Advanced Card Systems Ltd.
Card & Reader Technologies

ACR1281S-C1
双界面读写器
（串口）

通信协议 V1.03

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

版本历史

发布日期	修订说明	版本号
2012-09-03	• 初始发布	1.00
2013-03-07	• 更新 1.0 节：简介 • 更新 4.1.5 节和 4.1.6 节：设置/读取默认 LED 和蜂鸣器操作 • 更新 4.1.7 节和 4.1.8 节：读取/更新卡片插拔计数器 • 更新 4.1.16 节和 4.1.17 节：设置/读取自动 PPS	1.01
2014-12-13	• 更新 1.0 节：简介 • 更新 1.4 节：仿 CCID 命令	1.02
2017-09-07	• 更新 1.1 节：特性	1.03

目录

表目录

1.0. 简介

ACR1281S-C1 串行协议定义了 PC 与读写器之间的接口，以及 PC 与符合 ISO 14443 的非接触式卡（PICC）以及符合 ISO 7816 的全尺寸接触式卡（ICC）和 SIM 尺寸接触式卡（SAM）之间的通信通道。

1.1. 特性

- RS-232串行接口：波特率 = 9.6 kbps（默认）、19.2 kbps、38.4 kbps、57.6 kbps、115.2 kbps、230.4 kbps
- USB接口取电
- 仿CCID架构（二进制格式）
- 智能卡读写器：
  - 非接触接口：
    - 读写速度达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）
    - 支持扩展的APDU（最大64K字节）
  - 接触式接口：
    - 1 个全尺寸接触式卡槽
    - 支持ISO 7816的A类、B类和C类（5 V、3 V、1.8 V）卡
    - 支持符合T=0或T=1协议的微处理器卡
    - 支持各类存储卡
  - SAM 接口：
    - 1个SAM卡槽
    - 支持ISO 7816 A类 SAM卡
- 内置外围设备：
  - 2 个用户可控的 LED 指示灯
  - 1 个用户可控的蜂鸣器
- 具有USB固件升级能力
- 符合下列标准：
  - ISO 14443
  - ISO 7816
  - CE
  - FCC
  - RoHS

1.2. 串行接口

ACR1281S-C1 通过一个串行接口（RS-232 或 RS-485）与计算机建立连接。

1.2.1. 通信参数

ACR1281S-C1 通过串行接口（RS-232 或 RS-485）与主机建立连接，支持以下通讯波特率：9,600 bps （默认）、19,200 bps、38,400 bps、57,600 bps、115,200 bps 和 230,400 bps.

引脚	信号	功能
1	VCC	为读写器提供+5 V 的电源
2	TXD	主机向读写器发送的信号
3	RXD	读写器向主机发送的信号
4	GND	参考电压等级

表 1：RS-232 接口配线

引脚	信号	功能
1	VCC	为读写器提供+5 V 的电源
2	A	读写器和主机间以差分信号传输数据
3	B	读写器和主机间以差分信号传输数据
4	GND	参考电压等级

表 2：RS-485 接口配线

1.3. 串行协议

ACR1281S-C1 通过串行接口与主机连接。采用仿 CCID 架构用于通信。

命令格式如下：

STX (02h)	Bulk-OUT 头	APDU 命令或参数	校验和	ETX (03h)
1 个字节	10 个字节	M 个字节（如有）	1 个字节	1 个字节

其中：

STX 起始字元，通知读写器开始接收命令，必须为 02h

ETX 结束字元，通知读写器命令结束，必须为 03h

Bulk-OUT 头 10 字节的仿 CCID 头

APDU 命令或参数 用于访问读写器或卡片的 APDU 命令或参数

校验和 错误检查，等于 XOR {Bulk-OUT 头，APDU 命令或参数}

收到指令后，ACR1281S 会先返回一个状态帧，通知主机命令的状态。

状态帧的格式如下：

STX (02h)	状态	校验和	ETX (03h)
1 个字节	1 个字节	1 个字节	1 个字节

注：*校验和 = 状态*

有几种情况可能发生：

Case1 确认（ACK）帧 = {02 00 00 03h}
通知主机该帧已经被正确接收。主机必须等待命令的响应，而 ACR1281S 在命令处理期间不会接受其它的帧。

Case2 校验和错误帧 = {02 FF FF 03h}

接收到的数据校验和错误。

Case3 长度错误帧 = {02 FE FE 03h}

数据长度大于 275 个字节，

Case4 ETX 错误帧 = {02 FD FD 03h}

最后一个字节不等于 ETX“03h”。

Case5 超时错误帧 = {02 99 99 03h}

长时间没有接收到数据。

否定（NAK）帧 = {02 00 00 00 00 00 00 00 00 00 00 00 03h} // 11 个“0”

由主机使用来获取最后一个应答或卡片插/拔事件消息。

正确接收到帧后（例如：主机接收到确认帧），ACR1281S 会立即发送一个应答帧。

应答帧的格式如下：

STX (02h)	Bulk-IN 头	APDU 响应或 abData	校验和	ETX (03h)
1 个字节	10 个字节	N 个字节（如有）	1 个字节	1 个字节

其中：

STX	起始字元，通知主机接收响应，必须为 02h
ETX	结束字元，通知主机响应结束，必须为 03h
Bulk-IN 头	10 个字节的仿 CCID 头，请参考 **1.4 节– 仿 CCID 命令**。
APDU 响应或 abData	被访问的命令的 APDU 响应或数据
校验和	错误检查，等于 XOR {Bulk-OUT 头，APDU 响应或 abData}

1.4. 仿 CCID 命令

1.4.1. Bulk-OUT 消息

ACR1281S 应当遵循 CCID 协议第 6.1 部分有关 CCID 类 Bulk-OUT 消息的规定。该规范还定义了一些操作附加功能的扩展命令。本节将列举 ACR1281S 支持的 CCID Bulk-OUT 消息。

1.4.1.1. 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。 SAM 接口：*bSlot* = 2。 ICC 接口：*bSlot* = 1。 PICC 接口：*bSlot* = 0。
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）。

注：*访问接触式卡之前必须先激活 ICC 接口和 SAM 接口。*

1.4.1.2. 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	
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 SAM 接口：*bSlot* = 2。 ICC 接口：*bSlot* = 1。 PICC 接口：*bSlot* = 0。
6	*bSeq*	1		命令的序号。

偏移	数据域	大小	值	描述
7	*abRFU*	3		保留为将来使用。

此消息的响应是 *RDR_to_PC_SlotStatus* 消息。

1.4.1.3. 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。 SAM 接口：*bSlot* = 2。 ICC 接口：*bSlot* = 1。 PICC 接口：*bSlot* = 0。
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。

1.4.1.4. 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。 SAM 接口：*bSlot* = 2。 ICC 接口：*bSlot* = 1。 PICC 接口：*bSlot* = 0。
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*	字节型数组		发送给 CCID 的数据块。信息是“按原样”发送至 ICC（TPDU 交换级别）的。

