

NFC & MIFARE & ISO14443A/B & ISO7816 & ISO15693 标准型 PC/SC 接口读卡器

# PC/SC 标准型读卡器 通用技术手册

---

(Revision 2.70)

北京金木雨电子有限公司

2022/8/18



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

文件修改记录.....	5
1 简介.....	6
2 驱动安装和系统识别.....	6
2.1 Contactless Reader.....	6
2.2 SAM Reader.....	7
2.3 Device Controller.....	7
3 软件识别.....	8
4 PICC 接口描述.....	9
4.1 ATR 的生成.....	9
4.2 ATR 信息格式（适用于 ISO14443-3 PICC）.....	9
4.3 ATR 信息格式（适用于 ISO14443-4 PICC）.....	10
5 Contactless Reader 指令.....	11
5.1 操作准备及返回的状态信息.....	11
5.2 获取数据（GET DATA）.....	11
5.3 ISO14443A/B CPU 卡命令.....	12
5.4 直接 RF 命令（DIRECT TRANSACTION）.....	12
5.5 MIFARE Classic/UltraLight/ISO15693 存储卡命令（T=CL 模拟）.....	13
5.5.1 加载认证密钥（LOAD AUTHENTICATION KEYS）.....	13
5.5.2 认证（AUTHENTICATION）.....	14
5.5.3 读二进制块（READ BINARY BLOCKS）.....	14
5.5.4 更新二进制块（UPDATE BINARY BLOCKS）.....	15
5.5.5 钱包块操作（VALUE BLOCK OPERATION）.....	15
5.5.6 读取值块（READ VALUE BLOCK）.....	15
5.5.7 备份值块（RESTORE VALUE BLOCK）.....	16
5.5.8 ISO15693 Write AFI.....	16
5.5.9 ISO15693 LockAFI.....	16
5.5.10 ISO15693 WriteDSFID.....	17
5.5.11 ISO15693 LockDSFID.....	17
5.5.12 ISO15693 GetSysteminfo.....	17
5.5.13 ISO15693 Get Blocks Security.....	18
5.5.14 ISO15693 Lock Block.....	18
5.6 非接触式智能卡操作流程.....	19
5.6.1 ISO14443-4 卡操作.....	19
5.6.2 MIFARE 1K/4K 卡操作.....	19



5.6.3	MIFAREUltra Light 卡操作 .....	20
5.6.4	ISO15693 卡操作.....	21
6	SAM Reader 指令.....	24
7	DEVICE 操作指令 .....	24
7.1	切换当前操作智能卡 .....	24
7.2	复位 SAM 卡 .....	24
7.3	设置 SAM 卡上电复位参数.....	25
7.4	读取 SAM 卡上电复位参数.....	26
7.5	设置卡片操作方式 .....	27
7.6	初始化 RTC 时间.....	28
7.7	读 RTC 时间.....	28
7.8	显示时间 .....	29
7.9	显示日期 .....	29
7.10	设定日期显示格式 .....	29
7.11	设定 LCD 显示字体类型 .....	30
7.12	读取 LCD 显示字体类型 .....	31
7.13	LCD 点阵设定 .....	31
7.14	LCD 显示文本信息 .....	32
7.15	LCD 任意位置显示文本信息 .....	33
7.16	LCD 显示图片 .....	34
7.17	LCD 清除显示 .....	35
7.18	LCD 设定开机画面 .....	36
7.19	LCD 设定待机画面 .....	37
7.20	LCD 背光控制 .....	40
7.21	LCD 显示 Flash 中存储画面.....	41
7.22	读片外 Flash .....	42
7.23	写片外 Flash .....	43
7.24	获取产品序列号 .....	43
7.25	获取硬件版本和版本号 .....	44
7.26	LED 灯控制 .....	44
7.27	蜂鸣器控制 .....	45
7.28	卡片加密方法设置 .....	46
7.29	恢复出厂默认值（系统重新启动） .....	46
7.30	系统重新启动 .....	46
7.31	设置快速启动 .....	47
7.32	读取快速启动设置状态 .....	47
7.33	获取 PICC 操作信息指示参数 .....	48



---

7.34	设置 PICC 操作信息指示参数 .....	48
------	------------------------	----



# 文件修改记录

日期	版本号	修改内容
2016.06.14	R2.30	增加 Device 操作部分 修改了文档例程错误
2016.06.16	R2.31	修改目录描述, 去掉 MR801/811 说明 背光灯亮时间指令信息注释的修改 支持字体指令的信息添加
2016.07.01	R2.32	增加了简介部分中对 PC/SC 的说明 增加设置卡片功能中的注释
2016.07.04	R2.33	重新整理了章节 增加了部分说明
2017.08.18	R2.35	增加了章节说明与 APDU 的对象 修改了符号和英文拼写错误
2018.03.27	R2.37	增加章节 7.22 设置 PICC 操作信息指示参数章节 增加章节 7.23 获取 PICC 操作信息指示参数章节
2018.05.05	R2.38	增加章节 5.5, 直接 RF 命令传输
2018.07.24	R2.39	增加对 MR881 的命令说明
2018.07.26	R2.40	直接发送命令做了扩充
2018.08.01	R2.51	修改了文档例程和指令说明错误
2018.10.10	R2.52	增加快速启动的设置和读取
2020.08.28	R2.53	整理文档格式和示例, 修改描述性错误 合并两条擦除 LCD 指令, 在指令中根据 LCD 分辨率进行区分
2020.09.16	R2.54	修正文档格式
2021.01.04	R2.60	新增 SAM 卡通信波特率复位波特率等参数设定功能
2021.01.12	R2.61	新增自动设置 SAM 卡通信波特率指令介绍
2021.02.01	R2.63	修正文档图表格式 修改文档描述性错误
2022.08.18	R2.70	添加对 ISO15693 的支持



# 1 简介

PC/SC 规范由微软公司与世界其它著名的智能卡厂商组成的 PC/SC 工作组提出的。PC/SC 规范是一个基于 WINDOWS 平台的一个标准用户接口 (API)，提供了一个从个人电脑 (Personal Computer) 到智能卡 (SmartCard) 的整合环境。PC/SC 体系由三个主要部件组成，分别规定的操作系统厂商、读写器 (IFD) 厂商、智能卡 (ICC) 厂商的职责。

本公司读写器作为 PC 端对智能卡进行操作的中间环节的硬件部分，提供了简单、高效、可靠的通讯方式。安装 CCID 驱动后，将读写器 USB 连入 PC 端，放入卡片后即可完成智能卡与 PC 端的硬件连接。客户只需通过对基于 WINDOWS 平台的 PC/SC 规范的 API 函数进行调用即可完成对智能卡的操作。

PC/SC 的 API 函数的使用方法可参考我司提供的《PC/SC 函数使用说明》。

本公司的读写器主要负责与 PC 端的 PC/SC 接口的数据通信和对智能卡的读写等操作。读写器支持 MIFARE 1K/4K 卡、MIFARE UltraLight 卡、ISO14443-3 的 CPU 卡和 ISO14443-4 的 CPU 卡。使用者只需按照下面提供的通讯协议对卡片进行操作即可。我司也提供了测试软件来完成 PC 对智能卡的操作。

本手册适用于 MR791、MR7911、MR801、MR811、MR8111、MR881 等型号读写卡器。

本系列读写卡器完全符合 PC/SC 标准，可以与现有其他厂家的 PC/SC 读卡器相兼容，采用标准的 Microsoft CCID 驱动来简化驱动安装程序。

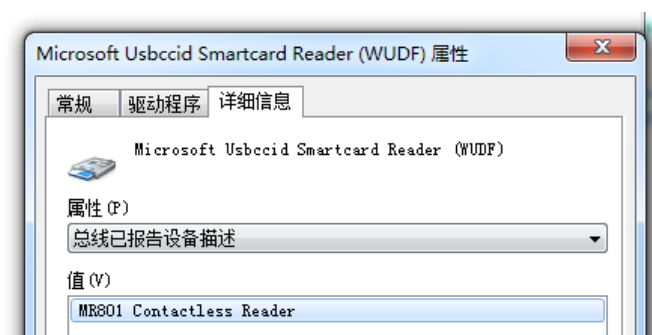
## 2 驱动安装和系统识别

PC/SC 读写卡器设备第一次通过 USB 接入 PC 机后，若系统中没有安装 CCID 驱动程序，会提示安装驱动程序，请安装光盘上带的驱动程序。安装目录：CD-ROM: \Chinese\桌面 IC 卡读写器\标准 USB PCSC 接口 IC 卡读写器\CCID Driver。

