Advanced Card Systems Ltd.
Card & Reader Technologies

ACR1283L

脱机式
非接触读写器

参考手册 V1.00

info@acs.com.hk
www.acs.com.hk

目录

图目录

表目录

1.0. 简介

ACR1283L 脱机式非接触读写器是一款用于读写非接触式智能卡的设备。它的非接触式接口可用于读写符合 ISO 14443 标准的 A 类和 B 类卡以及 Mifare 系列卡片。另外它还带有安全存取模块（SAM）接口以确保非接触式智能卡应用的高度安全性。

ACR1283L 可以支持连机和脱机两种工作模式。在连机应用中，它被用作计算机和智能卡的中间设备，读写器可以通过 USB 端口与计算机建立连接，并执行计算机发出的指令——无论是用于与非接触式卡或 SAM 卡通信的命令，还是用于控制外围设备（例如液晶显示屏、键盘、LED 和蜂鸣器）的命令。本手册详细介绍了如何按照 PC/SC 规范来执行 PC/SC APDU 命令控制读写器的外围设备和操作非接触式卡片。

2.0. 特性

- 双工作模式：
  - 连机
  - 脱机
- 连机操作：
  - USB 2.0 全速接口
  - 符合 CCID 标准
  - 支持 PC/SC
  - 支持 CT-API（通过 PC/SC 上一层的封装）
- 脱机操作：
  - 支持第三方应用程序编程
  - 超过 400 KB 的第三方应用程序存储空间
  - 超过 500 KB 的数据存储空间
  - 支持的开发平台：
    - IAR 嵌入式工作台（5.50 或以上版本）
    - CoIDE(GCC)（1.3.0 或以上版本）
- 智能卡读写器：
  - 读写速率高达 848 kbps
  - 内置天线用于读写非接触式标签，读取智能卡的距离可达到 50 mm（视标签的类型而定）
  - 支持 ISO 14443 第 4 部分的 A 类和 B 类卡，以及 Mifare 系列卡
  - 内建防冲突特性（任何时候都只能访问 1 张标签）
  - 4 个符合 ISO 7816 标准的 SAM 卡卡槽
- 内置外围设备：
  - 2 行图形液晶显示屏
  - 4 个用户可控的 LED 指示灯
  - 1 个用户可控的蜂鸣器
  - 12 键电容式触摸键盘
- 带独立备用电池的实时时钟（RTC）
- 设备内 AES（128 或 256）、DES 和 3DES 加密算法
- 支持 Android™ OS 3.1 及以上版本
- 具有 USB 固件升级能力

- 符合下列标准：
  - ISO 14443
  - CE
  - FCC
  - PC/SC
  - CCID
  - Microsoft® WHQL
  - RoHS

3.0. 体系结构

ACR1283L 与计算机之间的数据通讯采用 CCID 协议，而 PICC 和 SAM 间的通信则完全符合 PC/SC 标准。

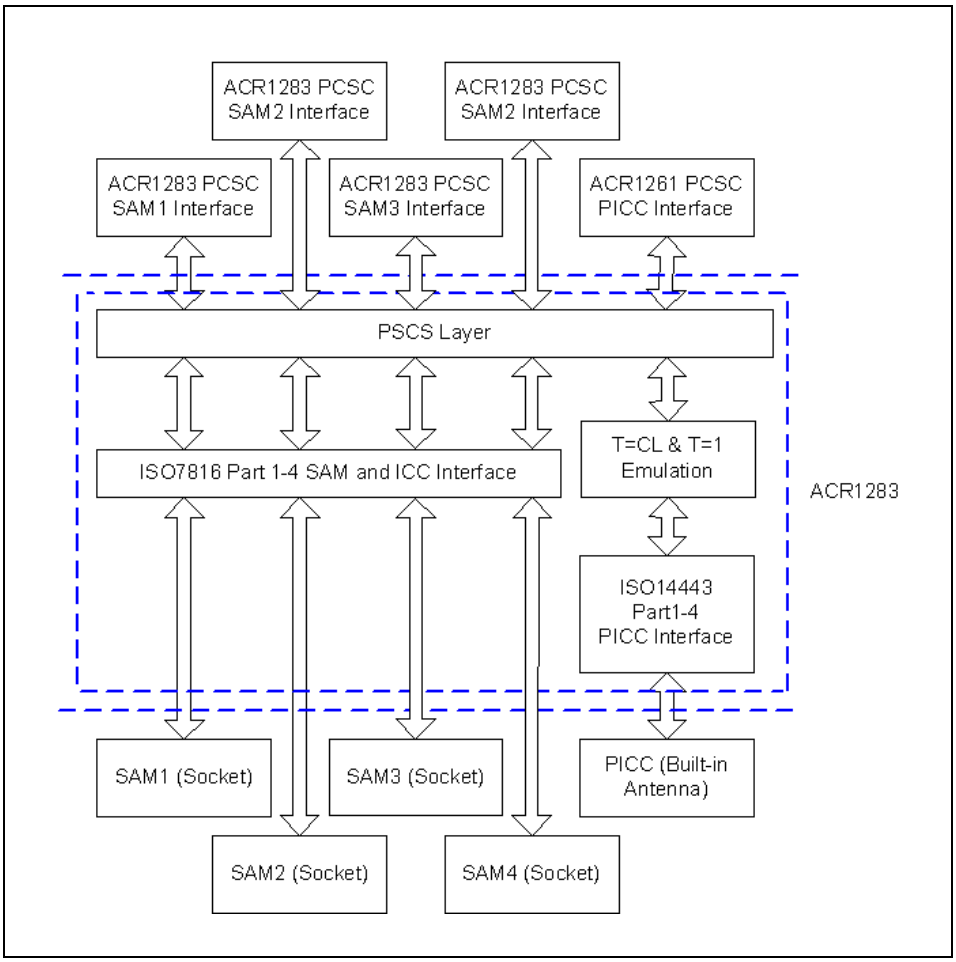

图1：ACR1283L 的架构

4.0. 硬件设计

4.1. USB

ACR1283L 通过符合 USB 标准的 USB 接口与计算机连接。

4.1.1. 通信参数

ACR1283L 通过根据 USB2.0 规范设计的 USB 接口与计算机建立连接，它支持 USB 全速模式，速率为 12Mbps。

引脚	信号	功能
1	V_{BUS}	为读写器提供+5 V 的电源。
2	D-	ACR1283L 和 PC 间以差分信号传输数据
3	D+	ACR1283L 和 PC 间以差分信号传输数据
4	GND	参考电压等级。

表1：USB 接口配线

注：*为了使 ACR1283L 通过 USB 接口正常运行，应该先安装设备驱动程序。*

4.1.2. 端点

ACR1283L 通过如下端点与主计算机进行通信：

Control Endpoint 用于设置和控制

Bulk OUT 用于从主计算机发送至 ACR1283L 的命令（数据包大小为 64 字节）

Bulk IN 用于从 ACR1283L 发送至主计算机的响应（数据包大小为 64 字节）

Interrupt IN 用于从 ACR1283L 发送至主计算机的卡片状态报文（数据包大小为 8 字节）

4.2. 接触式智能卡接口

ACR1283L 与插入的智能卡之间的接口符合 ISO 7816-3 标准协议，并进行了某些限制或提升来增强 ACR1283L 的实用功能。

4.2.1. 智能卡电源 VCC（C1）

插入的智能卡的电流消耗不得大于 50 mA。

4.2.2. 卡片类型选择

激活插入的卡片之前，处于控制地位的计算机需要向 ACR1283L 发送适当的命令来选择卡片类型。这些卡片包括存储卡和基于 MCU 的卡。

对于基于 MCU 的卡片来说，读写器允许从 T=0 或 T=1 中选择首选的协议。但是只有当插入读写器的卡片对这两种协议类型都支持时，读写器才可以通过协议与参数选择（PPS）接受并执行这样的选择。若基于 MCU 的卡仅支持一种协议类型——T=0 或 T=1 时，读写器会自动采用该协议类型，而不管应用程序选择了哪种协议。

4.2.3. 微控制器卡接口

基于微控制器的智能卡只使用触点 C1（VCC）、C2（RST)、C3（CLK）、C5（GND）和 C7

（I/O）。时钟信号（C3）的频率为 4.8 MHz。

4.3. 非接触式智能卡接口

ACR1283L 与非接触卡之间的接口符合 ISO 14443 标准协议，并进行了某些限制或提升来增强 ACR1283L 的实用功能。

4.3.1. 载波频率

ACR1283L 的载波频率为 13.56MHz。

4.3.2. 卡片轮询

ACR1283L 会自动检测进入工作场内的非接触卡。此功能支持 ISO 14443-4 的 A 类卡和 B 类卡，以及 Mifare 卡。

5.0. 软件设计

5.1. CCID 协议

ACR1283L 通过 USB 与主机端（host）建立连接。现在的行业内规范 -- CCID 标准，已经为 USB 芯片-智能卡接口设备定义了与此相关的协议。CCID 涵盖了操作智能卡和 PIN 所需的全部协议。