此消息的响应是 *RDR_to_PC_DataBlock* 消息。

1.4.1.5. PC_to_RDR_Escape

此命令用于访问扩展特性。

偏移	数据域	大小	值	描述
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。 SAM 接口：*bSlot* = 2。 ICC 接口：*bSlot* = 1。 PICC 接口：*bSlot* = 0。
6	*bSeq*	1		命令的序号。
7	*abRFU*	3		保留为将来使用。
10	*abData*	字节型数组		发送给 CCID 的数据块。

此命令消息的响应是 *RDR_to_PC_Escape* 消息。

1.4.2. Bulk-IN 消息

Bulk-IN 消息用于对 Bulk-OUT 消息做出响应。ACR1281S 应当遵循 CCID 标准第 6.2 部分有关 CCID Bulk-IN 消息的规定。本节将列举 ACR1281S 支持的 CCID Bulk-IN 消息。

1.4.2.1. RDR_to_PC_DataBlock

此消息由 ACR1281S 发出，是对 *PC_to_RDR_IccPowerOn* 和 *PC_to_RDR_XfrBlock* 消息的响应。

偏移	数据域	大小	值	描述
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。 SAM 接口：*bSlot* = 2。 ICC 接口：*bSlot* = 1。 PICC 接口：*bSlot* = 0。
6	*bSeq*	1		与 Bulk-OUT 消息中的值相同。
7	*bStatus*	1		CCID 规范 6.2.6 节定义的插槽状态寄存器。
8	*bError*	1		CCID 规范 6.2.6 节定义的插槽错误寄存器。
9	*bChainParameter*	1	00h	RFU（TPDU 交换级别）。
10	*abData*	字节型数组		本数据域包含由 CCID 返还的数据。

1.4.2.2. RDR_to_PC_Escape

此消息由 ACR1281S 发出，是对 *PC_to_RDR_Escape* 消息的响应。

偏移	数据域	大小	值	描述
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。 SAM 接口：*bSlot* = 2。 ICC 接口：*bSlot* = 1。 PICC 接口：*bSlot* = 0。
6	*bSeq*	1		与 Bulk-OUT 消息中的值相同。
7	*bStatus*	1		CCID 规范 6.2.6 节定义的插槽状态寄存器。
8	*bError*	1		CCID 规范 6.2.6 节定义的插槽错误寄存器。
9	*bRFU*	1	00h	RFU.
10	*abData*	字节型数组		本数据域包含由 CCID 返还的数据。

1.4.2.3. RDR_to_PC_SlotStatus

此消息由 ACR1281S 发出，是对 *PC_to_RDR_IccPowerOff* 和 *PC_to_RDR_GetSlotStatus* 消息，以及类特定 ABORT 请求的响应。

偏移	数据域	大小	值	描述
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。 SAM 接口：*bSlot* = 2。 ICC 接口：*bSlot* = 1。 PICC 接口：*bSlot* = 0。
6	*bSeq*	1		与 Bulk-OUT 消息中的值相同。
7	*bStatus*	1		CCID 规范 6.2.6 节定义的插槽状态寄存器。
8	*bError*	1		CCID 规范 6.2.6 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。

2.0. 接触式智能卡协议

私有APDU用于访问存储标签和外围设备。

私有 APDU 通过 *PC_to_RDR_XfrBlock* 消息发送，*bSlot* = 1。

2.1. 存储卡 – 1、2、4、8、16 kbits I2C 卡

2.1.1. Select Card Type

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。

命令格式

Pseudo-APDU					
CLA	**INS**	**P1**	**P2**	**Lc**	**Card Type**
FFh	A4h	00h	00h	01h	01h

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.1.2. Select Page Size

此命令会选择用于读取智能卡的页面大小。默认值是 8 字节页写。当卡片被移出，或者当读写器被下电时会重置为默认值。

命令格式

Pseudo-APDU					
CLA	**INS**	**P1**	**P2**	**Lc**	**Page size**
FFh	01h	00h	00h	01h	

其中：

Page size (1 个字节) = 03h：8 字节页写
= 04h：16 字节页写
= 05h：32 字节页写
= 06h：64 字节页写
= 07h：128 字节页写

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.1.3. Read Memory Card

此命令会从指定地址读取存储卡。

命令格式

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FFh	B0h			

其中：

Byte Address (2 个字节) = 存储卡的内存地址位置

MEM_L (1 个字节) = 待从存储卡内读取的数据的长度

应答数据格式

BYTE 1	…	…	BYTE N	SW1	SW2

其中：

BYTE (1…N) = 从存储卡中读取的数据

SW1 SW2 = 90 00h（如果未发生错误）

2.1.4. Write Memory Card

此命令用于将内容写入存储卡的指定地址。

命令格式

Pseudo-APDU								
CLA	INS	Byte Address		MEM_L	Byte 1	...	...	Byte N
		MSB	LSB					
FFh	D0h							

其中：

Byte Address (2 个字节) = 存储卡的内存地址位置

MEM_L (1 个字节) = 待写入存储卡的数据的长度

BYTE (1…N) = 待写入存储卡的数据

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.2. 存储卡 – 32、64、128、256、512、1024 kbits I2C 卡

2.2.1. Select Card Type

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。

命令格式

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FFh	A4h	00h	00h	01h	02h

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.2.2. Select Page Size

此命令会选择用于读取智能卡的页面大小。默认值是 8 字节页写。当卡片被移出，或者当读写器被下电时会重置为默认值。

命令格式

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Page size
FFh	01h	00h	00h	01h	

其中：

Page size (1 个字节) = 03h：8 字节页写
= 04h：16 字节页写
= 05h：32 字节页写
= 06h：64 字节页写
= 07h：128 字节页写

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.2.3. Read Memory Card

此命令会通过指定地址读取存储卡的内容。

命令格式

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FFh				

其中：

INS (1 个字节): 如果是 32, 64, 128, 256, 512 kbit I2C 卡，INS = B0h
如果是 1024kbit I2C 卡，INS = 1011 000*b
其中 * 表示 17 位地址的 MSB。

Byte Address (2 个字节) = 存储卡的内存地址位置

MEM_L (1 个字节) = 待从存储卡读取的数据的长度

应答数据格式

BYTE 1	…	…	BYTE N	SW1	SW2

其中：

BYTE (1…N) = 从存储卡读取的数据

SW1 SW2 = 90 00h（未发生错误）

2.2.4. Write Memory Card

此命令用于将内容写入存储卡的指定地址位置。

命令格式

Pseudo-APDU								
CLA	INS	Byte Address		MEM_L	Byte 1	…	…	Byte N
		MSB	LSB					
FFh								

