在每个命令的说明最后都有示例，示例命令是 JPC05 规范的，是我们以 JMY6801H 作为试验机得到的结果。有关 JCP04 协议规范的部分，不建议在新产品中使用。



## 5.2 系统设置命令

### 5.2.1 复位模块配置至出厂状态

功能：模块内部具有可保存的配置信息，当配置混乱后，可以使用此命令复位模块的各种配置状态至出厂默认状态。恢复默认配置后，读卡模块的看门狗电路会将模块复位。

上位机发送：

帧头	0x0F	52 45 53 45 54	校验字
----	------	----------------	-----

模块回应成功：

帧头	0x0F	校验字
----	------	-----

模块回应失败：

帧头	0xF0	校验字
----	------	-----

我们以 JMY6801H 为目标机，分别使用 JCP04 和 JCP05 协议，给出如下命令例子：

JCP04 发送：0x07 0F 52 45 53 45 54 5D

JCP04 返回：0x02 0F 0D

在后续的指令例子中，将不再给出 JCP04 的例子，因为 JCP04 通讯协议是为了兼容旧有产品，在新产品设计中推荐使用 JCP05 通讯协议。

JCP05 发送：0x00 09 00 0F 52 45 53 45 5453

JCP05 返回：0x00 04 01 0F 0A

### 5.2.2 读产品信息

功能：读取当前产品的产品信息，包含：产品名称，软件版本号，软件日期，以及配置信息。

上位机发送：

帧头	0x10	校验字
----	------	-----

模块回应成功：

帧头	0x10	产品信息	校验字
----	------	------	-----

产品信息：共 30 字节，8 字节产品名称，4 字节固件版本号，8 字节固件日期，1 字节 UART 波特率代码，1 字节 UART 多机通讯地址，1 字节 IIC 地址，1 字节多卡使能状态，1 字节 ISO15693 自动寻卡的 AFI，1 字节 ISO15693 自动寻卡的 AFI 使能状态，1 字节自动寻卡时间间隔，1 字节上电默认自动寻卡状态，1 字节上电默认自动输出 SNR 设定，1 字



节射频输出强度设定。

模块回应失败：

帧头	0xEF	校验字
----	------	-----

示例：

发送：0x00 04 00 10 14

返回：0x00 22 01 10 4A 4D 59 36 38 30 31 48 34 2E 33 30 32 30 31 33 31 31 32 33 00 01

A0 00 00 00 0A 00 00 00 99

### 5.2.3 读 PCD 的产品信息（包含 UID）

功能：读取当前 PCD 的产品信息，包含：产品类别识别码，UID，以及配置信息。

上位机发送：

帧头	0x03	校验字
----	------	-----

模块回应成功：

帧头	0x03	产品信息	校验字
----	------	------	-----

产品信息：共 16 字节，5 字节产品识别码，3 字节保留，4 字节 UID，3 字节保留，1

字节 CRC 校验。

模块回应失败：

帧头	0xFC	校验字
----	------	-----

示例：

发送：0x00 04 00 03 07

返回：0x00 14 01 03 30 FF FF 0F 04 00 00 00 25 54 75 5B A6 57 5E 38 E5

### 5.2.4 模块工作模式设定

功能：模块功能临时设定，模块不保存配置，所有设置将在下一次上电时丢失。设定天线电场开或关，设定自动寻卡开或关，自动寻卡并输出 UID 开或关。在自动寻卡并输出 UID(这个命令仅仅在 RS232C 或者 UART 接口时可用)的状态下，寻到卡片后就通过 RS232 或 UART 输出 UID，然后就会将刚刚寻到的卡片休眠。如果设定了连续输出 UID，那么当寻到卡片后就会连续输出卡号信息，直到卡片移开。UID 的输出格式还可以设定为 ASCII 格式。

上位机发送：



帧头	0x11	模式	校验字
----	------	----	-----

模式：1 字节

天线状态：                    BIT0=0：关；    BIT0=1：开 默认开  
 自动寻卡：                    BIT1=0：关；    BIT1=1：开 默认关  
 自动寻卡并输出 UID：          BIT2=0：关；    BIT2=1：开 默认关  
 自动寻卡并连续输出 UID：      BIT3=0：关；    BIT3=1：开 默认关  
 自动寻卡并输出 UID 数据格式：BIT4=0：HEX；  BIT4=1：ASCII 默认 HEX

模块回应成功：

帧头	0x11	校验字
----	------	-----

模块回应失败：

帧头	0xEE	校验字
----	------	-----

示例：

发送：0x00 05 00 11 03 17

返回：0x00 04 01 11 14

## 5.2.5 设置模块为空闲模式

功能：将模块设定为空闲模式。空闲模式下，模块的天线电场关闭，射频基站关闭，CPU 进入空闲模式，模块功耗将降低到 100uA 左右。向模块发送下一条命令即可唤醒模块到工作状态，唤醒后，天线状态和自动寻卡功能将恢复默认设置。模块进入空闲模式前，需要完成向上位机发送执行结果。在 IIC 和 SPI 通讯模式中，上位机需要将执行结果读取完毕，然后模块才会进入空闲模式。模块工作在 USB 接口时不支持空闲模式。

上位机发送：

帧头	0x12	随机数	校验字
----	------	-----	-----

随机数：1 字节随机数，如：0x55

模块回应成功：

帧头	0x12	校验字
----	------	-----

模块回应失败：

帧头	0xED	校验字
----	------	-----

示例：

发送：0x0005 00 12 55 42

返回：0x00 0401 12 17





## 5.2.6 设置 LED

功能：点亮或熄灭模块上的 LED 灯。

上位机发送：

帧头	0x13	状态	校验字
----	------	----	-----

状态：1 字节，位编码如下

LED1      BIT0=0: 关;      BIT0=1: 开

LED2      BIT1=0: 关;      BIT1=1: 开

LED3      BIT2=0: 关;      BIT2=1: 开

LED4      BIT3=0: 关;      BIT3=1: 开

模块回应成功：

帧头	0x13	校验字
----	------	-----

模块回应失败：

帧头	0xEC	校验字
----	------	-----

示例：

发送：0x0005 00 13 0F 19

返回：0x00 04 01 13 16

## 5.2.7 设置蜂鸣器

功能：鸣响模块上的蜂鸣器。

上位机发送：

帧头	0x14	时间	校验字
----	------	----	-----

时间：1 字节，单位是 10 毫秒，如果时间是 0x0A 则蜂鸣时间是 100 毫秒

模块回应成功：

帧头	0x14	校验字
----	------	-----

模块回应失败：

帧头	0xEB	校验字
----	------	-----

示例：

发送：0x0005 00 14 0A 1B

返回：0x00 04 01 14 11



## 5.2.8 读片内数据 FLASH

功能：读取模块 MCU 内部的数据 FLASH 中的数据。

上位机发送：

帧头	0x15	地址	字节数	校验字
----	------	----	-----	-----

地址：2 字节，读取起始地址，地址范围：0x0000 ~ 0x01FF，高字节在前

字节数：1 字节，读取字节数

模块回应成功：

帧头	0x15	数据	校验字
----	------	----	-----

数据：读到的数据

模块回应失败：

帧头	0xEA	校验字
----	------	-----

示例：

发送：0x0007 00 15 00 00 10 02

返回：0x00 14 01 15 FF FF FF FF FF FF FF FF FF FF FF FF FF FF 00

## 5.2.9 写片内数据 FLASH

功能：将数据写入到模块 MCU 内部的数据 FLASH 中。

上位机发送：

帧头	0x16	地址	字节数	数据	校验字
----	------	----	-----	----	-----

地址：2 字节，写入起始地址，地址范围：0x0000 ~ 0x01FF，高字节在前

字节数：1 字节，写入字节数

数据：要写入的数据

模块回应成功：

帧头	0x16	校验字
----	------	-----

模块回应失败：

帧头	0xE9	校验字
----	------	-----

示例：

发送：0x0017 00 16 00 00 10 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 11

返回：0x00 04 01 16 13



## 5.2.10 读片外数据 FLASH

功能：读取模块上的 FLASH 芯片中的数据。

上位机发送：

帧头	0x05	块号	块内偏移	字节数	校验字
----	------	----	------	-----	-----

块号：2 字节，读取起始地址，地址范围：0x0000 ~ 0x01FF，高字节在前

块内偏移：2 字节，数据块内部地址偏移，取值范围：0x0000~0x0107，高字节在前

字节数：2 字节，写入字节数，取值范围：0x0001~0x0108 字节，高字节在前

模块回应成功：

帧头	0x05	数据	校验字
----	------	----	-----

数据：读到的数据

模块回应失败：

帧头	0xFA	校验字
----	------	-----

示例：

发送：0x00 0A 00 05 00 02 00 10 00 08 15

返回：0x00 0C 01 05 00 01 02 03 04 05 06 07 08

## 5.2.11 写片外数据 FLASH

功能：将数据写入到模块上的 FLASH 芯片中。FLASH 的数据共 2048（0x0800）块，每块 264（0x0108）字节。

上位机发送：

帧头	0x06	块号	块内偏移	字节数	数据	校验字
----	------	----	------	-----	----	-----

块号：2 字节，写入块号，地址范围：0x0000 ~ 0x07FF，高字节在前

块内偏移：2 字节，数据块内部地址偏移，取值范围：0x0000~0x0107，高字节在前

字节数：2 字节，写入字节数，取值范围：0x0001~0x0108 字节

数据：要写入的数据

模块回应成功：

帧头	0x06	校验字
----	------	-----

模块回应失败：

帧头	0xF9	校验字
----	------	-----

示例：



发送: 0x00 12 00 06 00 02 00 10 00 08 00 01 02 03 04 05 06 07 0E

返回: 0x00 04 01 06 03

## 5.2.12 设定 UART 通讯波特率

功能: 设定模块的 UART 通讯波特率, 模块接收到命令后, 首先保存新波特率的设定值, 然后按照原来的波特率发送执行结果, 然后使设定值生效。模块的 UART 通讯波特率默认为 19200bps, 设定值将被保存在模块中, 掉电不受影响。

上位机发送:

帧头	0x17	波特率	校验字
----	------	-----	-----

波特率: 1 字节, 波特率代码; 0: 19200bps (默认); 1: 115200bps; 2: 9600bps; 3: 38400bps; 4: 57600bps;

模块回应成功:

帧头	0x17	校验字
----	------	-----

模块回应失败:

帧头	0xE8	校验字
----	------	-----

示例:

发送: 0x0005 00 17 00 12

返回: 0x00 04 01 17 12

## 5.2.13 设定 UART 多机通讯地址

功能: 设定模块的 UART 多机通讯地址, 模块接收到命令后, 首先保存新地址的设定值, 然后按照原来的地址发送执行结果, 最后使设定值生效。模块的 UART 多机通讯地址默认为 1, 设定值将被保存在模块中, 掉电不受影响。

上位机发送:

帧头	0x18	地址	校验字
----	------	----	-----

地址: 1 字节, UART 多机通讯地址; 1~0xFF

模块回应成功:

帧头	0x18	校验字
----	------	-----

模块回应失败:

帧头	0xE7	校验字
----	------	-----



示例:

发送: 0x0005 00 18 02 1F

返回: 0x00 04 02 18 1E

## 5.2.14 设定 IIC 通讯地址

功能: 设定模块的 IIC 通讯地址, 模块接收到命令后, 首先保存新地址, 然后按照原来的地址发送执行结果, 最后使设定值生效。模块的 IIC 地址为一字节 HEX 数据, 最低位为零, 即模块的地址必须为偶数, 不符合规定的地址不被接受, 设定值将被保存在模块中, 掉电不受影响。模块的地址默认为 0xA0。

上位机发送:

帧头	0x19	地址	校验字
----	------	----	-----

地址: 1 字节, 最低位为 0, 即地址必须为偶数

模块回应成功:

帧头	0x19	校验字
----	------	-----

模块回应失败:

帧头	0xE6	校验字
----	------	-----

示例:

发送: 0x00 05 00 19 A0 BC

返回: 0x0004 01 19 1C

## 5.2.15 设定多卡操作

功能: 设定多卡操作。如果用户从多张卡片中挑出一张来操作, 则需要使能多卡操作功能。如果用户设定了自动寻卡, 那么多卡操作将被强制禁止, 在天线区域内如果有多张卡片, 那么读卡模块就会报错, 从而避免卡片数据错乱。设定将被保存到模块中, 掉电不影响。多卡操作默认是使能的。多卡操作设定在 ISO14443A 和 ISO15693 适用。

上位机发送:

帧头	0x1A	多卡使能	校验字
----	------	------	-----

多卡使能: 1 字节, 0: 禁止多卡; 1: 使能多卡 (默认); 其他值: 保留

模块回应成功:

帧头	0x1A	校验字
----	------	-----



模块回应失败:

帧头	0xE5	校验字
----	------	-----

示例:

发送: 0x0005 00 1A 00 1F

返回: 0x0004 01 1A 1F

## 5.2.16 设定 ISO15693 自动寻卡的 AFI 和 AFI 使能

功能: 设定在 ISO15693 模式的自动寻卡中的 AFI 和 AFI 使能。如果用户设定了 AFI 并且设定了 AFI 使能, 那么在自动寻卡时只能寻到与设定的 AFI 相等的卡片。设定将被保存到模块中, 掉电不影响。默认 AFI 为 0, 默认 AFI 被禁止。

上位机发送:

帧头	0x1B	AFI	AFI 使能	校验字
----	------	-----	--------	-----

AFI: 1 字节, AFI, 取值 0 ~ 0xFF

AFI 使能: 1 字节, 0: 禁止; 1: 使能; 其他值: 保留

模块回应成功:

帧头	0x1B	校验字
----	------	-----

模块回应失败:

帧头	0xE4	校验字
----	------	-----

示例:

发送: 0x0006 00 1B 0A 01 16

返回: 0x0004 01 1B 1E

## 5.2.17 设定自动寻卡时间间隔

功能: 设定自动寻卡时两次寻卡间的时间间隔, 默认 100ms。设定后模块会保存此配置, 不受断电影响。

上位机发送:

帧头	0x1C	间隔时间	校验字
----	------	------	-----

间隔时间: 1 字节, 0x00 ~ 0xFF, 单位为 10ms, 即: 0x01 代表间隔 10ms

模块回应成功:

帧头	0x1C	校验字
----	------	-----



模块回应失败:

帧头	0xE3	校验字
----	------	-----

示例:

发送: 0x0005 00 1C 01 18

返回: 0x0004 01 1C 19

## 5.2.18 设定开机时的默认自动寻卡状态

功能: 设定开机时的默认自动寻卡状态, 设定后模块会保存此配置, 不受断电影响。

临时开启或关闭自动寻卡, 请使用 0x11 命令。

上位机发送:

帧头	0x1D	状态	校验字
----	------	----	-----

状态: 1 字节, 0x00: 关闭; 0x01: 开启; 其他值: 保留

模块回应成功:

帧头	0x1D	校验字
----	------	-----

模块回应失败:

帧头	0xE2	校验字
----	------	-----

示例:

发送: 0x0005 00 1D 01 19

返回: 0x0004 01 1D 18

## 5.2.19 设定自动寻卡并输出卡片序列号开机默认状态

功能: 设定自动寻卡并输出卡片序列号功能的开机默认状态。设定后模块会保存此配置, 不受断电影响。

上位机发送:

帧头	0x1E	状态	校验字
----	------	----	-----

状态: 1 字节, 位编码如下

自动寻卡并输出 UID:      BIT0=0: 关;      BIT0=1: 开

连续输出 UID:              BIT1=0: 关;      BIT1=1: 开

以 ASCII 编码输出:        BIT2=0: 关;      BIT2=1: 开

输出通讯协议选择:



	BIT4:BIT3 = 0:0	JCP04
	BIT4:BIT3 = 0:1	JCP05
	BIT4:BIT3 = 1:0	JCP02
	BIT4:BIT3 = 1:1	JCP03
寻卡 ISO15693:	BIT5=0: 开;	BIT5=1: 关
寻卡 ISO14443 TYPE A:	BIT6=0: 开;	BIT6=1: 关
读取身份证 UID:	BIT7=0: 开;	BIT7=1: 关

模块回应成功:

帧头	0x1E	校验字
----	------	-----

模块回应失败:

帧头	0xE1	校验字
----	------	-----

示例:

发送: 0x0005 00 1E 00 1B

返回: 0x00 04 01 1E 1B

## 5.2.20 设定射频输出功率

功能: 设定模块的射频输出功率, 最大输出功率 250mW (24dBm), 设定功率减小后, 对于不同的卡片, 有不同的读卡距离反应, 用户可以根据自己的实际情况设定。设定后模块会保存此配置, 不受断电影响。