ACR1283L 的 USB 端点配置和使用应当符合 CCID 标准第 3 部分的规定。概述总结如下：

- **控制命令**通过控制通道（缺省通道）发送。其中包括类特定请求和 USB 标准请求。由缺省通道发送的命令会通过缺省通道向主机反馈信息。
- CCID **事件**通过中断通道发送。
- CCID **命令**由 BULK-OUT 端点发出。发送至 ACR1283L 的每个命令都有一个相关的最终响应，一些命令也可以有中间响应。
- CCID **响应**由 BULK-IN 端点发出。所有发送至 ACR1283L 的命令都必须同步发送（即：对于 ACR1283L 来说，*bMaxCCIDBusySlots* 等同于 1）。

ACR1283L 支持的 CCID 功能由其类别描述符定义：

偏移	数据域	大小	值	描述
0	*bLength*	1	36h	描述符的字节数。
1	*bDescriptorType*	1	21h	CCID 功能描述符的类别。
2	*bcdCCID*	2	0110h	CCID 以二进制编码的十进制指定的版本号码。
4	*bMaxSlotIndex*	1	04h	ACR1283L 有 5 个卡槽。
5	*bVoltageSupport*	1	07h	ACR1283L 可支持 1.8V、3.0V 和 5.0V 的槽位电压。
6	*dwProtocols*	4	00000003h	ACR1283L 支持 T=0 和 T=1 协议
10	*dwDefaultClock*	4	00000FA0h	默认 ICC 时钟频率为 4 MHz。
14	*dwMaximumClock*	4	00000FA0h	ICC 支持的最大时钟频率为 4 MHz。
18	*bNumClockSupported*	1	00h	不支持手动设置时钟频率。
19	*dwDataRate*	4	00002A00h	默认 ICC I/O 波特率为 10752 bps。
23	*dwMaxDataRate*	4	0001F808h	ICC I/O 支持的最大波特率为 344100 bps。
27	*bNumDataRatesSupported*	1	00h	不支持手动设置波特率。
28	*dwMaxIFSD*	4	00000200h	ACR1283L T1 支持的最大 IFSD 为 512。
32	*dwSynchProtocols*	4	00000000h	ACR1283L 不支持同步卡。
36	*dwMechanical*	4	00000000h	ACR1283L 不支持特殊机制特性。

偏移	数据域	大小	值	描述
40	*dwFeatures*	4	00040040h	ACR1283L 支持以下特性： • CCID 自动进行参数协商 • 与 CCID 进行短 APDU 级和扩展 APDU 级交换
44	*dwMaxCCIDMessageLength*	4	0000020Ah	ACR1283L 可接受的最大报文长度为 522 字节。
48	*bClassGetResponse*	1	00h	表示 CCID 回应 APDU 的类别。
49	*bClassEnvelope*	1	00h	表示 CCID 回应 APDU 的类别。
50	*wLCDLayout*	2	0000h	无 LCD。
52	*bPINSupport*	1	00h	无 PIN 校验。
53	*bMaxCCIDBusySlots*	1	01h	可以同时忙的槽位数为 1。

5.2. CCID 命令

5.2.1. CCID 命令通道，Bulk-OUT 消息

ACR1283L 应当遵循 CCID 协议第四部分定义的 Bulk-OUT 消息。此外，该规范还定义了一些用于操作附加功能的扩展命令。此节列举了 ACR1283L 支持的 CCID 类 Bulk-OUT 消息。

5.2.1.1. PC_to_RDR_IccPowerOn

此命令用于激活卡槽并返回卡片的 ATR。

偏移	数据域	大小	值	描述
0	*bMessageType*	1	62h	
1	*dwLength*	4	00000000h	此消息的额外字节的大小。
2	*bSlot*	1		标识命令的插槽号。
5	*bSeq*	1		命令的序号。
6	*bPowerSelect*	1		ICC 上的电压值 00h = 自动电压选择 01h = 5 V 02h = 3 V
7	*abRFU*	2		保留为将来使用。

此消息的响应是 *RDR_to_PC_DataBlock* 消息，返回的数据是复位应答（ATR）。

5.2.1.2. PC_to_RDR_IccPowerOff

此命令用于取消激活卡槽。

偏移	数据域	大小	值	描述
0	*bMessageType*	1	63h	

偏移	数据域	大小	值	描述
1	*dwLength*	4	00000000h	此消息的额外字节的大小。
5	*bSlot*	1		标识命令的插槽号。
6	*bSeq*	1		命令的序号。
7	*abRFU*	3		保留为将来使用。

此消息的响应是 *RDR_to_PC_SlotStatus* 消息。

5.2.1.3. PC_to_RDR_GetSlotStatus

此命令用于获取当前的卡槽状态。

偏移	数据域	大小	值	描述
0	*bMessageType*	1	65h	
1	*dwLength*	4	00000000h	此消息的额外字节的大小。
5	*bSlot*	1		标识命令的插槽号。
6	*bSeq*	1		命令的序号。
7	*abRFU*	3		保留为将来使用。

此消息的响应是 *RDR_to_PC_SlotStatus* 消息。

5.2.1.4. PC_to_RDR_XfrBlock

此命令用于向 ICC 传输数据块。

偏移	数据域	大小	值	描述
0	*bMessageType*	1	6Fh	
1	*dwLength*	4		此消息的 abData 数据域的大小
5	*bSlot*	1		标识命令的插槽号
6	*bSeq*	1		命令的序号
7	*bBWI*	1		用于为当前传输延长 CCID 块的超时等待时间。“该数值乘以块等待时间”的时间段过去后，CCID 将超时该块。
8	*wLevelParameter*	2	0000h	RFU (TPDU 交换级别)
10	*abData*	Byte array		送至 CCID 的数据块。信息是“按原样”发送至 ICC（TPDU 交换级别）

此消息的响应是 *RDR_to_PC_DataBlock* 消息。

5.2.1.5. PC_to_RDR_Escape

此命令用于访问扩展特性。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	6Bh	
1	*dwLength*	4		此消息的 *abData* 数据域的大小
5	*bSlot*	1		标识命令的插槽号。
6	*bSeq*	1		命令的序号。
7	*abRFU*	3		保留为将来使用
10	*abData*	Byte array		送至 CCID 的数据块。

此命令消息的响应是 *RDR_to_PC_Escape* 消息。

5.2.1.6. PC_to_RDR_GetParameters

此命令用于获取卡槽参数。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	6Ch	
1	*dwLength*	4	00000000h	此消息的额外字节的大小。
5	*bSlot*	1		标识命令的插槽号。
6	*bSeq*	1		命令的序号。
7	*abRFU*	3		保留为将来使用。

此消息的响应是 *RDR_to_PC_Parameters* 消息。

5.2.1.7. PC_to_RDR_ResetParameters

此命令用于将卡槽参数重置为默认值。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	6Dh	
1	*DwLength*	4	00000000h	此消息的额外字节的大小。
5	*BSlot*	1		标识命令的插槽号。
6	*BSeq*	1		命令的序号。
7	*AbRFU*	3		保留为将来使用。

此消息的响应是 *RDR_to_PC_Parameters* 消息。

5.2.1.8. PC_to_RDR_SetParameters

此命令用于设置卡槽参数。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	61h	
1	*dwLength*	4		此消息的额外字节的大小。
5	*bSlot*	1		标识命令的插槽号。
6	*bSeq*	1		命令的序号。
7	*bProtocolNum*	1		指定后面的协议数据结构。 00h = T=0 协议的结构 01h = T=1 协议的结构 以下值保留为将来使用： 80h = 2 线协议结构 81h = 3 线协议结构 82h = I2C 协议结构
8	*abRFU*	2		保留为将来使用。
10	*abProtocolDataStructure*	Byte array		协议数据结构。

T=0 协议的协议数据结构（*dwLength*=00000005h）

偏移	数据域	大小	值	说明
10	*bmFindexDindex*	1		B7-4 – FI – ISO/IEC 7816-3:1997 中表 7 的索引，选择一个时钟频率转换因子 B3-0 – DI - ISO/IEC 7816-3:1997 中表 8 的索引，选择一个波特率转换因子
11	*bmTCCKST0*	1		B0 – 0b, B7-2 – 000000b B1 – 使用的约定（b1=0：正向约定；b1=1：反向约定）***注：CCID 忽略该位。***
12	*bGuardTimeT0*	1		两个字符间的额外保护时间。在通常的保护时间（12etu）基础上增加 0-254 个 etu。FFh 与 00h 相同。
13	*bWaitingIntegerT0*	1		WI for T=0，用于定义 WWT
14	*bClockStop*	1		支持 ICC 时钟停止。 00h = 不允许停止时钟 01h = 时钟信号为低时停止 02h = 时钟信号为高时停止 03h = 时钟信号为高或为低时停止