其中：

INS (1 个字节): 如果是 32, 64, 128, 256, 512 kbit I2C 卡，INS = D0h
如果是 1024 kbit I2C 卡，INS = 1101 000* b
其中 * 表示 17 位地址的 MSB。

Byte Address (2 个字节) = 存储卡的内存地址位置

MEM_L (1 个字节) = 待写入存储卡的数据的长度

BYTE (1…N) = 待写入存储卡的数据

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.3. 存储卡 – ATMEL AT88SC153

2.3.1. Select Card Type

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。还将选择页面大小为 8 字节页写。

命令格式

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FFh	A4h	00h	00h	01h	03h

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.3.2. Read Memory Card

此命令会通过指定地址位置读取存储卡。

命令格式

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
FFh		00h		

其中：

INS (1 个字节): 读取分区 00b，INS = B0h
读取分区 01b，INS = B1h
读取分区 10b，INS = B2h
读取分区 1b，INS = B3h
读取熔丝标志，INS = B4h

Byte Address (1 个字节) = 存储卡的内存地址位置

MEM_L (1Byte) = 待从存储卡读取的数据的长度

应答数据格式

BYTE 1	…	…	BYTE N	SW1	SW2

其中：

BYTE (1…N) = 从存储卡读取的数据

SW1 SW2 = 90 00h（未发生错误）

2.3.3. Write Memory Card

此命令用于将内容写入存储卡的指定地址位置。

命令格式

Pseudo-APDU								
CLA	INS	P1	Byte Address	MEM_L	Byte 1	…	…	Byte N
FFh		00h						

其中：

INS (1 个字节): 读取分区 00b，INS = D0h

读取分区 01b，INS = D1h

读取分区 10b，INS = D2h

读取分区 11b，INS = D3h

读取熔丝标志，INS = D4h

Byte Address (1 个字节) = 存储卡的内存地址位置

MEM_L (1 个字节) = 待写入存储卡的数据的长度

BYTE (1…N) = 待写入存储卡的数据

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.3.4. Verify Password

此命令用于验证存储卡的密码是否与用户输入的 PIN 相匹配。

命令格式

Pseudo-APDU							
CLA	INS	P1	P2	Lc	PW (0)	PW (1)	PW (2)
FFh	20h	00h		03h			

其中：

PW (0), PW (1), PW (2) = 待发送给存储卡的密码

P2 (1 Byte) = 0000 00r p b

其中的“rp”位表示待比较的密码

r = 0：写密码

r = 1：读密码

p = 密码集编号

r p = 01：安全密码

应答数据格式

SW1	ErrorCnt
90	

其中：

ErrorCnt (1 个字节) = 错误计数器

“FFh”表示验证正确，“00h”表示密码被锁定（超过最大重试次数）。其它值表示当前验证失败。

2.3.5. Initialize Authentication

此命令用于初始化存储卡认证。

命令格式

Pseudo-APDU								
CLA	INS	P1	P2	Lc	Q (0)	Q (1)	…	Q (7)
FFh	84h	00h	00h	08h				

其中：

Q (0…7) = 主机随机数，8 个字节

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.3.6. Verify Authentication

此命令用于校验存储卡认证。

命令格式

Pseudo-APDU								
CLA	**INS**	**P1**	**P2**	**Lc**	**Ch (0)**	**Ch (1)**	**…**	**Ch (7)**
FFh	82h	00h	00h	08h				

其中：

Ch (0…7) = 主机挑战数，8 个字节

应答数据格式

SW1	**SW2**

其中：

SW1 SW2 = 90 00h（未发生错误）

2.4. 存储卡 – ATMEL AT88SC1608

2.4.1. Select Card Type

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。还将选择页面大小为 16 字节页写。

命令格式

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FFh	A4h	00h	00h	01h	04h

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.4.2. Read Memory Card

此命令会通过指定地址位置读取存储卡。

命令格式

Pseudo-APDU				
CLA	INS	Zone Address	Byte Address	MEM_L
FFh				

其中：

INS (1 个字节): 读取用户区，INS = B0h

读取配置区或读取熔丝标志，INS = B1h

Zone Address (1 个字节) = 00000 A10 A9 A8b，其中 A10 表示分区地址的 MSB

**读取熔丝标志时无需关注

Byte Address (1 个字节) = A7 A6 A5 A4 A3 A2 A1 A0b 是存储卡的内存地址位置

读取熔丝标志时，Byte Address = 1000 0000b

MEM_L (1Byte) = 待从存储卡读取的数据的长度

应答数据格式

BYTE 1	…	…	BYTE N	SW1	SW2

其中：

BYTE (1…N) = 从存储卡读取的数据

SW1 SW2 = 90 00h（如果未发生错误）

2.4.3. Write Memory Card

此命令用于将内容写入存储卡的指定地址位置。

命令格式

Pseudo-APDU								
CLA	INS	Zone Address	Byte Address	MEM_L	Byte 1	...	...	Byte N
FFh								

其中：

INS (1 个字节): 读取用户区，INS = D0h

读取配置区或读取熔丝标志，INS = D1h

Zone Address (1 个字节) = 00000 A10 A9 A8b，其中 A10 表示分区地址的 MSB

**读取熔丝标志时无需关注

Byte Address (1 个字节) = A7 A6 A5 A4 A3 A2 A1 A0b 是存储卡的内存地址位置

读取熔丝标志时，Byte Address = 1000 0000b

MEM_L (1 个字节) = 待写入存储卡的数据的长度

BYTE (1…N) = 待写入存储卡的数据

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.4.4. Verify Password

此命令用于验证存储卡的密码是否与用户输入的 PIN 相匹配。

命令格式

Pseudo-APDU								
CLA	INS	P1	P2	Lc	RP	PW (0)	PW (1)	PW (2)
FFh	20h	00h	00h	04h				

其中：

PW (0), PW (1), PW (2) = 待发送给存储卡的密码

RP (1 个字节) = 0000 r p2 p1 p0 b

其中的两个位“r p2 p1 p0”表示待比较的密码

r = 0：写密码

r = 1：读密码

p2 p1 p0 = 密码集编号

r p2 p1 p0 = 0111：安全密码

应答数据格式

SW1	ErrorCnt
90h	

其中：

ErrorCnt (1 个字节) = 错误计数器

“FFh”表示验证正确，“00h”表示密码被锁定（超过最大重试次数）。其它值表示当前验证失败。

2.4.5. Initialize Authentication

此命令用于初始化存储卡认证。

命令格式

Pseudo-APDU								
CLA	INS	P1	P2	Lc	Q (0)	Q (1)	…	Q (7)
FFh	84h	00h	00h	08h				

其中：

Q (0…7) = 主机随机数，8 个字节

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.4.6. Verify Authentication

此命令用于校验存储卡认证。

命令格式

Pseudo-APDU								
CLA	INS	P1	P2	Lc	Ch (0)	Ch (1)	…	Ch (7)
FFh	82h	00h	00h	08h				

其中：

Ch (0…7) = 主机挑战数，8 个字节

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.5. 存储卡 – SLE4418/SLE4428/SLE5518/SLE5528

