

NFC & MIFARE & ISO14443A & ISO14443B 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 2.52)

北京金木雨电子有限公司

2018/10/10



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..... | 4 |
| 1 简介..... | 5 |
| 2 驱动安装和系统识别..... | 5 |
| 2.1 Contactless Reader | 5 |
| 2.2 SAM Reader | 6 |
| 2.3 Device Controller | 6 |
| 3 软件识别..... | 7 |
| 4 PICC 接口描述..... | 8 |
| 4.1 ATR 的生成..... | 8 |
| 4.2 ATR 信息格式（适用于 ISO14443-3 PICC） | 8 |
| 4.3 ATR 信息格式（适用于 ISO14443-4 PICC） | 9 |
| 5 Contactless Reader 指令..... | 10 |
| 5.1 操作准备及返回的状态信息 | 10 |
| 5.2 获取数据（GET DATA） | 10 |
| 5.3 ISO14443A/B CPU 卡命令 | 11 |
| 5.4 直接 RF 命令（DIRECT TRANSACTION） | 11 |
| 5.5 MIFARE Classic/UltraLight 存储卡命令（T=CL 模拟） | 12 |
| 5.5.1 加载认证密钥（LOAD AUTHENTICATION KEYS） | 12 |
| 5.5.2 认证（AUTHENTICATION） | 13 |
| 5.5.3 读二进制块（READ BINARY BLOCKS） | 13 |
| 5.5.4 更新二进制块（UPDATE BINARY BLOCKS） | 14 |
| 5.5.5 钱包块操作（VALUE BLOCK OPERATION） | 14 |
| 5.5.6 读取值块（READ VALUE BLOCK） | 15 |
| 5.5.7 备份值块（RESTORE VALUE BLOCK） | 16 |
| 5.6 非接触式智能卡操作流程..... | 17 |
| 5.6.1 ISO14443-4 卡操作 | 17 |
| 5.6.2 MIFARE 1K/4K 卡操作 | 17 |
| 5.6.3 MIFARE Ultra Light 卡操作 | 18 |
| 6 SAM Reader 指令..... | 19 |
| 7 DEVICE 操作指令..... | 19 |
| 7.1 切换当前操作智能卡（仅 MR881 支持） | 19 |
| 7.2 设置卡片操作方式..... | 19 |
| 7.3 初始化 RTC 时间（仅 MR801/MR881 支持） | 20 |
| 7.4 读 RTC 时间（仅 MR801/MR881 支持） | 20 |



| | | |
|------|--|----|
| 7.5 | 设定 RTC 时间显示-时间（仅 MR801/MR881 支持） | 21 |
| 7.6 | 设定 RTC 时间显示-日期（仅 MR801/MR881 支持） | 21 |
| 7.7 | 设定 LCD 显示字体类型（仅 MR801/MR881 支持） | 22 |
| 7.8 | 读取 LCD 显示字体类型（仅 MR801/MR881 支持） | 22 |
| 7.9 | LCD 点阵设定（仅 MR881 支持） | 23 |
| 7.10 | LCD 显示文本信息（仅 MR801/MR881 支持） | 23 |
| 7.11 | LCD 任意位置显示文本信息(仅 MR881 支持)..... | 24 |
| 7.12 | LCD 显示图片（直接发送图片数据）（仅 MR801/MR881 支持） | 25 |
| 7.13 | LCD 擦除行（仅 MR801/MR881 支持） | 26 |
| 7.14 | LCD 设定开机画面（仅 MR801/MR881 支持） | 26 |
| 7.15 | LCD 设定待机画面（仅 MR801/MR881 支持） | 29 |
| 7.16 | LCD 背光控制（仅 MR801/MR881 支持） | 30 |
| 7.17 | LCD 显示 Flash 中存储画面（仅 MR801/MR881 支持） | 31 |
| 7.18 | 读片外 Flash | 32 |
| 7.19 | 写片外 Flash | 32 |
| 7.20 | 获取产品序列号 | 33 |
| 7.21 | 获取硬件版本和版本号 | 33 |
| 7.22 | LED 灯控制(仅 MR881 支持)..... | 34 |
| 7.23 | 蜂鸣器控制 | 35 |
| 7.24 | 卡片加密方法设置 | 35 |
| 7.25 | 恢复出厂默认值（系统重新启动） | 36 |
| 7.26 | 系统重新启动 | 36 |
| 7.27 | 设置快速启动 | 36 |
| 7.28 | 读取快速启动设置状态 | 37 |
| 7.29 | 获取 PICC 操作信息指示参数 | 37 |
| 7.30 | 设置 PICC 操作信息指示参数 | 38 |



文件修改记录

| 日期 | 版本号 | 修改内容 |
|------------|-------|--|
| 2016.06.14 | V2.30 | 增加 Device 操作部分 修改了文档例程错误 |
| 2016.06.16 | V2.31 | 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 | V2.32 | 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 | V2.33 | 重新整理了章节 增加了部分说明 |
| 2017.08.18 | V2.35 | 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 | V2.37 | 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 |
| 2018.5.5 | V2.38 | 增加章节 5.5，直接 RF 命令传输 |
| 2018.7.24 | V2.39 | 增加对 MR881 的命令说明 |
| 2018.7.26 | V2.40 | 直接发送命令做了扩充 |
| 2018.08.01 | V2.51 | 修改了文档例程和指令说明错误。 |
| 2018.10.10 | V2.52 | 增加快速启动的设置和读取 |