T=1 协议的协议数据结构(*dwLength*=00000007h)

偏移	数据域	大小	值	说明
10	*bmFindexDindex*	1		B7-4 – FI – ISO/IEC 7816-3:1997 中表 7 的索引，选择一个时钟频率转换因子 B3-0 – DI - ISO/IEC 7816-3:1997 中表 8 的索引，选择一个波特率转换因子
11	*BmTCCKST1*	1		B7-2 – 000100b B0 – 校验和的类型（b0=0：LRC； b0=1：CRC） B1 – 使用的约定（b1=0：正向约定；b1=1：反向约定）***注：CCID 忽略该位。***
12	*BGuardTimeT1*	1		额外保护时间（两个字符间为 0-254 个 etu）若值为 FFh，则保护时间减少 1 个 etu。
13	*BwaitingIntegerT1*	1		B7-4 = BWI 值 0-9 有效 B3-0 = CWI 值 0-Fh 有效
14	*bClockStop*	1		支持 ICC 时钟停止。 00h = 不允许停止时钟 01h = 时钟信号为低时停止 02h = 时钟信号为高时停止 03h = 时钟信号为高或为低时停止
15	*bIFSC*	1		商定的 IFSC 的大小
16	*bNadValue*	1	00h	只支持 NAD = 00h

此消息的响应是 *RDR_to_PC_Parameters* 消息。

5.2.2. CCID 响应通道，Bulk-IN 消息

Bulk-IN 消息用于对 Bulk-OUT 消息做出响应。ACR1283L 应当遵循 CCID 协议第四部分定义的 Bulk-IN 消息。此节列举了 ACR1283L 支持的 CCID 类 Bulk-IN 消息。

5.2.2.1. RDR_to_PC_DataBlock

此消息由 ACR1283L 发出，是对 *PC_to_RDR_IccPowerOn*、*PC_to_RDR_XfrBlock* 和 *PC_to_RDR_Secure* 消息的响应。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	80h	表示正在从 CCID 发送一个数据块。
1	*dwLength*	4		此消息的额外字节的大小。
5	*bSlot*	1		与 Bulk-OUT 消息中的值相同。
6	*bSeq*	1		与 Bulk-OUT 消息中的值相同。
7	*bStatus*	1		CCID 规范 4.2.1 节定义的插槽状态寄存器。
8	*bError*	1		CCID 规范 4.2.1 节和本规范 5.2.8 节定义的插槽错误寄存器。

偏移	数据域	大小	值	说明
9	*bChainParameter*	1	00h	RFU（TPDU 交换级别）。
10	*abData*	Byte array		本数据域包含由 CCID 返还的数据。

5.2.2.2. RDR_to_PC_Escape

此消息由 ACR1283L 发出，是对 *PC_to_RDR_Escape* 消息的响应。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	83h	
1	*dwLength*	4		此消息的 *abData* 数据域的大小
5	*bSlot*	1		与 Bulk-OUT 消息中的值相同。
6	*bSeq*	1		与 Bulk-OUT 消息中的值相同。
7	*bStatus*	1		CCID 规范 4.2.1 节定义的插槽状态寄存器。
8	*bError*	1		CCID 规范 4.2.1 节和本规范 5.2.8 节定义的插槽错误寄存器。
9	*bRFU*	1	00h	RFU.
10	*abData*	Byte array		本数据域包含由 C 的 CID 返还的数据。

5.2.2.3. RDR_to_PC_SlotStatus

此消息由 ACR1283L 发出，是对 *PC_to_RDR_IccPowerOff*、*PC_to_RDR_GetSlotStatus* 和 *PC_to_RDR_Abort* 消息，以及类特定 ABORT 请求的响应。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	81h	
1	*dwLength*	4	00000000h	此消息的额外字节的大小。
5	*bSlot*	1		与 Bulk-OUT 消息中的值相同。
6	*bSeq*	1		与 Bulk-OUT 消息中的值相同。
7	*bStatus*	1		CCID 规范 4.2.1 节定义的插槽状态寄存器。
8	*bError*	1		CCID 规范 4.2.1 节和本规范 5.2.8 节定义的插槽错误寄存器。
9	*bClockStatus*	1		值： 00h = 时钟运行 01h = 时钟停于 L 状态 02h = 时钟停于 H 状态 03h = 时钟停止于未知状态 所有其他值保留为将来使用。

5.2.2.4. RDR_to_PC_Parameters

此消息由 ACR1283L 发出，是对 *PC_to_RDR_GetParameters*、*PC_to_RDR_ResetParameters* 和 *PC_to_RDR_SetParameters* 消息的响应。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	82h	
1	*dwLength*	4		此消息的额外字节的大小。
5	*bSlot*	1		与 Bulk-OUT 消息中的值相同。
6	*bSeq*	1		与 Bulk-OUT 消息中的值相同。
7	*bStatus*	1		CCID 规范 4.2.1 节定义的插槽状态寄存器。
8	*bError*	1		CCID 规范 4.2.1 节和本规范 5.2.8 节定义的插槽错误寄存器。
9	*bProtocolNum*	1		指定后面的协议数据结构。 00h = T=0 协议的结构 01h = T=1 协议的结构 以下值保留为将来使用： 80h = 2 线协议结构 81h = 3 线协议结构 82h = I2C 协议结构
10	*abProtocolDataStructure*	Byte array		协议数据结构如 5.2.3 节汇总

5.3. 非接触式智能卡协议

5.3.1. ATR 的生成

读写器检测到 PICC 后，一个 ATR 会被发送至 PCSC 驱动来识别 PICC。

5.3.1.1. ATR 信息格式（适用于 ISO 14443-3 PICC）

字节	值（十六进制）	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（HistByte 0 - HistByte N-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

字节	值（十六进制）	标记	说明
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 数据对象中可能存在状态指示。
	4F	Tk	应用标识符存在标识。
	0C		长度。
	RID		注册的应用提供商标识(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 的所有字符按位异或

例如：

Mifare 1K 卡的 ATR = {3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 01 00 00 00 00 6Ah}

长度 (YY) = 0x0Ch

RID = {A0 00 00 03 06h} (PC/SC 工作组)

标准(SSh) = 03h (ISO 14443A，第 3 部分)

卡片名称(C0 ..C1h) = {00 01h} (Mifare 1K)

卡片名称(C0 ..C1)

00 01：Mifare 1K　　FF 28：JCOP 30

00 02：Mifare 4K　　FF [SAK]：未定义的标签

00 03：Mifare Ultralight

00 26：Mifare Mini

5.3.1.2. ATR 信息格式（适用于 ISO 14443-4 PICC）

字节	值（十六进制）	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（HistByte 0 - HistByte N-1)

<table>
<tr><th>字节</th><th>值（十六进制）</th><th>标记</th><th>说明</th></tr>
<tr><td>2</td><td>80</td><td>TD1</td><td>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</td></tr>
<tr><td>3</td><td>01</td><td>TD2</td><td>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</td></tr>
<tr><td rowspan="2">4 至 3 + N</td><td>XX</td><td>T1</td><td rowspan="2">历史字节：

ISO 14443-A：
来自 ATS 响应的历史字节。参考 ISO 14443-4 标准。

ISO 14443-B：<table><tr><th>Byte1-4</th><th>Byte5-7</th><th>Byte8</th></tr><tr><td>ATQB 的应用数据</td><td>ATQB 的协议信息字符</td><td>高半字节 =ATTRIB 命令的 MBLI；低半字节 (RFU)=0</td></tr></table></td></tr>
<tr><td>XX
XX
XX</td><td>Tk</td></tr>
<tr><td>4+N</td><td>UU</td><td>TCK</td><td>T0 至 Tk 的所有字符按位异或</td></tr>
</table>

例 1：

DESFire 的 ATR = { 3B 81 80 01 80 80h } // 6 个字节的 ATR

注：使用 APDU "FF CA 01 00 00h"来区分是符合 ISO 14443A-4 的 PICC 还是符合 ISO 14443B-4 的 PICC，并且如果有的话，取回完整的 ATS。符合 ISO 14443A-3 或 ISO 14443B-3/4 类的 PICC 会返回 ATS。