上位机发送:

帧头	0x02	功率	校验字
----	------	----	-----

功率: 1 字节, 0x00: 最强; 0x01: 次强; 0x02: 次弱; 0x03: 最弱; 其他值: 保留

模块回应成功:

帧头	0x02	校验字
----	------	-----

模块回应失败:

帧头	0xFD	校验字
----	------	-----

JCP05 发送: 0x00 05 00 02 00 07

JCP05 返回: 0x00 04 01 02 07





## 5.2.21 设定读卡模式

功能：设定读卡模块非接触界面的读卡模式。读卡模式默认为 ISO14443A，如果模块不支持 ISO14443A，那么读卡模式默认为 ISO15693，设定不保存，下一次上电会恢复到默认状态，部分新的读卡模块会根据用户指令自动设定读卡类型。

上位机发送：

帧头	0x70	模式	校验字
----	------	----	-----

模式：1 字节，0：ISO14443A；1：ISO14443B；2：ISO15693；3：I.CODE 1；4：ISO18000-3M3；其他值：保留

模块回应成功：

帧头	0x70	校验字
----	------	-----

模块回应失败：

帧头	0x8F	校验字
----	------	-----

示例：

发送：0x0005 00 70 00 75

返回：0x0004 01 70 75

## 5.2.22 设定当前读卡天线

功能：设定多天线读卡模块的当前读卡天线，通讯口下发的所有指令都是针对当前天线的。

上位机发送：

帧头	0x04	编号	校验字
----	------	----	-----

编号：1 字节，取值范围 0~255，具体数值由读卡模块的具体配置决定。

模块回应成功：

帧头	0x04	校验字
----	------	-----

模块回应失败：

帧头	0xFB	校验字
----	------	-----

示例：

发送：0x0005 00 04 01 00

返回：0x0004 01 04 01



## 5.3 读卡模块安全认证

### 5.3.1 取认证码

功能：读卡模块认证的第一步，从模块取回加密的 8 字节认证码。

上位机发送：

帧头	0x07	校验字
----	------	-----

模块回应成功：

帧头	0x07	认证码	校验字
----	------	-----	-----

认证码：8 字节使用读卡器内的密钥加密的认证码

示例：

发送：0x00 04 00 07 03

返回：0x00 0C 01 07 D0 9B 09 33 2C B7 9F FB 84

### 5.3.2 模块安全认证

功能：控制器对读卡模块的身份认证。通过此认证，用户可以确保使用的是正规产品。

有关读卡模块认证的相关方法，请联系我公司，签订保密协议后即可获得。

上位机发送：

帧头	0x08	认证数据	校验字
----	------	------	-----

认证数据：16 字节，使用认证算法得出的 16 字节认证数据

模块回应成功：

帧头	0x08	认证结果	校验字
----	------	------	-----

认证结果：8 字节，读卡模块返回的认证结果。

模块回应失败：

帧头	0xF7	校验字
----	------	-----

示例：

发送：0x00 14 00 08 83 68 42 DB 08 9B FD CD 92 25 31 1A EE 04 31 FC 76

返回：0x00 0C 01 08 1F 7A 61 4D 9D 65 35 1D 9C



### 5.3.3 修改模块认证密钥

功能：修改模块内部存储的认证密钥，此操作需要在认证通过之后进行。

上位机发送：

帧头	0x09	密钥信息	校验字
----	------	------	-----

密钥信息：32 字节，加密的密钥信息

模块回应成功：

帧头	0x09	校验字
----	------	-----

模块回应失败：

帧头	0xF6	校验字
----	------	-----

示例：

发送：0x00 24 00 09 1F 7A 61 4D 9D 65 35 1D B4 D0 A7 45 41 14 72 61 82 2B BF ED C0

AD 98 86 B0 0B 5F 5A 80 0C 18 50 86

返回：0x00 04 01 09 0C

### 5.3.4 加密通讯通道

功能：通过加密方式传输命令。

上位机发送：

帧头	0xFC	密文信息	校验字
----	------	------	-----

密文信息：加密后的命令信息

模块回应成功：

帧头	0xFC	密文信息	校验字
----	------	------	-----

模块回应失败：

帧头	0x03	校验字
----	------	-----

示例：

发送：0x00 0C 00 FC B0 A5 56 01 11 46 D7 F1 C3

返回：0x00 14 01 FC 69 B6 C4 D2 1C 70 CA 6A 0F 1D 64 2B D6 D1 68 36 E8



## 5.4 ISO14443A/B CPU 卡命令

### 5.4.1 ISO14443 TYPE A 寻卡

功能：ISO14443A 寻卡，包含 MIFARE 系列和其他所有符合 ISO14443A 标准的卡片。在返回结果中，用户可以通过返回数据包的长度来判断序列号的长度，也可以通过 ATQA 来判断卡片的类型，还可以通过 SAK 来判断卡片是否支持 ISO14443-4。如果开启了自动寻卡，那么此命令是取自动寻卡的结果，而不是在接收到命令后进行寻卡。

上位机发送：

帧头	0x20	模式	校验字
----	------	----	-----

模式：1 字节，0：WUPA（寻所有卡）；1：REQA（寻未休眠的卡）；其它值：保留

模块回应成功：

帧头	0x20	数据	校验字
----	------	----	-----

数据：4，7 或 10 字节卡片序列号+2 字节 ATQA+1 字节 SAK

模块回应失败：

帧头	0xDF	校验字
----	------	-----

示例：

发送：0x00 05 00 20 00 25

返回：0x00 0B 01 20 32 41 00 21 04 00 28 54

### 5.4.2 ISO14443 TYPE A 寻全部卡

功能：ISO14443A 寻卡，将天线区域内所有能寻到的卡的 UID 输出。通常用于用 UID 做物品识别。

上位机发送：

帧头	0x7D	校验字
----	------	-----

模块回应成功：

帧头	0x7D	N * 数据	校验字
----	------	--------	-----

N：操作过程中寻到的卡片数量，最大值 16

数据：10 字节卡片序列号（不足部分用 0x00 补足）+2 字节 ATQA+1 字节 SAK + 1 字节实际 UID 长度

模块回应失败：



帧头	0x82	校验字
----	------	-----

示例:

发送: 0x00 04 00 7D 79

返回: 0x00 20 01 7D 76 3F 82 A2 00 00 00 00 00 00 00 00 00 08 04 52 90 32 22 00 00 00 00  
00 00 00 00 28 04 C7

### 5.4.3 ISO14443-4 TYPE A 卡复位 (RATS)

功能: 把一张符合 ISO14443-4 规格的卡片进行复位。在执行此命令前需要进行寻卡(命令代码: 0x20), 并在卡片的 SAK 中确认卡片支持 ISO14443-4。如果开启了自动寻卡功能, 那么在一次成功的执行 RATS 命令后, 自动寻卡功能将被强制关闭。

上位机发送:

帧头	0x30	EN_NADCID	校验字
----	------	-----------	-----

EN\_CID: 1: 使能 CID; 0: 禁止 CID (EMV 和 PBOC 标准是禁止的)

模块回应成功:

帧头	0x30	信息	校验字
----	------	----	-----

信息: 卡片复位信息, 字节长度由卡片决定

模块回应失败:

帧头	0xCF	校验字
----	------	-----

示例:

发送: 0x0005 00 30 00 35

返回: 0x00 14 01 30 10 78 80 90 02 20 90 00 00 00 00 00 21 00 41 32 BD

### 5.4.4 ISO14443-4 TYPE B 寻卡

功能: ISO14443B 寻卡, 寻到卡后并对卡片与读卡模块的通讯进行设定。

上位机发送:

帧头	0x60	模式	AFI	EN_NADCID	校验字
----	------	----	-----	-----------	-----

模式: 1 字节, 0: WUPB (寻所有卡); 1: REQB (寻未休眠的卡); 其他值: 保留

AFI: 要寻卡的 AFI, 如果不指定 AFI, 请使用 0x00

EN\_NADCID: 1: 使能 NADCID; 0: 禁止 NADCID (EMV 和 PBOC 标准是禁止的)

模块回应成功:



帧头	0x60	信息	校验字
----	------	----	-----

信息：12 字节，卡复位信息，构成内容如下：

0x50（1 字节），PUPI（4 字节），应用数据（4 字节），协议信息（3 字节）

更多详细内容请参考 ISO14443-3 “ATQB Response” 部分。

模块回应失败：

帧头	0x9F	校验字
----	------	-----

示例：

发送：0x0007 00 60 00 00 00 67

返回：0x00 12 01 60 50 18 12 02 23 00 00 00 00 00 00 81 00 00 89

## 5.4.5 按照 EMV 或 PBOC 的规范寻卡

功能：按照 EMV 或 PBOC 规范的要求进行寻卡操作，寻到卡后对卡片与读卡模块的通讯进行设定。本命令的寻卡是针对 T=CL 的 CPU 卡的，包含 ISO14443A 和 ISO14443B。通过此命令寻到卡后，即可对卡片发送 APDU 进行操作。此时，CID 和 NAD 都是禁止的。

上位机发送：

帧头	0x32	校验字
----	------	-----

模块回应成功：

帧头	0x32	卡类型	信息	校验字
----	------	-----	----	-----

卡类型：0x41、0x42 或 0x4D，0x41 代表寻到的卡片为 TYPE A；0x42 为 TYPE B；0x4D 为天线区域内有多张卡，此时不能继续操作卡片。

信息：TYPE A 的返回信息构成内容如下：

1 字节 UID 长度（取值为 4，7 或 10），UID（长度由前一字节决定），2 字节 ATQA，1 字节 SAK，ATS（卡的 ATS 信息长度不固定，请参照卡片厂商提供的手册）；

TYPE B 卡返回信息构成内容如下：

0x50（1 字节），PUPI（4 字节），应用数据（4 字节），协议信息（3 字节），

ATTRIB 信息（1 或多字节，由卡片决定）。

模块回应失败：

帧头	0xCD	校验字
----	------	-----

示例：

发送：0x00 04 00 32 36



返回: 0x00 12 01 32 42 50 18 12 02 23 00 00 00 00 00 81 81 00 18

## 5.4.6 发送 APDU 到 ISO14443-4 卡片

功能: 给一张符合 ISO14443-4 规格的卡片发送 APDU 命令。在执行此命令前需要对卡片进行复位。如果要操作 ISO14443-4 的卡片, 需要关闭自动寻卡, 因为自动寻卡的操作会使 ISO14443-4 卡片的状态丢失。

上位机发送:

帧头	0x31	APDU	校验字
----	------	------	-----

APDU: 要发送的 APDU

模块回应成功:

帧头	0x31	回应	校验字
----	------	----	-----

回应: 卡片回应, 长度由具体的命令决定

模块回应失败:

帧头	0xCE	校验字
----	------	-----

示例:

发送: 0x0009 00 31 00 84 00 00 08 B4

返回: 0x000E 01 31 B9 89 3A B0 16 40 7E D0 90 00 EC

## 5.4.7 ISO14443-4 TYPE B 卡休眠

功能: 将当前操作的 ISO14443B 卡置入休眠状态。不是所有卡都支持这个命令, 尤其是新卡, 多数不支持。

上位机发送:

帧头	0x62	PUPI	校验字
----	------	------	-----

PUPI: 4 字节, 需要休眠的卡的 PUPI

模块回应成功:

帧头	0x62	校验字
----	------	-----

模块回应失败:

帧头	0x9D	校验字
----	------	-----

示例:

发送: 0x0008 01 62 00 00 00 00 6B



返回: 0x0004 01 62 67





## 5.5 MIFARE 1K/4K/mini 卡命令

### 5.5.1 MIFARE 寻卡

MIFARE 系列卡片的寻卡请参照：[ISO14443A 寻卡](#)

### 5.5.2 MIFARE 1K/4K 读数据块

功能：读取 MIFARE 1K/4K 的一块数据。

上位机发送：

帧头	0x21	密钥标识	块号	密钥	校验字
----	------	------	----	----	-----

密钥标识：1 字节，密钥标识

BIT0=0：密钥 A；BIT0=1：密钥 B；

BIT1=0：使用指令中的密钥；BIT1=1：使用由 0x2D 命令下载到模块中的密钥；

BIT6:BIT5:BIT4:BIT3:BIT2：如果使用已经下载的密钥，在这里指明密钥编号

BIT7=0：使用上述密钥进行认证

BIT7=1：该块已经认证过，本次操作不需要认证密钥。这可以加快读写速度，但需要使用者严密的思路来保证流程的正确性（本操作与自动寻卡不能同时使用）；

（注意，详细的信息请参照：有关密钥标识）

块号：1 字节，要读取的数据块逻辑编号，S50 卡从 0 到 0x3F；S70 卡从 0 到 0xFF

密钥：6 字节，卡的密钥

模块回应成功：

帧头	0x21	数据	校验字
----	------	----	-----

数据：16 字节卡片的数据

模块回应失败：

帧头	0xDE	校验字
----	------	-----

示例：

发送：0x00C 00 21 00 01 FF FF FF FF FF FF 2C

返回：0x0014 01 21 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 34



### 5.5.3 MIFARE 1K/4K 读多个数据块

功能：读取 MIFARE 1K/4K 的一个扇区内的多块数据。如果跨扇区，读取操作将失败。

上位机发送：

帧头	0x2A	密钥标识	起始块号	块数	密钥	校验字
----	------	------	------	----	----	-----

密钥标识：1 字节，密钥标识

起始块号：1 字节，要读取的起始块号

块数：1 字节，要读取的块数（取决于要读的卡片结构，S50 取值 1-4）

密钥：6 字节，卡片的密钥

模块回应成功：

帧头	0x2A	数据	校验字
----	------	----	-----

数据：16 字节卡片的数据/块 \* 块数

模块回应失败：

帧头	0xD5	校验字
----	------	-----

示例：

发送：0x00 0D 00 2A 00 01 02 FF FF FF FF FF FF 24

返回：0x00 24 01 2A 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15

16 17 18 19 1A 1B 1C 1D 1E 1F 0F

### 5.5.4 MIFARE 1K/4K 写数据块

功能：将数据写入 MIFARE 1K/4K 的一个块。

上位机发送：

帧头	0x22	密钥标识	块号	密钥	数据	校验字
----	------	------	----	----	----	-----

密钥标识：1 字节，密钥标识

块号：1 字节，要写入的数据块逻辑编号，S50 卡从 0 到 0x3F；S70 卡从 0 到 0xFF

密钥：6 字节，卡片的密钥

数据：要写的 16 字节数据

模块回应成功：

帧头	0x22	校验字
----	------	-----

模块回应失败：

帧头	0xDD	校验字
----	------	-----



示例:

发送: 0x001C 00 22 00 01 FF FF FF FF FF 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D  
0E 0F 3F

返回: 0x0004 01 22 27

### 5.5.5 MIFARE 1K/4K 写多个数据块

功能: 写入 MIFARE 1K/4K 多个块的数据。如果跨扇区, 在跨扇区后的第一块将写入失败, 并在返回结果中提示出错。

上位机发送:

帧头	0x2B	密钥标识	起始块号	块数	密钥	数据	校验字
----	------	------	------	----	----	----	-----

密钥标识: 1 字节, 密钥标识

起始块号: 1 字节, 要写入的起始块号

块数: 1 字节, 需要写入的块数

密钥: 6 字节, 卡片的密钥

数据: 16 字节需要写入的数据/块 \* 块数

模块回应成功:

帧头	0x2B	校验字
----	------	-----

模块回应失败:

帧头	0xD4	校验字
----	------	-----

示例:

发送: 0x00 2D 00 2B 00 01 02 FF FF FF FF FF 00 01 02 03 04 05 06 07 08 09 0A 0B  
0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 05

返回: 0x0004 01 2B 2E

### 5.5.6 MIFARE 1K/4K 初始化钱包

功能: 将 MIFARE 1K/4K 的一个块初始化为一个钱包, 钱包的格式使用 MIFARE 1K/4K 默认的格式, 卡片的第 0 块和密钥块不能作为钱包使用。有关 MIFARE1K/4K 卡片的详细信息请阅读卡片资料。

上位机发送:

帧头	0x23	密钥标识	块号	密钥	钱包值	校验字
----	------	------	----	----	-----	-----



密钥标识: 1 字节, 密钥标识

块号: 1 字节, 要初始化的钱包块逻辑编号, S50 卡从 0 到 0x3F; S70 卡从 0 到 0xFF

密钥: 6 字节, 卡的密钥

钱包值: 4 字节, 初始钱包数值, 低字节在前

模块回应成功:

帧头	0x23	校验字
----	------	-----

模块回应失败:

帧头	0xDC	校验字
----	------	-----

示例:

发送: 0x0010 00 23 00 01 FF FF FF FF FF FF 04 03 02 01 36

返回: 0x0004 01 23 26

### 5.5.7 MIFARE 1K/4K 读钱包

功能: 读 MIFARE 1K/4K 的一个钱包。钱包的格式使用 MIFARE 1K/4K 默认的格式。

读出卡片内容后, 会按照钱包的格式去做验证, 如果格式不正确就返回失败。

上位机发送:

帧头	0x24	密钥标识	块号	密钥	校验字
----	------	------	----	----	-----

密钥标识: 1 字节, 密钥标识

块号: 1 字节, 要读取的钱包块逻辑编号, S50 卡从 0 到 0x3F; S70 卡从 0 到 0xFF

密钥: 6 字节, 卡的密钥

模块回应成功:

帧头	0x24	数据	校验字
----	------	----	-----

数据: 4 字节数值数据, 低字节在前

模块回应失败:

帧头	0xDB	校验字
----	------	-----

示例:

发送: 0x000C 00 24 00 01 FF FF FF FF FF FF 29

返回: 0x0008 01 24 04 03 02 01 29



## 5.5.8 MIFARE 1K/4K 钱包充值

功能：把 MIFARE 1K/4K 的一个钱包进行充值。钱包的格式使用 MIFARE 1K/4K 默认的格式。充值的意思是在原有钱包值的基础上增加。

上位机发送：

帧头	0x25	密钥标识	块号	密钥	数值	校验字
----	------	------	----	----	----	-----

密钥标识：1 字节，密钥标识

块号：1 字节，要充值的钱包块逻辑编号，S50 卡从 0 到 0x3F；S70 卡从 0 到 0xFF

密钥：6 字节，卡的密钥

数值：4 字节，钱包增加值，低字节在前

模块回应成功：

帧头	0x25	校验字
----	------	-----

模块回应失败：

帧头	0xDA	校验字
----	------	-----

示例：

发送：0x0010 00 25 00 01 FF FF FF FF FF FF 01 00 00 00 35

返回：0x0004 01 25 20

## 5.5.9 MIFARE 1K/4K 钱包扣款

功能：把 MIFARE 1K/4K 的一个钱包进行减值。钱包的格式使用 MIFARE 1K/4K 默认的格式。减值的意思是在原有钱包值的基础上减少，扣款只需要卡片的“读”权限就可进行。

上位机发送：

帧头	0x26	密钥标识	块号	密钥	数值	校验字
----	------	------	----	----	----	-----

密钥标识：1 字节，密钥标识

块号：1 字节，要扣款的钱包块逻辑编号，S50 卡从 0 到 0x3F；S70 卡从 0 到 0xFF

密钥：6 字节，卡的密钥

数值：4 字节，扣款值，低字节在前

模块回应成功：

帧头	0x26	校验字
----	------	-----

模块回应失败：



帧头	0xD9	校验字
----	------	-----

示例:

发送: 0x0010 00 26 00 01 FF FF FF FF FF FF 02 00 00 00 35

返回: 0x0004 01 26 23

### 5.5.10 MIFARE 1K/4K 备份钱包

功能: 把 MIFARE 1K/4K 的一个钱包备份到同一扇区中的另外一块中。钱包的格式使用 MIFARE 1K/4K 默认的格式。

上位机发送:

帧头	0x27	密钥标识	来源	目标	密钥	校验字
----	------	------	----	----	----	-----

密钥标识: 1 字节, 密钥标识

来源: 1 字节, 需要备份的钱包块逻辑编号, S50 卡从 0 到 0x3F; S70 卡从 0 到 0xFF

目标: 1 字节, 钱包目的块逻辑编号 (来源和目标需要在同一个扇区里)

密钥: 6 字节, 卡片的密钥

模块回应成功:

帧头	0x27	校验字
----	------	-----

模块回应失败:

帧头	0xD8	校验字
----	------	-----

示例:

发送: 0x000D 00 27 00 01 02 FF FF FF FF FF FF 29

返回: 0x0004 01 27 22

### 5.5.11 ISO14443A 卡休眠

功能: 把当前操作的包含 MIFARE 系列和其他所有符合 ISO14443A 标准的卡片设定为休眠状态。

上位机发送:

帧头	0x28	校验字
----	------	-----

模块回应成功:

帧头	0x28	校验字
----	------	-----

模块回应失败:



帧头	0xD7	校验字
----	------	-----

示例:

发送: 0x0004 00 28 2C

返回: 0x0004 01 28 2D

### 5.5.12 下载 MIFARE 1K/4K 卡片密钥到模块中

功能: 把 MIFARE 1K/4K 卡片的密钥下载到模块中。模块中有 32 个密钥存储空间, 可以存储 32 个不同的密钥。在使用下载到模块中的密钥时, 这个密钥不会出现在射频基站的引脚上, 可以防止被不法分子窃取, 保密性更强。因为 EEPROM 是有写的次数限制的, 因此, 不可频繁使用此命令, 这会造成 EEPROM 的失效而导致设备故障。

上位机发送:

帧头	0x2D	密钥索引	密钥	校验字
----	------	------	----	-----

密钥索引: 1 字节, 在模块中储存此密钥的编号, 编号取值从 0 到 0x1F

密钥: 6 字节, 需要保存在模块里的密钥

模块回应成功:

帧头	0x2D	校验字
----	------	-----

模块回应失败:

帧头	0xD2	校验字
----	------	-----

示例:

发送: 0x000B 00 2D 00 FF FF FF FF FF FF 26

返回: 0x0004 01 2D 28

### 5.5.13 有关密钥标识

在 MIFARE 1K/4K 读卡写卡等指令序列中有一字节密钥标识, 此字节用于模块来识别用什么方式获得操作卡片的密钥。

KeyIdentification							
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0

BIT0=0: 密钥 A; BIT0=1: 密钥 B;

BIT1=0: 使用指令中的密钥; BIT1=1: 使用由命令 0x2D 预先下载到模块中的密钥

BIT6:BIT5:BIT4:BIT3:BIT2: 如果使用已经下载的密钥, 在这里指名密钥编号



**BIT7=0:** 该块使用上述密钥进行认证

**BIT7=1:** 该块已经认证通过，本次操作不需认证操作

如果指令中的 **BIT1=0**，则 **BIT6:BIT5:BIT4:BIT3:BIT2** 的数据与本次操作卡片无关，如果指令中的 **BIT1=1**，则使用已经下载的密钥，密钥编号为 **BIT6:BIT5:BIT4:BIT3:BIT2**，同时，指令中的后续“6 字节密钥”就变成与本次操作卡片无关的数据了。

例如：一个密钥标识为 **0x00**，二进制为：00000000，此时：

**BIT0 = 0;** 代表认证卡片的 A 密钥

**BIT1 = 0;** 代表使用已经命令中的密钥

**BIT6:BIT5:BIT4:BIT3:BIT2** 为：00000，由于不使用已经下载的密钥，此时这个密钥索引在本条命令中无用。

例如：一个密钥标识为 **0x33**，二进制为：00110011，此时：

**BIT0 = 1;** 代表认证卡片的 B 密钥

**BIT1 = 1;** 代表使用已经下载到模块中的密钥

**BIT6:BIT5:BIT4:BIT3:BIT2** 为：01100，那么，就使用已经下载的第 01100 号密钥，16 进制为 **0x0C**，10 进制就是 12。





## 5.6 MIFARE Ultralight/Ultralight C/Ultralight EV1 卡命令

### 5.6.1 MIFARE Ultralight/Ultralight C/Ultralight EV1 寻卡

MIFARE Ultralight/Ultralight C/Ultralight EV1 的寻卡请参照：[ISO14443A 寻卡](#)

### 5.6.2 MIFARE Ultralight/Ultralight C/Ultralight EV1 读卡

功能：读取 Ultralight 或 Ultra Light C 卡中数据。对于 Ultralight，每次读可以得到 4 块或指定数量块数的数据，如果读取起始块为最后一块，那么读出块号是循环的。

上位机发送：

帧头	0x41	读取起始块	校验字
----	------	-------	-----

或上位机发送：

帧头	0x41	读取起始块	N	校验字
----	------	-------	---	-----

读取起始块：1 字节，读取起始块号

N：读取块数，如不输入块数则为默认 4 块，参数仅支持 4 的倍数，如：4，8，12，16，最大数量仅受卡片存储空间限制。

模块回应成功：

帧头	0x41	数据	校验字
----	------	----	-----

数据：4×N 字节数据，每次读操作读取起始块号开始的 4 块数据

模块回应失败：

帧头	0xBE	校验字
----	------	-----

示例：

发送：0x00 05 00 41 05 41 （默认读取 4 块，共 16 字节）

返回：0x00 14 01 41 00 06 01 10 11 FF 00 00 00 00 00 00 88 88 88 88 AD

发送：0x00 06 00 41 05 08 4A （读取 8 块，共 32 字节）

返回：0x00 24 01 41 00 06 01 10 11 FF 00 00 00 00 00 00 88 88 88 88 00 06 01 10 11 FF 00 00 00 00 00 88 88 88 88 AD

### 5.6.3 MIFARE Ultralight/Ultralight C/Ultralight EV1 写卡

功能：写入数据到 Ultra Light 或 Ultra Light C 卡中。每次写 1 块数据。



上位机发送:

帧头	0x42	块号	数据	校验字
----	------	----	----	-----

块号: 1 字节, 需要写入的数据块逻辑编号

数据: 写入的 4 字节数据

模块回应成功:

帧头	0x42	校验字
----	------	-----

模块回应失败:

帧头	0xBD	校验字
----	------	-----

示例:

发送: 0x00 09 00 42 05 55 55 55 55 4E

返回: 0x00 04 01 42 47

## 5.6.4 MIFARE UltraLightC 认证密钥

功能: 输入 UltraLight C 的密钥, 设备直接认证密钥, 认证过程由模块控制。

上位机发送:

帧头	0x43	密钥	校验字
----	------	----	-----

密钥: 16 字节的 UltraLight C 密钥

模块回应成功:

帧头	0x43	校验字
----	------	-----

模块回应失败:

帧头	0xBC	校验字
----	------	-----

示例:

发送: 0x0014 00 43 49 45 4D 4B 41 45 52 42 21 4E 41 43 55 4F 59 46 21

返回: 0x0004 01 43 46

## 5.6.5 MIFARE UltraLightC 读取 Ek(RndB)

功能: 读取 UltraLight C 卡片产生的加密 RndB; 0x44 和 0x45 指令是 Ultra Light C 认证的分离指令, 因为单片机计算 3DES 会比较慢, 因此认证也就比较耗时, 因此把认证部分分离出来可以让用户去计算 3DES 以节约认证的时间。用户可以先考虑用 0x43 去认证, 如果有认证速度上的要求, 那么可以联系我们获得技术支持后使用分离认证指令。



上位机发送:

帧头	0x44	校验字
----	------	-----

模块回应成功:

帧头	0x44	Ek(RndB)	校验字
----	------	----------	-----

Ek(RndB): 卡片返回的 RndB 加密数据。使用卡片的密钥对 RndB 进行 DES 解密; 解密后进行字节移位, 首字节移动到后, 得到 RndB'; 此时 Ek(RndB)为后续的 3DES CBC 算法的初始向量。

模块回应失败:

帧头	0xBB	校验字
----	------	-----

示例:

发送: 0x00 04 00 44 40

返回: 0x00 0C 01 44 93 AB 8B F9 42 56 AF 14 AC

## 5.6.6 MIFARE UltraLight C 认证 Ek(RndA + RndB')

功能: 输入加密好的 RndA + RndB'

上位机发送:

帧头	0x45	Ek(RndA + RndB')	校验字
----	------	------------------	-----

Ek(RndA + RndB'): 使用 DES CBC 对 RndA + RndB'进行加密的 16 字节结果; RndA 为用户自己指定的 8 字节随机数, RndB'由 0x44 指令获得的 RndB 经解密移位获得 (首字节移动到后);

模块回应成功:

帧头	0x45	Ek(RndA)	校验字
----	------	----------	-----

模块回应失败:

帧头	0xBA	校验字
----	------	-----

Ek(RndA): 卡片返回的加密 RndA; 3DES CBC 解密并移位后与 RndA 比较, 若相等则认证通过。

示例:

发送: 0x00 14 00 45 6E 30 F6 C1 17 05 C0 BE 48 40 DE 68 71 6C 9F F0 98

返回: 0x00 0C 01 45 30 F6 C1 17 05 C0 BE 6E 4D





## 5.6.9 Ultralight EV1 读计数器

功能：读取 Ultralight EV1 卡片内计数器的数据

上位机发送：

帧头	0x48	地址	校验字
----	------	----	-----

地址：1 字节，计数器的地址

模块回应成功：

帧头	0x48	数据	校验字
----	------	----	-----

数据：3 字节计数器数据

模块回应失败：

帧头	0xB7	校验字
----	------	-----

示例：

发送：0x0005 00 48 00 4D

返回：0x0007 01 48 06 00 00 48

## 5.6.10 Ultralight EV1 计数器递增

功能：Ultralight EV1 卡片内计数器的数据递增

上位机发送：

帧头	0x49	地址	数据	校验字
----	------	----	----	-----

地址：1 字节，计数器的地址

数据：3 字节计数器递增数据

模块回应成功：

帧头	0x49	校验字
----	------	-----

模块回应失败：

帧头	0xB6	校验字
----	------	-----

示例：

发送：0x0008 00 49 00 01 00 00 40

返回：0x0004 01 49 4C



### 5.6.11 Ultralight EV1 认证密钥

功能：Ultralight EV1 卡片认证密钥

上位机发送：

帧头	0x4A	密钥	校验字
----	------	----	-----

密钥：4 字节，卡片密钥

模块回应成功：

帧头	0x4A	校验字
----	------	-----

模块回应失败：

帧头	0xB5	校验字
----	------	-----

示例：

发送：0x0008 00 4A FF FF FF FF 42

返回：0x00 06 01 4A 00 00 4D

### 5.6.12 Ultralight EV1 读取签名

功能：读取 Ultralight EV1 卡片签名

上位机发送：

帧头	0x4B	校验字
----	------	-----

模块回应成功：

帧头	0x4B	签名	校验字
----	------	----	-----

签名：32 字节签名数据

模块回应失败：

帧头	0xB4	校验字
----	------	-----

示例：

发送：0x0004 00 4B 4F

返回：0x0024 01 4B 6F 44 89 DA 45 59 EF C9 87 1A DB D0 CF 66 4D 47 F7 0A C9 EC  
2E EC 7C FD 81 FF 74 4B 2E 28 97 1F 4D

### 5.6.13 Ultralight EV1 读取卡片计数器的 Tearing 事件

功能：读取 Ultralight EV1 卡片计数器的 Tearing 事件



上位机发送:

帧头	0x8C	地址	校验字
----	------	----	-----

地址: 1 字节, 计数器地址

模块回应成功:

帧头	0x8C	标志	校验字
----	------	----	-----

标志: 1 字节, 计数器合法标志

模块回应失败:

帧头	0x73	校验字
----	------	-----

示例:

发送: 0x0005 00 8C 00 89

返回: 0x0005 01 8C BD 35

## 5.6.14 Ultralight EV1 读取虚拟卡片信息

功能: 读取 Ultralight EV1 卡片的虚拟卡片标识

上位机发送:

帧头	0x8D	安装标识	射频描述	校验字
----	------	------	------	-----

安装标识: 16 字节, 安装标识 (IID)

射频描述: 4 字节, 射频基站能力标识 (PCDCAPS)

模块回应成功:

帧头	0x8D	标识	校验字
----	------	----	-----

标识: 1 字节, 虚拟卡片标识符 (VCTID)

模块回应失败:

帧头	0x72	校验字
----	------	-----

示例:

发送: 0x0018 00 8D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 95

返回: 0x0005 01 8D 05 8C



## 5.7 MIFARE Plus 卡命令

JMY600 系列读卡模块提供了操作 MIFARE Plus 卡的命令通道, MIFARE Plus 卡是 NXP 公司推出的用于代替 MIFARE 1 卡的产品, 我们提供的应用层指令都是基于层级 3 的, 在卡片的层级 3 中, 加密认证采用 AES 加密算法。JMY600 系列读卡模块在与卡片通讯的过程中, 全部采用加密数据+命令带 MAC+响应带 MAC 的方式, 与卡片的射频通讯过程中的数据安全性极高。