2.5.1. Select Card Type

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。

命令格式

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FFh	A4h	00h	00h	01h	05h

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.5.2. Read Memory Card

此命令会通过指定的地址位置读取存储卡的内容。

命令格式

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FFh	B0h			

其中：

MSB Byte Address (1 个字节) = 0000 00 A9 A8b 是存储卡的内存地址位置

LSB Byte Address (1 个字节) = A7 A6 A5 A4 A3 A2 A1 A0b 是存储卡的内存地址位置

MEM_L (1Byte) = 待从存储卡读取的数据的长度

应答数据格式

BYTE 1	…	…	BYTE N	SW1	SW2

其中：

BYTE (1…N) = 从存储卡读取的数据

SW1 SW2 = 90 00h（未发生错误）

2.5.3. Presentation Error Counter Memory Card (仅限 SLE4428 和 SLE5528)

此命令用于读取密码输入错误计数器。

命令格式

Pseudo-APDU				
CLA	INS	P1	P2	MEM_L
FFh	B1h	00h	00h	03h

应答数据格式

ERRCNT	DUMMY 1	DUMMY 2	SW1	SW2

其中：

ERRCNT (1 个字节) = 密码输入错误计数器的值。“FFh” 表示最后一次验证正确。“00h” 表示密码被锁定（超过最大重试次数）。其它值表示最后一次验证失败。

DUMMY1, DUMMY2 (2 个字节) = 从卡片读取的 2 个字节的虚拟数据

SW1 SW2 = 90 00h（未发生错误）

2.5.4. Read Protection Bit

此命令用于读取保护位。

命令格式

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FFh	B2h			

其中：

MSB Byte Address (1 个字节) = 0000 00 A9 A8b 是存储卡的内存地址位置

LSB Byte Address (1 个字节) = A7 A6 A5 A4 A3 A2 A1 A0b 是存储卡的内存地址位置

MEM_L (1 个字节) = 要从卡片中读取的保护位的长度，位数是 8 的倍数，（最大值为 32）

MEM_L = 1 + INT ((位数-1)/8)

例如，要读取始于内存 0010h 的 8 个保护位，应当发送下面的 pseudo-APDU：FF B1 00 10 01h.

应答数据格式

PROT 1	…	…	PROT L	SW1	SW2

其中：

PROT (1…L) = 含有保护位的字节

SW1 SW2 = 90 00h（未发生错误）

在 PROT 字节中，保护位的排列如下：

PROT 1								PROT 2								....							
P8	P7	P6	P5	P4	P3	P2	P1	P16	P15	P14	P13	P12	P11	P10	P9	..	..	..	..	..	..	P18	P17

其中：

Px 是响应数据中 BYTE x 的保护位

‘0’字节被写保护

‘1’字节可以被写入

2.5.5. Write Memory Card

此命令用于将内容写入存储卡的指定地址位置。

命令格式

Pseudo-APDU								
CLA	INS	Byte Address		MEM_L	字节 1	…	...	Byte N
		MSB	LSB					
FFh	D0h							

其中：

MSB Byte Address (1 个字节) = 0000 00 A9 A8b 是存储卡的内存地址位置

LSB Byte Address (1 个字节) = A7 A6 A5 A4 A3 A2 A1 A0b 是存储卡的内存地址位置

MEM_L (1 个字节) = 待写入存储卡的数据的长度

Byte (1…N) = 待写入存储卡的数据

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.5.6. Write Protection Memory Card

命令中指定的每一个字节与存储在特定地址中的字节进行内部对比。如果数据相符，则相应的保护位就会不可逆地被设定为“0”。

命令格式

Pseudo-APDU								
CLA	INS	Byte Address		MEM_L	Byte 1	...	...	Byte N
		MSB	LSB					
FFh	D1h							

其中：

MSB Byte Address (1 个字节) = 0000 00 A9 A8b 是存储卡的内存地址位置

LSB Byte Address (1 个字节) = A7 A6 A5 A4 A3 A2 A1 A0b 是存储卡的内存地址位置

MEM_L (1 个字节) = 待写入存储卡的数据的长度

Byte (1…N) = 要与卡片内始于 Byte Address 的数据做比较的 Byte 值。BYTE 1 与在 Byte Address 的数据比较；BYTE N 与在（Byte Address + N -1）的数据比较

应答数据格式

SW1	SW2

其中:

SW1 SW2 = 90 00h（未发生错误）

2.5.7. Present Code Memory Card（仅限 SLE 4428 和 SLE5528）

此命令用于向存储卡提交密码，从而启用对 SLE4428 卡和 SLE5528 卡的写操作。执行的操作如下：

1. 搜索密码输入错误计数器中值为‘1’的位，然后将该位写为‘0’。
2. 向卡片提交指定的密码。
3. 擦除密码输入错误计数器。

命令格式

Pseudo-APDU						
CLA	INS	P1	P2	MEM_L	CODE	
					Byte 1	Byte 2
FFh	20h	00h	00h	02h		

其中：

CODE（2 个字节） = 密码（PIN）

应答数据格式

SW1	ErrorCnt
90h	

其中：

ErrorCnt (1 个字节) = 错误计数器。“FFh”表示验证正确。“00h”表示密码被锁定（超过最大重试次数）。其它值表示当前验证失败。

2.6. 存储卡 – SLE4432/SLE4442/SLE5532/SLE5542

2.6.1. Select Card Type

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。

命令格式

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FFh	A4h	00h	00h	01h	06h

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.6.2. Read Memory Card

此命令会通过指定地址读取存储卡的内容。

命令格式

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
FFh	B0h	00h		

其中：

Byte Address (1 个字节) = A7 A6 A5 A4 A3 A2 A1 A0b 是存储卡的内存地址位置

MEM_L (1 个字节) = 待从存储卡读取的数据的长度

应答数据格式

BYTE 1	…	…	BYTE N	PROT 1	PROT 2	PROT3	PROT 4	SW1	SW2

其中：

BYTE (1…N) = 从存储卡读取的数据

PROT (1…4) = 含有保护位的字节

SW1 SW2 = 90 00h（未发生错误）

在 PROT 字节中，保护位的排列如下：

PROT 1								PROT 2								…							
P8	P7	P6	P5	P4	P3	P2	P1	P16	P15	P14	P13	P12	P11	P10	P9	..	..	..	..	..	..	P18	P17

其中：

Px 是响应数据中 BYTE x 的保护位

‘0’字节被写保护

‘1’字节可以被写入

2.6.3. Read Present Error Counter Memory Card（仅限 SLE4442 和 SLE5542）

此命令用于读取密码输入错误计数器。

命令格式

Pseudo-APDU				
CLA	INS	P1	P2	MEM_L
FFh	B1h	00h	00h	04h

应答数据格式

ERRCNT	DUMMY 1	DUMMY 2	DUMMY 3	SW1	SW2

其中：

ERRCNT (1 个字节) = 密码输入错误计数器的值。“07h” 表示最后一次验证正确。“00h”表示密码被锁定（超过最大重试次数）。其它值表示最后的验证失败。