APDU 命令 = FF CA 01 00 00h

APDU 响应 = 06 75 77 81 02 80 90 00h

ATS = {06 75 77 81 02 80h}

例 2：

EZ-Link 的 ATR = {3B 88 80 01 1C 2D 94 11 F7 71 85 00 BEh}

ATQB 的应用数据 = 1C 2D 94 11h

ATQB 的协议信息 = F7 71 85h

ATTRIB 的 MBLI = 00h

5.3.2. 非接触接口的私有 APDU 指令

5.3.2.1. GET DATA 命令

此命令用于获取“已建立连接的 PICC”的序列号或 ATS。

GET UID 命令的 APDU 结构（5 个字节）

命令	CLA	INS	P1	P2	Le
Get Data	FFh	CAh	00h 01h	00h	00h (最大长度)

若 P1 = 0x00h，Get UID 的响应报文结构（UID + 2 个字节）

响应	响应数据域					
结果	UID (LSB)	…	…	UID (MSB)	SW1	SW2

如果 ***P1 = 0x01h，***获取 ISO 14443 A 类卡的 ATS（ATS + 2 个字节）

响应	响应数据域		
结果	ATS	SW1	SW2

响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
警告	62 82h	UID/ATS 的长度小于 Le（Le 大于 UID 的长度）。
错误	6C XXh	长度错误（错误的 *Le*：‘XX’表示确切的数字），如果 Le 小于 UID 的长度。
错误	63 00h	操作失败。
错误	6A 81h	不支持此功能。

例如：

// 获取“已经建立连接的 PICC”的序列号

UINT8 GET_UID[5]={0xFFh, 0xCAh, 0x00h, 0x00h, 0x00h};

//获取“已经建立连接的 ISO14443-A PICC”的 ATS

UINT8 GET_ATS[5]={0xFFh, 0xCAh, 0x01h, 0x00h, 0x00h};

5.3.2.2. Mifare 1K/4K 存储卡的 PICC 命令（T=CL 模拟）

5.3.2.2.1. LOAD AUTHENTICATION KEYS 命令

此命令用于向读写器加载认证密钥。该认证密钥用于验证 Mifare 1K/4K 存储卡的特定扇区。读写器提

供了两种认证密钥位置，分别是易失密钥位置和非易失密钥位置。

Load Authentication Keys 命令的 APDU 结构（11 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Load Authentication Keys	FFh	82h	密钥结构	密钥号	06h	密钥（6 个字节）

其中：

密钥结构（1 个字节）： 0x00h = 密钥被载入读写器的易失性存储器

0x20h = 密钥被载入读写器的非易失性存储器

其它 = 保留

密钥号（1 个字节）：

0x00h ~ 0x1Fh = 用于存储密钥的非易失性存储器。密钥被永久地存在读写器中，即使读写器与电脑断开连接也不会消失。读写器的非易失性存储器内可以存储最多 32 个密钥。

0x20h（过程密钥）= 用于存储临时密钥的易失性存储器。一旦读写器与电脑断开连接，密钥就会消失。易失密钥只有一个，可以用作不同会话的过程密钥。

注：默认值 = {FF FF FF FF FF FFh}

密钥（6 个字节）： 载入读写器的密钥值，例如：{FF FF FF FF FF FFh}

Load Authentication Keys 命令的响应结构（2 个字节）

响应	响应数据域	
结果	SW1	SW2

Load Authentication Keys 命令的响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

例如：

// 向非易失性存储器位置 0x05h 加载密钥 {FF FF FF FF FF FFh}。

APDU = {FF 82 20 05 06 FF FF FF FF FF FFh}

// 向易失性存储器位置 0x20h 加载密钥 {FF FF FF FF FF FFh}。

APDU = {FF 82 00 20 06 FF FF FF FF FF FFh}

注：

1. *基本上，应用程序需要了解所有正在被使用的密钥。出于安全方面的考虑，建议将所有需要的*

密钥存储在非易失性存储器内。易失性存储器和非易失性存储器的内容都无法从外部读取。

2. *直到读写器复位或下电，易失性存储器的内容"过程密钥 0x20h"才会失效。过程密钥适于存储经常变化的密钥值。它们被存储在"内部 RAM"中。而非易失密钥被存储在"EEPROM"中。EEPROM 相对于内部 RAM 存储速度稍慢。*
3. *我们不建议使用"非易失密钥位置 0x00h ~ 0x1Fh"来存储任何经常变化的"临时密钥值"。"非易失密钥"主要是用于存储不经常变化的"密钥值"。如果"密钥值"会不时的变化，请将其存储在"易失密钥位置 0x020h"。*

5.3.2.2.2. Authentication for Mifare 1K/4K 命令

此命令使用存储在读写器内的密钥来验证 Mifare 1K/4K 卡（PICC）。其中会用到两种认证密钥：TYPE_A 和 TYPE_B。

Load Authentication Keys 命令的 APDU 结构（6 个字节）（弃用）

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

Load Authentication Keys 命令的 APDU 结构（10 个字节）

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 个字节）

字节 1	字节 2	字节 3	字节 4	字节 5
版本 0x01h	0x00h	块号	密钥类型	密钥号

其中：

块号（1 个字节）： 待验证的存储块。

Mifare 1K 卡的内存划分为 16 个扇区，每个扇区包含 4 个连续的块。例如，扇区 0x00h 包含块{0x00h、0x01h、0x02h 和 0x03h}；扇区 0x01h 包含块{0x04h、0x05h、0x06h 和 0x07h}；最后一个扇区 0x0Fh 包含块{0x3Ch、0x3Dh、0x3Eh 和 0x3Fh}。验证通过后，读取同一扇区内的其他块不需要再次进行验证。详情请参考 Mifare 1K/4K 卡标准。

***注：** 一旦该块被成功验证，即可访问属于同一扇区的所有块。*

密钥类型（1 个字节）： 0x60h = 该密钥被用作 TYPE A 密钥进行验证

0x61h = 该密钥被用作 TYPE B 密钥进行验证

密钥号（1 个字节）：

0x00h ~ 0x1Fh = 用于存储密钥的非易失性存储器。密钥被永久地存在读写器中，即使读写器与电脑断开连接也不会消失。读写器的非易失性存储器可以存储 32 个密钥。

0x20h（过程密钥） = 用于存储密钥的易失性存储器。读写器与电脑断开连接后，密钥即会消失。易失密钥只有一个，易失密钥可以用作不同会话的过程密钥。

Load Authentication Keys 的响应结构（2 个字节）

响应	响应数据域	
结果	SW1	SW2

Load Authentication Keys 命令的响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

扇区 (共 16 个扇区，每个扇区包含 4 个连续的块)	数据块 (3 个块，每块 16 个字节)	尾部块 (1 个块，16 个字节)
扇区 0	0x00h ~ 0x02h	0x03h
扇区 1	0x04h ~ 0x06h	0x07h
..		
..		
扇区 14	0x38h ~ 0x0Ah	0x3Bh
扇区 15	0x3Ch ~ 0x3Eh	0x3Fh

1K 字节

表2：Mifare 1K 卡的内存结构

扇区 (共 32 个扇区，每个扇区包含 4 个连续的块)	数据块 (3 个块，每块 16 个字节)	尾部块 (1 个块，16 个字节)
扇区 0	0x00h ~ 0x02h	0x03h
扇区 1	0x04h ~ 0x06h	0x07h
..		
..		
扇区 30	0x78h ~ 0x7Ah	0x7Bh
扇区 31	0x7Ch ~ 0x7Eh	0x7Fh

2K 字节

扇区 (共 8 个扇区，每个扇区包含 16 个连续的块)	数据块 (15 个块，每块 16 个字节)	尾部块 (1 个块，16 个字节)

扇区 (共 8 个扇区，每个扇区包含 16 个连续的块)	数据块 (15 个块，每块 16 个字节)	尾部块 (1 个块，16 个字节)
扇区 32	0x80h ~ 0x8Eh	0x8Fh
扇区 33	0x90h ~ 0x9Eh	0x9Fh
..		
..		
扇区 38	0xE0h ~ 0xEEh	0xEFh
扇区 39	0xF0h ~ 0xFEh	0xFF

2K 字节

表3：Mifare 4K 卡的内存结构

例如：

// 要使用{TYPE A，密钥号 0x00h}验证块 0x04h。

// PC/SC V2.01，弃用

APDU = {FF 88 00 04 60 00h};