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。

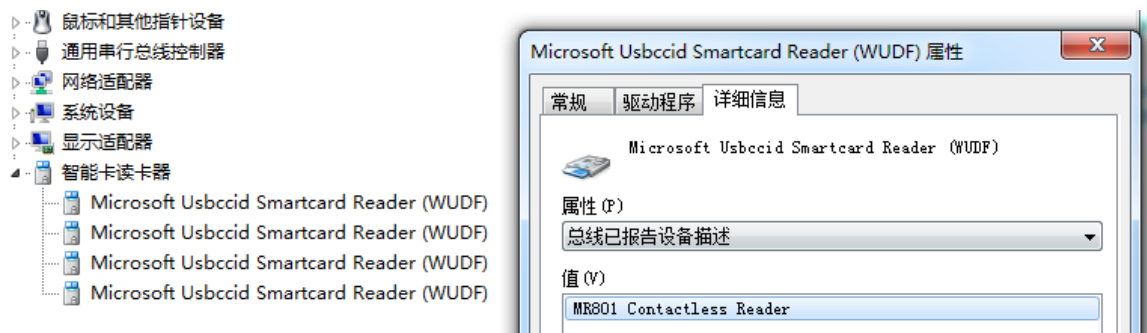
2 驱动安装和系统识别

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

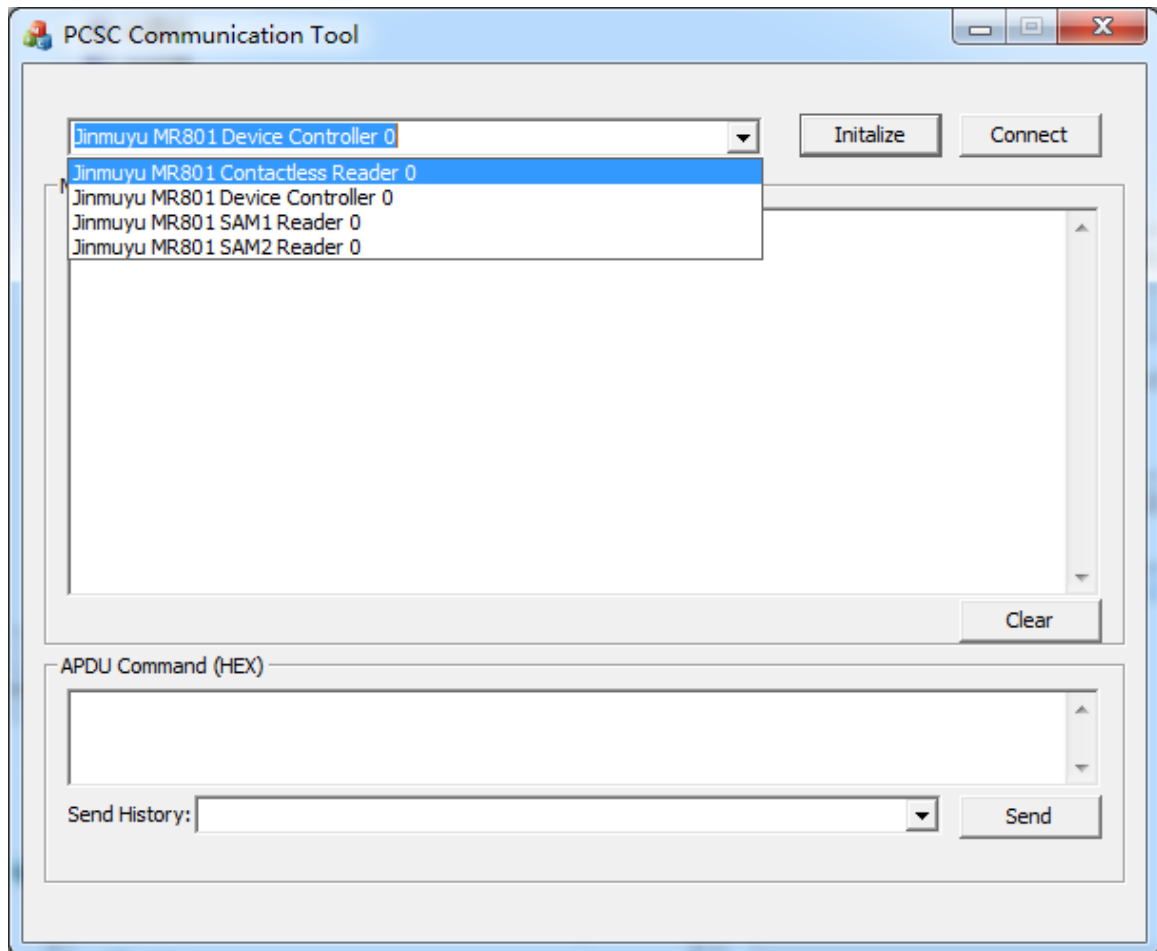
处理对读卡器的操作，例如 LED、Flash 等。





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 | 3Bh | 初始字符 | |
| 1 | 8Nh | 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 | 80h | 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 | 01h | 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 | 80h | 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 |
| | 4Fh | Tk | 应用标识符存在标识 |
| | 0Ch | | 长度 |
| | RID | | 注册应用供应商标识 (RID) # A0 00 00 03 06h |
| | SS | | 标准字节 |
| | C0...C1h | | 卡片名称字节 |
| | 00 00 00 00h | RFU | RFU # 00 00 00 00h |
| 4+N | UUh | 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: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 **00 01** 00 00 00 00 6A