安装驱动后，设备管理器中识别到的读卡器设备。

### 2.1 Contactless Reader

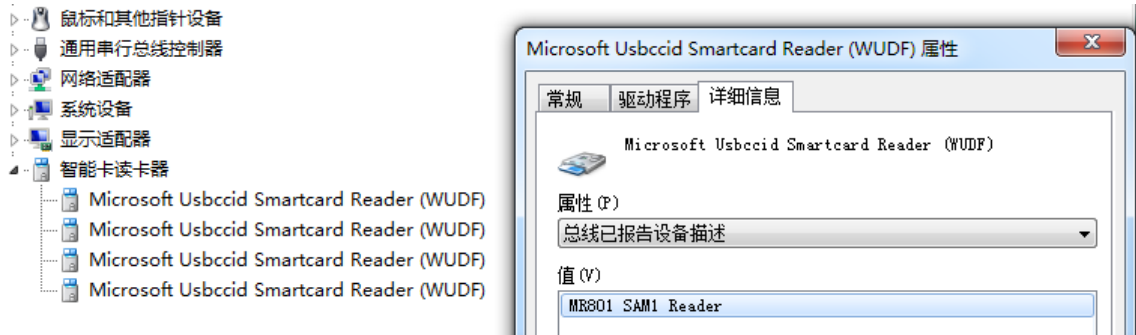
处理外部天线区域非接触式智能卡和存储卡。





## 2.2 SAM Reader

处理读卡器内部 SAM 卡槽中的智能卡。



## 2.3 Device Controller

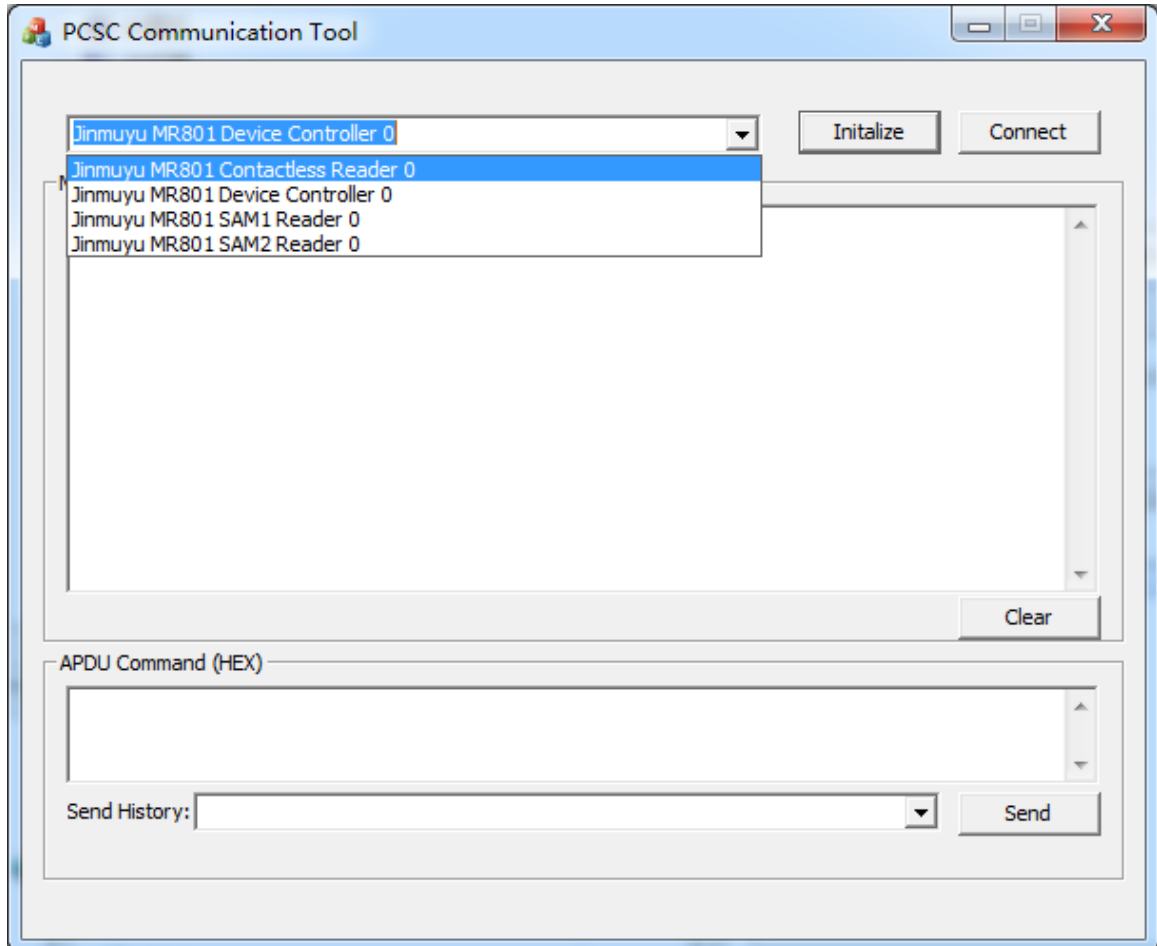
处理对读卡器的操作，例如 LED、Flash 等。





### 3 软件识别

以 MR801 为例，将读写器连接至 PC，打开 PCSC Communication Tool 工具软件，点击 Initialize(初始化)，识别到 4 个读卡器设备。



这 4 个读卡器设备解释如下：

Jinmuyu MR801 Contactless Reader 0: MR801 IC 卡读写器的非接触卡通道

Jinmuyu MR801 Device Controller 0: MR801 IC 卡读写器的设备控制通道

Jinmuyu MR801 SAM1 Reader 0: MR801 IC 卡读写器的 SAM1 通道

Jinmuyu MR801 SAM2 Reader 0: MR801 IC 卡读写器的 SAM2 通道





## 4 PICC 接口描述

### 4.1 ATR 的生成

当一个符合 ISO14443 标准的 PICC 放入读卡器的读卡区域，读写器检测到 PICC 后，一个 ATR 会被准备好，当 PC 对此通道进行连接时，读写器会把此 ATR 发送到 PC 来识别 PICC。

### 4.2 ATR 信息格式（适用于 ISO14443-3 PICC）

字节	值(Hex)	标记	说明
0	3B	初始字符	
1	8N	T0	高半字节 8 表示：后续不存在 TA1、TB1 和 TC1，只存在 TD1 低半字节 N 指出历史字符的个数（HisByte0 – HistByteN-1）
2	80	TD1	高半字节 8 表示：后续不存在 TA2、TB2 和 TC2，只存在 TD2 低半字节 0 表示协议类型为 T=0
3	01	TD2	高半字节 0 表示后续不存在 TA3、TB3、TC3 和 TD3。 低半字节 1 表示协议类型为 T=1
4 至 3+N	80	T1	类别指示字节，80 表示在可选的 COMPACT-TLV 数据对象中或许存在一个状态标识符
	4F	Tk	应用标识符存在标识
	0C		长度
	RID		注册应用供应商标识 (RID) # A0 00 00 03 06
	SS		标准字节
	C0...C1		卡片名称字节
	00 00 00 00	RFU	RFU # 00 00 00 00
4+N	UU	TCK	T0 至 Tk 的所有字节按位异或

由于 MIFARE 卡属于 ISO14443-3 卡片，卡片本身没有 ATR，因此这个 ATR 是按照 PC/SC 的规范组装出来的。

例如：

MIFARE S50: 3B8F8001804F0CA000000306030001000000006A

MIFARE Ultralight: 3B8F8001804F0CA0000003060300030000000068



### 4.3 ATR 信息格式（适用于 ISO14443-4 PICC）

字节	值(Hex)	标记	说明
0	3B	初始字符	
1	8N	T0	高半字节 8 表示：后续不存在 TA1、TB1 和 TC1，只存在 TD1 低半字节 N 指出历史字符的个数（HisByte0 – HistByteN-1）
2	80	TD1	高半字节 8 表示：后续不存在 TA2、TB2 和 TC2，只存在 TD2
3	01	TD2	高半字节 0 表示后续不存在 TA3、TB3、TC3 和 TD3。 低半字节 1 表示协议类型为 T=1
4 至 3+N	XX	T1	历史字节： ISO 14443A：来自 ATS 应答的历史字节。请参考 ISO14443-4 规范。
	XX	Tk	ISO 14443B：来自 ATTRIB 应答（ATQB）的上层响应。请参考 ISO14443-3 规范。
	xx		
4+N	UU	TCK	T0 至 Tk 的所有字节按位异或

对于 ISO14443-4 的智能卡，此 ATR 并非来自卡片，而是经过组装的，对于卡片的 ATR，可以用 GetData 命令获取。

TYPE A 智能卡：3B 8B 80 01 20 90 00 00 00 00 00 26 1E 9A A6 BE

TYPE B 智能卡：3B 8C 80 01 50 20 16 21 EE 55 55 55 55 00 81 C1 E4



## 5 Contactless Reader 指令

### 5.1 操作准备及返回的状态信息

在操作非接触 IC 卡时，请先确认卡片被此读卡器支持。

将读卡器连接至电脑，并将卡片放置在读卡器的读卡区域内，读卡器寻到卡片后会有声光提示，启动我们提供的 PC/SC 操作软件：“PCSC Communication Tool”，在软件界面点击“Connect”，成功后即可输入 APDU 进行操作。