// 要使用{TYPE A，密钥号 0x00h}验证块 0x04h。

// PC/SC V2.07

APDU = {FF 86 00 00 05 01 00 04 60 00h}

注：Mifare Ultralight 不需要进行验证，其内存可以自由访问。

字节号	0	1	2	3	页
序列号	SN0	SN1	SN2	BCC0	0
序列号	SN3	SN4	SN5	SN6	1
内部 / 锁	BCC1	Internal	Lock0	Lock1	2
OTP	OPT0	OPT1	OTP2	OTP3	3
数据读/写	Data0	Data1	Data2	Data3	4
数据读/写	Data4	Data5	Data6	Data7	5
数据读/写	Data8	Data9	Data10	Data11	6
数据读/写	Data12	Data13	Data14	Data15	7
数据读/写	Data16	Data17	Data18	Data19	8
数据读/写	Data20	Data21	Data22	Data23	9
数据读/写	Data24	Data25	Data26	Data27	10
数据读/写	Data28	Data29	Data30	Data31	11
数据读/写	Data32	Data33	Data34	Data35	12
数据读/写	Data36	Data37	Data38	Data39	13
数据读/写	Data40	Data41	Data42	Data43	14
数据读/写	Data44	Data45	Data46	Data47	15

512 位
或
64 字节

表4：Mifare Ultralight 卡的内存结构

5.3.2.2.3. READ BINARY BLOCKS 命令

此命令用于从 PICC 卡片中取回多个"数据块"。执行 *Read Binary Blocks* 命令前，必须先对数据块/尾部块进行验证。

Read Binary 命令的 APDU 结构（5 个字节）

命令	CLA	INS	P1	P2	Le
Read Binary Blocks	FFh	B0h	00h	块号	待读取的字节数

其中：

块号（1 个字节）： 起始块。

待读取的字节数（1 个字节）：

MIFARE 1K/4K 卡的待读字节的长度应是 16 字节的倍数；Mifare Ultralight 卡应是 4 字节的倍数。

- Mifare Ultralight 卡的待读字节数最大为 16。
- Mifare 1K 卡的待读字节数最大为 48。（多块模式；3 个连续的块）
- Mifare 4K 卡的待读字节数最大为 240。（多块模式；15 个连续的块）

例 1：0x10h（16 个字节）。仅起始块。（单块模式）

例 2：0x40h（64 个字节）。从起始块至起始+3 块。（多块模式）

注：*出于安全因素考虑，多块模式仅用于访问数据块。尾部块不能在多块模式下被访问，请使用单块模式对其进行访问。*

Read Binary Block 命令的响应结构（4/16 的倍数 + 2 个字节）

响应	响应数据域		
结果	数据（4/16 字节的倍数）	SW1	SW2

Read Binary Block 命令的响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

例如：

// 从二进制块 0x04h 中读取 16 个字节（Mifare 1K 或 4K）

APDU = {FF B0 00 04 10h}

从二进制块 0x80h 开始读取 240 个字节（Mifare 4K）

// 块 0x80h——块 0x8Eh（15 个块）

APDU = {FF B0 00 80 F0h}

5.3.2.2.4. UPDATE BINARY BLOCKS 命令

此命令用于向 PICC 卡写入多个"数据块"。执行 *Update Binary Blocks* 命令前，必须先对数据块/尾部块进行验证。

Update Binary 命令的 APDU 结构（16 的倍数 + 5 个字节）

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 (16 字节的倍数)

其中：

块号（1 个字节）： 待更新的起始块

待更新的字节数（1 个字节）：

- MIFARE 1K/4K 卡的待更新字节的长度应该是 16 字节的倍数；Mifare Ultralight 卡是 4 字节的倍数。
- Mifare 1K 卡的待更新字节数最大为 48（多块模式；3 个连续的块）

- Mifare 4K 卡的待更新字节数最大为 240（多块模式；15 个连续的块）

例 1：0x10h（16 个字节）。仅起始块。（单块模式）

例 2：0x30h（48 个字节）。从起始块至起始+2 块。（多块模式）

注：*出于安全因素考虑，多块模式仅用于访问数据块。尾部块不能在多块模式下被访问，请使用单块模式对其进行访问。*

块数据（16 的倍数 + 2 个字节，或 6 个字节）：待写入二进制块的数据。

Update Binary Block 命令的响应状态码（2 个字节）

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

例如：

// 将 Mifare 1K/4K 卡中的二进制块 0x04h 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 0Fh}

// 将 Mifare Ultralight 卡中的二进制块 0x04h 的数据更新为{00 01 02 03h}

APDU = {FF D6 00 04 04 00 01 02 03h}

5.3.2.2.5. VALUE BLOCK OPERATION（INC, DEC, STORE）命令

此命令用于对基于数值的交易进行操作（例如：增加值块的值等）。

Value Block Operation 命令的 APDU 结构（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 个字节）{MSB ..LSB}

其中：

块号（1 个字节）： 待操作的值块

VB_OP（1 个字节）： 0x00h =将 VB_Value 存入该块，然后该块变为一个值块。

0x01h =使值块的值增加 VB_Value。此命令仅适用于对值块的操作。

0x02h =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 个字节）： 用于算数运算的数值，是一个有符号长整数（4 个字节）。

例 1： Decimal –4 = {0xFFh, 0xFFh, 0xFFh, 0xFCh}

VB_Value			
MSB			LSB
FFh	FFh	FFh	FCh

例 2： Decimal 1 = {0x00h, 0x00h, 0x00h, 0x01h}

VB_Value			
MSB			LSB
00h	00h	00h	01h

Value Block Operation 命令的响应结构（2 个字节）

响应	响应数据域	
结果	SW1	SW2

Value Block Operation 响应码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

5.3.2.2.6. READ VALUE BLOCK 命令

此命令用于获取值块中的数值，仅适用于对值块的操作。

Read Value Block 命令的 APDU 结构（5 个字节）

命令	CLA	INS	P1	P2	Le
Read Value Block	FFh	B1h	00h	块号	00h

其中：

块号（1 个字节）： 待访问的值块

Read Value Block 的响应报文结构（4+2 个字节）

响应	响应数据域		
结果	值 {MSB ..LSB}	SW1	SW2

其中：

值（4 个字节）： 卡片返回的数值，是一个有符号长整数（4 个字节）。

例 1： Decimal –4 = {0xFFh, 0xFFh, 0xFFh, 0xFCh}

值			
MSB			LSB
FFh	FFh	FFh	FCh

例 2： Decimal 1 = {0x00h, 0x00h, 0x00h, 0x01h}

值			
MSB			LSB
00h	00h	00h	01h

Read Value Block 命令的响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

5.3.2.2.7. COPY VALUE BLOCK 命令

此命令用于将一个值块中的数值复制到另外一个值块。

Copy Value Block 命令的 APDU 结构（7 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域	
Value Block Operation	FFh	D7h	00h	源块号	02h	03h	目标块号

其中：

源块号（1 个字节）： 源值块中的值会被复制到目标值块。

目标块号（1 个字节）： 要恢复的值块。源值块和目标值块必须位于同一个扇区。

Copy Value Block 的响应报文结构（4+2 个字节）

响应	响应数据域	
结果	SW1	SW2

Copy Value Block 命令的响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

例如：

// 将数值“1”存入块 0x05h

APDU = {FF D7 00 05 05 00 00 00 00 01h}

// 读取值块 0x05h

APDU = {FF B1 00 05 00h}

将值块 0x05h 的值复制到值块 0x06h

APDU = {FF D7 00 05 02 03 06h}

// 使值块 0x05h 的值增加“5”

APDU = {FF D7 00 05 05 01 00 00 00 05h}

5.3.2.3. 访问符合 PCSC 标准的标签（ISO 14443-4）

基本上，所有符合 ISO 14443-4 标准的卡片（PICC）都可以理解符合 ISO 7816-4 规定的 APDU。ACR1283L 读写器与符合 ISO 14443-4 标准的卡片进行通信时，需要对 ISO 7816-4 规定的 APDU 和响应进行转换。它会在内部处理 ISO 14443 第 1-4 部分的协议。

另外 Mifare 1K, 4K, MINI 和 Ultralight 标签是通过 T=CL 模拟进行支持的，只要将 Mifare 标签视作标准的 ISO 14443-4 标签即可。

ISO 7816-4 规定的 APDU 报文的结构

命令	CLA	INS	P1	P2	Lc	命令数据域	Le
ISO 7816 第 4 部分规定的命令					命令数据域的长度		期望返回的响应数据的长度

ISO 7816-4 规定的响应报文的结构（数据 + 2 个字节）

响应	响应数据域		
结果	响应数据	SW1	SW2

通用的 ISO 7816-4 命令的响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

典型的操作顺序为：

1. 出示标签，并连接 PICC 界面。
2. 读取/更新标签的存储内容。

步骤 1：与标签建立连接。

标签的 ATR 为 3B 88 80 01 00 00 00 00 33 81 81 00 3Ah

其中，

ATQB 应用数据 = 00 00 00 00h，ATQB 协议信息 = 33 81 81h。这是一个 ISO 14443-4 Type B 标签。

步骤 2：发送 APDU，取随机数

<< 00 84 00 00 08h

>> 1A F7 F3 1B CD 2B A9 58h [90 00h]