MIFARE Ultralight: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 **00 03** 00 00 00 00 68



4.3 ATR 信息格式（适用于 ISO14443-4 PICC）

| 字节 | 值(Hex) | 标记 | 说明 |
|---------------|------------------|------|---|
| 0 | 3Bh | 初始字符 | |
| 1 | 8Nh | 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 | 80h | TD1 | 高半字节 8 表示：后续不存在 TA2、TB2 和 TC2，只存在 TD2 |
| 3 | 01h | 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 | XXh | T1 | 历史字节： ISO 14443A：来自 ATS 应答的历史字节。请参考 ISO14443-4 规范。 |
| | XXh xx XXh | Tk | ISO 14443B：来自 ATTRIB 应答（ATQB）的上层响应。 请参考 ISO14443-3 规范。 |
| 4+N | UUh | 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 | FFh | CAh | 00h 01h | 00h | 00h(全长) |

当 P1=0x00h，响应格式为获取 UID (UID + SW)

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

当 P1 = 0x01h，则获取 ISO 14443 A 卡的 ATS (ATS + SW)

| Response | Data Out | | |
|----------|----------|-----|-----|
| Result | ATS | SW1 | SW2 |

例如:

1. 获取“已经建立连接的 PICC”序列号

APDU = FF CA 00 00 00

2. 获取“已经建立连接的 ISO 14443-4 PICC”的 ATS

APDU = FF CA 01 00 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。

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 | 命令数据域 |
|--------------------|-----|-----|-----|-----|-----|--------|
| Direct Transaction | FFh | 00h | 00h | 00h | LEN | RF 数据流 |

LEN: 后续 RF 数据流的长度

RF 数据流: 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 | 命令数据域 |
|-----------------|-------|-----|----|----|----------------|-----|-----|-------|
| DIRECT TRANSMIT | FF | 00 | FF | FF | 待发送的字节数 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: 命令为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

命令数据域: 经由 RF 发出的命令和数据

应答:

| Response | 响应数据域 |
|-----------------|-------|
| DIRECT TRANSMIT | 响应数据 |

例如:

MIFARE Ultralight C 卡片的数据块读写操作:

Send: FF CA 00 00 00 (寻卡)

Receive: 07 04 15 BA 8A 7C 3B 80 44 00 00 90 00

Send: FF 00 FF FF 08 01 00 A2 04 01 02 03 04 (写块09)

Receive: 90 00

Send: FF 00 FF FF 04 00 05 30 04 (读块09起始的4个块)

Receive: 01 02 03 04 00 00 00 00 00 00 00 00 00 00 90 00



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

本部分的 MIFARE Classic/UltraLight 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 | 命令数据域 |
|-----------------|-----|-----|------|-----|-------|-------|
| Load Auth. Keys | FFh | 82h | 密钥结构 | 密钥号 | 密钥字节数 | 密钥 |

密钥结构: 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。

密钥号: 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。

密钥字节数: 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。

密钥: N 字节

载入读写器的读卡器或卡片密钥值。

应答:

| Response | Data Out | |
|----------|----------|-----|
| Result | SW1 | SW2 |

例如:

1. 向读卡器密钥存储位置 00h, 用明文方式加载密钥{00 11 22 ... FFh}

APDU = FF 82 A0 00 10 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF

2. 向卡片密钥缓存位置 00h, 用明文方式加载密钥{FF FF FF FF FF FFh}



APDU = FF 82 00 00 06 FF FF FF FF FF FF

3. 向卡片密钥存储位置 01h, 用密文方式加载密钥{FF FF FF FF FF FFh}, 加密密钥{00 11 22 ... FFh}

6 字节密钥后补 2 字节 00h, 使用加密密钥加密后的值为 C0 D6 1E B0 84 F9 43 57

APDU = FF 82 60 01 08 C0 D6 1E B0 84 F9 43 57

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 (6 字节) (旧的 PC/SC 标准, 不推荐使用):

| 命令 | CLA | INS | P1 | P2 | P3 | 命令数据域 |
|----------------|-----|-----|-----|----|------|-------|
| Authentication | FFh | 88h | 00h | 块号 | 密钥类型 | 密钥号 |

发送 APDU 格式 (10 字节) (新的 PC/SC 标准, 建议使用):

| 命令 | CLA | INS | P1 | P2 | Lc | 命令数据域 |
|----------------|-----|-----|-----|-----|-----|--------|
| Authentication | FFh | 86h | 00h | 00h | 05h | 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 01h | 00h | 块号 | 密钥类型 | 密钥号 |

块号: 1 字节 验证的存储块

密钥类型: 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

密钥号: 1 字节

00h ~ 01Fh = 密钥位置

应答:

| Response | Data Out | |
|----------|----------|-----|
| Result | SW1 | SW2 |

例如:

1. M1 卡, 密钥号 00h 验证块 04h。PC/SC V2.01 (已弃用)

APDU = FF 88 00 04 60 00

2. M1 卡, 密钥号 00h 验证块 04h。PC/SC V2.07

APDU = FF 86 00 00 05 01 00 04 60 00

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 Binary Blocks | FFh | B0h | 00h | 块号 | 待读取的字节数 |

块号: 1 字节 待访问的块

待读取的字节数: 1 字节 16 字节

应答 (N + 2 字节):

| Response | Data Out | | |
|----------|----------|-----|-----|
| Result | N=16 | SW1 | SW2 |

例如:

1. 读取 MIFARE 1K/4K 卡的 04h 块 16 字节二进制数据

APDU = FF B0 00 04 10

2. 读取 MIFARE Ultralight 卡, 04h 块开始 16 字节二进制数据 (读取 4、5、6、7)

APDU = FF B0 00 04 10

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 (4 或 16 + 5 字节):

| 命令 | CLA | INS | P1 | P2 | Lc | 命令数据域 |
|----------------------|-----|-----|-----|----|---------|--|
| Update Binary Blocks | FFh | D6h | 00h | 块号 | 待更新的字节数 | 块数据: MIFARE Ultralight: 4 字节 MIFARE 1K/4K: 16 字节 |

块号: 1 字节 待更新的起始块

待更新的字节数: 1 字节

MIFARE 1K/4K 的更新字节数为 16

MIFARE Ultralight 的更新字节数据位 4

数据块: 4/16 字节 待写入二进制块的数据

例如:

1. 将 MIFARE 1K/4K 卡中二进制块 04h 的数据更新为{00 01 ..0Fh}

APDU = FF D6 00 04 10 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

2. 将 MIFARE Ultralight 中二进制块 04h 的数据更新为{00 01 02 03}

APDU = FF D6 00 04 04 00 01 02 03

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 (10 个字节)



| 命令 | CLA | INS | P1 | P2 | Lc | 命令数据域 | |
|-----------------------|-----|-----|-----|----|-----|-------|------------------------------|
| Value Block Operation | FFh | D7h | 00h | 块号 | 05h | VB_OP | VB_Value: 4 字节 (LSB..MSB) |

块号: 1 字节 待操作的值块

VB_OP: 1 字节

00h = 将 VB_Value 存入该块。然后该块变为一个值块

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: 4 字节 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 |
|------------------|-----|-----|-----|----|-----|
| Read Value Block | FFh | B1h | 00h | 块号 | 04h |

块号: 1 字节 待访问的值块

应答 (4 + 2 字节):

| Response | Data Out | | |
|----------|--------------|-----|-----|
| Result | 值(LSB...MSB) | SW1 | SW2 |

值: 4 字节 卡片返回的数值，是一个有符号长整数 (4 字节)



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 | 命令数据域 | |
|---------------------|-----|-----|-----|-----|-----|-------|------|
| Restore Value Block | FFh | D7h | 00h | 源块号 | 02h | 03h | 目标块号 |

源块号: 1 字节 源值块中的值会被复制到目标值块

目标块号: 1 字节 要恢复的值块。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 |

例如:

1. 将数值 “1”存入块 04h。

APDU = FF D7 00 04 05 00 01 00 00 00

2. 读取值块

APDU = FF B1 00 04 00

3. 将值块 04h 的值复制到值块 05h

APDU = FF D7 00 04 02 03 05

4. 使值块 05h 的值增加“5”

APDU = FF D7 00 05 05 01 05 00 00 00



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

得到 8 个字节的随机数

Send: 0x00 84 00 00 08

Receive: 0x1A 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

加载卡片密钥:

Send: 0xFF 82 00 00 06 FFFFFFFFFFFFFFFF

Receive: 0x90 00

使用加载在 0x00 位置存储的密钥对 0x04 块进行身份验证:

Send: 0xFF 86 00 00 05 01 00 04 60 00

Receive: 0x90 00

从 0x04 块读取数据:

Send: 0xFF B0 00 04 10

Receive: 0xF6 FF FF FF 09 00 00 00 F6 FF FF FF 04 FB 04 FB 90 00

将数据写入 0x04 块:

Send: 0xFF D6 00 04 10 00112233445566778899AABBCCDDEEFF

Receive: 0x90 00:

钱包的初始化:

Send: 0xFF D7 00 04 05 00 00000000

Receive: 0x90 00

钱包充值:

Send: 0xFF D7 00 04 05 01 05000000

Receive: 0x90 00

钱包减值:



Send: 0xFF D7 00 04 05 02 0A000000

Receive: 0x90 00

备份钱包:

Send: 0xFF D7 00 04 02 03 05

Receive: 0x90 00

读钱包:

Send: 0xFF B1 00 05 04

Receive: 0xF6 FF FF FF 90 00

5.6.3 MIFARE Ultra Light 卡操作

基本操作步骤:

步骤 1: 将 MIFARE Ultra 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 MIFARE Ultra Light 卡片操作命令

写数据块:

Send: 0xFF D6 00 04 04 00 11 22 33

Receive: 0x90 00

读数据块:

Send: 0xFF B0 00 04 10

Receive: 0x00 11 22 33 00 00 00 00 44 55 66 77 88 99 AA BB 90 00



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。

对于 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 均为针对读写器的操作，使用者只需将 DEVICE 设备成功连接即可发送下列 APDU。