向卡片发送 APDU 发生错误时，请确认卡片支持您发送的 APDU。

返回信息状态如下（SW1/SW2）：

结果	SW1	SW2	错误注释
成功	90	00	操作成功
错误	63	00	操作失败
错误	6A	81	功能不支持

### 5.2 获取数据（GET DATA）

GET DATA 命令会返回“已建立连接的卡片”序列号或 ATS。

发送 APDU 格式：

命令	CLA	INS	P1	P2	Le
Get Data	FF	CA	00/01	00	00(全长)

应答：

P1=0x00h，响应格式为获取 UID（UID + SW）

Response	Data Out				
Result	UID(LSB)	--	UID(MSB)	SW1	SW2

P1 = 0x01h，响应格式为获取 ISO14443 A 卡的 ATS（ATS + SW）

Response	Data Out		
Result	ATS	SW1	SW2

示例：

获取“已经建立连接的PICC”序列号

Send: FF CA 00 00 00

Receive: 03 12 94 DD 90 00

获取“已经建立连接的ISO 14443-4 PICC”的ATS

Send: FF CA 01 00 00

Receive: 10 78 80 90 02 20 90 00 00 00 00 00 00 D0 19 47 B6 90 00



## 5.3 ISO14443A/B CPU 卡命令

对于卡片的标准 APDU，把 APDU 直接向读卡器发送即可完成对卡片的操作。

示例：

获取随机数

Send: 00 84 00 00 08

Receive: 02 57 BB 9F 51 C6 6E 89 90 00

## 5.4 直接 RF 命令 (DIRECT TRANSACTION)

直接发送命令到非接触通道并取得返回结果。

发送 APDU 格式一：

命令	CLA	INS	P1	P2	Lc	Data
Direct Transaction	FF	00	00	00	LEN	RF 数据流

Lc: 后续 RF 数据流的长度。

Data: 要通过非接触通道发送出去的命令。

应答：

Response	Data Out		
Result	RF返回数据	SW1	SW2

发送 APDU 格式二：

命令	Class	INS	P1	P2	Lc	CMD	TMO	Data
Direct Transaction	FF	00	FF	FF	2+n	命令	FWI	数据包 n

Lc: 1个字节，待发送的字节数，最大值为255。

CMD: 命令。0: 发送且接收；1: 只发送。

TMO: 超时参数，FWI 值，对于 M1 卡的读写，FWI=4。当 CMD=1 时此字节无意义。

Data: 经由 RF 发出的命令和数据

应答：

Response	Data Out
Result	响应数据

示例：

MIFARE Ultralight C 卡片的数据块读写操作：

Send: FF CA 00 00 00 (寻卡)

Receive: 04 0E 8B 8A 7C 3B 80 90 00

Send: FF 00 FF FF 08 01 00 A2 0401 02 03 04 (写块04)



**Receive:** 90 00  
**Send:** FF 00 FF FF 04 00 05 30 04 (读块04起始的4个块)  
**Receive:** 01 02 03 04 00 00 00 00 00 00 00 00 00 00 00 00 90 00

## 5.5 MIFARE Classic/UltraLight/ISO15693 存储卡命令 (T=CL 模拟)

本部分的 MIFARE Classic/UltraLight/ISO15693 存储卡操作的 APDU 是由 PC 发送给读写器，读写器做分析处理后按照卡片的命令进行处理，与 CPU 卡的 APDU 不同。

### 5.5.1 加载认证密钥 (LOAD AUTHENTICATION KEYS)

LOAD AUTHENTICATION KEYS 用于向读写器加载认证密钥和通讯加密密钥。卡片认证密钥用于验证 MIFARE 1K/4K 存储卡的特定扇区。通讯加密密钥用于对指令中的密文解密。下载的密钥可指定保存方式为非易失性。

**发送 APDU 格式:**

命令	CLA	INS	P1	P2	Lc	Data
LoadKeys	FF	82	KeyStructure	KeyIndex	Len	KeyData

**KeyStructure:** 1 字节

b7	b6	b5	b4	b3	b2	b1	b0	描述
X								0: 卡片密钥; 1: 读卡器密钥
	X							0: 明文方式传输; 1: 密文方式传输
		X						0: 密钥保存方式为易失性的 1: 密钥保存方式为非易失性的
			X	X	X	X	X	RFU

非易失性的密钥保存是有寿命限制的，用户使用中应注意。

**KeyIndex:** 1 字节

当密钥结构 b7=0: 数值范围 00~1Fh, 读卡器可以存储 32 个卡片密钥。

当密钥结构 b7=1: 数值范围 00h, 读卡器可以存储 1 个读卡器密钥。

**Len:** 1 字节

当下载读卡器密钥时，密钥长度要求 16 字节，否则读卡器会报错。

当以明文方式下载卡片密钥时，读卡器对密钥长度没有要求。

当以密文方式下载卡片密钥时，密钥长度要求 8 字节或 16 字节，否则读卡器会报错。

**KeyData:** N 字节

载入读写器的读卡器/卡片密钥值。

**应答:**

Response	Data Out
----------	----------



<b>Result</b>	SW1	SW2
---------------	-----	-----

## 5.5.2 认证 (AUTHENTICATION)

AUTHENTICATION 命令使用存储在读写器内的密钥来验证 MIFARE 1K/4K 卡。其中会用到两种认证密钥：KEY A 和 KEY B。

**发送 APDU 格式：**

命令	CLA	INS	P1	P2	Lc	Data
Auth	FF	86	00	00	05	认证数据字节

**认证数据字节 (5 字节)：**

字节 1	字节 2	字节 3	字节 4	字节 5
版本 01	00	Address	KeyType	KeyIndex

**Address:** 1 字节，验证的存储块地址。

**KeyType:** 1 字节。

60h = 密钥用作 A 密钥进行认证。

61h = 密钥用作 B 密钥进行认证。

**KeyIndex:** 1 字节，密钥编号。取值范围 00h ~ 01Fh。

**应答：**

Response	Data Out	
Result	SW1	SW2

## 5.5.3 读二进制块 (READ BINARY BLOCKS)

READ BINARY BLOCKS 命令用于从存储型 IC 卡中取回多个数据块的数据。如果卡片有密钥保护，那么执行该命令前必须先对数据块进行密钥验证。

**发送 APDU 格式：**

命令	CLA	INS	P1	P2	Le
Read Blocks	FF	B0	00	Address	LEN

**Address:** 1 字节，待访问的块地址。

**LEN:** 1 字节，待读取的字节数（16 字节的整数倍，需在同一块区）。

**应答：**

Response	Data Out		
Result	Data	SW1	SW2



### 5.5.4 更新二进制块 (UPDATE BINARY BLOCKS)

UPDATE BINARY BLOCKS 命令用于向 Classic/UltraLight 写入多个“数据块”。执行该命令前必须先对数据块/尾部块进行验证。

发送 APDU 格式:

命令	CLA	INS	P1	P2	Lc	Data
Update Blocks	FF	D6	00	Address	LEN	Data

**Address:** 1 字节, 待更新的块地址。

**LEN:** 1 字节, 待更新的字节数。

MIFARE 1K/4K 的更新字节数为 (16 字节的整数倍, 需在同一块区)。

MIFARE Ultralight 的更新字节数据为 (4 字节的整数倍)。

**Data:** 待写入二进制块的数据。

应答:

Response	Data Out	
Result	SW1	SW2

### 5.5.5 钱包块操作 (VALUE BLOCK OPERATION)

VALUE BLOCK OPERATION 命令用于向 Classic 卡进行数值操作, 例如: 增加值块的值等。

发送 APDU 格式:

命令	CLA	INS	P1	P2	Lc	Data
Value Block	FF	D7	00	Address	05	Data

**Address:** 1 字节, 待操作的块地址。

**Data:** VB\_OP(1Byte) + VB\_Value(4Bytes, LSB...MSB,有符号长整数)。

**VB\_OP:** 00h = 将块变为值块并初始化为 VB\_Value。

01h = 使值块的值增加 VB\_Value。此命令仅适用于对值块的操作。

02h = 使值块的值减少 VB\_Value。此命令仅适用于对值块的操作。

**VB\_Value:** 用于算数运算的数值, 是一个有符号长整数。

应答:

Response	Data Out	
Result	SW1	SW2

### 5.5.6 读取值块 (READ VALUE BLOCK)

READ VALUE BLOCK 命令用于向 Classic 卡获取值块中的数值, 仅适用于对值块的操作。

发送 APDU 格式:



命令	CLA	INS	P1	P2	Lc
ReadValueBlock	FF	B1	00	Address	04

**Address:** 1 字节, 待访问的值块。

**应答:**

Response	Data Out		
Result	Value	SW1	SW2

**Value:** 4 字节(LSB...MSB)。读取的块值, 是一个有符号长整数。

### 5.5.7 备份值块 (RESTORE VALUE BLOCK)

RESTORE VALUE BLOCK 命令用于向 Classic 卡将一个值块中的数值复制到另外一个值块。

**发送 APDU 格式:**

命令	CLA	INS	P1	P2	Lc	Data	
Restore	FF	D7	00	SourceAdd	02	03	TargetAdd

**SourceAdd:** 1 字节, 源值块地址。

**TargetAdd:** 1 字节, 目标值块地址。源值块和目标值块必须位于同一个扇区。

**应答:**

Response	Data Out	
Result	SW1	SW2

### 5.5.8 ISO15693 Write AFI

写 ISO15693 Tag AFI。

**发送 APDU 格式:**

Command	Class	INS	P1	P2	Lc	Data
Write AFI	FF	00	40	06	01	AFI

**AFI:** 新的 AFI。

**应答:**

Response	Data Out	
Result	SW1	SW2

### 5.5.9 ISO15693 LockAFI

锁定 ISO15693 Tag AFI。

**发送 APDU 格式:**





Command	Class	INS	P1	P2	Lc
LockAFI	FF	00	40	07	00

应答:

Response	Data Out	
Result	SW1	SW2

### 5.5.10 ISO15693 WriteDSFID

写 ISO15693 Tag DSFID。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
WriteDSFID	FF	00	40	08	01	DSFID

DSFID: 新的 DSFID。

应答:

Response	Data Out	
Result	SW1	SW2

### 5.5.11 ISO15693 LockDSFID

锁定 ISO15693 Tag DSFID。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc
LockDSFID	FF	00	40	09	00

应答:

Response	Data Out	
Result	SW1	SW2

### 5.5.12 ISO15693 GetSystemInfo

获取 ISO15693 Tag 系统信息

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc
GetSysInfo	FF	00	40	0A	00

应答:



Response	Data Out		
Result	System Info	SW1	SW2

**SystemInfo:** InfoFlag (1Byte)+UID (8Byte)+DSFID (1Byte)+AFI (1Byte)+Other (nByte)。

### 5.5.13 ISO15693 Get Blocks Security

获取 ISO15693 Tag 块安全状态

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
GetBlkSec	FF	00	40	0B	02	Data

**Data:** StartAddr(1Byte) + Num(1Byte)。

**StartAddr:** 起始块地址。

**Num:** 块数量 (n+1, n=0 时只读取起始块的安全状态)。

应答:

Response	Data Out		
Result	BlockSecSta*Num	SW1	SW2

### 5.5.14 ISO15693 Lock Block

锁定块。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
LockDSFID	FF	00	40	0C	01	Data

**Data:** BlockNO(1Byte)。

**BlockNO:** 待锁块号。

应答:

Response	Data Out	
Result	SW1	SW2



## 5.6 非接触式智能卡操作流程

### 5.6.1 ISO14443-4 卡操作

#### 基本操作步骤:

步骤 1: 将 CPU 卡放入非接触式读卡器天线上。

步骤 2: 电脑连接非触阅读器。

步骤 3: 发送 APDU 命令。

#### 示例:

取 8 字节随机数

**Send:** 00 84 00 00 08

**Receive:** 1A F7 F3 1B CD 2B A9 58 90 00

### 5.6.2 MIFARE 1K/4K 卡操作

#### 基本操作步骤:

步骤 1: 将 MIFARE 1K/4K 卡放入非接触式读卡器天线上。

步骤 2: 电脑连接非触阅读器。

步骤 3: 发送 MIFARE 1K/4K 卡片操作命令。

#### 示例:

**//Get Data**

**Send:** FF CA 00 00 00

**Receive:** 03 12 94 DD 90 00

**//加载读写器密钥**

**Send:** FF 82 A0 00 10 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF

**Receive:** 90 00

**//明文加载卡片密钥, 编号 00h**

**Send:** FF 82 20 00 06 FF FF FF FF FF FF

**Receive:** 90 00

**//密文加载卡片密钥, 编号 01, 密钥明文{FF FF FF FF FF FF}**

**Send:** FF 82 60 01 08 C0 D6 1E B0 84 F9 43 57

**Receive:** 90 00

**//验证块 4**

**Send:** FF 86 00 00 05 01 00 04 60 00

**Receive:** 90 00

**//写 4 块**



```
Send:      FF D6 00 04 10 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 00
Receive:   90 00
           //读 4/5 块

Send:      FF B0 00 04 20
Receive:   01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 00 FF FF FF FF FF FF
           FF FF FF FF FF FF FF FF FF FF 90 00
           //初始化值块 (00000000h)

Send:      FF D7 00 04 05 00 00 00 00 00
Receive:   90 00
           //读值块

Send:      FF B1 00 04 04
Receive:   00 00 00 00 90 00
           //增加值块 (02000000h)

Send:      FF D7 00 04 05 01 00 00 00 02
Receive:   90 00
           //读值块

Send:      FF B1 00 04 04
Receive:   00 00 00 02 90 00
           //减少值块 (01000000h)

Send:      FF D7 00 04 05 02 00 00 00 01
Receive:   90 00
           //读值块

Send:      FF B1 00 04 04
Receive:   00 00 00 01 90 00
           //备份值块

Send:      FF D7 00 04 02 03 05
Receive:   90 00
           //读备份值块

Send:      FF B1 00 05 04
Receive:   00 00 00 01 90 00
```

### 5.6.3 MIFAREUltra Light 卡操作

#### 基本操作步骤:

步骤 1: 将 MIFAREUltra Light 卡放入非接触式读卡器天线上

步骤 2: 电脑连接非触阅读器



步骤 3: 发送 MIFAREUltra Light 卡片操作命令

示例:

```
//Get Data
Send:      FF CA 00 00 00
Receive:   04 0E 8B 8A 7C 3B 80 90 00
//读 4~7 块
Send:      FF B0 00 04 10
Receive:   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 90 00
//写第 4 块
Send:      FF D6 00 04 04 00 01 02 03
Receive:   90 00
//写 5~7 块
Send:      FF D6 00 05 0C 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
Receive:   90 00
//读 4~12 块
Send:      FF B0 00 04 20
Receive:   00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 90 00
```

## 5.6.4 ISO15693 卡操作

基本操作步骤:

步骤 1: 将 ISO15693 卡片放入非接触式读卡器天线上

步骤 2: 电脑连接非触阅读器

步骤 3: 发送 ISO15693 卡片操作命令

示例:

```
//Get Data
Send:      FF CA 00 00 00
Receive:   00 AC 9C FC A3 00 01 04 E0 90 00
//读 8~11 块
Send:      FF B0 00 08 10
Receive:   FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 90 00
//写 8~11 块
Send:      FF D6 00 08 10 12 34 56 78 87 65 43 21 11 22 33 44 55 66 77 88
Receive:   90 00
//读 8~11 块
```



```
Send:      FF B0 00 08 10
Receive:   12 34 56 78 87 65 43 21 11 22 33 44 55 66 77 88 90 00
           //写 AFI = 0x12

Send:      FF 00 40 06 01 12
Receive:   90 00
           //锁定 AFI

Send:      FF 00 40 07 00
Receive:   90 00
           //写 AFI = 0x34

Send:      FF 00 40 06 01 34
Receive:   63 00(已锁定, 写 AFI 失败)
           //写 DSFID = 0x34

Send:      FF 00 40 08 01 34
Receive:   90 00
           //锁定 DSFID

Send:      FF 00 40 09 00
Receive:   90 00
           //写 DSFID = 0x56

Send:      FF 00 40 08 01 56
Receive:   63 00(已锁定, 写 DSFID 失败)
           //读系统信息

Send:      FF 00 40 0A 00
Receive:   0F AC 9C FC A3 00 01 04 E0 34 12 1B 03 01 90 00
           //块锁定测试
           //写 0x19 块

Send:      FF D6 00 19 04 19 19 19 19
Receive:   90 00
           //读 0x19 块

Send:      FF B0 00 19 04
Receive:   19 19 19 19 90 00
           //锁定 0x19 块

Send:      FF 00 40 0C 01 19
Receive:   90 00
           //读 0x00~0x19 块安全状态

Send:      FF 00 40 0B 02 00 19
Receive:   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
           01 90 00
```