使用以下的命令可以使用户快速开展 MIFARE Plus 的研发任务, 对于高级用户, 也可以使用 ISO14443-4 的通道对卡片进行操作, 这样就可以避开我们在下面提供的这些操作 MIFARE Plus 的命令, 完全自己操控加密过程, 完全自己掌握安全性。当然, 这样也会让用户在研发过程中有巨大投入, 如果不是大型系统, 可以不必考虑这种方式。

### 5.7.1 卡片准备命令

#### 5.7.1.1 MIFARE Plus 寻卡

MIFARE Plus 的寻卡请参照: [ISO14443A 寻卡](#)

#### 5.7.1.2 MIFARE Plus 卡复位

请参照: [ISO14443-4 TYPE A 卡复位 \(RATS\)](#)

#### 5.7.1.3 MIFARE Plus 寻卡并复位

此命令会寻卡并对卡片复位, 请参照: [按照 EMV 或 PBOC 的规范寻卡](#)

### 5.7.2 卡片初始化命令

#### 5.7.2.1 MIFARE Plus Write Perso

功能: 初始化 AES 密钥及其所有块的初始值, 有关这些块的地址范围和应选择默认值, 请参考 MIFARE Plus 的 datasheet 或咨询我们。

上位机发送:

帧头	0x33	块地址	数据	校验字
----	------	-----	----	-----

块地址: 2 字节 (高字节在前),

数据: 16 字节

模块回应成功:





帧头	0x33	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0xCC	校验字
----	------	-----

示例：

发送：0x00 16 00 33 40 00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF 65

返回：0x00 04 01 33 36

### 5.7.2.2 MIFARE Plus Commit Perso

功能：Level 0 指令，从 Level0 切换到 Level1 或者 Level3，切换到的 Level 取决于卡片，需要在卡片订货时说明是 Level 1 或 Level 3，在使用此指令前需要先使用 [MIFARE Plus Write Perso](#) 指令写入所有 AES Key 及其他块的初始值，然后才能使用此指令使改动生效。

上位机发送：

帧头	0x34	校验字
----	------	-----

模块回应成功：

帧头	0x34	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0xCB	校验字
----	------	-----

示例：

发送：0x00 04 00 34 30

返回：0x00 04 01 34 31

### 5.7.2.3 MIFARE Plus 切换到 Level2/3

功能：Level 1 或 Level 2 指令，将卡片功能切换到 Level2 或者 Level3

上位机发送：

帧头	0x35	目标层级	密钥	校验字
----	------	------	----	-----

目标层级：1 字节，2：层级 2；3：层级 3

密钥：16 字节

模块回应成功：

帧头	0x35	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，



见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0xCA	校验字
----	------	-----

示例：

发送：0x00 15 00 35 03 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 23

返回：0x00 04 01 35 30

## 5.7.3 卡片 LEVEL 3 应用层命令

### 5.7.3.1 MIFARE Plus 授权数据块

功能：Level 3 指令，只对数据块进行授权

上位机发送：

帧头	0x36	密钥类型	块地址	密钥	校验字
----	------	------	-----	----	-----

密钥类型：0：A 密钥；1：B 密钥

块地址：2 字节（高字节在前）

密钥：16 字节

模块回应成功：

帧头	0x36	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0xC9	校验字
----	------	-----

示例：

发送：0x00 17 00 36 01 00 04 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 24

返回：0x00 04 01 36 33

### 5.7.3.2 MIFARE Plus 读数据块

功能：Level 3 指令，对数据块进行读操作，读前要对相应块进行授权。

上位机发送：

帧头	0x37	起始块号	块数	校验字
----	------	------	----	-----

起始块号：2 字节（高字节在前）

要读的块数：1 字节

模块回应成功：



帧头	0x37	状态码	数据	校验字
----	------	-----	----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

数据：块数×16 字节（不成功无此部分）

模块回应失败：

帧头	0xC8	校验字
----	------	-----

示例：

发送：0x00 07 00 37 00 05 01 34

返回：0x00 14 01 37 00 00 00 00 FF FF FF FF 00 00 00 00 05 FA 05 FA 22

### 5.7.3.3 MIFARE Plus 写数据块

功能：Level 3 指令，对数据块进行写操作，写前要对相应块进行授权。

上位机发送：

帧头	0x38	起始块号	块数	数据	校验字
----	------	------	----	----	-----

起始块号：2 字节（高字节在前）

块数：1 字节，要写入的块数

数据：要写的数据，块数×16 字节

模块回应成功：

帧头	0x38	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0xC7	校验字
----	------	-----

示例：

发送：0x00 17 00 38 00 04 01 00 00 00 00 FF FF FF FF 00 00 00 00 05 FA 05 FA 2A

返回：0x00 04 01 38 3D

### 5.7.3.4 MIFARE Plus 创建钱包

功能：Level 3 指令，用初始值创建钱包。

上位机发送：

帧头	0x39	块号	钱包值	校验字
----	------	----	-----	-----

块号：2 字节（高字节在前），要建立钱包的数据块编号

钱包值：4 字节（低字节在前），钱包初始值



模块回应成功:

帧头	0x39	状态码	校验字
----	------	-----	-----

状态码: 由卡片返回的执行结果代码, 指与卡片通讯成功, 但执行条件不符合等错误,

见表格: [卡片返回的状态码](#)

模块回应失败:

帧头	0xC6	校验字
----	------	-----

示例:

发送: 0x00 0A 00 39 00 04 00 01 00 00 36

返回: 0x00 04 01 39 3C

### 5.7.3.5 MIFARE Plus 读钱包余额

功能: Level 3 指令读取钱包余额。

上位机发送:

帧头	0x3A	块号	校验字
----	------	----	-----

块号: 2 字节 (高字节在前)

模块回应成功:

帧头	0x3A	状态码	数据	校验字
----	------	-----	----	-----

状态码: 由卡片返回的执行结果代码, 指与卡片通讯成功, 但执行条件不符合等错误,

见表格: [卡片返回的状态码](#)

数据: 4 字节钱包余额 (不成功无此部分)

模块回应失败:

帧头	0xC5	校验字
----	------	-----

示例:

发送: 0x00 06 01 3A 00 04 39

返回: 0x00 08 01 3A 00 01 00 00 32

### 5.7.3.6 MIFARE Plus 钱包充值

功能: Level 3 指令, 给钱包充值。

上位机发送:

帧头	0x3B	块号	钱包增加值	校验字
----	------	----	-------	-----

块号: 2 字节 (高字节在前)

钱包增加值: 4 字节 (低字节在前)

模块回应成功:



帧头	0x3B	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0xC4	校验字
----	------	-----

示例：

发送：0x00 0A 00 3B 00 04 00 01 00 00 34

返回：0x00 04 01 3B 3E

### 5.7.3.7 MIFARE Plus 钱包扣款

功能：Level 3 指令，给钱包扣款。

上位机发送：

帧头	0x3C	块号	钱包扣款值	校验字
----	------	----	-------	-----

块号：2 字节（高字节在前）

钱包扣款值：4 字节（低字节在前）

模块回应成功：

帧头	0x3C	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0xC3	校验字
----	------	-----

示例：

发送：0x00 0A 00 3C 00 04 00 01 00 00 33

返回：0x00 04 01 3C 39

### 5.7.3.8 MIFARE Plus 备份钱包

功能：Level3 指令，备份钱包到同一扇区的另一块。

上位机发送：

帧头	0x3D	源块号	目地块号	校验字
----	------	-----	------	-----

源块号：2 字节（高字节在前）

目地块号：2 字节（高字节在前）

模块回应成功：

帧头	0x3D	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，



见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0xC2	校验字
----	------	-----

示例：

发送：0x00 08 00 3D 00 04 00 05 34

返回：0x00 04 01 3D 38

### 5.7.3.9 MIFARE PlusFirst Authenticate

功能：Level 1/3 指令，在 Level 3 该指令主要在对块读写等操作前对块进行 AES 密钥认证授权。包括数据块、配置块、AES 密钥块等所有块进行授权操作，然后进行读写。

上位机发送：

帧头	0x3E	AES Key 地址	AES Key 值	校验字
----	------	------------	-----------	-----

AES Key 地址：2 字节 (高字节在前)

AES Key 值：16 字节

模块回应成功：

帧头	0x3E	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0xC1	校验字
----	------	-----

示例：

发送：0x00 16 00 3E 40 02 FF FF FF FF FF FF FF FF FF FF FF FF FF 6A

返回：0x00 04 01 3E 3B

### 5.7.3.10 MIFARE PlusFollowing Authenticate

功能：Level 1/3 指令，在 Level 3 该指令主要在对非数据块读写等操作前对块重新进行 AES 密钥认证授权，一般应用场合可以不用。主要在 First authenticate 认证后，需要认证时，可以用此指令

上位机发送：

帧头	0x3F	AES Key 地址	AES Key 值	校验字
----	------	------------	-----------	-----

AES Key 地址：2 字节 (高字节在前)

AES Key 值：16 字节

模块回应成功：



帧头	0x3F	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0xC0	校验字
----	------	-----

示例：

发送：0x00 16 00 3F 40 02 FF FF FF FF FF FF FF FF FF FF FF FF FF FF 6B

返回：0x00 04 01 3F 3A



## 5.7.4 卡片返回的状态码

MIFARE Plus 卡片操作后会给出执行结果，下表指示了这些可能的状态。

HEX 编码	状态	解释
0x90	OPERATION_SUCCESS	操作正常结束
0x06	AUTHENTICATION_ERROR	认证条件不满足、不存在的块、以数值格式访问的块不是一个数值块
0x07	COMMAND_OVERFLOW	在一个任务中的明文读写命令次数溢出
0x08	INVALID_MAC	下发给卡片的 MAC 错
0x88	INVALID_MAC of return	卡片返回的 MAC 错
0x09	INVALID_BLOCK_NUMBER	非法的块号
0x0A	NOT_EXIST_BLOCK_NUMBER	不存在的块号
0x0B	CONDITIONS_NOT_SATISFIED	使用条件不满足
0x0C	LENGTH_ERROR	长度错误
0x0F	GENERAL_MANIPULATION_ERROR	卡片内部错误





## 5.8 DESfire 卡命令

我们在此提供了分立的操作 DESfire 卡片的命令，DESfire 卡片的加密认证与通迅过程采用 DES 加密算法，JMY600 系列读卡模块在与卡片通讯的过程中，完全按照用户对卡片的加密设定进行与卡片的通讯，如果用户设定了与卡片的射频通讯过程是加密的，那么与卡片的射频通讯过程中的数据安全性极高。

使用以下的命令可以使用户快速开展 DESfire 的研发任务，当然，对于高级用户，也可以使用 ISO14443-4 的通道对 DESfire 卡片进行操作，这样就可以完全绕开我们在下面提供的这些操作 DESfire 的命令，完全自己操控加密过程，完全自己掌握安全性。当然，这样也会让用户在研发过程中有巨大投入，如果不是大型系统，可以不必考虑这种方式。

### 5.8.1 卡片准备命令

#### 5.8.1.1 DESfire 寻卡

DESfire 的寻卡请参照：[ISO14443A 寻卡](#)

#### 5.8.1.2 DESfire 卡复位

请参照：[ISO14443-4 TYPE A 卡复位 \(RATS\)](#)

#### 5.8.1.3 DESfire 寻卡并复位

此命令会寻卡并对卡片复位，请参照：[按照 EMV 或 PBOC 的规范寻卡](#)

#### 5.8.1.4 DESFire Authenticate

功能：卡片与读写器三重相互认证，认证的密钥编号可以是主控密钥或任何其他密钥。

本命令是上位机把密钥发给读卡模块，由模块进行加解密并与卡片交互获得认证结果。对于高级用户，用户可以自己控制认证过程以提高安全性，我们提供了额外的认证接口，参见 0x8E：[DESFire Authenticate first step Get ekNo\(RndB\)](#)和 0x8F：[DESFire Authenticate second step get ekNo\(RndA'\)](#)，对于提高保密性能有帮助，但操作相对复杂。

对于所有有关卡片操作中的加密解密方法，请参照 datasheet 进行加密计算，或使用我们提供的工具以及可参考的源代码，如果需要帮助，请联系我们。

上位机发送：

帧头	0x90	密钥号	密钥	校验字
----	------	-----	----	-----

密钥号：1 字节



密钥：16 字节

模块回应成功：

帧头	0x90	状态码	过程密钥	校验字
----	------	-----	------	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

过程密钥：16 字节，只有认证通过才会有过程密钥返回，过程密钥在后续的卡片操作中会用到，在加密的通讯过程中的加密数据都需要使用此密钥来加解密。

模块回应失败：

帧头	0x6F	校验字
----	------	-----

示例：

发送：0x00 15 00 90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 85

返回：0x00 15 01 90 00 CC 6C E1 74 46 42 09 8D 1B 78 17 03 49 4C 67 A1 85

### 5.8.1.5 DESFire Authenticate first step Get ekNo(RndB)

功能：模块发起认证，从卡片获得 ekNo(RndB)。

上位机发送：

帧头	0x8E	密钥号	校验字
----	------	-----	-----

密钥号：1 字节

模块回应成功：

帧头	0x8E	状态码	ekNo(RndB)	校验字
----	------	-----	------------	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)。在此状态码为 0xAF 是正确的，需要由上位机提供进一步的数据，后续命令必须为：[DESFire Authenticate second step get ekNo\(RndA'\)](#)才能进行继续进行认证。

ekNo(RndB)：8 字节，卡片产生的随机数，由指定认证的密钥进行加密后的结果。模块得到后使用正确密钥进行解密计算，可以得到 RndB。

模块回应失败：

帧头	0x71	校验字
----	------	-----

示例：

发送：0x00 05 00 8E 00 8B

返回：0x00 0D 01 8E AF 28 EA 37 7B 60 A0 DC F8 47



### 5.8.1.6 DESFire Authenticate second step get ekNo(RndA')

功能：上位机产生随机数 RndA，并与 RndB 进行组合后使用正确密钥加密，通过模块发送给卡片，从卡片获得 RndA'，RndA'经过移位后，如果与 RndA 相等，此时即可与卡片获得相互认证。

上位机发送：

帧头	0x8F	dkNo(RndA + RndB')	校验字
----	------	--------------------	-----

dkNo(RndA + RndB')：16 字节

模块回应成功：

帧头	0x8F	状态码	ekNo(RndA')	校验字
----	------	-----	-------------	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，见表格：[卡片返回的状态码](#)

ekNo(RndA')：加密的上位机随机数，使用正确密钥解密后即可还原，经过移位后如果与 RndA 相等，那么认证通过。

后续过程密钥：16 字节，由 RndA 和 RndB 组合得到如下公式：

$$\text{Session key} = \text{RndA}[0..3] + \text{RndB}[0..3] + \text{RndA}[4..7] + \text{RndB}[4..7]$$

过程密钥在后续的卡片操作中会用到，在加密的通讯过程中的加密数据都需要使用此过程密钥来加解密。

模块回应失败：

帧头	0x70	校验字
----	------	-----

示例：

发送：0x00 14 00 8F 42 FF CF C7 87 A1 90 32 B9 CC B8 A5 E7 70 C6 F5 66

返回：0x00 0D 01 8F 00 28 4C 45 14 2A 60 17 67 8C

### 5.8.1.7 DESFire Select Application

功能：选择指定的卡片应用。以后的指令将针对这个应用产生作用。

上位机发送：

帧头	0x98	应用标识	校验字
----	------	------	-----

应用标识：3 字节，低字节在前

模块回应成功：

帧头	0x98	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)



模块回应失败:

帧头	0x67	校验字
----	------	-----

示例:

发送: 0x00 07 00 98 12 34 56 EF

返回: 0x00 05 01 98 00 9C

## 5.8.2 卡片初始化命令

### 5.8.2.1 DESfireFormat Card

功能: 格式化卡片, 卡片所有应用和应用下的所有文件都被删除。

上位机发送:

帧头	0x99	校验字
----	------	-----

模块回应成功:

帧头	0x99	状态码	校验字
----	------	-----	-----

状态码: 由卡片返回的执行结果代码, 指与卡片通讯成功, 但执行条件不符合等错误,

见表格: [卡片返回的状态码](#)

模块回应失败:

帧头	0x67	校验字
----	------	-----

示例:

发送: 0x00 04 00 99 9D

返回: 0x00 05 01 99 00 9D

### 5.8.2.2 DESFire Create Application

功能: 建立新的卡片应用。

上位机发送:

帧头	0x95	应用标识	密钥设置	密钥数量	校验字
----	------	------	------	------	-----

应用标识: 3 字节, 低字节在前

密钥设置: 1 字节

密钥数量: 1 字节

模块回应成功:

帧头	0x95	状态码	校验字
----	------	-----	-----

状态码: 由卡片返回的执行结果代码, 指与卡片通讯成功, 但执行条件不符合等错误,

见表格: [卡片返回的状态码](#)



模块回应失败:

帧头	0x6A	校验字
----	------	-----

示例:

发送: 0x00 09 00 95 12 34 56 EF 03 00

返回: 0x00 05 01 95 00 91

### 5.8.2.3 DESFire Change Key Settings

功能: 修改主控密钥/应用主密钥的密钥设置信息。指令执行过程中数据采用 DES/3DES 加密和 CRC 校验。

上位机发送:

帧头	0x91	新的密钥设置信息	校验字
----	------	----------	-----

新的密钥设置信息: 经过加密的 8 字节信息

模块回应成功:

帧头	0x91	状态码	校验字
----	------	-----	-----

状态码: 由卡片返回的执行结果代码, 指与卡片通讯成功, 但执行条件不符合等错误,

见表格: [卡片返回的状态码](#)

模块回应失败:

帧头	0x6E	校验字
----	------	-----

示例:

发送: 0x00 0C 00 91 84 87 0F C4 44 83 B1 C9 EA

返回: 0x00 05 01 91 00 95

### 5.8.2.4 DESFire Get Key Settings

功能: 取得主控密钥/应用主密钥的当前密钥设置信息。

上位机发送:

帧头	0x92	校验字
----	------	-----

密钥号: 1 字节

模块回应成功:

帧头	0x92	状态码	密钥信息	密钥个数	校验字
----	------	-----	------	------	-----

状态码: 由卡片返回的执行结果代码, 指与卡片通讯成功, 但执行条件不符合等错误,

见表格: [卡片返回的状态码](#)

密钥设置信息: 1 字节

密钥个数: 1 字节



模块回应失败:

帧头	0x6D	校验字
----	------	-----

示例:

发送: 0x00 04 00 92 96

返回: 0x00 07 01 92 00 EF 0C 77

### 5.8.2.5 DESFire Change Key

功能: 修改存储在卡片内的任何密钥。该指令执行过程中数据采用 DES/3DES 加密和 CRC 校验。

上位机发送:

帧头	0x93	密钥 ID	加密密钥数据	校验字
----	------	-------	--------	-----

密钥 ID: 1 字节

加密密钥数据: 24 字节 (请参照 datasheet 进行加密计算, 或使用我们提供的工具以及可参考的源代码)

模块回应成功:

帧头	0x93	状态码	校验字
----	------	-----	-----

状态码: 由卡片返回的执行结果代码, 指与卡片通讯成功, 但执行条件不符合等错误,

见表格: [卡片返回的状态码](#)

模块回应失败:

帧头	0x6C	校验字
----	------	-----

示例:

发送: 0x00 1D 00 93 01 B5 30 7D 8F 42 7E D4 E3 C2 9B 0E 2B F0 A8 9D 49 59 35 9E 62

1F FE C8 00 BA

返回: 0x00 05 01 93 00 97

### 5.8.2.6 DESFire Get Key Version

功能: 取得当前密钥版本信息。

上位机发送:

帧头	0x94	密钥号	校验字
----	------	-----	-----

密钥号: 1 字节

模块回应成功:

帧头	0x94	状态码	密钥版本信息	校验字
----	------	-----	--------	-----

状态码: 由卡片返回的执行结果代码, 指与卡片通讯成功, 但执行条件不符合等错误,



见表格：[卡片返回的状态码](#)

密钥版本信息：1 字节

模块回应失败：

帧头	0x6B	校验字
----	------	-----

示例：

发送：0x00 05 00 94 01 90

返回：0x00 06 01 94 00 F0 63

### 5.8.2.7 DESFire Delete Application

功能：删除指定的卡片应用。

上位机发送：

帧头	0x96	应用标识	校验字
----	------	------	-----

应用标识：3 字节，低字节在前

模块回应成功：

帧头	0x96	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0x69	校验字
----	------	-----

示例：

发送：0x00 07 00 96 4A 4D 59 CF

返回：0x00 05 01 96 00 92

### 5.8.2.8 DESFire Get Version

功能：该函数返回卡片的制造业的相关数据。

上位机发送：

帧头	0x9A	校验字
----	------	-----

模块回应成功：

帧头	0x9A	状态码	数据	校验字
----	------	-----	----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

数据：28 字节卡片的制造业的相关数据

模块回应失败：



帧头	0x65	校验字
----	------	-----

示例:

发送: 0x00 04 00 9A 9E

返回: 0x00 21 01 9A 00 04 01 01 01 00 16 05 04 01 01 01 04 16 05 04 0B 30 9A 4F 22 80

BA 24 17 A9 20 07 11 E0

### 5.8.2.9 DESFire Get Application IDs

功能: 该函数返回卡片当前所有应用的应用标识符。

上位机发送:

帧头	0x97	校验字
----	------	-----

模块回应成功:

帧头	0x97	状态码	应用标识	校验字
----	------	-----	------	-----

状态码: 由卡片返回的执行结果代码, 指与卡片通讯成功, 但执行条件不符合等错误,

见表格: [卡片返回的状态码](#)

应用标识: 每个应用标识长度为 3 字节, 长度为: 标识个数×3 字节

模块回应失败:

帧头	0x68	校验字
----	------	-----

示例:

发送: 0x00 04 00 97 93

返回: 0x00 0B 01 97 00 4A 4D 07 4A 4D 59 C3

### 5.8.2.10 DESFire Get File IDs

功能: 取卡片当前应用目录下的所有文件的文件标识符。

上位机发送:

帧头	0x9B	校验字
----	------	-----

模块回应成功:

帧头	0x9B	状态码	文件标识	校验字
----	------	-----	------	-----

状态码: 由卡片返回的执行结果代码, 指与卡片通讯成功, 但执行条件不符合等错误,

见表格: [卡片返回的状态码](#)

文件标识: 每个文件标识长度为 1 字节, 总长度为文件个数×1 字节

模块回应失败:

帧头	0x64	校验字
----	------	-----

示例:

发送: 0x00 04 00 9B 9F





返回：0x00 08 01 9B 00 03 04 01 94

### 5.8.2.11 DESFire Get FileSettings

功能：取当前应用目录下的指定文件的设置信息。

上位机发送：

帧头	0x9C	文件标识	校验字
----	------	------	-----

文件标识：1 字节

模块回应成功：

帧头	0x9C	状态码	数据	校验字
----	------	-----	----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

数据：根据文件类型的不同长度有所不同，具体情况如下：

- 数据文件：1 字节文件类型+1 字节通讯设置+2 字节访问权限+3 字节文件大小
- 数值文件：1 字节文件类型+1 字节通讯设置+2 字节访问权限+4 字节最小值+4 字节最大值+4 字节当前最大限存金额+1 字节是否支持限存
- 记录文件：1 字节文件类型+1 字节通讯设置+2 字节访问权限+3 字节记录大小+3 字节最大记录数+3 字节当前记录数

(注：以上多字节数据均为低字节在前)

模块回应失败：

帧头	0x63	校验字
----	------	-----

示例：

发送：0x00 05 00 9C 01 98

返回：0x00 16 01 9C 00 02 00 EE EE 00 00 00 00 77 77 77 77 00 00 00 00 89

### 5.8.2.12 DESFire Change File Settings

功能：修改卡片当前应用目录下的指定文件的设置信息。

上位机发送：

明文：

帧头	0x9D	文件标识	通信设置	文件权限	校验字
----	------	------	------	------	-----

密文：

帧头	0x9D	文件标识	设置信息	校验字
----	------	------	------	-----

文件标识：1 字节

通信设置：1 字节，0：明文传输；1：MAC 码校验；3：DES/3DES 加密



文件权限：2 字节(低字节在前)

加密的设置信息：8 字节，1 字节通讯设置+2 字节文件权限+2 字节 CRC+3 字节 0x00

经加密得到

模块回应成功：

帧头	0x9D	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0x62	校验字
----	------	-----

示例：

发送：0x00 08 00 9D 03 00 11 11 96

返回：0x00 05 01 9D 00 99

### 5.8.2.13 DESFire Create Std Data File

功能：在卡片当前应用目录下的建新的标准数据文件。

上位机发送：

帧头	0x9E	文件标识	通信设置	操作权限	文件长度	校验字
----	------	------	------	------	------	-----

文件标识：1 字节

通信设置：1 字节，0：明文传输；1：MAC 码校验；3：DES/3DES 加密

操作权限：2 字节(低字节在前)

文件长度：3 字节(低字节在前)

模块回应成功：

帧头	0x9E	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0x61	校验字
----	------	-----

示例：

发送：0x00 0B 00 9E 03 00 EE EE 00 01 00 97

返回：0x00 05 01 9E 00 9A

### 5.8.2.14 DESFire Create Backup Data File

功能：在卡片当前应用目录下建新的数据文件，支持备份机制。卡片给该文件分配的



空间总是大于或等于文件大小的两倍，且为 32 的倍数。

上位机发送：

帧头	0x9F	文件标识	通信设置	操作权限	文件长度	校验字
----	------	------	------	------	------	-----

文件标识：1 字节

通信设置：1 字节，0：明文传输；1：MAC 码校验；3：DES/3DES 加密

操作权限：2 字节(低字节在前)

文件长度：3 字节(低字节在前)

模块回应成功：

帧头	0x9F	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0x60	校验字
----	------	-----

示例：

发送：0x00 0B 00 9F 04 00 EE EE 00 01 00 91

返回：0x00 05 01 9F 00 9B

### 5.8.2.15 DESFire Create Value File

功能：在卡片当前应用目录下的建新的数值文件，支持备份机制。

上位机发送：

帧头	0xA0	文件标识	通信设置	操作权限	最小值	最大值	初始值	是否限存	校验字
----	------	------	------	------	-----	-----	-----	------	-----

文件标识：1 字节

通信设置：1 字节，0：明文传输；1：MAC 码校验；3：DES/3DES 加密

操作权限：2 字节(低字节在前)

最小值：4 字节(有符号整型，低字节在前)

最大值：4 字节(有符号整型，低字节在前)

初始值：4 字节(有符号整型，低字节在前)

是否限存：1 字节，0：不限制；1：限制

模块回应成功：

帧头	0xA0	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，



见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0x5F	校验字
----	------	-----

示例：

发送：0x00 15 00 A0 01 00 EE EE 00 00 00 00 77 77 77 77 22 22 22 22 00 B4

返回：0x00 05 01 A0 00 A4

### 5.8.2.16 DESFire Create Linear Record File

功能：在卡片当前应用目录下的建新的线性记录文件，支持备份机制。

上位机发送：

帧头	0xA1	文件标识	通信设置	操作权限	记录大小	最大记录个数	校验字
----	------	------	------	------	------	--------	-----

文件标识：1 字节

通信设置：1 字节，0：明文传输；1：MAC 码校验；3：DES/3DES 加密

操作权限：2 字节(低字节在前)

记录大小：3 字节(低字节在前)，单个记录的字节数

最大记录个数：3 字节(低字节在前)，记录文件的条目数量

模块回应成功：

帧头	0xA1	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0x5E	校验字
----	------	-----

示例：

发送：0x00 0E 00 A1 08 00 00 00 20 00 00 20 00 00 A7

返回：0x00 05 01 A1 00 A5

### 5.8.2.17 DESFire Create Cyclic Record File

功能：在卡片当前应用目录下的建新的循环记录文件。

上位机发送：

帧头	0xA2	文件标识	通信设置	操作权限	记录大小	最大记录个数	校验字
----	------	------	------	------	------	--------	-----

文件标识：1 字节

通信设置：1 字节，0：明文传输；1：MAC 码校验；3：DES/3DES 加密



操作权限：2 字节(低字节在前)

记录大小：3 字节(低字节在前)，单个记录的字节数

最大记录个数：3 字节(低字节在前)，记录文件的条目数量

模块回应成功：

帧头	0xA2	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0x5D	校验字
----	------	-----

示例：

发送：0x00 0E 00 A2 06 00 EE EE 10 00 00 10 00 00 AA

返回：0x00 05 01 A2 00 A6

### 5.8.2.18 DESFire Delete File

功能：删除卡片当前应用目录下指定的文件。

上位机发送：

帧头	0xA3	文件标识	校验字
----	------	------	-----

文件标识：1 字节

模块回应成功：

帧头	0xA3	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0x5C	校验字
----	------	-----

示例：

发送：0x00 05 00 A3 06 A0

返回：0x00 05 01 A3 00 A7

## 5.8.3 卡片应用层命令

### 5.8.3.1 DESFire Read Data

功能：读取卡片当前目录下指定数据文件（标准数据文件或备份数据文件）的数据。

上位机发送：



帧头	0xA4	文件标识	偏移地址	读取长度	校验字
----	------	------	------	------	-----

文件标识：1 字节

偏移地址：3 字节(低字节在前)，文件内的偏移地址

读取长度：3 字节(低字节在前)，需要读取的字节数

模块回应成功：

帧头	0xA4	状态码	数据	校验字
----	------	-----	----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

数据：实际读出的数据

模块回应失败：

帧头	0x5B	校验字
----	------	-----

示例：

发送：0x00 0B 00 A4 03 00 00 00 10 00 00 BC

返回：0x00 15 01 A4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 B0

### 5.8.3.2 DESFire Write Data

功能：向卡片当前目录下的指定数据文件（标准数据文件或备份数据文件）写数据，备份数据文件写执行后需要执行：[DESFire Commit Transaction](#) 才能生效。

上位机发送：

帧头	0xA5	文件标识	偏移地址	写入长度	数据	校验字
----	------	------	------	------	----	-----

文件标识：1 字节

偏移地址：3 字节(低字节在前)，文件内的偏移地址

写入长度：3 字节(低字节在前)，需要写入的字节数

数据：要写入的数据

模块回应成功：

帧头	0xA5	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0x5A	校验字
----	------	-----

示例：

发送：0x00 14 00 A5 07 00 00 00 08 00 00 00 11 22 33 44 55 66 77 88 36



返回: 0x00 05 01 A5 00 A1

### 5.8.3.3 DESFire Get Value

功能: 读取卡片当前目录下指定值文件的当前值。

上位机发送:

帧头	0xA6	文件标识	校验字
----	------	------	-----

文件标识: 1 字节

模块回应成功:

帧头	0xA6	状态码	数据	校验字
----	------	-----	----	-----

状态码: 由卡片返回的执行结果代码, 指与卡片通讯成功, 但执行条件不符合等错误,

见表格: [卡片返回的状态码](#)

数据: 根据是否加密有两种长度

不加密: 4 字节数值(低字节在前)

加密: 8 字节加密数据, 解密后得到 4 字节数值(低字节在前)、2 字节 CRC 和 2 字节 0x00

模块回应失败:

帧头	0x59	校验字
----	------	-----

示例:

发送: 0x00 05 00 A6 01 A2

返回: 0x00 09 01 A6 00 22 22 22 22 AE

### 5.8.3.4 DESFire Credit

功能: 当前目录下指定值文件加值, 执行后需要执行: [DESFire Commit Transaction](#) 才能生效。

上位机发送:

帧头	0xA7	文件标识	数据	校验字
----	------	------	----	-----

文件标识: 1 字节

数据: 根据是否加密有两种长度

不加密: 4 字节数值(低字节在前)

加密: 8 字节加密数据, 4 字节数值(低字节在前)、2 字节 CRC 和 2 字节 0x00 经加密得到。

模块回应成功:

帧头	0xA7	状态码	校验字
----	------	-----	-----



状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0x58	校验字
----	------	-----

示例：

发送：0x00 09 00 A7 01 01 00 00 00 AE

返回：0x00 05 01 A7 00 A3

### 5.8.3.5 DESFire Debit

功能：当前目录下指定值文件减值，执行后需要执行：[DESFire Commit Transaction](#)才能生效。

上位机发送：

帧头	0xA8	文件标识	数据	校验字
----	------	------	----	-----

文件标识：1 字节

数据：根据是否加密有两种长度

不加密：4 字节数值(低字节在前)

加密：8 字节加密数据，4 字节数值(低字节在前)、2 字节 CRC 和 2 字节 0x00

经加密得到。

模块回应成功：

帧头	0xA8	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0x57	校验字
----	------	-----