DUMMY1, DUMMY2, DUMMY3 (3 个字节) = 从卡片读取的虚拟数据

SW1 SW2 = 90 00h（未发生错误）

2.6.4. Read Protection Bits

此命令用于读取前 32 个字节的保护位。

命令格式

Pseudo-APDU				
CLA	INS	P1	P2	MEM_L
FFh	B2h	00h	00h	04h

应答数据格式

PROT 1	PROT 2	PROT3	PROT 4	SW1	SW2

其中：

PROT (1…4) = 含有保护位的字节

SW1 SW2 = 90 00h（未发生错误）

在 PROT 字节中，保护位的排列如下：

PROT 1								PROT 2								…							
P8	P7	P6	P5	P4	P3	P2	P1	P16	P15	P14	P13	P12	P11	P10	P9	..	..	..	..	..	..	P18	P17

其中：

Px 是响应数据中 BYTE x 的保护位

‘0’字节被写保护

‘1’字节可以被写入

2.6.5. Write Memory Card

此命令用于将内容写入存储卡的指定地址位置。

命令格式

Pseudo-APDU								
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**	**Byte 1**	**...**	**...**	**Byte N**
FFh	D0h	00h						

其中：

Byte Address (1 个字节) = A7 A6 A5 A4 A3 A2 A1 A0b 是存储卡的内存地址位置

MEM_L (1 个字节) = 待写入存储卡的数据的长度

Byte (1…N) = 待写入存储卡的数据

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.6.6. Write Protection Memory Card

命令中指定的每一个字节与存储在特定地址中的字节进行内部对比。如果数据相符，则相应的保护位就会不可逆地被设定为“0”。

命令格式

Pseudo-APDU								
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**	**Byte 1**	**...**	**...**	**Byte N**
FFh	D1h	00h						

其中：

Byte Address (1 个字节) = 000A4 A3 A2 A1 A0b (00h-1Fh)是存储卡的保护内存地址位置

MEM_L (1 个字节) = 待写入存储卡的数据的长度

Byte (1…N) = 要与卡片内始于 Byte Address 的数据做比较的 Byte 值。BYTE 1 与在 Byte Address 的数据比较；BYTE N 与在（Byte Address + N -1）的数据比较

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.6.7. Present Code Memory Card（仅限 SLE4442 和 SLE5542）

此命令用于向存储卡提交密码，从而启用对 SLE4442 卡和 SLE5542 卡的写操作。执行的操作如下：

1. 搜索密码输入错误计数器中值为‘1’的位，然后将该位写为‘0’。
2. 向卡片提交指定的密码。
3. 擦除密码错误计数器。

命令格式

Pseudo-APDU							
CLA	INS	P1	P2	MEM_L	CODE		
					Byte 1	Byte 2	Byte 3
FFh	20h	00h	00h	03h			

其中：

CODE (3 个字节) = 密码(PIN)

应答数据格式

SW1	ErrorCnt

其中：

ErrorCnt (1 个字节) = 错误计数器。“07h” 表示验证正确。“00h”表示密码被锁定（超过最大重试次数）。其它值表示当前验证失败。

2.6.8. Change Code Memory Card（仅限 SLE4442 和 SLE5542）

此命令用于将特定数据作为新密码写入卡片。执行此命令之前，需要先使用 PRESENT_CODE 命令向卡片提交当前密码。

命令格式

Pseudo-APDU							
CLA	INS	P1	P2	MEM_L	CODE		
					Byte 1	Byte 2	Byte 3
FFh	D2h	00h	01h	03h			

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.7. 存储卡 – SLE4406/SLE4436/SLE5536/SLE6636

2.7.1. Select Card Type

此命令用于对选定的插入读写器的卡片进行上电/下电，同时进行卡片复位操作。

命令格式

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FFh	A4h	00h	00h	01h	07h

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.7.2. Read Memory Card

此命令会通过指定地址读取存储卡的内容。

命令格式

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
FFh	B0h	00h		

其中：

Byte Address (1 个字节) = 存储卡的内存地址位置

MEM_L (1Byte) = 待从存储卡读取的数据的长度

应答数据格式

BYTE 1	…	…	BYTE N	SW1	SW2

其中：

BYTE (1…N) = 从存储卡读取的数据

SW1 SW2 = 90 00h（未发生错误）

2.7.3. Write One Byte Memory Card

此命令用于向所插入卡片的特定地址写一个字节。该字节从 LSB 开始写入卡片，也就是说，卡片地址 bit 0 被视为 byte 0 的 LSB。

此类卡片有四种不同的写入模式，通过命令数据域内的标志加以区分。

1. **Write** - 命令中指定的字节值被写入特定的地址，可用于向卡片写入个人化信息和计数器值。
2. **Write with carry** - 命令中指定的字节值被写入特定的地址，且命令被送至卡片来擦除下一个低位计数器。因此，该模式仅适用于卡内计数器的值的更新。
3. **Write with backup enabled (SLE4436, SLE5536 and SLE6636 only)** - 命令中指定的字节值被写入特定的地址，可用于向卡片写入个人化信息和计数器值。同时启用备份位，保护数据免受卡片插拔导致的损失。
4. **Write with carry and backup enabled (SLE4436, SLE5536 and SLE6636 only)** - 命令中指定的字节值被写入特定的地址，且命令被送至卡片来擦除下一个低位计数器。因此，该模式仅适用于卡内计数器的值的更新。同时启用备份位，保护数据免受卡片插拔导致的损失。

在这四种模式下，指定地址上的字节在写操作前不会被擦除，所以存储位只能由“1”设为“0”。

SLE4436 卡和 SLE5536 卡的备份模式可以在写操作中被启用或禁用。

命令格式

Pseudo-APDU						
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**	**MODE**	**BYTE**
FFh	D0h	00h		02h		

其中：

Byte Address (1 个字节) = 存储卡的内存地址位置

MODE (1 个字节) = 指定写入模式和备份选项

00h： write

01h： write with carry

02h： write with backup enabled (SLE4436, SLE5536 and SLE6636 only)

03h： write with carry and with backup enabled (SLE4436, SLE5536 and SLE6636 only)