//写 0x19 块

**Send:** FF D6 00 19 04 FF FF FF FF

**Receive:** 63 00(数据块已经锁定, 写操作失败)



## 6 SAM Reader 指令

重要提示：SAM 卡需要在读卡器上电前插入到读写器的 SAM 卡槽中。连接到 SAM 卡读写器后就可以直接对 SAM 发送 APDU 命令。

不支持 SAM Reader 设备驱动的产品，可在 Device、Contactless Reader 等设备驱动下使用。

例如：MR881 读卡器，SAM 卡的操作需要连接 DEVICE 设备，默认自动选定 SAM1 卡操作。

## 7 DEVICE 操作指令

本部分的 APDU 均为针对读写器（即“设备”通道）的操作，使用者只需将 DEVICE 设备成功连接即可发送下列 APDU。

RTC, LCD, Flash, LED, BEEP 等相关指令需要读写器硬件支持。字库编码：简体中文编码（GB2312），繁体中文（BIG5）。

其他参数详见产品手册。

### 7.1 切换当前操作智能卡

该指令用于切换“当前”SAM 卡，以实现对不同的 SAM 卡发送 APDU。

发送 APDU 格式：

Command	Class	INS	P1	P2	Lc	Data
SwitchCard	FF	00	61	00	01	CurSmartCard

CurSmartCard:

0x00 - 接触式 CPU 大卡

0x01 - SAM1 卡

0x02 - SAM2 卡

0x03 - SAM3 卡

0x04 - SAM4 卡

应答：

Response	Data Out	
Result	SW1	SW2

### 7.2 复位 SAM 卡

该功能用于 SAM 卡的手动复位。

发送 APDU 格式：

Command	Class	INS	P1	P2	Lc	Data
RstCard	FF	00	60	10	03	Parameter





**Parameter:** SAMNO(1Byte) + BaudRate(1Byte) + PPSPa(1Byte)。

**SAMNO:** SAM 卡编号

0x00 -接触式 CPU 大卡

0x01 - SAM1 卡

0x02 - SAM2 卡

0x03 - SAM3 卡

0x04 - SAM4 卡

**BaudRate:** 复位波特率

0x00 - 9600

0x01 - 19200

0x02 - 38400

0x03 - 55800

0x04 - 57600

0x05 - 115200

0x06 - 230400

**PPSPa:** 通信波特率 (PPS)

0x00 - 9600

0x01 - 19200

0x02 - 38400

0x03 - 55800

0x04 - 57600

0x05 - 115200

0x06 - 230400

0xFE - 以 ATR 中指定的波特率进行 PPS 操作

0xFF - 禁止 PPS 操作

应答:

Response	Data Out		
Result	ATR	SW1	SW2

## 7.3 设置 SAM 卡上电复位参数

设备上电时,会对所有的 SAM 卡进行复位操作,该功能是设置 SAM 上电复位时采用的波特率和 PPS 功能,默认复位波特率为 9600bps 和 PPS 禁止。此参数掉电后保存。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data		
SetRstSamBaud	FF	00	60	11	03	SAMNO	Baudrate	PPSPa

**SAMNO:**



0x00 -接触式 CPU 大卡

0x01 - SAM1 卡

0x02 - SAM2 卡

0x03 - SAM3 卡

0x04 - SAM4 卡

**Baudrate:**

0x00 - 9600

0x01 - 19200

0x02 - 38400

0x03 - 55800

0x04 - 57600

0x05 - 115200

0x06 - 230400

**PPSPa:**

0x00 - 9600

0x01 - 19200

0x02 - 38400

0x03 - 55800

0x04 - 57600

0x05 - 115200

0x06 - 230400

0xFE - 以 ATR 中指定的波特率进行 PPS 操作

0xFF - 禁止 PPS 操作

**应答:**

Response	Data Out	
Result	SW1	SW2

## 7.4 读取 SAM 卡上电复位参数

读取 SAM 复位时参数。

**发送 APDU 格式:**

Command	Class	INS	P1	P2	Lc	Data
ReadRstBaud	FF	00	60	12	01	SAMNO

**SAMNO:**

0x00 -接触式 CPU 大卡

0x01 - SAM1 卡

0x02 - SAM2 卡



0x03 - SAM3 卡

0x04 - SAM4 卡

**应答:**

Response	Data Out			
Result	RstBaud	PPSPa	SW1	SW2

**RstBaud:**

0x00 - 9600

0x01 - 19200

0x02 - 38400

0x03 - 55800

0x04 - 57600

0x05 - 115200

0x06 - 230400

**PPSPa:**

0x00 - 9600

0x01 - 19200

0x02 - 38400

0x03 - 55800

0x04 - 57600

0x05 - 115200

0x06 - 230400

0xFE - 以 ATR 中指定的波特率进行 PPS 操作

0xFF - 禁止 PPS 操作

## 7.5 设置卡片操作方式

有些 TYPE A CPU 卡与 M1 卡合并在一个芯片上，例如复旦微电子的 FM1208 卡，可以用 CPU 卡的方式操作，也可以用 M1 卡的方式操作，在一般的 PC/SC 读卡器上会默认按照 CPU 卡方式去操作。此命令可以指定按照 M1 卡方式操作。

**发送 APDU 格式:**

Command	Class	INS	P1	P2	Le	Data
Set Mode	FF	00	FF	10	01	Status

**Status:**

0x00: 使用 CPU 卡方式操作

0x01: 使用 M1 卡方式操作

**注:** 每次切换模式后需要重新放入卡片并重新连接智能卡。**应答:**



Response	Data Out	
Result	SW1	SW2

示例:

**Send:** FF 00 FF 10 01 01

**Receive:** 90 00

## 7.6 初始化 RTC 时间

该功能实现对读卡器内部时钟初始化操作。若需要时间能掉电保持，需要配备电池。指令的执行需要读写器具有 RTC 功能，详见产品手册。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
InitialRTC	FF	00	FB	00	08	Time

**Time:** 年 (HighByte) + 年 (LowByte) + 月 (Month) + 日 (Date) + 时 (Hour) + 分 (Minute) + 秒 (Second) + 星期 (Week)。

如: 时间数据: 2010-4-12 12:01:00 星期一指令数据: 07 DA 04 0C 0C 01 00 01。

应答:

Response	Data Out	
Result	SW1	SW2

## 7.7 读 RTC 时间

该功能实现读取读卡器内部时钟。指令的执行需要读写器具有 RTC 功能，详见产品手册。

发送 APDU 格式:

Command	Class	INS	P1	P2	Le
ReadRTC	FF	00	FB	01	08

应答:

Response	Data Out		
Result	Time	SW1	SW2

**Time:** 年 (HighByte) + 年 (LowByte) + 月 (Month) + 日 (Date) + 时 (Hour) + 分 (Minute) + 秒 (Second) + 星期 (Week)

如: 时间数据: 2010-4-12 12:01:00 星期一指令数据: 07 DA 04 0C 0C 01 00 01。

示例:

设置 RTC 时间, 2020-02-02 12:00:00 星期日

**Send:** FF 00 FB 00 08 07 E4 02 02 0C 00 00 07

**Receive:** 90 00

延时几秒后读取 RTC 时间



**Send:** FF 00 FB 01 08  
**Receive:** 07 E4 02 02 0C 00 05 07 90 00

## 7.8 显示时间

该功能主要是设置时间在 LCD 上的显示模式。若需要时间能掉电保持，需要配备电池。本指令的执行需要读写器具有 RTC，LCD 功能，详见产品手册。

### 发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
DisTime	FF	00	FB	02	03	Data

**Data:** EnableFag(1Byte)+Line(1Byte)+Column(1Byte)。

**EnableFag:** 时间显示使能 (0-Disable, 1-Enable)。

**Line:** 显示起始行 (0~7/0~12)。(对应 LCD 分辨率: 128\*64/240\*128)。

**Column:** 显示起始列 (0~127/0~239)。(同上)。

应答:

Response	Data Out	
Result	SW1	SW2

## 7.9 显示日期

该功能主要是设置日期在 LCD 上的显示模式。本指令的执行需要读写器具有 RTC，LCD 功能，详见产品手册。

### 发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
DisDate	FF	00	FB	03	03	Data

**Data:** EnableFag(1Byte)+Line(1Byte)+Column(1Byte)。

**EnableFag:** 日期显示使能 (0-Disable, 1-Enable)。

**Line:** 显示起始行 (0~7/0~12)。(对应 LCD 分辨率: 128\*64/240\*128)。

**Column:** 显示起始列 (0~127/0~239)。(同上)。

应答:

Response	Data Out	
Result	SW1	SW2

## 7.10 设定日期显示格式

**MR88x 专用指令。**该功能主要是设置日期在 LCD 上的显示格式。本指令的执行需要读写器具有 RTC，LCD 功能，详见产品手册。

**发送 APDU 格式:**

Command	Class	INS	P1	P2	Lc	Data
DateFormat	FF	00	FB	04	01	USdataformat

**USdataformat:** 日期显示格式。**0x00** – YYYY-MM-DD(默认)。**0x01** – MM-DD-YYYY。**应答:**

Response	Data Out	
Result	SW1	SW2

**示例:**

//设置 RTC

**Send:** FF 00 FB 00 08 07 E4 02 02 0C 00 00 07**Receive:** 90 00

//延时几秒后读取 RTC

**Send:** FF 00 FB 01 08**Receive:** 07 E4 02 02 0C 00 0F 07 90 00

//显示时间 (从 0 行 0 列开始显示)