注：*对于 ISO 14443-4 Type A 标签来说，可以通过 APDU"FF CA 01 00 00h"来获取 ATS。*

例如：

// 从 ISO 14443-4 Type B PICC (ST19XR08E)中读取 8 个字节

APDU ={80 B2 80 00 08h}

CLA = 0x80h

INS = 0xB2h

P1 = 0x80h

P2 = 0x00h

Lc = None

命令数据域 = 无

Le = 0x08h

应答： 00 01 02 03 04 05 06 07h [$9000h]

5.4. 外设控制

读写器的外设控制命令通过使用 *PC_to_RDR_Escape* 来实现。Escape 命令的供应商 IOCTL 是 3500 。

5.4.1. GET FIRMWARE VERSION 命令

此命令用于获取读写器的固件信息。

Get Firmware Version 命令的结构

命令	CLA	INS	P1	P2	Lc
Get Firmware Version	0xE0h	0x00h	0x00h	0x18h	0x00h

Get Firmware Version 命令的响应结构

响应	CLA	INS	P1	P2	Le	响应数据域
结果	0xE1h	0x00h	0x00h	0x00h	待收到的字节数	固件版本

5.4.2. SET DEFAULT LED AND BUZZER BEHAVIORS 命令

此命令用于设置 LED 和蜂鸣器的默认行为。

Set Default LED and Buzzer Behaviors 命令的结构（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Set Default LED and Buzzer Behaviors	0xE0h	0x00h	0x00h	0x21h	0x01h	默认行为

默认行为（1 个字节）

默认行为	模式	说明
Bit 0	RFU	RFU
Bit 1	PICC 轮询状态 LED	显示 PICC 轮询状态 1 = 启用；0 = 禁用
Bit 2	PICC 激活状态 LED	显示 PICC 界面的激活状态 1 = 启用；0 = 禁用
Bit 3	卡片插入和卡片移出事件蜂鸣器	每次检测到卡片插入或者卡片移出就会发出哔的一声。 1 = 启用；0 = 禁用
Bit 4 - 6	RFU	RFU
Bit 7	卡片操作闪烁 LED	使 LED 在卡片被访问时会闪烁。

注：*默认行为的默认值 = 0x08h*

Set Default LED and Buzzer Behaviors 命令的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	0xE1h	0x00h	0x00h	0x00h	0x01h	默认行为

5.4.3. READ DEFAULT LED AND BUZZER BEHAVIORS 命令

此命令用于读取 LED 和蜂鸣器的当前默认行为。

Read Default LED and Buzzer Behaviors 命令的结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read Default LED and Buzzer Behaviors	0xE0h	0x00h	0x00h	0x21h	0x00h

Read Default LED and Buzzer Behaviors 命令的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	0xE1h	0x00h	0x00h	0x00h	0x01h	默认行为

默认行为（1 个字节）

默认行为	模式	说明
Bit 0	RFU	RFU
Bit 1	PICC 轮询状态 LED	显示 PICC 轮询状态 1 = 启用；0 = 禁用
Bit 2	PICC 激活状态 LED	显示 PICC 界面的激活状态 1 = 启用；0 = 禁用
Bit 3	卡片插入和卡片移出事件蜂鸣器	每次检测到卡片插入或者卡片移出就会发出哔的一声。 1 = 启用；0 = 禁用
Bit 4 - 6	RFU	RFU
Bit 7	卡片操作闪烁 LED	使 LED 在卡片被访问时会闪烁。

注：*默认行为的默认值 = 0x08h*

5.4.4. SET AUTOMATIC PICC POLLING 命令

此命令用于设置读写器的轮询模式。

每当读写器连接到电脑上，读写器的 PICC 轮询功能就会启动 PICC 扫描，以确定是否有 PICC 被放置于/移出了内置天线的范围。

我们可以发送一个命令来停用 PICC 轮询功能。该命令通过 PC/SC Escape Command 界面发送。为了满足节能要求，PICC 闲置，或者找不到 PICC 的时候，我们提供了几种关闭天线场的特殊模式。在省电模式下，读写器会消耗更低的电能。

Set Automatic PICC Polling 命令的结构

命令	CLA	INS	P1	P2	Lc	命令数据域
Set Automatic PICC Polling	0xE0h	0x00h	0x00h	0x23h	0x01h	轮询设置

Set Automatic PICC Polling 命令的响应结构

响应	CLA	INS	P1	P2	Le	响应数据域
结果	0xE1h	0x00h	0x00h	0x00h	0x01h	轮询设置

其中：

轮询设置： 默认值 = 8Fh（1 个字节）

轮询设置	说明	说明
Bit 0	自动 PICC 轮询	1 = 启用 0 = 停用
Bit 1	如果没有找到 PICC，关闭天线场	1 = 启用 0 = 停用
Bit 2	如果 PICC 闲置，关闭天线场	1 = 启用 0 = 停用
Bit 3	RFU	RFU
Bit 5 – 4	PICC 轮询间隔	Bit 5 – Bit 4： 0 – 0 = 250 毫秒 0 – 1 = 500 毫秒 1 – 0 = 1000 毫秒 1 – 1 = 2500 毫秒
Bit 6	RFU	RFU
Bit 7	执行 ISO14443A 第 4 部分	1 = 启用 0 = 停用

注：

1. *建议启用"如果 PICC 闲置，关闭天线场"选项，这样闲置的 PICC 就不会一直暴露在天线场中，*

可以防止 PICC“发热”。

2. *PICC 轮询间隔时间越长，节能效果越好。然而，PICC 轮询的响应时间也会增加。在节能状态下，空闲时的电流消耗约为 60 mA；而在非节能状态下，空闲时的电流消耗约为 130 mA。空闲时的电流消耗=PICC 处于闲置状态。*
3. *读写器会自动激活“ISO14443A-4 PICC”的 ISO 14443A-4 模式。B 类的 PICC 不会受此选项影响。*
4. *JCOP30 卡片有两种模式：ISO 14443A-3（Mifare 1K）和 ISO 14443A-4 模式。一旦 PICC 被激活，应用必须要选定一种模式。*

5.4.5. READ AUTOMATIC PICC POLLING 命令

此命令用于检查当前的自动 PICC 轮询设置。

Read Automatic PICC Polling 命令的结构

命令	CLA	INS	P1	P2	Lc
Read Automatic PICC Polling	0xE0h	0x00h	0x00h	Ox23h	0x00h

Read Automatic PICC Polling 命令的响应结构

响应	CLA	INS	P1	P2	Le	响应数据域
结果	0xE1h	0x00h	0x00h	0x00h	0x01h	轮询设置

其中：

轮询设置：默认值 = 8Fh（1 个字节）

轮询设置	说明	说明
Bit 0	自动 PICC 轮询	1 = 启用 0 = 停用
Bit 1	如果没有找到 PICC，关闭天线场	1 = 启用 0 = 停用
Bit 2	如果 PICC 闲置，关闭天线场	1 = 启用 0 = 停用
Bit 3	RFU	RFU
Bit 5 -4	PICC 轮询间隔	Bit 5 – Bit 4： 0 – 0 = 250 毫秒 0 – 1 = 500 毫秒 1 – 0 = 1000 毫秒 1 – 1 = 2500 毫秒
Bit 6	RFU	RFU
Bit 7	执行 ISO14443A 第 4 部分	1 = 启用 0 = 停用

5.4.6. SET THE PICC OPERATING PARAMETER 命令。

此命令用于设置 PICC 操作参数。

Set the PICC Operating Parameter 命令的结构（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Set the PICC Operating Parameter	0xE0h	0x00h	0x00h	0x20h	0x01h	操作参数

Set the PICC Operating Parameter 命令的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	0xE1h	0x00h	0x00h	0x00h	0x01h	操作参数

操作参数（1 个字节）

操作参数	参数	说明	选项
Bit0	ISO 14443 A 类	PICC 轮询要检测的标签类别	1 = 检测 0 = 跳过
Bit1	ISO 14443 B 类		1 = 检测 0 = 跳过
Bit2 - 7	RFU	RFU	RFU

注：*操作参数的默认值 = 0x03h*

5.4.7. READ THE PICC OPERATING PARAMETER 命令

此命令用于检查当前的 PICC 操作参数。

Read the PICC Operating Parameter 命令的结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read the PICC Operating Parameter	0xE0h	0x00h	0x00h	0x20h	0x00h

Read the PICC Operating Parameter 命令的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	0xE1h	0x00h	0x00h	0x00h	0x01h	操作参数