7.1 切换当前操作智能卡（仅 MR881 支持）

该功能主要实现接触的 SAM 之间切换。

发送 APDU 格式：

| Command | Class | INS | P1 | P2 | Lc | Data |
|-----------------|-------|-----|----|----|----|--------------|
| SwitchSmartCard | FF | 00 | FA | 00 | 01 | CurSmartCard |

CurSmartCard:

- 1 - SAM1 卡
- 2 - SAM2 卡
- 3 - SAM3 卡
- 4 - SAM4 卡

7.2 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 |
|---------------|------------|------------|

例如:

Send: FF 00 FF 10 01 01

Receive: 90 00

7.3 初始化 RTC 时间（仅 MR801/MR881 支持）

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。

发送 APDU 格式:

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

Time: 年 (High Byte) + 年 (Low Byte) + 月 (Month) + 日 (Date) + 时 (Hour) + 分 (Minute) + 秒 (Second) + 星期 (Week)

如: 2010-4-12 12: 01: 00 星期一 时间数据是: 07 DA 04 0C 0C 01 00 01

应答:

| Response | Data Out | |
|---------------|----------|-----|
| Result | SW1 | SW2 |

例如:

设置时间并读出时间:

Send: FF 00 FB 00 08 07 DA 04 0C 0C 01 00 01

Receive: 90 00

Send: FF 00 FB 01 08

Receive: 07 DA 04 0C 0C 03 15 01 90 00

7.4 读 RTC 时间（仅 MR801/MR881 支持）

该功能实现读取读卡器内部时钟。若需要时间能掉电保持，需要配备电池。

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 (High Byte) + 年 (Low Byte) + 月 (Month) + 日 (Date) + 时 (Hour) + 分 (Minute) + 秒 (Second) + 星期 (Week)

如: 2010-4-12 12: 01: 00 星期一 时间数据是: 07 DA 04 0C 0C 01 00 01



7.5 设定 RTC 时间显示-时间（仅 MR801/MR881 支持）

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。

发送 APDU 格式：

| Command | Class | INS | P1 | P2 | Lc | Data |
|---------|-------|-----|----|----|----|-----------------------|
| DisTime | FF | 00 | FB | 02 | 03 | EnableFag+Line+Column |

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

Line: 显示起始行（0~7/12）

Column: 显示起始列（0~127/239）

如显示 12 点 10 分 10 秒， 显示为：12: 10: 10

应答：

| Response | Data Out | |
|----------|----------|-----|
| Result | SW1 | SW2 |

例如：

关闭时间显示

Send: FF 00 FB 02 03 00 00 00

Receive: 90 00

设定时间显示

Send: FF 00 FB 02 03 01 03 05

Receive: 90 00

7.6 设定 RTC 时间显示-日期（仅 MR801/MR881 支持）

该功能主要是设置日期在 LCD 上的显示模式。若需要时间能掉电保持，需要配备电池。

发送 APDU 格式：

| Command | Class | INS | P1 | P2 | Lc | Data |
|---------|-------|-----|----|----|----|-----------------------|
| DisDate | FF | 00 | FB | 03 | 03 | EnableFag+Line+Column |

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

Line: 显示起始行（0~7/12）

Column: 显示起始列（0~127/239）

如显示 2010-04-16， 显示为：10/04/16

应答：

| Response | Data Out | |
|----------|----------|-----|
| Result | SW1 | SW2 |

例如：



关闭日期显示

Send: FF 00 FB 03 03 00 00 00

Receive: 90 00

设定日期显示

Send: FF 00 FB 03 03 01 03 05

Receive: 90 00

7.7 设定 LCD 显示字体类型（仅 MR801/MR881 支持）

MR801 支持简体中文、繁体中文和俄文三种字体。通过该指令可实现非英文显示字库切换。

发送 APDU 格式:

| Command | Class | INS | P1 | P2 | Lc | Data |
|-------------|-------|-----|----|----|----|-----------------|
| SetFontType | FF | 00 | FC | 00 | 01 | ChineseFontType |

ChineseFontType:

0x01 简体中文（默认）

0x02 繁体中文

0x03 俄文

应答:

| Response | Data Out | |
|----------|----------|-----|
| Result | SW1 | SW2 |

例如:

简体中文显示

Send: FF 00 FC 00 01 01

Receive: 90 00

7.8 读取 LCD 显示字体类型（仅 MR801/MR881 支持）

MR800 支持简体中文、繁体中文和俄文三种字体。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



例如:

Send: FF 00 FC 00 01 02
 Receive: 90 00
 Send: FF 00 FC 01 01
 Receive: 02 90 00

7.9 LCD 点阵设定 (仅 MR881 支持)

MR881 支持三种点阵显示, 开机默认 32 点阵。通过改指令可以自由切换点阵大小

发送 APDU 格式:

| Command | Class | INS | P1 | P2 | Lc | Data |
|-----------------|-------|-----|----|----|----|--------------|
| Display Picture | 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(系统默认, 重新上电会重置)

应答:

| Response | Data Out | |
|----------|----------|-----|
| Result | SW1 | SW2 |

注: 俄文仅支持 32 点阵, 中文简体繁体支持 16,24,32 点阵。

例如:

Send: FF 00 FC 0A 01 00
 Receive: 90 00

7.10 LCD 显示文本信息 (仅 MR801/MR881 支持)

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 2 Byte, 英文字符占 1 Byte, 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 | Configure + Row + Column + Display Data |

Configure:

Bit0 (NegativeDis) : 正反显示 0-正显, 1-反显

Bit2~1:

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

Bit3 (BackLight) : 0-背光不亮, 1-背光亮

Bit4~7: RFU

Row:

0~7 (MR801 (1Row = 16 dot High))

0~7 (MR880 32 点阵(1Row = 32 dot High))

0~0x09 (MR880 24 点阵(1Row = 24 dot High))

0~0x0F (MR880 16 点阵(1Row = 16 dot High))

Column: 0~127 / 239

DisplayData: 显示内容 (汉字相当于 2Byte, ASCII 和俄文相当于 1 个字节)

应答:

| Response | Data Out | |
|----------|----------|-----|
| Result | SW1 | SW2 |

例如:

在左上角显示“金木雨”三个字, 正显、显示画面不清屏幕、背光不亮。

Send: FF 00 FC 02 09 00 00 00 BD F0 C4 BE D3 EA

Receive: 90 00

7.11 LCD 任意位置显示文本信息(仅 MR881 支持)

此指令跟“LCD 显示文本信息”指令功能基本相同, 只不过它可以任意的点的位置 (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、中文等), 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)。

发送 APDU 格式:

| Command | Class | INS | P1 | P2 | Lc | Data |
|--------------|-------|-----|----|----|-------|---|
| Display Font | FF | 00 | FC | 09 | nByte | Configure + Row + Column + Display Data |

Configure:

Bit0 (NegativeDis) : 正反显示 0-正显, 1-反显

Bit1: RFU

Bit2: 此位置 1 则显示之前清整个液晶屏, 为 0 则不清屏

Bit3 (BackLight) : 0-背光不亮, 1-背光亮

Bit4~5: (此点阵大小只对该命令有效)

01-用 16 点高的字符显示 (汉字宽 16 点, ASC 码宽 8 点)

10-用 24 点高的字符显示 (汉字宽 24 点, ASC 码宽 12 点)

11-用 32 点高的字符显示 (汉字宽 32 点, ASC 码宽 16 点)

Bit6~7: RFU



Row 字节:

范围 0~127 ， 即表示字符串显示的点行位置

Column 字节:

范围 0~239 ， 即表示字符串显示的点列位置

DisplayData: 显示内容，显示不能超过 240 点。

应答:

| Response | Data Out | |
|----------|----------|-----|
| Result | SW1 | SW2 |

例如：在第 03 点行第 04 点列的位置显示 24 点阵的“金木雨”三个字，正显、显示画面、不清屏幕、背光亮。

Send: FF 00 FC 09 09 28 03 04 BD F0 C4 BE D3 EA

Receive: 90 00

7.12 LCD 显示图片(直接发送图片数据) (仅 MR801/MR881 支持)

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。

发送 APDU 格式:

| Command | Class | INS | P1 | P2 | Lc | Data |
|-----------------|-------|-----|----|----|-------|---|
| Display Picture | FF | 00 | FC | 03 | nByte | Configure + Row + Column + PictureWidth + PictureHigh+ Display Data |

Configure:

Bit0 (Negative Dis) : 正反显示 0-正显, 1-反显

Bit2~1:

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

Bit3 (Back Light) : 0-背光不亮, 1-背光亮

Bit4~7: RFU

Row (1row = 8 dot High) : 0~7 / 15 (开始列)

Column: 0~127 / 239 (开始行)

Picture Width: 1~128 / 240, 图片宽度

Picture High: 1~8 / 16, 图片高度

Display Data: 显示图片内容 (字节数= 宽度 x 高度)



应答:

| Response | Data Out | |
|----------|----------|-----|
| Result | SW1 | SW2 |

7.13 LCD 擦除行 (仅 MR801/MR881 支持)

以行为单位清除 LCD 显示。

发送 APDU 格式:

| Command | Class | INS | P1 | P2 | Lc | Data |
|----------|-------|-----|----|----|-------|------|
| EraseLCD | FF | 00 | FC | 04 | 01/02 | Row |

Row (1row = 8 dot High) : Bit0~Bit7 分别代表 0~7 行 (0-保持不变, 1-擦除)

注: MR881 的 Lc=02, Row 为双字节变量, 是 Bit0~Bit15 分别代表 0~15 行。

应答:

| Response | Data Out | |
|----------|----------|-----|
| Result | SW1 | SW2 |

例如:

LCD 擦除所有行

Send: FF 00 FC 04 01 FF

Receive: 90 00

Send: FF 00 FC 04 02 FF FF

Receive: 90 00

7.14 LCD 设定开机画面 (仅 MR801/MR881 支持)

该功能实现默认开机画面设置。若没有设置, 则开机默认显示金木雨开机画面。所有显示画面都保存于读卡器内。

发送 APDU 格式:

| Command | Class | INS | P1 | P2 | Lc | Data |
|------------|-------|-----|----|----|----|---|
| PowerOnPIC | FF | 00 | FC | 05 | 08 | Enable+SaveAddr+Width+High+StartLine+StartColumn+Time |