BYTE (1 个字节) = 待写入卡片的字节值

应答数据格式

SW1	**SW2**

其中：

SW1 SW2 = 90 00h（未发生错误）

2.7.4. Present Code Memory Card

此命令用于向存储卡提交密码，从而启用卡片个人化模式。执行的操作如下：

1. 搜索密码输入错误计数器中值为‘1’的位，然后将该位写为‘0’。
2. 向卡片提交指定的密码。

密码提交后，ACR1281S 不会尝试擦除密码计数器，除非通过应用软件单独使用‘*Write with carry*’命令来进行。

命令格式

Pseudo-APDU								
CLA	INS	P1	P2	MEM_L	CODE			
					ADDR	Byte 1	Byte 2	Byte 3
FFh	20h	00h	00h	04h	09h			

其中：

ADDR (1 个字节) = 输入错误计数器的字节地址

CODE（3 个字节） = 密码（PIN）

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.7.5. Authenticate Memory Card（仅限 SLE4436、SLE5536 和 SLE6636）

此命令用于从 SLE5536 或 SLE6636 卡中读取卡片认证证书。ACR1281S 执行以下操作：

1. 根据命令在卡片中选择 Key 1 或 Key 2。
2. 将命令中指定的随机数提交给卡片。
3. 为卡片计算出的每位认证数据生成指定数量的时钟脉冲。
4. 从卡片中读取 16 位的认证数据。
5. 将卡片复位回正常的操作模式。

认证的过程分为两步：步骤 1 是将认证证书发送至卡片。步骤 2 是取回卡片计算出的 2 个字节的认证数据。

步骤 1：向卡片发送认证证书

命令格式

Pseudo-APDU												
CLA	INS	P1	P2	MEM_L	CODE							
					KEY	CLK_ CNT	Byte1	Byte 2	…	Byte 5	Byte 6
FFh	84h	00h	00h	08h								

其中：

KEY (1 个字节) = 用于计算认证证书的密钥：

00h = key 1，不带密码块链接

01h = key 2，不带密码块链接

80h = key 1，带密码块链接 (仅限 SLE5536 和 SLE6636）

81h = key 2，带密码块链接 (仅限 SLE5536 和 SLE6636）

CLK_CNT (1 个字节) = 待提供给卡片的时钟脉冲的个数，卡片将该脉冲用于计算认证证书的每个位。通常为 160（A0h）。

BYTE (1...6) = 卡片随机数据

应答数据格式

SW1	SW2
61h	02h

如果未发生错误，则表示两个字节的认证数据准备就绪。可以通过 *GET_RESPONSE* 命令检索认证数据。

步骤 2：取回认证数据（GET_RESPONSE）

命令格式

Pseudo-APDU				
CLA	INS	P1	P2	MEM_L
FFh	C0h	00h	00h	02h

应答数据格式

CERT		SW1	SW2

其中：

CERT (2 个字节) = 卡片计算出的 16 位的认证数据。BYTE 1 的 LSB 是从卡片中读取的第一个认证位。

SW1 SW2 = 90 00h（未发生错误）

2.8. 存储卡– SLE4404

2.8.1. Select Card Type

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。

命令格式

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FFh	A4h	00h	00h	01h	08h

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.8.2. Read Memory Card

此命令会通过指定的地址位置读取存储卡的内容。

命令格式

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
FFh	B0h	00h		

其中：

Byte Address (1 个字节) = 存储卡的内存地址位置

MEM_L (1 个字节) = 待从存储卡读取的数据的长度

应答数据格式

BYTE 1	…	…	BYTE N	SW1	SW2

其中：

BYTE (1…N) = 从存储卡读取的数据

SW1 SW2 = 90 00h（未发生错误）

2.8.3. Write Memory Card

此命令用于向所插入卡片的特定地址写入数据。该字节从 LSB 开始写入卡片，也就是说，卡片地址 bit 0 被视为 byte 0 的 LSB。

指定地址上的字节在写操作前不会被擦除，所以存储位只能由“1”设为“0”。

命令格式

Pseudo-APDU								
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**	**Byte 1**	**...**	**...**	**Byte N**
FFh	D0h	00h						

其中：

Byte Address (1 个字节) = 存储卡的内存地址位置

MEM_L (1 个字节) = 待写入存储卡的数据的长度

BYTE (1…N) = 待写入卡片的字节值

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.8.4. Erase Scratch Pad Memory Card

此命令用于擦除所插入卡片的暂存存储器的数据。暂存存储器内所有的存储位都会被设定为状态“1”。

命令格式

Pseudo-APDU				
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**
FFh	D2h	00h		00h

其中：

Byte Address (1 个字节) = 暂存存储区的内存字节地址位置
通常为 02h

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.8.5. Verify User Code

此命令用于向插入的卡片提交用户密码（2 个字节）。用户密码旨在使卡的内存能够被访问。

执行的操作如下：

1. 向卡片提交指定的密码。
2. 搜索密码输入错误计数器中值为‘1’的位，然后将该位写为‘0’。
3. 擦除密码输入错误计数器。提交的密码验证正确后，用户错误计数器可被擦除。

命令格式

Pseudo-APDU						
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE	
					Byte 1	Byte 2
FFh	20h	04h	08h	02h		

其中：

Error Counter LEN (1 个字节) = 密码输入错误计数器的长度，单位为比特

Byte Address (1 个字节) = 卡片中密钥的字节地址

CODE (1 个字节) = 用户密码

应答数据格式

SW1	SW2

SW1 SW2 = 90 00h（未发生错误）

如果不再有重试的机会，则该状态字= 63 00h。

***注：** 收到响应 SW1 SW2 = 90 00h 后，应当再次读取用户错误计数器，检查 VERIFY_USER_CODE 是否正确。如果用户错误计数器被擦除并且等于‘FFh’，证明先前的验证成功。*

2.8.6. Verify Memory Code

此命令用于向插入的卡片提交存储密码（4 个字节）。该存储密码可授权用户重新载入用户内存及用户密码。

执行的操作如下：

1. 向卡片提交指定的密码
2. 搜索密码输入错误计数器中值为‘1’的位，然后将该位写为‘0’
3. 擦除密码输入错误计数器。请注意，存储错误计数器的内容不能被擦除。

命令格式

Pseudo-APDU								
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE			
					Byte 1	Byte 2	Byte 3	Byte 4
FFh	20h	40h	28h	04h				

其中：

Error Counter LEN (1 个字节) = 密码输入错误计数器的长度，单位为比特

Byte Address (1 个字节) = 卡片中密钥的字节地址

CODE (4 个字节) = 存储密码

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

如果不再有重试的机会，则该状态字= 63 00h。

注：*收到响应 SW1 SW2 = 90 00h 后，应当再次读取应用区，检查 VERIFY_MEMORY_CODE 是否正确。如果应用区域的全部数据都被擦除并且等于'FFh'，证明先前的验证成功。*

2.9. 存储卡 – AT88SC101/AT88SC102/AT88SC1003

2.9.1. Select Card Type

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。

命令格式

Pseudo-APDU					
CLA	**INS**	**P1**	**P2**	**Lc**	**Card Type**
FFh	A4h	00h	00h	01h	09h