操作参数（1 个字节）

操作参数	参数	说明	选项
Bit0	ISO 14443 A 类	PICC 轮询要检测的标签类别	1 = 检测 0 = 跳过
Bit1	ISO 14443 B 类		1 = 检测 0 = 跳过
Bit2 - 7	RFU	RFU	RFU

5.4.8. SET AUTO PPS 命令

此命令用于设定读写器的 PPS 设置。

每次识别出 PICC，读写器都会尝试改变最快连接速度定义的从 PCD 到 PICC 的通信数据速率。若卡片不支持建议的连接速度，读写器会尝试以较慢的速度与卡片建立连接。

Set Auto PPS 命令的结构（7 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Set Auto PPS	0xE0h	0x00h	0x00h	0x24h	0x01h	最大速度

Set Auto PPS 命令的响应结构（9 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	0xE1h	0x00h	0x00h	0x00h	0x02h	最大速度	当前速度

其中：

最大速度（1 个字节）： 最高速率

当前速度（1 个字节）： 当前速率

可以为： 106k bps = 0x00h（等同于没有设置自动 PPS）

212k bps = 0x01h

424k bps = 0x02h（默认设置）

848k bps = 0x03h

注：

1. *通常来讲，应用程序应当知道正在被使用的 PICC 的最大连接速率，周围环境也会对最大可达速率有所影响。读写器只是使用建议的通信速率来与 PICC 进行对话。如果 PICC 或周围环境不能满足建议的通信速率的要求，PICC 将变得不能访问。*
2. *读写器支持不同的数据发送速度和接收速度。*

5.4.9. READ AUTO PPS 命令

此命令用于检查当前的自动 PPS 设置。

Read Auto PPS 命令的结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read Auto PPS	0xE0h	0x00h	0x00h	0x24h	0x00h

Set Auto PPS 命令的响应结构（9 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	0xE1h	0x00h	0x00h	0x00h	0x02h	最大速度	当前速度

其中：

最大速度（1 个字节）： 最高速率

当前速度（1 个字节）： 当前速度

可以为： 106k bps = 0x00h（等同于没有设置自动 PPS）

212k bps = 0x01h

424k bps = 0x02h（默认设置）

848k bps = 0x03h

5.4.10. SET ANTENNA FIELD 命令

此命令用于打开/关闭天线场。

Antenna Field Control 命令的结构（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Antenna Field Control	0xE0h	0x00h	0x00h	0x25h	0x01h	状态

Antenna Field Control 命令的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	0xE1h	0x00h	0x00h	0x00h	0x01h	状态

其中：

状态（1 个字节）： 0x01h = 启用天线场

0x00h = 停用天线场

注：*关闭天线场前要确保自动 PICC 轮询功能已经停用。*

5.4.11. READ ANTENNA FIELD STATUS 命令

此命令用于检查当前的天线场状态。

Read Antenna Field Status 命令的结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read Antenna Field Status	0xE0h	0x00h	0x00h	0x25h	0x00h

Read Antenna Field Status 命令的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	0xE1h	0x00h	0x00h	0x00h	0x01h	状态

其中：

状态（1 个字节）： 0x01h = 启用天线场

0x00h = 停用天线场

5.4.12. TWO LEDs CONTROL 命令

此命令用于控制 LED 输出。

LED Control 命令的结构（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
LED Control	0xE0h	0x00h	0x00h	0x29h	0x01h	LED 状态

LED Control 命令的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	0xE1h	0x00h	0x00h	0x00h	0x01h	LED 状态

LED 状态（1 个字节）

LED 状态	说明	说明
Bit 0	蓝色 LED	1 = 开；0 = 关
Bit 1	橙色 LED	1 = 开；0 = 关
Bit 2 - 7	RFU	RFU

5.4.13. LED STATUS 命令

此命令用于检查现有 LED 的状态。

LED Status 命令的结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
LED Status	0xE0h	0x00h	0x00h	0x29h	0x00h

LED Status 命令的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	0xE1h	0x00h	0x00h	0x00h	0x01h	LED 状态

LED 状态（1 个字节）

LED 状态	说明	说明
Bit 0	蓝色 LED	1 = 开；0 = 关
Bit 1	橙色 LED	1 = 开；0 = 关
Bit 2 - 7	RFU	RFU

5.4.14. FOUR LEDs CONTROL 命令

此命令用于对四个 LED 进行控制。

LEDs Control 命令的结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
LEDs Control	0xFFh	0x00h	0x44h	bLEDsState	0x00h

其中：

P2： bLEDsState

LED_0、LED_1、LED_2 和 LED_3 控制结构（1 个字节）

CMD	项目	说明
Bit 0	LED_0 状态	1 = 开；0 = 关
Bit 1	LED_1 状态	1 = 开；0 = 关
Bit 2	LED_2 状态	1 = 开；0 = 关
Bit 3	LED_3 状态	1 = 开；0 = 关
Bits 4 – 7	保留	-

其中：

响应数据： SW1 SW2

状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

5.4.15. BUZZER CONTROL 命令

此命令用于控制蜂鸣器的输出。

Buzzer Control 命令的结构（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Buzzer Control	0xE0h	0x00h	0x00h	0x28h	0x01h	蜂鸣器持续

其中：

蜂鸣器持续（1 个字节）： 0x00h = 关闭
0x01h - 0xFEh = 持续时间（单位：10 ms）
0xFFh = 开启

Buzzer Control 命令的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	0xE1h	0x00h	0x00h	0x00h	0x01h	0x00h

其中：

蜂鸣器持续时间（1 个字节）： 保持“开启”的持续时间（单位：10 ms）

5.4.16. LCD CONTROL 命令

5.4.16.1. CLEAR LCD

此命令用于清除 LCD 上显示的全部内容。

Clear LCD 命令的结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Clear LCD	0xFFh	0x00h	0x60h	0x00h	0x00h

其中：

响应数据： SW1 SW2

状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

5.4.16.2. LCD DISPLAY（ASCII 模式）

此命令用于 ASCII 模式字符信息在 LCD 上的显示。

此 LCD Display 命令的结构（5 个字节 + 要在 LCD 上显示的消息的长度）

命令	CLA	INS	P1	P2	Lc	命令数据域 (最多 16 个字节)
LCD Display	0xFFh	选项字节	0x68h	LCD 坐标	LCD 消息长度	LCD 消息

其中：

INS： 选项字节（1 个字节）

CMD	项	说明
Bit 0	字符加粗	1 = 加粗；0 = 常规
Bit 1 - 3	保留	-
Bit 4 - 5	字库索引	00 = 字体 A 01 = 字体 B 10 = 字体 C
Bits 6 – 7	保留	-

P2： LCD 坐标。字符在 LCD 上的显示位置，通过 DDRAM 地址进行指定

请参考下方的 DDRAM 地址表以了解 LCD 字符位置的表示。

对于字体 1 和字体 2，

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	显示位置
第 1 行	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	LCD 坐标
第 2 行	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	

对于字体 3，

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	显示位置
第 1 行	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	LCD 坐标
第 2 行	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	
第 3 行	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	
第 4 行	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	