**Send:** FF 00 FB 02 03 01 00 00**Receive:** 90 00

//显示日期 (从 4 行 0 列开始显示)

**Send:** FF 00 FB 03 03 01 04 00**Receive:** 90 00

//更改日期显示格式

**Send:** FF 00 FB 04 01 01**Receive:** 90 00

## 7.11 设定 LCD 显示字体类型

通过该指令可实现非英文显示字库切换。支持简体中文、繁体中文和俄文三种字体的切换。

**发送 APDU 格式:**

Command	Class	INS	P1	P2	Lc	Data
SetFontType	FF	00	FC	00	01	FontType

**FontType:**

0x01 简体中文 (默认)

0x02 繁体中文

0x03 俄文



应答:

Response	Data Out	
Result	SW1	SW2

## 7.12 读取 LCD 显示字体类型

通过该指令可获知当前支持的非英文字体类型。

发送 APDU 格式:

Command	Class	INS	P1	P2	Le
ReadFontType	FF	00	FC	01	01

应答:

Response	Data Out		
Result	FontType	SW1	SW2

FontType: 0x01 简体中文 (默认), 0x02 繁体中文, 0x03 俄文。

示例:

```

//设定简体中文
Send:      FF 00 FC 00 01 01
Receive:   90 00

//读取当前字体类型
Send:      FF 00 FC 01 01
Receive:   01 90 00

```

## 7.13 LCD 点阵设定

MR88x 专用指令。支持三种点阵显示, 默认 32 点阵。通过改指令可以自由切换点阵大小。

注: 俄文仅支持 32 点阵, 中文简体繁体支持 16,24,32 点阵。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
Lattice Set	FF	00	FC	0A	01	DefineBitmap

DefineBitmap:

```

0x00  16 点阵大小
0x01  24 点阵大小
0x02  32 点阵大小

```

应答:

Response	Data Out
----------	----------



<b>Result</b>	SW1	SW2
---------------	-----	-----

## 7.14 LCD 显示文本信息

在 LCD 上显示指定个数的字符(包括英文字符和非英文字符)。注意一个中文字符占 2Byte, 英文字符占 1Byte, 俄文字符占 1 byte。

**发送 APDU 格式:**

Command	Class	INS	P1	P2	Lc	Data
Display Font	FF	00	FC	02	nByte	Data

**Data:** Configure (1Byte) + Row(1Byte) + Column(1Byte) + DisplayData(nBytes)。

**Configure:**

位 Bit	值	说明
<b>B<sub>0</sub></b>	0	正显(黑字白底)
	1	反显(白字黑底)
<b>B<sub>2</sub>~B<sub>1</sub></b>	00	显示画面前不清屏幕
	01	显示画面前只清除显示画面的行
	10	显示画面前全部清屏
<b>B<sub>3</sub></b>	0	LCD 背光不亮
	1	LCD 背光亮
<b>B<sub>7</sub>~b<sub>4</sub></b>	RFU	RFU

**Row:**

值	说明
<b>0~7</b>	LCD 分辨率 128*64 16 点阵(1Row = 16 dot High)
<b>0~7</b>	LCD 分辨率 240*128 32 点阵(1Row = 32 dot High)
<b>0~0x09</b>	LCD 分辨率 240*128 24 点阵(1Row = 24 dot High)
<b>0~0x0F</b>	LCD 分辨率 240*128 16 点阵(1Row = 16 dot High)

**Column:** 0~127 / 0~239。(对应 LCD 分辨率: 128\*64/240\*128)。

**DisplayData:** 显示内容(注: 1 个汉字占位 2 个字符)。

**应答:**

Response	Data Out	
<b>Result</b>	SW1	SW2





## 7.15 LCD 任意位置显示文本信息

**MR88x 专用指令。**此指令跟“LCD 显示文本信息”指令功能基本相同，但它可以在任意点开始（指定点的 X 坐标和 Y 坐标位置）进行显示。该指令显示指定个数的字符（包括英文或中文），同时指定此字符串的点阵大小（共有 16 点阵，24 点阵和 32 点阵三种点阵可以选择，此点阵数据仅当前命令有效）。

**发送 APDU 格式：**

Command	Class	INS	P1	P2	Lc	Data
Display Font	FF	00	FC	09	nByte	Data

**Data: Configure(1Byte) + Row(1Byte) + Column(1Byte) + DisplayData(nBytes)。**

**Configure:**

位 Bit	值	说明
<b>B<sub>0</sub></b>	0	正显（黑字白底）
	1	反显（白字黑底）
<b>B<sub>1</sub></b>	RFU	RFU
<b>B<sub>2</sub></b>	0	不清屏
	1	清屏
<b>B<sub>3</sub></b>	0	LCD 背光不亮
	1	LCD 背光亮
<b>B<sub>5</sub>~b<sub>4</sub></b>	01	16 点阵字符显示
	10	24 点阵字符显示
	11	32 点阵字符显示
<b>B<sub>7</sub>~b<sub>6</sub></b>	RFU	RFU

**Row:** 0~127，显示字符的起始行地址。

**Column:** 0~239，显示字符的起始列地址。

**DisplayData:** 显示内容，显示不能超过 240 点。

**应答：**

Response	Data Out	
Result	SW1	SW2

**示例：**

//设定简体中文

**Send:** FF 00 FC 00 01 01

**Receive:** 90 00

//设定 16 点阵

**Send:** FF 00 FC 0A 01 00

**Receive:** 90 00



```

//显示“金木雨”
Send: FF 00 FC 02 09 00 00 80 BD F0 C4 BE D3 EA
Receive: 90 00

//从点（48*165）开始,显示 24 点阵的“金木雨”.正显,不清屏,背光亮
Send: FF 00 FC 09 09 28 30 A5 BD F0 C4 BE D3 EA
Receive: 90 00

//显示“金木雨”，此时还是 16 点阵格式
Send: FF 00 FC 02 09 00 00 80 BD F0 C4 BE D3 EA
Receive: 90 00

```

## 7.16 LCD 显示图片

该功能实现显示规定大小的图片，大的图片可以分多次显示。图片取模方式：纵向取模。

发送 APDU 格式：

Command	Class	INS	P1	P2	Lc	Data
DisPicture	FF	00	FC	03	nByte	Data

**Data:** Configure(1Byte) + Row(1Byte) + Column(1Byte) + PictureWidth(1Byte) + PictureHigh(1Byte) + DisplayData(nBytes)。

**Configure:**

位 Bit	值	说明
B <sub>0</sub>	0	正显（黑字白底）
	1	反显（白字黑底）
B <sub>2</sub> ~B <sub>1</sub>	00	显示画面前不清屏幕
	01	显示画面前只清除显示画面的行
	10	显示画面前全部清屏
B <sub>3</sub>	0	LCD 背光不亮
	1	LCD 背光亮
B <sub>7</sub> ~b <sub>4</sub>	RFU	RFU

**Row (1 row = 8 dot High) :** 0~7 / 0~15。（对应 LCD 分辨率：128\*64/240\*128）。

**Column:** 0~127 / 0~239。（同上）。

**PictureWidth:** 1~128 / 1~240，图片宽度。（同上）。

**PictureHigh:** 1~8 / 1~16，图片高度。（同上）。

**DisplayData:** 显示图片内容（字节数= 宽度 x 高度）。

应答：

Response	Data Out	
Result	SW1	SW2







应答:

Response	Data Out	
Result	SW1	SW2

备注:

- ❖ 若设置开机画面禁止，则后面参数无效。
- ❖ 开机画面保存在读卡器片外Flash中，字库占据部分空间，用户不可使用。例如：MR80x系列可用空间地址是1303~8191块区间，每块大小是512字节。MR88x系列可用空间地址是10360~16383块区间，每块大小是512字节。
- ❖ 在使能开机画面前，需用FlashWrite APDU 写入画面数据到Flash SaveAddr地址中，否则显示画面为不确定，若画面大于512字节，则多余字节写入紧接的第2块。
- ❖ 画面大小=Width\*High。

## 7.19 LCD 设定待机画面

该功能实现待机画面设置，若没有设置，则显示完毕用户界面后不会回到待机画面。所有显示画面都保存于读卡器 Flash 内。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
IdlePIC	FF	00	FC	06	08	Data

**Data:** Configure(1Byte) + SaveAddr(2Byte) + Width(1Byte) + High(1Byte) + StartLine(1Byte) + StartColumn(1Byte) + Time(1Byte)。

Configure:

位 Bit	值	说明
B <sub>0</sub>	0	禁止显示待机画面
	1	显示待机画面
B <sub>2</sub> ~B <sub>1</sub>	00	显示画面前不清屏幕
	01	显示画面前只清除显示画面的行
	10	显示画面前全部清屏
B <sub>3</sub>	0	LCD 背光不亮
	1	LCD 背光亮
B <sub>7</sub> ~b <sub>4</sub>	RFU	RFU