示例：

发送：0x00 09 00 A8 01 01 00 00 00 A1

返回：0x00 05 01 A8 00 AC

### 5.8.3.6 DESFire Limited Credit

功能：当前目录下指定值文件加限定的值。该操作必须在 Debit 操作后执行，而且每执行完一次 Debit 操作只能执行一次 Limited Credit 操作，而且该操作加的值必须小于或等于 Debit 操作减去的值。执行后需要执行：[DESFire Commit Transaction](#)才能生效。

上位机发送：





帧头	0xA9	文件标识	数据	校验字
----	------	------	----	-----

文件标识：1 字节

数据：根据是否加密有两种长度

不加密：4 字节数值(低字节在前)

加密：8 字节加密数据，4 字节数值(低字节在前)、2 字节 CRC 和 2 字节 0x00 经加密得到。

模块回应成功：

帧头	0xA9	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0x56	校验字
----	------	-----

示例：

发送：0x00 09 00 A9 01 01 00 00 00 A0

返回：0x00 05 01 A9 00 AD

### 5.8.3.7 DESFire Write Record

功能：卡片当前目录下指定记录文件（线性记录文件或循环记录文件）写数据到下一条记录中。对于记录文件，写记录到最后，如果文件满则出错；对于循环记录文件，如果满则会写到最老的记录上。执行后需要执行：[DESFire Commit Transaction](#) 才能生效。

上位机发送：

帧头	0xAA	文件标识	偏移地址	写入长度	数据	校验字
----	------	------	------	------	----	-----

文件标识：1 字节

偏移地址：单个记录中的偏移地址 3 字节(低字节在前)

写入长度：3 字节(低字节在前)，大于 0 且小于等于（记录大小减去记录内的偏移）

数据：要写入的数据

模块回应成功：

帧头	0xAA	状态码	校验字
----	------	-----	-----

状态码：由卡片返回的执行结果代码，指与卡片通讯成功，但执行条件不符合等错误，

见表格：[卡片返回的状态码](#)

模块回应失败：

帧头	0x55	校验字
----	------	-----



示例:

发送: 0x00 1B 00 AA 06 00 00 00 10 00 00 00 11 22 33 44 55 66 77 88 99 AA BB CC DD  
EE FF A7

返回: 0x00 05 01 AA 00 AE

### 5.8.3.8 DESFire Read Record

功能: 读取卡片当前目录下指定记录文件一条或多条记录信息。

上位机发送:

帧头	0xAB	文件标识	起始记录	读取数目	校验字
----	------	------	------	------	-----

文件标识: 1 字节

起始记录: 3 字节(低字节在前)

读取数目: 读取的记录数目, 3 字节(低字节在前)

模块回应成功:

帧头	0xAB	状态码	数据	校验字
----	------	-----	----	-----

状态码: 由卡片返回的执行结果代码, 指与卡片通讯成功, 但执行条件不符合等错误,

见表格: [卡片返回的状态码](#)

数据: 实际读出的数据, 用户需要根据设定的记录文件的规格拆分数据。

模块回应失败:

帧头	0x54	校验字
----	------	-----

示例:

发送: 0x00 0B 00 AB 06 00 00 00 01 00 00 A7

返回: 0x00 15 01 AB 00 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF BF

### 5.8.3.9 DESFire Clear Record File

功能: 将卡片当前目录下指定记录文件清空, 执行后需要执行: [DESFire Commit Transaction](#) 才能生效。

上位机发送:

帧头	0xAC	文件标识	校验字
----	------	------	-----

文件标识: 1 字节

模块回应成功:

帧头	0xAC	状态码	校验字
----	------	-----	-----

状态码: 由卡片返回的执行结果代码, 指与卡片通讯成功, 但执行条件不符合等错误,

见表格: [卡片返回的状态码](#)



模块回应失败:

帧头	0x53	校验字
----	------	-----

示例:

发送: 0x00 05 00 AC 05 AC

返回: 0x00 05 01 AC 00 A8

### 5.8.3.10 DESFire Commit Transaction

功能: 提交当前目录下的备份数据文件、值文件和记录文件前面所有的写访问操作, 使之前的修改有效。

上位机发送:

帧头	0xAD	校验字
----	------	-----

模块回应成功:

帧头	0xAD	状态码	校验字
----	------	-----	-----

状态码: 由卡片返回的执行结果代码, 指与卡片通讯成功, 但执行条件不符合等错误,

见表格: [卡片返回的状态码](#)

模块回应失败:

帧头	0x52	校验字
----	------	-----

示例:

发送: 0x00 04 00 AD A9

返回: 0x00 05 01 AD 00 A9

### 5.8.3.11 DESFire Abort Transaction

功能: 取消当前目录下的备份数据文件、值文件和记录文件前面所有的写访问操作, 使之前的修改无效。

上位机发送:

帧头	0xAE	校验字
----	------	-----

模块回应成功:

帧头	0xAE	状态码	校验字
----	------	-----	-----

状态码: 由卡片返回的执行结果代码, 指与卡片通讯成功, 但执行条件不符合等错误,

见表格: [卡片返回的状态码](#)

模块回应失败:

帧头	0x51	校验字
----	------	-----

示例:

发送: 0x00 04 00 AE AA



返回: 0x00 05 01 AE 00 AA



## 5.8.4 卡片返回的状态码

DESfire 卡片操作后会给出执行结果，下表指示了这些可能的状态。

HEX 编码	状态	解释
0x00	OPERATION_OK	操作正常结束
0x0C	NO_CHANGES	Backup 文件没有改变，不需要 CommitTransaction 或者 AbortTransaction
0x0E	OUT_OF_EEPROM_ERROR	存储器不足以完成该指令
0x1C	ILLEGAL_COMMAND_CODE	该命令码不被支持
0x1E	INTEGRITY_ERROR	CRC 或 MAC 码不匹配，填补的字节无效
0x40	NO_SUCH_KEY	无此密钥
0x7E	LENGTH_ERROR	命令的字符串长度错误
0x9D	PERMISSION_DENIED	当前配置或状态不允许该请求
0x9E	PARAMETER_ERROR	参数的值无效
0xA0	APPLICATION_NOT_FOUND	请求的应用标识不存在
0xA1	APPL_INTEGRITY_ERROR	在应用内不可恢复的错误，该应用将无效
0xAE	AUTHENTICATION_ERROR	当前认证状态不允许该请求指令
0xAF	ADDITIONAL_FRAME	期望再发数据帧
0xBE	BOUNDARY_ERROR	尝试超出文件范围读/写数据
0xC1	PICC_INTEGRITY_ERROR	卡片内不可恢复的错误，卡片将无效
0xCA	COMMAND_ABORTED	前一指令尚未完成，PCD 并未请求或提供全部数据帧
0xCD	PICC_DISABLED_ERROR	卡片由于不可恢复的错误而无效
0xCE	COUNT_ERROR	应用的总数最大为 28，不可以再创建应用
0xDE	DUPLICATE_ERROR	不可以创建该文件或应用，该文件号或应用号已存在
0xEE	EEPROM_ERROR	因为掉电，内部备份/反转机制不能完成写操作
0xF0	FILE_NOT_FOUND	指定的文件号不存在
0xF1	FILE_INTEGRITY_ERROR	文件内不可恢复的错误，该文件无效



## 5.9 SR176 卡命令

### 5.9.1 SR 系列卡单通道初始化

功能：SR 系列卡（指：SR176/SRI512/SRI1K/SRI2K/SRI4K/SRIX4K，下同）单通道初始化。在使用“SR 系列卡选定卡”命令后，才可对卡片进行读写等操作。详细的卡片操作，请参照卡片的说明书。

上位机发送：

帧头	0x63	校验字
----	------	-----

模块回应成功：

帧头	0x63	卡片 ID	校验字
----	------	-------	-----

卡片 ID：1 字节，卡片 ID

模块回应失败：

帧头	0x9C	校验字
----	------	-----

### 5.9.2 SR 系列卡选定卡

功能：选定一张 SR 卡为当前操作的卡。选定后可以对当前卡片进行读写等操作。

上位机发送：

帧头	0x65	卡片 ID	校验字
----	------	-------	-----

卡片 ID：1 字节，需要选定的卡片 ID

模块回应成功：

帧头	0x65	卡片 ID	校验字
----	------	-------	-----

卡片 ID：1 字节，被选定的卡片 ID

模块回应失败：

帧头	0x9A	校验字
----	------	-----

### 5.9.3 SR 系列卡操作结束

功能：将当前操作的 SR 卡置入操作结束状态。此时如果要再次操作卡片，需要把卡片移出天线感应区域后重新进行初始化。

上位机发送：

帧头	0x67	校验字
----	------	-----



模块回应成功:

帧头	0x67	校验字
----	------	-----

模块回应失败:

帧头	0x98	校验字
----	------	-----

## 5.9.4 SR176 读卡

功能: SR176 卡读数据块。

上位机发送:

帧头	0x68	块号	块数	校验字
----	------	----	----	-----

块号: 1 字节, 需要读的数据块首地址

块数: 1 字节, 需要读取的块数,

模块回应成功:

帧头	0x68	数据	校验字
----	------	----	-----

数据: 2 字节 \* 读取的块数, 读出来的数据

模块回应失败:

帧头	0x97	校验字
----	------	-----

## 5.9.5 SR176 写卡

功能: SR176 卡写数据块。写卡后读卡模块会读出数据进行比较, 如果不相等则报错。

上位机发送:

帧头	0x69	块号	块数	数据	校验字
----	------	----	----	----	-----

块号: 1 字节, 需要写的数据块首地址

块数: 1 字节, 需要写入的块数

数据: 2 字节 \* 块数, 需要写入的数据

模块回应成功:

帧头	0x69	校验字
----	------	-----

模块回应失败:

帧头	0x96	校验字
----	------	-----

## 5.9.6 SR176 块锁定

功能: SR176 卡写数据锁定寄存器。写操作后会验证锁定结果。



上位机发送:

帧头	0x6A	锁定值	校验字
----	------	-----	-----

锁定值: 1 字节, 需要写入的锁定寄存器值

模块回应成功:

帧头	0x6A	校验字
----	------	-----

模块回应失败:

帧头	0x95	校验字
----	------	-----





## 5.10 SRI512/1K/2K/4K 卡命令

### 5.10.1 SR 系列卡单通道初始化

请参照：[SR 系列卡单通道初始化](#)

### 5.10.2 SRI 系列卡 16 通道初始化

功能：SRI 系列卡（指：SRI512/SRI1K/SRI2K/SRI4K/SRIX4K，下同）16 通道初始化。

上位机发送：

帧头	0x64	校验字
----	------	-----

模块回应成功：

帧头	0x64	状态	卡片 ID	校验字
----	------	----	-------	-----

状态：16 字节，通道 0 到通道 15 的执行结果，

0x00：本通道成功；0xE8：本通道冲突；0xFF：本通道无卡

卡片 ID：16 字节，16 个通道的卡片 ID，当前通道的执行结果为成功时 ID 才有效

模块回应失败：

帧头	0x9B	校验字
----	------	-----

### 5.10.3 SR 系列卡选定卡

请参照：[SR 系列卡选定卡](#)

### 5.10.4 SRI 系列卡回到初始化状态

功能：设定一张已经被选定的卡回到初始化状态。

上位机发送：

帧头	0x66	校验字
----	------	-----

模块回应成功：

帧头	0x66	校验字
----	------	-----

模块回应失败：

帧头	0x99	校验字
----	------	-----



## 5.10.5 SR 系列卡操作结束

请参照：[SR 系列卡操作结束](#)

## 5.10.6 SRI 系列卡读卡

功能：SRI 系列卡读数据块。

上位机发送：

帧头	0x6B	块号	块数	校验字
----	------	----	----	-----

块号：1 字节，需要读的数据块首地址

块数：1 字节，需要读取的块数

模块回应成功：

帧头	0x6B	数据	校验字
----	------	----	-----

数据：4 字节 \* 块数，读出来的数据

模块回应失败：

帧头	0x94	校验字
----	------	-----

## 5.10.7 SRI 系列卡写卡

功能：SRI 系列卡写数据块。卡后读卡模块会读出数据进行比较，如果不相等则报错。

上位机发送：

帧头	0x6C	起始块号	块数	数据	校验字
----	------	------	----	----	-----

起始块号：1 字节，需要写的数据块首地址

块数：1 字节，需要写入的块数

数据：4 字节 \* 块数，需要写入的数据

模块回应成功：

帧头	0x6C	校验字
----	------	-----

模块回应失败：

帧头	0x93	校验字
----	------	-----

## 5.10.8 SRI 系列卡块锁定

功能：SRI 系列卡写数据锁定寄存器。写操作后会验证锁定结果。



上位机发送:

帧头	0x6D	锁定值	校验字
----	------	-----	-----

锁定值: 1 字节, 需要写入的锁定寄存器值

模块回应成功:

帧头	0x6D	校验字
----	------	-----

模块回应失败:

帧头	0x92	校验字
----	------	-----

### 5.10.9 SRI 系列卡读 UID

功能: SRI 系列卡读取 UID。

上位机发送:

帧头	0x6E	校验字
----	------	-----

模块回应成功:

帧头	0x6E	UID	校验字
----	------	-----	-----

UID: 8 字节, 当前卡片的 UID

模块回应失败:

帧头	0x91	校验字
----	------	-----

### 5.10.10 SRIX 系列卡认证

功能: SRIX 系列卡认证。这是 SRIX 系列卡提供的防复制功能。

上位机发送:

帧头	0x6F	数据	校验字
----	------	----	-----

数据: 6 字节, 输入数据

模块回应成功:

帧头	0x6F	结果	校验字
----	------	----	-----

结果: 3 字节, 返回结果

模块回应失败:

帧头	0x90	校验字
----	------	-----



## 5.11 SAM 或 CPU 卡命令

### 5.11.1 SAM 或 CPU 卡复位

功能：SAM 卡复位，取得复位信息并按照复位信息设置读卡器的相关通讯参数。

上位机发送：

帧头	0x4D	卡座编号	速率代码	校验字
----	------	------	------	-----

卡座编号：1 字节，0 为大卡座，1 以后为 SAM 卡座编号，与板上的编号对应。

速率代码：1 字节，在 ISO7816 中，卡片默认的复位通讯速率是 9600bps，而有些卡片，如在国内大量使用的建设部一卡通的 SAM 卡，复位通讯速率为 38400bps，在此，用这个参数来指定复位时的通讯速率。取值范围，0：9600bps；1：19200bps；2：38400bps；3：55800bps；4：57600bps；5：115200bps；6：230400bps，其他值：保留。

模块回应成功：

帧头	0x4D	卡座编号	信息	校验字
----	------	------	----	-----

信息：SAM 卡复位信息，长度由卡片决定

模块回应失败：

帧头	0xB2	校验字
----	------	-----

示例：

发送：0x0006 00 4D 01 00 4A

返回：0x00 15 01 4D 01 3B 6C 00 02 43 21 86 38 07 54 42 00 16 0E 5A 2F AD

### 5.11.2 设定 SAM 或 CPU 卡复位后通讯速率

功能：在 SAM 卡复位后使用 PPS 设定改变通讯速率。

上位机发送：

帧头	0x4E	卡座编号	速率代码	校验字
----	------	------	------	-----

卡座编号：1 字节，取值 0-4，对应板上的 SAM1-SAM4。

速率代码：1 字节，SAM 卡通讯速率代码，0：9600bps；1：19200bps；2：38400bps；3：55800bps；4：57600bps；5：115200bps；6：230400bps，其他值：保留。

模块回应成功：

帧头	0x4E	卡座编号	校验字
----	------	------	-----

模块回应失败：



帧头	0xB1	校验字
----	------	-----

示例:

发送: 0x0006 00 4E 01 05 4C

返回: 0x00 05 01 4E 01 4B

### 5.11.3 发送 APDU 到 SAM 或 CPU 卡

功能: 发送 APDU (COS 命令) 到 SAM 卡并取回结果。

上位机发送:

帧头	0x4F	卡座编号	APDU	校验字
----	------	------	------	-----

卡座编号: 1 字节, 取值 0-4, 对应板上的 SAM1-SAM4。

APDU: 要发送的 APDU

模块回应成功:

帧头	0x4F	卡座编号	回应	校验字
----	------	------	----	-----

回应: SAM 卡回应, 长度由具体命令决定

模块回应失败:

帧头	0xB0	校验字
----	------	-----

示例:

发送: 0x000A 00 4F 01 00 84 00 00 08 C8

返回: 0x00 0F 01 4F 01 58 CE 18 13 43 E3 6B 10 90 00 96

### 5.11.4 弹出 CPU 卡

功能: 将大卡座中的卡片下电并弹出。

上位机发送:

帧头	0x4C	卡座编号	校验字
----	------	------	-----

卡座编号: 1 字节, 取值 0-4, 对应板上的 SAM1-SAM4。

模块回应成功:

帧头	0x4C	卡座编号	校验字
----	------	------	-----

回应: SAM 卡回应, 长度由具体命令决定

模块回应失败:

帧头	0xB3	校验字
----	------	-----

示例:



发送: 0x00 05 00 4C 00 49

返回: 0x00 05 00 4C 00 49



## 5.12 ISO15693 电子标签命令

### 5.12.1 ISO15693 inventory

功能：ISO15693 寻卡，如果成功则将其设定为“当前卡片”。如果开启了自动寻卡，那么此命令是取自动寻卡的结果，而不会在接收到命令后进行寻卡。

上位机发送：

帧头	0x5C	AFI	校验字
----	------	-----	-----

AFI：1 字节 AFI，只寻 AFI 相等的卡片

如果不想启用 AFI，请用如下的命令格式：

帧头	0x5C	校验字
----	------	-----

模块回应成功：

帧头	0x5C	DSFID	UID	校验字
----	------	-------	-----	-----

DSFID：1 字节，当前卡片的 DSFID

UID：8 字节，当前卡片的 UID，低字节在前

模块回应失败：

帧头	0xA3	校验字
----	------	-----

示例：

发送：0x0005 00 5C 00 59

返回：0x00 0D 01 5C 33 E3 DB CF 19 00 00 07 E0 6A

### 5.12.2 ISO15693 inventory 所有在天线区域内的卡

功能：ISO15693 寻卡，将所有可以寻到的卡的 UID 输出。执行此命令后，会将卡片置入休眠状态，已经输出 UID 的卡片如果没有移开并重新进入天线区域，再次执行此命令将不会输出卡片信息。

上位机发送：

帧头	0x7C	AFI	校验字
----	------	-----	-----

AFI：1 字节 AFI，只寻 AFI 相等的卡片

如果不想启用 AFI，请用如下的命令格式：

帧头	0x7C	校验字
----	------	-----

模块回应成功：



帧头	0x7C	N * (DSFID + UID)	校验字
----	------	-------------------	-----

N: 当前所有被发现的卡片数量

DSFID: 1 字节, 当前卡片的 DSFID

UID: 8 字节, 当前卡片的 UID, 低字节在前

模块回应失败:

帧头	0x83	校验字
----	------	-----

示例:

发送: 0x00 05 00 7C 00 79

返回: 0x00 16 01 7C 00 3B 8B FC A3 00 01 04 E0 00 3B A1 D9 12 00 01 04 E0 D5

### 5.12.3 ISO15693 stay quiet

功能: 设置当前卡片保持安静。这个命令仅仅对 Inventory 和 get system information 有效, 读写卡片等命令是基于地址的, 因此即使使用了本命令也可以读写。

上位机发送:

帧头	0x5D	校验字
----	------	-----

模块回应成功:

帧头	0x5D	校验字
----	------	-----

模块回应失败:

帧头	0xA2	校验字
----	------	-----

示例:

发送: 0x0004 00 5D 59

返回: 0x0004 01 5D 58

### 5.12.4 ISO15693 get system information

功能: 取当前卡片系统信息。

上位机发送:

帧头	0x5E	校验字
----	------	-----

模块回应成功:

帧头	0x5E	数据	校验字
----	------	----	-----

数据: 系统信息, 内容依靠于卡片本身的功能, 请参照卡片本身的数据手册

模块回应失败:





帧头	0xA1	校验字
----	------	-----

示例:

发送: 0x0004 00 5E 5A

返回: 0x00 12 01 5E 0F E3 DB CF 19 00 00 07 E0 33 23 3F 03 8B EC

### 5.12.5 ISO15693 reset to ready

功能: 唤醒一个保持安静的卡片。

上位机发送:

帧头	0x5F	UID	校验字
----	------	-----	-----

UID: 8 字节, 需要唤醒的卡片的 UID

模块回应成功:

帧头	0x5F	校验字
----	------	-----

模块回应失败:

帧头	0xA0	校验字
----	------	-----

示例:

发送: 0x00 0C 00 5F E3 DB CF 19 00 00 07 E0 5A

返回: 0x0004 01 5F 5A

### 5.12.6 ISO15693 read blocks

功能: 读当前卡片的数据。

上位机发送:

帧头	0x54	起始块号	读取块数	校验字
----	------	------	------	-----

起始块号: 1 字节, 需要读取的起始数据块逻辑编号

读取块数: 1 字节, 读取的块数, 一次最多 62 块

模块回应成功:

帧头	0x54	数据	校验字
----	------	----	-----

数据: 4 字节数据/块 \* 读取块数

模块回应失败:

帧头	0xAB	校验字
----	------	-----

示例:

发送: 0x00 06 00 54 08 02 58



返回: 0x00 0C 01 54 88 88 88 88 99 99 99 99 59

### 5.12.7 ISO15693 write blocks

功能: 写数据到当前卡片。

上位机发送:

帧头	0x55	起始块号	写入块数	数据	校验字
----	------	------	------	----	-----

起始块号: 1 字节, 需要写入起始数据块逻辑编号

写入块数: 1 字节, 写入的块数, 一次最多 62 块

数据: 4 字节数据/块 \* 写入块数

模块回应成功:

帧头	0x55	校验字
----	------	-----

模块回应失败:

帧头	0xAA	校验字
----	------	-----

示例:

发送: 0x00 0E 00 55 08 02 11 11 11 11 22 22 22 22 51

返回: 0x0004 01 55 50

### 5.12.8 ISO15693 lock block

功能: 锁定当前卡片的一块。

上位机发送:

帧头	0x56	块号	校验字
----	------	----	-----

块号: 1 字节, 需要锁定的数据块逻辑编号

模块回应成功:

帧头	0x56	校验字
----	------	-----

模块回应失败:

帧头	0xA9	校验字
----	------	-----

示例:

发送: 0x00 05 00 56 08 5B

返回: 0x00 04 01 56 53



### 5.12.9 ISO15693 write AFI

功能：写当前卡片的 AFI 。

上位机发送：

帧头	0x57	AFI	校验字
----	------	-----	-----

AFI: 1 字节, 要写入的 AFI 数值

模块回应成功：

帧头	0x57	校验字
----	------	-----

模块回应失败：

帧头	0xA8	校验字
----	------	-----

示例：

发送：0x00 05 00 57 08 5A

返回：0x00 04 01 57 52

### 5.12.10 ISO15693 lock AFI

功能：锁定当前卡片的 AFI。

上位机发送：

帧头	0x58	校验字
----	------	-----

模块回应成功：

帧头	0x58	校验字
----	------	-----

模块回应失败：

帧头	0xA7	校验字
----	------	-----

示例：

发送：0x00 04 00 58 5C

返回：0x0004 01 58 5D

### 5.12.11 ISO15693 write DSFID

功能：写当前卡片的 DSFID 。

上位机发送：

帧头	0x59	DSFID	校验字
----	------	-------	-----

DSFID: 1 字节, 需要写入的 DSFID 数值



模块回应成功:

帧头	0x59	校验字
----	------	-----

模块回应失败:

帧头	0xA6	校验字
----	------	-----

示例:

发送: 0x00 05 00 59 09 55

返回: 0x00 04 01 59 5C

### 5.12.12 ISO15693 lock DSFID

功能: 锁定当前卡片的 DSFID 。

上位机发送:

帧头	0x5A	校验字
----	------	-----

模块回应成功:

帧头	0x5A	校验字
----	------	-----

模块回应失败:

帧头	0xA5	校验字
----	------	-----

示例:

发送: 0x00 04 00 5A 5E

返回: 0x0004 01 5A 5F

### 5.12.13 ISO15693 get blocks security

功能: 取当前卡片的安全 (锁定) 信息。

上位机发送:

帧头	0x5B	起始块号	块数	校验字
----	------	------	----	-----

起始块号: 1 字节, 起始块号

块数: 1 字节, 块数

模块回应成功:

帧头	0x5B	数据	校验字
----	------	----	-----

数据: 字节数等于发送的命令中的“块数”, 数据块的锁定信息

模块回应失败:

帧头	0xA4	校验字
----	------	-----





## 5.13 I.CODE 1 电子标签命令

### 5.13.1 I.CODE1 寻卡

功能：寻 ICODE1 卡。

上位机发送：

帧头	0x80	校验字
----	------	-----

模块回应成功：

帧头	0x80	卡序号	校验字
----	------	-----	-----

卡序号：8 字节

模块回应失败：

帧头	0x7F	校验字
----	------	-----

示例：

发送：0x0004 00 80 84

返回：0x000C 01 80 5D 9A D4 0F 00 00 00 01 90

### 5.13.2 I.CODE 1 读卡

功能：读 ICODE1 卡。

上位机发送：

帧头	0x81	块号	校验字
----	------	----	-----

块号：1 字节，取值 0x00 到 0x0F

模块回应成功：

帧头	0x81	数据	校验字
----	------	----	-----

数据：4 字节

模块回应失败：

帧头	0x7E	校验字
----	------	-----

示例：

发送：0x0005 00 81 08 8C

返回：0x0008 01 81 FF FF FF FF 88



### 5.13.3 I.CODE 1 写卡

功能：写 ICODE1 卡。

上位机发送：

帧头	0x82	块号	数据	校验字
----	------	----	----	-----

块号：1 字节

数据：4 字节

模块回应成功：

帧头	0x82	校验字
----	------	-----

模块回应失败：

帧头	0x7D	校验字
----	------	-----

示例：

发送：0x00 09 00 82 08 12 34 56 78 8B

返回：0x0004 01 82 87

### 5.13.4 I.CODE 1 卡休眠

功能：ICODE1 卡休眠。

上位机发送：

帧头	0x83	校验字
----	------	-----

块号：1 字节

数据：4 字节

模块回应成功：

帧头	0x83	校验字
----	------	-----

模块回应失败：

帧头	0x7C	校验字
----	------	-----

示例：

发送：0x0004 00 83 87

返回：0x0004 01 83 86



## 5.14 二代身份证命令

### 5.14.1 身份证寻卡并获取 UID

功能：寻身份证卡，并获取身份证的 UID。

上位机发送：

帧头	0x71	校验字
----	------	-----

模块回应成功：

帧头	0x71	UID	校验字
----	------	-----	-----

UID：8 字节

模块回应失败：

帧头	0x8E	校验字
----	------	-----

示例：

发送：0x0004 00 71 75

返回：0x00 0C 01 71 00 97 9B A0 0A B3 9C 7C 89

### 5.14.2 身份证寻卡

功能：身份证寻卡。

上位机发送：

帧头	0x72	校验字
----	------	-----

模块回应成功：

帧头	0x72	信息	校验字
----	------	----	-----

信息：12 字节，卡复位信息，构成内容如下：

0x50（1 字节），PUPI（4 字节），应用数据（4 字节），协议信息（3 字节）

更多详细内容请参考 ISO14443-3 “ATQB Response” 部分。

模块回应失败：

帧头	0x8D	校验字
----	------	-----

示例：

发送：0x00 04 00 72 76

返回：0x0010 01 72 50 00 00 00 00 D1 03 86 07 00 80 90 70





### 5.14.3 发送 APDU 到身份证卡片

功能：给身份证卡片发送命令。在执行此命令前需要先执行身份证寻卡指令([0x72 指令](#)或 [0x71 指令](#))。

上位机发送：

帧头	0x73	APDU	校验字
----	------	------	-----

APDU：要发送的 APDU

模块回应成功：

帧头	0x73	回应	校验字
----	------	----	-----

回应：卡片回应，长度由具体的命令决定

模块回应失败：

帧头	0x8C	校验字
----	------	-----

示例：

发送：0x0009 00 73 00 84 00 00 08 F6

返回：0x000E 01 73 0C 2D 76 58 5C 19 AA A4 90 00 A8



## 5.15 NFC 功能命令

### 5.15.1 设备 NFC 工作模式设置

功能：设定读卡模块的 NFC 通讯模式。

上位机发送：

帧头	0xC0	模式	校验字
----	------	----	-----

模块回应成功：

帧头	0xC0	校验字
----	------	-----

模式：1 字节

= 0x00 被动发起者模式（默认）

= 0x01 被动目标模式

= 0x02 主动发起者模式

= 0x03 主动目标模式

= 0x04 模拟 NFC Tag 模式

模块回应失败：

帧头	0x3F	校验字
----	------	-----

示例：

发送：0x00 05 00 C0 01 C4

返回：0x00 04 01 C0 C5

### 5.15.2 发起者端命令

#### 5.15.2.1 发起者寻找目标设备命令

功能：发起者通过 RF 界面发送 NFC 属性请求。

上位机发送：

帧头	0xC1	校验字
----	------	-----

模块回应成功：

帧头	0xC1	数据	校验字
----	------	----	-----

数据：目标返回的信息

模块回应失败：



帧头	0x3E	校验字
----	------	-----

示例:

发送: 0x00 04 00 C1 C5

返回: 0x00 20 01 C1 31 32 33 34 35 31 32 33 34 35 00 00 00 0C 32 46 66 6D 01 01 11 03

02 00 13 04 01 96 03

### 5.15.2.2 发起者与目标设备数据交换命令

功能: 发起者通过 RF 界面与目标做数据交换。

上位机发送:

帧头	0xC2	数据	校验字
----	------	----	-----

数据: 要发出的数据

模块回应成功:

帧头	0xC2	数据	校验字
----	------	----	-----

数据: 目标返回的数据

模块回应失败:

帧头	0x3D	校验字
----	------	-----

示例:

发送: 0x00 0C 00 C2 11 22 33 44 55 66 77 88 46

返回: 0x00 08 01 C2 12 34 56 78 C3

## 5.15.3 目标端命令

### 5.15.3.1 目标缓存交换数据命令

功能: 通过通讯端口把需要交换的数据放入目标设备缓存, 等待响应发起者发出的数据交换命令。

上位机发送:

帧头	0xC3	数据	校验字
----	------	----	-----

数据: 缓存的需要交换的数据。

模块回应成功:

帧头	0xC3	校验字
----	------	-----

模块回应失败:

帧头	0x3C	校验字
----	------	-----

示例:



发送: 0x00 08 00 C3 12 34 56 78 C3

返回: 0x00 04 01 C3 C6

### 5.15.3.2 目标读取状态及数据交换结果

功能: 通过通讯宽口读取目标设备的 RF 部分通讯状态, 如果有数据交换完成, 返回收到的交换数据。

上位机发送:

帧头	0xC4	校验字
----	------	-----

模块回应成功:

帧头	0xC4	状态	数据	校验字
----	------	----	----	-----

状态: 0x00: 数据交换已经完成, 空闲状态, 后续可能跟随交换的数据;

0xFF: 数据交换尚未完成;

数据: 读回来的数据, 可能为空。

模块回应失败:

帧头	0x3B	校验字
----	------	-----

示例:

发送: 0x00 04 00 C4 C0

返回: 0x00 0D 01 C4 00 11 22 33 44 55 66 77 88 40



## 5.16 NFC Tag 功能命令

### 5.16.1 综述

具有 NFC 目标功能的读卡模块，可以配置为 NFC Tag 模式，此时读卡模块就相当于一个 NFC Tag，可以使用 NFC Tag 读写命令对其进行读写。针对“目标”的命令是使用通讯口对配置为 NFC Tag 的读卡模块进行配置的命令。针对“发起者”的命令是通过对读卡模块发命令通过 RF 界面操作 NFC Tag 的。

NFC Tag 内部 FLASH 空间为 128 字节，组织结构为 4 字节/块×32 块，其中 0, 1 块和第 2 块前 2 字节为 UID 数据，是只读的。

### 5.16.2 发起者端命令

#### 5.16.2.1 发起者读取模拟 NFC Tag 数据

功能：发起者（读卡模块）通过 RF 界面读取模拟 NFC Tag 数据，指定起始数据页，每次读取连续 4 页数据，共 16 字节。如指定页到结束页不够 4 页，缺少页从第 0 页开始补齐，与 Ultralight 卡片操作方法类似。

上位机发送：

帧头	0xC8	起始块号	校验字
----	------	------	-----

起始块号：1 字节（取值范围 0x00 到 0x1F）

模块回应成功：

帧头	0xC8	数据	校验字
----	------	----	-----

数据：16 字节数据

模块回应失败：

帧头	0x37	校验字
----	------	-----

示例：

发送：0x00 05 00 C8 07 CA

返回：0x00 14 01 C8 07 07 07 07 08 08 08 08 FF FF FF FF FF FF FF DD

#### 5.16.2.2 发起者写入模拟 NFC Tag 数据

功能：发起者（读卡模块）通过 RF 界面写入模拟 NFC Tag 数据，指定写入数据页，每次写一页（4 字节）数据，与操作 Ultralight 卡片方法类似。



上位机发送:

帧头	0xC9	块号	数据	校验字
----	------	----	----	-----

块号: 1 字节 (取值范围 0x02 到 0x1F)

数据: 4 字节写入数据

模块回应成功:

帧头	0xC9	校验字
----	------	-----

模块回应失败:

帧头	0x36	校验字
----	------	-----

示例:

发送: 0x00 09 00 C9 07 07 07 07 07 C7

返回: 0x00 04 01 C9 CC

### 5.16.3 目标端命令

#### 5.16.3.1 目标读取模拟 NFC Tag 数据

功能: 通过通讯端口读取模拟 NFC Tag 数据。

上位机发送:

帧头	0xCA	起始块号	读取块数	校验字
----	------	------	------	-----

起始块号: 1 字节 (取值范围 0x00 到 0x1F)

读取块数: 1 字节

模块回应成功:

帧头	0xCA	数据	校验字
----	------	----	-----

数据: 长度为读取块数×4 字节

模块回应失败:

帧头	0x35	校验字
----	------	-----

示例:

发送: 0x00 06 00 CA 07 01 CA

返回: 0x00 08 01 CA 07 07 07 07 C3

#### 5.16.3.2 目标写入模拟 NFC Tag 数据

功能: 通过通讯端口写入模拟 NFC Tag 数据, 用于对 NFC Tag 的初始化。

上位机发送:

帧头	0xCB	起始块号	写入块数	写入数据	校验字
----	------	------	------	------	-----



起始块号：1 字节（取值范围 0x02 到 0x1F）

写入块数：1 字节

写入数据：块数×4 字节写入数据

模块回应成功：

帧头	0xCB	校验字
----	------	-----

模块回应失败：

帧头	0x34	校验字
----	------	-----

示例：

发送：0x00 0E 00 CB 07 02 07 07 07 07 08 08 08 08 C0

返回：0x00 04 01 CB CE

### 5.16.3.3 目标写入模拟 NFC Tag UID

功能：通过通讯端口写入模拟 NFC Tag 7 字节 UID，用于 NFC Tag 的初始化。

上位机发送：

帧头	0xCC	UID	校验字
----	------	-----	-----

UID：7 字节卡片 UID

模块回应成功：

帧头	0xCC	校验字
----	------	-----

模块回应失败：

帧头	0x33	校验字
----	------	-----

示例：

发送：0x00 0B 00 CC 01 02 03 04 05 06 07 C7

返回：0x00 04 01 CC C9



## 5.17 FeliCa 卡片命令

### 5.17.1 收发 FeliCa 卡片命令

功能：收发 FeliCa 卡片的命令。

上位机发送：

帧头	0x2F	发送数据	校验字
----	------	------	-----

发送数据：全面依据 FeliCa 的卡片规范

模块回应成功：

帧头	0x2F	返回数据	校验字
----	------	------	-----

返回数据：全面依据 FeliCa 的卡片规范

模块回应失败：

帧头	0xD0	校验字
----	------	-----

示例：

发送：0x00 0A 00 2F 06 00 FF FF 01 00 22

返回：0x00 18 01 2F 14 01 01 2E 30 C3 76 94 48 1F 03 35 0B 82 82 44 83 FF 12 FC A1





## 5.18 ISO18000-3M3 命令

### 5.18.1 错误代码

执行命令若发生错误，则在返回结果中返回错误代码，错误代码含义见下表

Error-Coder Support	Error Code (二进制)	Error-Code Name
Error specific	0000 0000	Other error
	0000 0011	Memory overrun
	0000 0100	Memory locked
	0000 1011	Insufficient power
Non specific	0000 1111	Non specific error
Reader operation Error	1111 1111	No tag in field

### 5.18.2 ActivateCard

功能：获取单张标签的 Handle

上位机发送：

帧头	0x85	0x00	校验字
----	------	------	-----

模块回应成功：

帧头	0x85	0x00	Handle	校验字
----	------	------	--------	-----

Handle: 标签的句柄，后续操作要用到。

模块回应失败：

帧头	0x7A	0x00	ErrorCode	校验字
----	------	------	-----------	-----

ErrorCode: 读卡器或卡片返回的错误信息，详见返回错误编码表。

示例：

发送：00 05 00 85 00 80

返回：00 07 01 85 00 CF 16 5A

### 5.18.3 ActivateCards

功能：在多个标签中识别一个并获取 handle

上位机发送：



帧头	0x85	0x01	Q	校验字
----	------	------	---	-----

Q: 1 字节, 用于指定时间槽数量 ( $2^Q$ , 即最多寻  $2^Q$  张卡), 取值范围 0x00~0x05 (最大 32 张卡), Q 值越大, 越耗时。

模块回应成功:

帧头	0x85	0x01	Handle	校验字
----	------	------	--------	-----

模块回应失败:

帧头	0x7A	0x01	ErrorCode	校验字
----	------	------	-----------	-----

ErrorCode: 读卡器或卡片返回的错误信息, 详见返回错误编码表。

示例:

发送: 00 06 00 85 01 01 83

返回: 00 07 01 85 01 C9 1C 57

## 5.18.4 Read

功能: 读取标签数据

上位机发送:

帧头	0x85	0x02	ReadInfo	校验字
----	------	------	----------	-----

ReadInfo: 6~9 字节读取参数

ReadInfo	Byte1	MemBank	Memory bank 00h: Reserved 01h: UII 02h: TID 03h: User
	Byte2	WordPtrlength	Length of wordpointer 00h: 8 bit 01h: 16 bit 02h: 24 bit 03h: 32 bit
	Byte3( <a href="#">Note1</a> )	WordPtr	Starting address pointer (高字节在前)
	Byte4	WordCount	The number of 16-bit words to be read
	Byte5_6	handle	2 bytes handle

**Note1:** WordPtr 的字节数是根据 WordPtrlength 参数决定的。如果 WordPtrlength 等于 02h, 则 WordPtr 占用 2 个字节, 之后的参数定义后移一个字节。

模块回应成功:

帧头	0x85	0x02	Data	校验字
----	------	------	------	-----



DATA: n words(16 bits) + handle, 读取的数据

模块回应失败:

帧头	0x7A	0x02	ErrorCode	校验字
----	------	------	-----------	-----

ErrorCode: 读卡器或卡片返回的错误信息, 详见返回错误编码表。

示例:

发送: 00 0B 00 85 02 00 00 00 04 A2 D4 FE

返回: 00 0F 01 85 02 00 00 00 00 00 00 00 00 A2 D4 FF

## 5.18.5 Write

功能: 将 1 个 word(16bits)型数据写入标签指定位置

上位机发送:

帧头	0x85	0x03	WriteInfo	校验字
----	------	------	-----------	-----

WriteInfo: 7~10 字节写入参数和数据

WriteInfo	Byte1	MemBank	Memory bank 00h: Reserved 01h: UII 02h: TID 03h: User
	Byte2	WordPtrlength	Length of wordpointer 00h: 8 bit 01h: 16 bit 02h: 24 bit 03h: 32 bit
	Byte3( <a href="#">Note1</a> )	WordPtr	Starting address pointer (高字节在前)
	Byte4_5	Data	A 16-bit word to be written
	Byte6_7	handle	2 bytes handle

模块回应成功:

帧头	0x85	0x03	Handle	校验字
----	------	------	--------	-----

模块回应失败:

帧头	0x7A	0x03	ErrorCode	校验字
----	------	------	-----------	-----

ErrorCode: 读卡器或卡片返回的错误信息, 详见返回错误编码表。

示例:

发送: 00 0C 00 85 03 00 00 00 11 11 A2 D4 FC



返回: 00 07 01 85 03 A2 D4 F6

发送: 00 0B 00 85 02 00 00 00 04 A2 D4 FE

返回: 00 0F 01 85 02 11 11 00 00 00 00 00 00 A2 D4 FF

## 5.18.6 KILL

功能: 终止(可选)或重新启用标签, 详见 Information technology — Radio frequency identification for item management —Part 3:Parameters for air interface communications at 13,56 MHz

上位机发送:

帧头	0x85	0x04	KillInfo	校验字
----	------	------	----------	-----

KillInfo: 7 字节参数

KillInfo	Byte1_4	Kill password	The kill password is a 32-bit value stored in Reserved memory 00h to 1Fh, MSB first.
	Byte5	Recom	Recommissioning bits.
	Byte6_7	handle	2 bytes handle

模块回应成功:

帧头	0x85	0x04	Handles	校验字
----	------	------	---------	-----

模块回应失败:

帧头	0x7A	0x04	ErrorCode	校验字
----	------	------	-----------	-----

ErrorCode: 读卡器或卡片返回的错误信息, 详见返回错误编码表。

示例:

发送: 00 0C 00 85 04 11 11 00 00 04 12 9E 05

返回: 00 07 01 85 04 12 9E 0B

## 5.18.7 Lock

功能: 按照参数设定锁定标签

上位机发送:

帧头	0x85	0x05	LockInfo	校验字
----	------	------	----------	-----

LockInfo: 6 字节参数



WriteInfo	Byte1_2	Mask	Kill pwd	Bit15	Skip/write
				Bit14	Skip/write
			Access pwd	Bit13	Skip/write
				Bit12	Skip/write
			UII memory	Bit11	Skip/write
				Bit10	Skip/write
			TID memory	Bit9	Skip/write
	Bit8	Skip/write			
	User memory	Bit7	Skip/write		
		Bit6	Skip/write		
	RFU	Bit5_0	RFU		
	Byte3_4	Action	Kill pwd	Bit15	Pwd read/write
				Bit14	Perma lock
			Access pwd	Bit13	Pwd read/write
				Bit12	Perma lock
			UII memory	Bit11	Pwd write
				Bit10	Perma lock
			TID memory	Bit9	Pwd write
Bit8	Perma lock				
User memory	Bit7	Pwd write			
	Bit6	Perma lock			
RFU	Bit5_0	RFU			
Byte5_6	handle	2 bytes handle			

Pwd write	Perma lock	Description
0	0	Associated memory bank is writeable from either the open or secured states
0	1	Associated memory bank is permanently writeable from either the open or secured states and may never be locked.
1	0	Associated memory bank is writeable from the secured state but not from the open state.
1	1	Associated memory bank is not writeable from any state.
Pwd read/write	Perma lock	Description
0	0	Associated password location is readable and writeable from either the open or secured states.
0	1	Associated password location is permanently readable and writeable from open or secured or secured states and may never be locked.
1	0	Associated password location is readable and



		writeable from the secured state but not from the open state.
1	1	Associated password location is not readable or writeable from any state.

模块回应成功:

帧头	0x85	0x05	Handles	校验字
----	------	------	---------	-----

模块回应失败:

帧头	0x7A	0x05	ErrorCode	校验字
----	------	------	-----------	-----

ErrorCode: 读卡器或卡片返回的错误信息, 详见返回错误编码表。

示例:

发送: 00 0B 00 85 05 30 00 20 00 12 9E 17

返回: 00 07 01 85 05 12 9E 0A

## 5.18.8 Access

功能: 获取权限

上位机发送:

帧头	0x85	0x06	AccessInfo	校验字
----	------	------	------------	-----

KillInfo: 6 字节参数

KillInfo	Byte1_4	Access password	The access password is a 32-bit value stored in Reserved memory 20h to 3Fh, MSB first.
	Byte5_6	handle	2 bytes handle

模块回应成功:

帧头	0x85	0x06	Handles	校验字
----	------	------	---------	-----

模块回应失败:

帧头	0x 7A	0x06	ErrorCode	校验字
----	-------	------	-----------	-----

ErrorCode: 读卡器或卡片返回的错误信息, 详见返回错误编码表。

示例:

发送: 00 0B 00 85 06 00 00 00 00 12 9E 04

返回: 00 07 01 85 06 12 9E 09



## 5.18.9 BlockWrite

功能：块写入

上位机发送：

帧头	0x85	0x07	WriteInfo	校验字
----	------	------	-----------	-----

WriteInfo：写入参数和数据

WriteInfo	Byte1	MemBank	Memory bank 00h: Reserved 01h: UII 02h: TID 03h: User
	Byte2	WordPtrlength	Length of wordpointer 00h: 8 bit 01h: 16 bit 02h: 24 bit 03h: 32 bit
	Byte3( <a href="#">Note1</a> )	WordPtr	Starting address pointer
	Byte4	WordCount	Number of words(16-bit) to write (对于I.CODE ILT-m, WordCount = 01h to 02h)
	Byte5_n	Data	Data to be written
	Byten+1_n+2	handle	2 bytes handle

模块回应成功：

帧头	0x85	0x07	Handles	校验字
----	------	------	---------	-----

模块回应失败：

帧头	0x 7A	0x07	ErrorCode	校验字
----	-------	------	-----------	-----

ErrorCode：读卡器或卡片返回的错误信息，详见返回错误编码表。

示例：

发送：00 0F 00 85 07 00 00 02 02 11 22 33 44 12 9E 45

返回：00 07 01 85 07 12 9E 08

发送：00 0B 00 85 02 00 00 00 04 12 9E 04

返回：00 0F 01 85 02 11 11 00 00 11 22 33 44 12 9E 41



## 5.18.10 BlockErase

功能：块擦除

上位机发送：

帧头	0x85	0x08	EraseInfo	校验字
----	------	------	-----------	-----

EraseInfo: 擦除参数和数据

EraseInfo	Byte1	MemBank	Memory bank 00h: Reserved 01h: UII 02h: TID 03h: User
	Byte2	WordPtrlength	Length of wordpointer 00h: 8 bit 01h: 16 bit 02h: 24 bit 03h: 32 bit
	Byte3( <a href="#">Note1</a> )	WordPtr	Starting address pointer
	Byte4	WordCount	Number of words(16-bit) to write (对于I.CODE ILT-m, WordCount = 01h to 02h)
	Byte5_6	handle	2 bytes handle

模块回应成功：

帧头	0x85	0x08	Handles	校验字
----	------	------	---------	-----

模块回应失败：

帧头	0x 7A	0x08	ErrorCode	校验字
----	-------	------	-----------	-----

ErrorCode: 读卡器或卡片返回的错误信息，详见返回错误编码表。

## 5.18.11 Inventory

功能：识别多个标签

上位机发送：

帧头	0x85	0x0A	Q	校验字
----	------	------	---	-----

Q: 1 字节，用于指定时间槽数量 ( $2^Q$ ，即最多寻  $2^Q$  张卡)，取值范围 0x00~0x05 (最大 32 张卡)。

模块回应成功：





帧头	0x85	0x0A	AcknowLedge	校验字
----	------	------	-------------	-----

AcknowLedge: 1 字节寻卡个数 n + n 字节卡片信息长度+卡片信息长度数据

模块回应失败:

帧头	0x 7A	0x0A	ErrorCode	校验字
----	-------	------	-----------	-----

ErrorCode: 读卡器或卡片返回的错误信息, 详见返回错误编码表。

示例:

发送: 00 06 00 85 0A 04 8D

返回: 00 3A 01 85 0A 04 0C 0C 0C 0C 00 00 00 00 00 00 40 00 FA B1 F5 23 00 00 00 00  
00 00 40 00 FA B1 4C AB 00 00 00 00 00 00 40 00 FA B1 39 46 00 00 00 00 00 00  
40 00 FA B1 58 92 34

(示例颜色说明: 寻到 4 张标签, 标签的信息长度均为 0C 0C 0C 0C h, 第一标签信息 00 00 00 00 00 00 40 00 FA B1 F5 23……)

### 5.18.12 ReqRnCard

功能: 获取标签的 handle, 配合 [Inventory](#) 函数使用

上位机发送:

帧头	0x85	0x0B	Info	校验字
----	------	------	------	-----

Info: 1 字节长度 n + n 字节卡片信息

模块回应成功:

帧头	0x85	0x0B	Handles	校验字
----	------	------	---------	-----

模块回应失败:

帧头	0x 7A	0x0B	ErrorCode	校验字
----	-------	------	-----------	-----

ErrorCode: 读卡器或卡片返回的错误信息, 详见返回错误编码表。

示例:

发送: 00 12 00 85 0B 0C 00 00 00 00 00 00 40 00 FA B1 F5 23 4D

返回: 00 06 01 85 8A 32 3A

发送: 00 0B 00 85 02 00 00 00 04 8A 32 30

返回: 00 0E 01 85 33 33 33 33 00 00 00 00 8A 32 32

----- End of file -----