其中：

Lc： 在 LCD 上显示的消息的长度（最大为 0x10）。如果消息的长度超过 LCD 可以显示的字符数，则多余的字符将不会在屏幕上显示。

命令数据域：LCD 消息。待发送至 LCD 的数据（每行最多 16 个字符）

有关 LCD 上显示的字符索引，请参考以下字库（通过 INS 的 Bit4-5 来选择）。

注：*字体 A 和字体 B 的字符大小为 8x16，而字体 C 的字符大小为 8x8。*

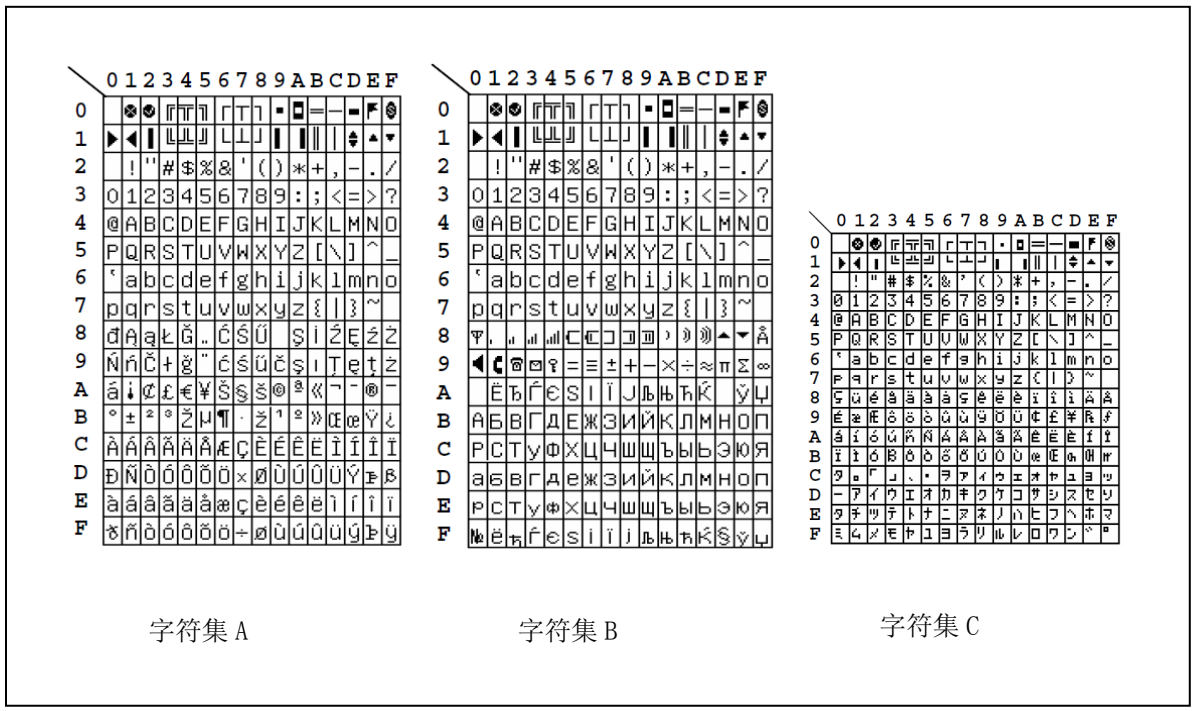

图2：LCD 显示字库

其中：

响应数据域： **SW1 SW2**

状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

5.4.16.3. LCD DISPLAY（GB 模式）

此命令用于 GB 模式字符信息在 LCD 上的显示。

此 LCD Display 命令的结构（5 个字节 + 要在 LCD 上显示的信息的长度）

命令	CLA	INS	P1	P2	Lc	命令数据域 (最多 16 个字节)
LCD Display	0xFFh	选项字节	0x69h	LCD 坐标	LCD 消息长度	LCD 消息

其中：

INS： 选项字节（1 个字节）

CMD	项	说明
Bit 0	字符加粗	1 = 加粗；0 = 常规
Bit 1 - 7	保留	-

P2： LCD 坐标。字符在 LCD 上的显示位置，通过 DDRAM 地址进行指定。

请参考下方的 DDRAM 地址表以了解 LCD 字符位置的表示。

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	显示位置
第 1 行	00		01		02		03		04		05		06		07		LCD 坐标
第 2 行	40		41		42		43		44		45		46		47		

其中：

Lc： 在 LCD 上显示的消息的长度（最大为 0x10）。如果消息的长度超过 LCD 可以显示的字符数，则多余的字符将不会在屏幕上显示。

由于每个中文字符（GB 码）包含两个字节，所以信息的长度应当乘以 2。

命令数据域： LCD 消息。要发送至 LCD 的数据，每行最多 8 个字符（每个字符 2x8 位）。请参加下方 GB 编码的字库。

响应数据域： **SW1 SW2**

状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

5.4.16.4. LCD Display（图形模式）

此命令用于图形模式字符信息在 LCD 上的显示。

此 LCD Display 命令的结构（5 个字节 + 要在 LCD 上显示的消息的长度）

命令	CLA	INS	P1	P2	Lc	命令数据域 (最多 128 个字节)
LCD Display	0xFFh	0x00h	0x6Ah	行索引	像素数据长度	像素数据

其中：

P2： 行索引。用来设置从 LCD 显示屏上的哪一行开始更新。LCD 的显示位置见下方。

Lc： 像素数据长度。像素数据的长度（最大为 0x80h）。

命令数据域： 像素数据。待发送到 LCD 进行显示的像素数据。

LCD 显示位置（LCD 总尺寸：128x32）

	字节 0x00h (X = 0x00h)								字节 0x01h (X = 0x01h)								…	字节 0x0Fh (X = 0x0Fh)							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	…	7	6	5	4	3	2	1	0
0x00h																									
0x01h																									
0x02h																									
0x03h																									
0x04h																									
0x05h																									
0x06h																									
0x07h																									
0x08h																									
0x09h																									
…	…																								

	字节 0x00h (X = 0x00h)								字节 0x01h (X = 0x01h)								...	字节 0x0Fh (X = 0x0Fh)							
0x1Fh																								

其中：

响应数据域： **SW1 SW2**

状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

5.4.16.5. SCROLL CURRENT LCD DISPLAY 命令

此命令用于设置当前 LCD 显示屏的字符滚动功能。

Scrolling LCD 命令的结构（5 个字节 + LCD 消息的长度）

命令	CLA	INS	P1	P2	Lc	命令数据域 (6 个字节)
Scrolling LCD	0xFFh	0x00h	0x6Dh	0x00h	0x06h	Scroll Ctrl

其中：

Scroll Ctrl： 6 个字节；滚动功能控制格式。

滚动功能控制格式（6 个字节）

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
X 位置	Y 位置	滚动范围（水平）	滚动范围（垂直）	刷新速度控制	滚动方向

其中：

X 位置： 水平起始位置，请参考下方关于 LCD 显示位置的说明。

Y 位置： 垂直起始位置，请参考下方关于 LCD 显示位置的说明。

LCD 显示位置（LCD 总尺寸：128x32）

	字节 0x00h (X = 0x00h)								字节 0x01h (X = 0x01h)								...	字节 0x0Fh (X = 0x0Fh)							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	...	7	6	5	4	3	2	1	0
0x00h																									
0x01h																									
0x02h																									
0x03h																									

	字节 0x00h (X = 0x00h)								字节 0x01h (X = 0x01h)								…	字节 0x0Fh (X = 0x0Fh)							
0x04h																									
0x05h																									
0x06h																									
0x07h																									
0x08h																									
0x09h																									
…	…																								
0x1Fh																									

其中：

滚动范围（水平）： 在 X 水平方向滚动的 8 像素数

滚动范围（垂直）： 在 Y 垂直方向滚动的像素数

刷新速度控制： Bit 0~Bit 3 – 在滚动前移动的像素数

Bit 4~Bit 7 – 滚动周期

Bit7	Bit6	Bit5	Bit4	滚动周期
0	0	0	0	1 Unit
0	0	0	1	3 Units
0	0	1	0	5 Units
0	0	1	1	7 Units
0	1	0	0	17 Units
0	1	0	1	19 Units
0	1	1	0	21 Units
0	1	1	1	23 Units
1	0	0	0	129 Units
1	0	0	1	131 Units
1	0	1	0	133 Units
1	0	1	1	135 Units
1	1	0	0	145 Units
1	1	0	1	147 Units
1	1	1	0	149 Units
1	1	1	1	151 Units

表5： 滚动周期

Bit1	Bit0	滚动方向
0	0	从左至右
0	1	从右至左

Bit1	Bit0	滚动方向
1	0	从上至下
1	1	从下至上

表6：滚动方向

其中：

响应数据域：　SW1 SW2

状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

5.4.16.6. PAUSE LCD SCROLLING 命令

此命令用于暂停之前设置的 LCD 字符滚动功能。要重启滚动功能，需要重新发送 Scrolling LCD 命令。

Pause Scrolling 命令的结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Pause LCD Scrolling	0xFFh	0x00h	0x6Eh	0x00h	0x00h

其中：

响应数据域：　SW1 SW2

状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

5.4.16.7. STOP LCD SCROLLING

此命令用于终止之前设置的 LCD 字符滚动功能，之后 LCD 显示屏恢复到正常显示状态。

Stop Scrolling LCD 命令的结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Stop Scrolling LCD	0xFFh	0x00h	0x6Fh	0x00h	0x00h

其中：

响应数据域： **SW1 SW2**

状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

5.4.16.8. LCD CONTRAST CONTROL

此命令用于控制 LCD 的对比度。

LCD Contrast Control 命令的结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
LCD Contrast Control	0xFFh	0x00h	0x6Ch	对比度控制	0x00h

其中：

P2： 对比度控制。对比度的区间范围是 0x00h-0x0Fh。值越大，对比越亮，否则对比变暗。

响应数据域： **SW1 SW2**

状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

5.4.16.9. LCD BACKLIGHT CONTROL

此命令用于控制 LCD 的背光。

LCD Backlight Control 命令的结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
LCD Backlight Control	0xFFh	0x00h	0x64h	背光控制	0x00h

其中：

P2： 背光控制

背光控制的格式（1 个字节）

CMD	说明

CMD	说明
0x00h	LCD 背光关闭
0xFFh	LCD 背光开启

其中：

响应数据域：　SW1 SW2

状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。