**SaveAddr:** 待机画面保存于 *Flash* 中的地址，地址低字节在前。

**Width:** 图片宽度 (1~128 / 1~240)。(对应 LCD 分辨率: 128\*64/240\*128)。

**High:** 图片高度 (1~8 / 1~16)。(同上)。

**StartLine:** 显示开始行 (0~7 / 0~15)。(同上)。

**StartColumn:** 显示开始列 (0~127 / 0~239)。(同上)。



**Time:** 设定多长时间未操作 LCD，进入待机画面（单位：秒）。

**应答:**

Response	Data Out	
Result	SW1	SW2

**备注:**

- ❖ 若设置待机画面禁止，则后面参数无效。
- ❖ 待机画面保存在读卡器片外Flash中，字库占据部分空间，用户不可使用。例如：MR80x系列可用空间地址是1303~8191块区间，每块大小是512字节。MR88x系列可用空间地址是10360~16383块区间，每块大小是512字节。
- ❖ 在使能待机画面前，需用FlashWrite APDU 写入画面数据到Flash SaveAddr地址中，否则显示画面为不确定，若画面大于512字节，则多余字节写入紧接的第2块。
- ❖ 画面大小=Width\*High。
- ❖ 指令方法可以参考LCD设定开机画面的例程，但是要注意Flash的存储地址不能重复。

**示例:**

设定开机/待机画面，需要在 FLASH 中先存储一张图片（以 MR88x 系列为例）。

**Send:** FF 00 FD 01 84 28 78 00 00  
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
 00 00 00 00 00 00 00 00 00 00 00 00 01 07 3F 3F 3F  
 1F 07 01 00 00 00 00 00 00 00 00 00 00 00 00 00  
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

**Receive:** 90 00

**Send:** FF 00 FD 01 84 28 78 00 80  
 00 00 00 00 00 00 00 00 00 00 7C 7F 7F 7F 3F 3F  
 3F 3F 1F 1F 1F 0F 0F 07 07 03 7F FF FF FF FF FF  
 FF FF FF 7D 03 07 07 0F 0F 1F 1F 1F 3F 3F 3F 3F  
 7F 7F 7F 78 00 00 00 00 00 00 00 00 00 00 00 00  
 00 00 00 00 00 00 01 03 03 0D 39 71 31 0D 07 07  
 03 03 01 00 00 04 04 04 04 05 07 7F 27 05 04 04  
 0C 0C 00 00 30 37 37 37 35 34 3F 3F 37 35 34 37  
 37 30 00 00 00 00 00 00 00 00 00 00 00 00 00 00

**Receive:** 90 00

**Send:** FF 00 FD 01 84 28 78 01 00  
 00 00 00 00 00 00 00 00 00 00 00 C0 F0 FC FF FF



FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  
BF 7F FF FF FF FF FF FF FF FF FF FF FF FF FF  
FC F0 80 00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 83 A2 32 3A 2E 26 FE FE 26 3E 3A  
62 22 02 00 04 0C 18 30 60 C0 00 FF 00 C0 60 30  
18 18 08 00 00 FF FE 20 B8 90 FE FE 20 BA 03 FF  
FC 00 00 00 00 00 00 00 00 00 00 00 00 00 00

**Receive:** 90 00

**Send:** FF 00 FD 01 84 28 78 01 80  
00 00 00 06 0F 0F 1F 1F 3F 3F 7F 7F 7F 7F BF  
FF EF FF FF FF FF FF FF FF FF FF FF FF FF FF  
FF FF FF FF FF FF FF FF FF FF FF F6 FF FF FF 7F  
7F 7F 7F 7F 3F 3F 1F 1F 0F 0F 07 00 00 00 00  
00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 00 00 00 00 00 00 00 00 00 00

**Receive:** 90 00

**Send:** FF 00 FD 01 84 28 79 00 00  
00 00 00 00 00 80 80 C0 C0 E0 E0 E0 E3 EF DF  
FF 7F FF FF FF FF FF FF FF FF FF FF FF FF FF  
FF FF FF FF FF FF FF FF FF FF FF FF FF FF EF  
E3 E0 E0 E0 C0 C0 80 80 00 00 00 00 00 00 00  
00 00 00 00 00 08 0E 06 01 05 05 05 1F 1D 05  
05 05 01 00 00 02 0E 0C 09 0B 08 08 08 08 0B  
0F 0C 00 00 00 0F 0F 09 0F 0F 00 0F 09 0F  
0F 00 00 00 00 00 00 00 00 00 00 00 00 00 00

**Receive:** 90 00

**Send:** FF 00 FD 01 84 28 79 00 80  
00 00 00 00 00 00 00 00 00 03 1F FF FF FF FF  
FF FF FF FF FF FF FE FE FC FF FF FF FF FF  
FF EF FF FB FC FE FE FF FF FF FF FF FF FF  
FF FF 1F 01 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 C0 C0 FF FF 87 36 5C 6C 27 7F 7D  
05 C4 8C 00 00 04 06 06 F6 D6 96 96 96 96 96  
BF B8 00 00 44 64 EF EF 5C F7 EF E0 EF B4 DC 6F  
6F 6C 28 00 00 00 00 00 00 00 00 00 00 00 00



**Receive:** 90 00

**Send:** FF 00 FD 01 84 28 79 01 00  
 00 00 00 00 00 00 00 00 00 00 00 E0 E0 E0 E0 E0 C0  
 C0 C0 80 80 80 00 00 00 00 00 00 F0 FC FE FF FF FF  
 FF FE F8 E0 00 00 00 00 00 80 80 80 C0 C0 C0 E0  
 E0 E0 E0 E0 00 00 00 00 00 00 00 00 00 00 00  
 00 00 00 00 00 00 80 80 40 40 C0 80 80 00 00  
 80 C0 40 00 00 00 00 00 00 00 00 80 80 C0 C0  
 80 00 00 00 00 00 C0 C0 80 C0 80 00 C0 80 80 C0  
 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00

**Receive:** 90 00

**Send:** FF 00 FD 01 84 28 79 01 80  
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
 00 00 00 00 00 00 00 00 00 00 00 00 00 C0 C0 C0  
 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

**Receive:** 90 00

//设定开机画面，重启时显示

**Send:** FF 00 FC 05 08 05 78 28 80 08 04 40 05

**Receive:** 90 00

//设定待机画面,5 秒后显示

**Send:** FF 00 FC 06 08 05 78 28 80 08 00 00 05

**Receive:** 90 00

## 7.20 LCD 背光控制

该功能对 LCD 的背光进行控制。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
LCDBackLight	FF	00	FC	07	02	Mode+Time

**Mode:**

00-灭

01-常亮





## 02-规定时间亮 (Time内容有效)

**Time:** 仅仅在 Mode =2 才有效 (单位: 秒)**应答:**

Response	Data Out	
Result	SW1	SW2

**示例:**

//LCD 背光灯亮 15 秒

**Send:** FF 00 FC 07 02 02 0F**Receive:** 90 00

## 7.21 LCD 显示 Flash 中存储画面

该功能实现保存画面显示。所有显示画面都保存于读卡器的串行 Flash 内。

**发送 APDU 格式:**

Command	Class	INS	P1	P2	Lc	Data
IdlePIC	FF	00	FC	08	07	Data

**Data:** Configure(1Byte) + DisAddr(2Byte) + Width(1Byte) + High(1Byte) + StartLine(1Byte) + StartColumn(1Byte)。

**Configure:**

位 Bit	值	说明
B <sub>0</sub>	RFU	RFU
B <sub>2</sub> ~B <sub>1</sub>	00	显示画面前不清屏幕
	01	显示画面前只清除显示画面的行
	10	显示画面前全部清屏
B <sub>3</sub>	0	LCD 背光不亮
	1	LCD 背光亮
B <sub>7</sub> ~b <sub>4</sub>	RFU	RFU

**DisAddr:** 显示画面保存于 Flash 中, 地址低字节在前**Width:** 图片宽度 (1~128 / 1~240)。(对应 LCD 分辨率: 128\*64/240\*128)。**High:** 图片高度 (1~8 / 1~16)。(同上)。**StartLine:** 显示开始行 (0~7 / 0~15)。(同上)。**StartColumn:** 显示开始列 (0~127 / 0~239)。(同上)。**应答:**

Response	Data Out	
Result	SW1	SW2

**备注:**

- ❖ 画面保存在读卡器片外Flash中，字库占据部分空间，用户不可使用。例如：MR80x系列可用空间地址是1303~8191块区间，每块大小是512字节。MR88x系列可用空间地址是10360 ~ 16383块区间，每块大小是512字节。
- ❖ 在显示画面前，需用FlashWrite APDU 写入画面数据到Flash SaveAddr地址中，否则显示画面为金木雨默认画面，若画面大于512字节，则多余字节写入紧接的第2块。
- ❖ 画面大小=Width\*High。