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.9.2. Read Memory Card

此命令会通过指定地址读取存储卡的内容。

命令格式

Pseudo-APDU				
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**
FFh	B0h	00h		

其中：

Byte Address (1 个字节) = 存储卡的内存地址位置

MEM_L (1 个字节) = 待从存储卡读取的数据的长度

应答数据格式

BYTE 1	…	…	BYTE N	SW1	SW2

其中：

BYTE (1…N) = 从存储卡读取的数据

SW1 SW2 = 90 00h（未发生错误）

2.9.3. Write Memory Card

此命令用于向所插入卡片的特定地址写入数据。该字节从 LSB 开始写入卡片，也就是说，卡片地址 bit 0 被视为 byte 0 的 LSB。

指定地址上的字节在写操作前不会被擦除，所以存储位只能由“1”设为“0”。

命令格式

Pseudo-APDU								
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**	**Byte 1**	**…**	**...**	**Byte N**
FFh	D0h	00h						

其中：

Byte Address (1 个字节) = 存储卡的内存地址位置

MEM_L (1 个字节) = 待写入存储卡的数据的长度

BYTE (1…N) = 待写入卡片的字节值

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.9.4. Erase Non-Application Zone

此命令用于擦除存储在非应用区的数据。EEPROM 内存由 16 位字构成。即使只擦除单独的一位，内存中的整个字都会被 ERASE 操作所清除。因此对某个字中的任何位执行 ERASE 命令，都会将该字的全部 16 位清除为状态‘1’。

要擦除错误计数器或是在应用区域存储的数据，请参考：

- 指定的 Erase Application Zone With Erase 命令。
- 指定的 Erase Application Zone With Write and Erase 命令。
- 指定的 Verify Security Code 命令。

命令格式

Pseudo-APDU				
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**
FFh	D2h	00h		00h

其中：

Byte Address (一个字节) = 待擦除的字的内存字节地址位置

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

2.9.5. Erase Application Zone with erase

此命令可用于下列情况：

- AT88SC101：擦除应用区域中的数据，EC 功能禁用
- AT88SC102：擦除应用区域 1 中的数据
- AT88SC102：擦除应用区域 2 中的数据，EC2 功能禁用
- AT88SC1003：擦除应用区域 1 中的数据
- AT88SC1003：擦除应用区域 2 中的数据，EC2 功能禁用
- AT88SC1003：擦除应用区域 3 中的数据

此命令执行以下操作：

1. 向卡片提交指定的密码。
2. 擦除密码输入错误计数器。提交的密码验证正确后，相应的应用区域中的数据可以被擦除。

命令格式

Pseudo-APDU									
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE				
					Byte 1	Byte 2	…	…	Byte N
FFh	20h	00h							

其中：

Error Counter LEN (1 个字节) = 密码输入错误计数器的长度，单位为比特，值始终是 00h。

Byte Address (1 个字节) = 卡片中的应用区密钥的字节地址。正确值请参阅下表：

MEM_L (1 个字节) = “擦除”密钥的长度。正确值请参阅下表：

CODE (1…N) = “擦除”密钥

例如	Byte Address	LEN
AT88SC101：擦除应用区域，EC 功能禁用	96h	04h
AT88SC102：擦除应用区域 1	56h	06h
AT88SC102：擦除应用区域 2，EC2 功能禁用	9Ch	04h
AT88SC1003：擦除应用区域 1	36h	06h
AT88SC1003：擦除应用区域 2，EC2 功能禁用	5Ch	04h
AT88SC1003：擦除应用区域 3	C0h	06h

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

注：*收到状态字 SW1SW2 = 90 00h 后，可重新读取应用区域内的数据来检查 Erase Application Zone with Erase 命令是否正确。如果应用区域的全部数据都被擦除并且等于'FFh'，则说明先前的验证成功。*

2.9.6. Erase Application Zone with Write and Erase

此命令可用于下列情况：

- AT88SC101：擦除应用区域中的数据，EC 功能启用。
- AT88SC102：擦除应用区域 2 中的数据，EC2 功能启用。
- AT88SC1003：擦除应用区域 2 中的数据，EC2 功能启用。

EC 或 EC2 功能启用后（即：ECEN 或 EC2EN 标识位没有被更改并处于"1"状态），会执行以下操作：

1. 向卡片提交指定的密码。
2. 搜索密码输入错误计数器中值为'1'的位，然后将该位写为'0'。
3. 擦除密码输入错误计数器。提交的密码验证正确后，相应的应用区域中的数据可以被擦除。

命令格式

Pseudo-APDU								
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE			
					Byte 1	Byte 2	Byte 3	Byte 4
FFh	20h	80h		04h				

其中：

Error Counter LEN (1 个字节) = 密码输入错误计数器的长度，单位为比特。值始终是 80h。

Byte Address (1 个字节) = 卡片中的应用区密钥的字节地址

例如	Byte Address
AT88SC101	96h
AT88SC102	9Ch
AT88SC1003	5Ch

其中：

CODE (4 个字节) = “擦除”密钥

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

如果不再有重试的机会，则该状态字= 63 00h。

注：*收到状态字 SW1 SW2 = 90 00h 后，可重新读取应用区域内的数据来检查 Erase Application Zone with Write and Erase 命令是否正确。如果应用区域的全部数据都被擦除并且等于‘FFh’，则说明先前的验证成功。*

2.9.7. Verify Security Code

此命令用于向插入的卡片提交安全密码（2 个字节）。安全密码旨在使卡的内存能够被访问。

执行的操作如下：

1. 向卡片提交指定的密码。
2. 搜索密码输入错误计数器中值为‘1’的位，然后将该位写为‘0’。
3. 擦除密码输入错误计数器。提交的密码验证正确后，安全密码尝试计数器可被擦除。

命令格式

Pseudo-APDU						
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE	
					Byte 1	Byte 2
FFh	20h	08h	0Ah	02h		

其中：

Error Counter LEN (1 个字节) = 密码输入错误计数器的长度，单位为比特

Byte Address (1 个字节) = 卡片中密钥的字节地址

CODE (2 个字节) = 安全密码

应答数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

如果不再有重试的机会，则该状态字= 63 00h。

注：*收到响应 SW1SW2 = 90 00h 后，应当再次读取安全密码尝试计数器（SCAC），检查 Verify User Code 是否正确。如果 SCAC 被擦除并且等于'FFh'，证明先前的验证成功。*

2.9.8. Blown Fuse

此命令用于更改所插入卡片的标识位。标识位可以是 EC_EN 标识位、EC2EN 标识位、发行商标识位或生产商标识位。

注：*改变标识位是一个不可逆的过程。*

命令格式

Pseudo-APDU								
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE			
					Fuse Bit Addr (High)	Fuse Bit Addr (Low)	State of FUS Pin	State of RST Pin
FFh	05h	00h	00h	04h			01h	00h 01h