Enable (1Byte) : 0-禁止显示开机画面, 1-显示开机画面

SaveAddr (2Byte) : 开机画面保存于串行 Flash 中, 地址低字节在前

Width (1Byte) : 图片宽度 (1~128 / 240)

High (1Byte) : 图片高度 (1~8 / 16)

StartLine (1Byte) : 显示开始行 (0~7 / 15)



StartColumn (1Byte) : 显示开始列 (0~127 / 239)

Time: 设定显示启动画面时间 (单位: S)

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, 字库占据开始的1303块 (0~1302) (MR881为10360(0 ~ 10359), 用户不能进行擦写, 供用户使用的块号是1303~8191 (MR881为10360 ~ 16383), 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。

例如:

设定一个开机画面, 本图片大小为 128*64.(要在 Flash 中先写入图片数据)

Send: FF 00 FD 01 84 05 17 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 05 17 00 80
 00 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 05 17 01 00
 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 FF
FC F0 80 00 00 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 00

Receive: 90 00

Send: FF 00 FD 01 84 05 17 01 80

00 00 00 06 0F 0F 1F 1F 3F 3F 7F 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 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 00 00 00 00 00 00 00 00 00

Receive: 90 00

Send: FF 00 FD 01 84 05 18 00 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 FF FF EF
E3 E0 E0 E0 C0 C0 80 80 00 00 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 08 0B
0F 0C 00 00 00 00 0F 0F 09 0F 0F 00 0F 09 09 0F
0F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Receive: 90 00

Send: FF 00 FD 01 84 05 18 00 80

00 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
FF EF FF FB FC FE FE FF 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 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 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 00



Receive: 90 00
Send: FF 00 FD 01 84 05 18 01 00
 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 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 05 18 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 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

Receive: 90 00
Send: FF 00 FC 05 08 01 17 05 80 08 00 00 05
Receive: 90 00

7.15 LCD 设定待机画面（仅 MR801/MR881 支持）

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内。

发送 APDU 格式：

| Command | Class | INS | P1 | P2 | Lc | Data |
|---------|-------|-----|----|----|----|---|
| IdlePIC | FF | 00 | FC | 06 | 08 | Configure +SaveAddr+ Width+ High+StartLine+StartColumn+Time |

Configure（1Byte）：

Bit0: 0-禁止显示待机画面， 1-显示待机画面

Bit2~1:

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

Bit3 (BackLight) : 0-背光不亮, 1-背光亮

Bit4~7: RFU

SaveAddr (2Byte) : 开机画面保存于串行 *Flash* 中, 地址低字节在前。

Width (1Byte) : 图片宽度 (1~128 / 240)

High (1Byte) : 图片高度 (1~8 / 16)

StartLine (1Byte) : 显示开始行 (0~7 / 15)

StartColumn (1Byte) : 显示开始列 (0~127 / 239)

Time: 设定多长时间未操作 LCD, 进入待机画面 (单位: S)

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, 字库占据开始的1303块 (0~1302) (MR881为10360块), 用户不能进行擦写, 供用户使用的块号是1303~8191 (MR881为10360~16383), 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。
- ❖

7.16 LCD 背光控制 (仅 MR801/MR881 支持)

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-设定时间内亮 (Time内容有效)

Time: 仅仅在 Mode =2 才有效 (单位: S)

应答:

| Response | Data Out | |
|----------|----------|-----|
| Result | SW1 | SW2 |



例如:

LCD 背光灯亮 15 秒

Send: FF 00 FC 07 02 02 0F

Receive: 90 00

7.17 LCD 显示 Flash 中存储画面（仅 MR801/MR881 支持）

该功能实现保存画面显示。所有显示画面都保存于读卡器内。

发送 APDU 格式:

| Command | Class | INS | P1 | P2 | Lc | Data |
|---------|-------|-----|----|----|----|---|
| IdlePIC | FF | 00 | FC | 08 | 09 | Configure +DisAddr +Width+ High + StartLine+StartColumn |

Configure (1Byte) :

Bit0: RFU

Bit2~1:

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

Bit3 (BackLight) : 0-背光不亮, 1-背光亮

Bit4~7: RFU

DisAddr (2Byte) : 显示画面保存于 Flash 中, 地址低字节在前

Width (1Byte) : 图片宽度 (1~128 / 240)

High (1Byte) : 图片高度 (1~8 / 16)

StartLine (1Byte) : 显示开始行 (0~7 / 15)

StartColumn (1Byte) : 显示开始列 (0~127 / 239)

应答:

| Response | Data Out | |
|----------|----------|-----|
| Result | SW1 | SW2 |

备注:

- ❖ 显示画面保存在读卡器片外Flash中, 字库占据开始的1303块 (0~1302) (MR881为10360块), 用户不能进行擦写, 供用户使用的块号是1303~8191 (MR881为10360~16383), 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。

例如:



显示 Flash 中地址 1303 的存储画面

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

Receive: 90 00

7.18 读片外 Flash

片外 Flash 容量是 4Mbytes(MR881 是 8Mbytes), 字库占据开始的 1303 块(0~1302) (MR881 为 10360 块), 用户不能进行擦写, 供用户使用的块号是 1303~8191(MR881 为 10360 ~ 16383), 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 | BlockAddr+ ByteAddr+ Len |