**示例:**

//显示 Flash 中地址 1303 的存储画面（以 MR80x 系列为例）

**Send:** FF 00 FC 08 09 0C 17 05 80 08 00 00**Receive:** 90 00

//显示 Flash 中地址 0x2878 的存储画面（以 MR88x 系列为例）

**Send:** FF 00 FC 08 07 0C 78 28 80 08 08 7F**Receive:** 90 00

## 7.22 读片外 Flash

片外 Flash 容量是 4Mbytes/8Mbytes，字库占据部分空间，用户不可使用。例如：MR80x 系列可用空间地址是 1303~8191 块区间，每块大小是 512 字节。MR88x 系列可用空间地址是 10360 ~ 16383 块区间，每块大小是 512 字节。

**发送 APDU 格式:**

Command	Class	INS	P1	P2	Lc	Data
ReadFlash	FF	00	FD	00	06	Data

**Data:** BlockAddr(2Bytes) + ByteAddr(2Bytes) + Len(2Byte)。**BlockAddr:** 块地址（高字节在前）。**ByteAddr:** 块内字节起始地址（高字节在前）。**Len:** 所读字节长度（高字节在前），len≤256。**应答:**

Response	Data Out		
Result	Flash Data	SW1	SW2

**示例:**

//读 Flash 的 02 块中的 2Byte，起始地址 0002

**Send:** FF 00 FD 00 06 00 02 00 02 00 02**Receive:** 18 08 90 00



## 7.23 写片外 Flash

片外 Flash 容量是 4Mbytes/8Mbytes，字库占据部分空间，用户不可使用。例如：MR80x 系列可用空间地址是 1303~8191 块区间，每块大小是 512 字节。MR88x 系列可用空间地址是 10360 ~ 16383 块区间，每块大小是 512 字节。

发送 APDU 格式：

Command	Class	INS	P1	P2	Lc	Data
WriteFlash	FF	00	FD	01	04+n	Data

Data: BlockAddr(2Bytes) + ByteAddr(2Bytes) + nData((n Bytes))

BlockAddr: 块地址（高字节在前）

ByteAddr: 块内字节起始地址（高字节在前）

nData: 所写数据

应答：

Response	Data Out	
Result	SW1	SW2

示例：

//给 0616 块写 1 字节数据，起始字节 00 02

Send: FF 00 FD 01 05 06 16 00 02 01

Receive: 90 00

## 7.24 获取产品序列号

不支持 DEVICE 设备驱动的产品，可在 SAM Reader、Contactless Reader 设备驱动下使用。

发送 APDU 格式：

Command	Class	INS	P1	P2	Le
GetSNR	FF	00	FF	00	0A

应答：

Response	Data Out		
Result	Product SNR	SW1	SW2

示例：

Send: FF 00 FF 00 0A

Receive: 01 05 07 09 09 04 03 08 06 09 90 00



## 7.25 获取硬件版本和版本号

不支持 DEVICE 设备驱动的产品, 可在 SAM Reader、Contactless Reader 设备驱动下使用。

发送 APDU 格式:

Command	Class	INS	P1	P2	Le
GetVer	FF	00	FF	01	04

应答:

Response	Data Out
Result	Hardware ver(2Bytes) + Software ver(2Bytes) SW1 SW2

示例:

Send: FF 00 FF 01 04

Receive: 01 00 01 05 90 00

## 7.26 LED 灯控制

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
LEDCtr	FF	00	FF	02	05	Data

Data: LEDStatus + LEDStatusMask + T1Duration + T2Duration + Number.

**LEDStatus:**

BIT0 = 红灯最终状态 (1-ON, 0-OFF)

BIT1 = 绿灯最终状态 (1-ON, 0-OFF)

BIT2 = 蓝灯最终状态 (1-ON, 0-OFF)

BIT3 = 黄灯最终状态 (1-ON, 0-OFF)

BIT4 = 红灯闪动初始状态 (1-ON, 0-OFF)

BIT5 = 绿灯闪动初始状态 (1-ON, 0-OFF)

BIT6 = 蓝灯闪动初始状态 (1-ON, 0-OFF)

BIT7 = 黄灯闪动初始状态 (1-ON, 0-OFF)

**LEDStatusMask:**

BIT0 = 红灯状态更新 (1-更新, 0-不改变)

BIT1 = 绿灯状态更新 (1-更新, 0-不改变)

BIT2 = 蓝灯状态更新 (1-更新, 0-不改变)

BIT3 = 黄灯状态更新 (1-更新, 0-不改变)

BIT4~7 RFU

**T1/T2:** T1 前半周期时间, T2 后半周期时间 (单位: 100ms)

**Number:** 次数



应答:

Response	Data Out	
Result	SW1	SW2

示例:

//四种颜色灯闪动两次, 最终状态为所有灯全关

Send: FF 00 FF 02 05 F0 0F 0F 02

Receive: 90 00

//红色灯闪动两次, 最终状态为红灯开

Send: FF 00 FF 02 05 01 01 0F 02

Receive: 90 00

//黄红灯交替闪动, 最终状态为红灯, 执行两次

Send: FF 00 FF 02 05 81 09 0F 02

Receive: 90 00

## 7.27 蜂鸣器控制

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
BuzzerCtr	FF	00	FF	03	05	Data

Data: BeepStatus + BeepStatusMask + T1Duration + T2Duration + Number。

**BeepStatue:**

BIT0 = BEEP最终状态 (1-ON, 0-OFF)

BIT4 = BEEP闪动初始状态 (1-ON, 0-OFF)

**BeepStatusMask:**

BIT0 = BEEP状态更新 (1-更新, 0-不改变)

BIT2~7 RFU

**T1/T2:** T1 前半周期时间, T2 后半周期时间 (单位: 100ms)

**Number:** 次数

应答:

Response	Data Out	
Result	SW1	SW2

示例:

//蜂鸣器闪动两次。

Send: FF 00 FF 03 05 08 01 0F 02

Receive: 90 00



## 7.28 卡片加密方法设置

设定 M1 卡认证加密标准。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
EncrMode	FF	00	FF	05	01	EncryMode

EncryMode:

0x00 – NXP加密标准

0x01-上海加密标准

应答:

Response	Data Out	
Result	SW1	SW2

示例:

设置上海标准加密方法

Send: FF 00 FF 05 01 01

Receive: 90 00

## 7.29 恢复出厂默认值（系统重新启动）

不支持 DEVICE 设备驱动的产品，可在 SAM Reader、Contactless Reader 设备驱动下使用。

发送 APDU 格式:

Command	Class	INS	P1	P2	Le
Reset	FF	00	FF	06	00

应答:

Response	Data Out	
Result	SW1	SW2

示例:

Send: FF 00 FF 06 00

Receive: 90 00

## 7.30 系统重新启动

发送 APDU 格式:

Command	Class	INS	P1	P2	Le
Reboot	FF	00	FF	07	00



应答:

Response	Data Out	
Result	SW1	SW2

示例:

Send: FF 00 FF 07 00

Receive: 90 00

## 7.31 设置快速启动

通过此命令可以设定读卡器省略设备自检的操作，快速进入到可以操作的状态。

发送 APDU 格式:

Command	Class	INS	P1	P2	Le	Data
SetFastBoot	FF	00	FF	08	01	Data

Data:

0x00 – 允许自检

0x01 – 跳过自检

应答:

Response	Data Out	
Result	SW1	SW2

例如:

Send: FF 00 FF 08 01 01

Receive: 90 00

## 7.32 读取快速启动设置状态

通过此命令读取快速启动的设定值。

发送 APDU 格式:

Command	Class	INS	P1	P2	Le
Reboot	FF	00	FF	08	00

应答:

Response	Data Out		
Result	Set	SW1	SW2

Set:

0x00 – 允许自检

0x01 – 跳过自检



例如:

Send: FF 00 FF 08 00

Receive: 00 90 00

### 7.33 获取 PICC 操作信息指示参数

该命令用于获取当前读卡器的 PICC 操作信息指示参数。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc
GetPICC	FF	00	FF	21	00

应答:

Response	Data Out		
Result	Parameter	SW1	SW2

例如:

Send: FF 00 FF 21 00

Receive: 03 90 00 (Parameter: 默认0x03)

### 7.34 设置 PICC 操作信息指示参数

该命令用于设置读卡器的 PICC 操作信息指示参数。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
SetPICC	FF	00	FF	21	01	Parameter

应答:

Response	Data Out	
Result	SW1	SW2

例如:

Send: FF 00 FF 21 01 03

Receive: 90 00

PICC 操作参数默认值 = 03h

位	参数	说明	选项
B7-B2	0	RFU	0
B1	LED提示	设置声光提示的使能与关闭	1 = 使能 0 = 关闭





<b>B0</b>	蜂鸣器提示		1 = 使能 0 = 关闭
-----------	-------	--	------------------

—————文档结束—————