BlockAddr: 块地址 (2Byte, 高字节在前)

ByteAddr: 块内字节起始地址 (2Byte, 高字节在前)

Len: 所读字节长度 (2Byte, 高字节在前) len ≤ 256

应答:

| Response | Data Out | | |
|----------|------------|-----|-----|
| Result | Flash Data | SW1 | SW2 |

例如:

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

7.19 写片外 Flash

片外 Flash 容量是 4Mbytes(MR881 是 8Mbytes), 字库占据开始的 1303 块(0~1302) (MR881 为 10360 块), 用户不能进行擦写, 供用户使用的块号是 1303~8191(MR881 为 10360 ~ 16383), 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 | BlockAddr+ ByteAddr+Data (n) |

BlockAddr: 块地址 (2Byte, 高字节在前)

ByteAddr: 块内字节起始地址 (2Byte, 高字节在前)

Data: 所写数据

应答:

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



| | | |
|---------------|-----|-----|
| Result | SW1 | SW2 |
|---------------|-----|-----|

例如:

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

7.20 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 |

例如:

Send: FF 00 FF 00 0A

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

7.21 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 (2Byte) +Software ver (2Byte) | SW1 | SW2 |

例如:

Send: FF 00 FF 01 04

Receive: 02 00 03 00 90 00



7.22 LED 灯控制(仅 MR881 支持)

发送 APDU 格式:

| Command | Class | INS | P1 | P2 | Lc | Data |
|---------|-------|-----|----|----|----|---|
| LEDCtr | FF | 00 | FF | 02 | 05 | LED state + state Mask +T1 duration + T2 Duration + Number |

LED Status:

- 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)

LED Status Mask:

- BIT0 = 红灯状态更新掩码 (1-更新, 0-不改变)
- BIT1 = 绿灯状态更新掩码 (1-更新, 0-不改变)
- BIT2 = 蓝灯状态更新掩码 (1-更新, 0-不改变)
- BIT3 = 黄灯状态更新掩码 (1-更新, 0-不改变)
- BIT4~7 RFU

T1/T2: T1, T2 时间 (单位: 100ms), T=T1+T2

Number: 次数

应答:

| Response | Data Out | |
|----------|----------|-----|
| Result | SW1 | SW2 |

例如:

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 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 11 01 0F 0F 02

Receive: 90 00

黄灯闪动, 最终状态为红灯, 执行两次

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



Receive: 90 00

7.23 蜂鸣器控制

发送 APDU 格式:

| Command | Class | INS | P1 | P2 | Lc | Data |
|-----------|-------|-----|----|----|----|--|
| BuzzerCtr | FF | 00 | FF | 03 | 05 | Beep state + state Mask +T1 duration + T2 Duration + Number |

BEEP Status:

BIT0 = BEEP最终状态 (1-ON, 0-OFF)

BIT4 = BEEP闪动初始状态 (1-ON, 0-OFF)

Status Mask:

BIT0 = Buzzer状态更新掩码 (1-更新, 0-不改变)

BIT4~7 RFU

T1/T2: T1,T2 时间 (单位: 100ms), T=T1+T2

Number: 次数

应答:

| Response | Data Out | |
|----------|----------|-----|
| Result | SW1 | SW2 |

例如:

蜂鸣器闪动, 更新状态掩码, 闪动两次, 重复两次。

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

Receive: 90 00

7.24 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 | Encrypt Standard |

Encrypt Standard:

0x00-NXP加密标准

0x01-上海加密标准

应答:

| Response | Data Out | |
|----------|----------|-----|
| Result | SW1 | SW2 |



例如：

设置上海标准加密方法

Send: FF 00FF 05 01 01

Receive: 90 00

7.25 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 |
|----------------|-------|-----|----|----|----|
| FactoryDefault | FF | 00 | FF | 06 | 00 |

应答：

| Response | Data Out | |
|----------|----------|-----|
| Result | SW1 | SW2 |

例如：

Send: FF 00 FF 06 00

Receive: 90 00

7.26 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 |

例如：

Send: FF 00 FF 07 00

Receive: 90 00

7.27 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 |
|---------|-------|-----|----|----|----|------|
| Reboot | FF | 00 | FF | 08 | 01 | 01 |



应答:

| Response | Data Out | |
|----------|----------|-----|
| Result | SW1 | SW2 |

例如:

Send: FF 00 FF 08 01 01

Receive: 90 00

7.28 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 | Lc |
|---------|-------|-----|----|----|----|
| Reboot | FF | 00 | FF | 08 | 00 |

应答:

| Response | Data Out | | |
|----------|----------|-----|-----|
| Result | Setting | SW1 | SW2 |

例如:

Send: FF 00 FF 08 00

Receive: 00 90 00

7.29 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 |
|-------------|-------|-----|----|----|----|
| MessageMode | 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)



7.30 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 |
|----------|-------|-----|----|----|----|--------------|
| EncrMode | FF | 00 | FF | 21 | 01 | 新的 PICC 操作参数 |

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 |
|------|-------|--------------|------------------|
| 7..2 | 0 | RFU | 0 |
| 1 | LED提示 | 设置声光提示的使能与关闭 | 1 = 使能 0 = 关闭 |
| 0 | 蜂鸣器提示 | | 1 = 使能 0 = 关闭 |

----- 文档结束 -----