其中：

Fuse Bit Addr (2 个字节) = 标识位的位地址。正确值请参阅下表

State of FUS Pin (1 个字节) = FUS pin 的状态。始终是 01h。

State of RST Pin (1 个字节) = RST pin 的状态。正确值请参阅下表。

例如	Fuse	Fuse Bit Addr (High)	Fuse Bit Addr (Low)	State of RST Pin
AT88SC101	生产商标识位	05h	80h	01h
	EC_EN 标识位	05h	C9h	01h
	发行商标识位	05h	E0h	01h
AT88SC102	生产商标识位	05h	B0h	01h
	EC2EN 标识位	05h	F9h	01h
	发行商标识位	06h	10h	01h
AT88SC1003	生产商标识位	03h	F8h	00h
	EC2EN 标识位	03h	FCh	00h
	发行商标识位	03h	E0h	00h

应答数据格式

SW1	SW2

其中

SW1 SW2 = 90 00h（未发生错误）

3.0. 非接触式智能卡协议

3.1. ATR 的生成

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

3.1.1. ATR 信息格式（适用于 ISO 14443-3 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（HistByte 0 - HistByte N-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		注册的应用提供商标识(RID) A0 00 00 03 06h
	SS		标准字节
	C0h ..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

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) = 0Ch

RID = {A0 00 00 03 06h} (PC/SC 工作组)

标准 (SS) = 03 (ISO 14443A，第 3 部分)

卡片名称(C0 ..C1) = {00 01h} (MIFARE 1K)

00 01h：MIFARE 1K
00 02h：MIFARE 4K
00 03h：MIFARE Ultralight
00 26h：MIFARE Mini
FF 28h：JCOP 30
FF [SAK]h：未定义的标签

3.1.2. ATR 信息格式（适用于 ISO 14443-4 PICC）

<table>
<tr><th>字节</th><th>值
(Hex)</th><th>标记</th><th>说明</th></tr>
<tr><td>0</td><td>3Bh</td><td>初始字符</td><td></td></tr>
<tr><td>1</td><td>8Nh</td><td>T0</td><td>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)</td></tr>
<tr><td>2</td><td>80h</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>01h</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>XXh</td><td>T1</td><td rowspan="2">历史字节：

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

ISO 14443B：
<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>XXh
XXh
XXh</td><td>Tk</td></tr>
<tr><td>4+N</td><td>UUh</td><td>TCK</td><td>T0 至 Tk 的所有字节按位异或</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 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

3.2. 非接触接口的私有 APDU 指令

私有APDU用于访问非接触标签和外围设备。

私有 APDU 通过 *PC_to_RDR_XfrBlock* 消息发送，其中 *bSlot* = 0。

3.2.1. Get Data

此命令用于获取“已建立连接的 PICC”的序列号或 ATS。

GET UID 的 APDU 结构（5 个字节）

命令	CLA	INS	P1	P2	Le
Get Data	FFh	CAh	00h 01h	00h	00h (Max Length)

若 P1 = 00h，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 = 01h，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 的末尾先于 Le 字节到达（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]={FFh, CAh, 00h, 00h, 00h};

// 获取“已经建立连接的 ISO 14443-A PICC”的 ATS

UINT8 GET_ATS[5]={FFh, CAh, 01h, 00h, 00h};

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

3.2.2.1. Load Authentication Keys

Load Authentication Keys 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：易失密钥位置和非易失密钥位置。

Load Authentication Keys 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）

00h = 密钥被载入读写器的易失性存储器

20h = 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）

00h ~ 1Fh = 使用非易失性存储器存储密钥。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。读写器的非易失存储器内可以存储最多 32 组密钥。

20h（过程密钥） = 使用易失性存储器来存储临时密钥。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（1）个，可以用作不同会话的过程密钥。**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 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	90h	00h	操作成功完成。
错误	63h	00h	操作失败。

例 1：

// 向非易失性存储器位置 05h 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 20h 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 20h"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 00h ~ 1Fh"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 020h"。*

3.2.2.2. Authentication for MIFARE 1K/4K

Authentication 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 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 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 01h	00h	块号	密钥类型	密钥号

其中：

块号（1 个字节） 待验证的存储块。一张 MIFARE 1K 卡分为 16 个扇区，每个扇区包含四（4）个连续的块。

例如：扇区 00h 包含块{00h、01h、02h 和 03h}；扇区 01h 包含块{04h、05h、06h 和 07h}；最后一个扇区 0Fh 包含块{3Ch、3Dh、3Eh 和 3Fh}。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） 60h = 该密钥被用作 TYPE A 密钥进行验证

61h = 该密钥被用作 TYPE B 密钥进行验证

密钥号（1 个字节） 00h ~ 1Fh = 使用非易失性存储器存储密钥。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。读写器的非易失存储器可以存储 32 个密钥。

20h（过程密钥） = 使用易失性存储器来存储密钥。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（1）个，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 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	00h ~ 02h	03h	
扇区 1	04h ~ 06h	07h	
..			
..			
扇区 14	38h ~ 0Ah	3Bh	
扇区 15	3Ch ~ 3Eh	3Fh	1K 字节

表 3：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	00h ~ 02h	03h	
扇区 1	04h ~ 06h	07h	
..			
..			
扇区 30	78h ~ 7Ah	7Bh	
扇区 31	7Ch ~ 7Eh	7Fh	2K 字节

表 4：MIFARE 4K 卡的内存结构

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	80h ~ 8Eh	8Fh
扇区 33	90h ~ 9Eh	9Fh
..		
..		
扇区 38	E0h ~ EEh	EFh
扇区 39	F0h ~ FEh	FFh

2K 字节

例如：

// 要使用{TYPE A，密钥号 00h}验证块 04h。

// PC/SC V2.01，弃用

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

<同样>

// 要使用{TYPE A，密钥号 00h}验证块 04h。

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

512 位
或
64 字节

表 5：MIFARE Ultralight 卡的内存结构

3.2.2.3. Read Binary Blocks

Read Binary Blocks 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 Block 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 1：10h（16 个字节）。仅起始块。（单块模式）

例 2：40h（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	90h	00h	操作成功完成。
错误	63h	00h	操作失败。

例如：

// 从二进制块 04h 中读取 16 个字节（MIFARE 1K 或 4K）

APDU = {FF B0 00 04 10h}

从二进制块 80h 开始读取 240 个字节（MIFARE 4K）

// 块 80h——块 8Eh（15 个块）

APDU = {FF B0 00 80 F0h}

3.2.2.4. Update Binary Blocks

Update Binary Blocks 命令用于向 PICC 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 1：10h（16 个字节）。仅起始块。（单块模式）

例 2：30h（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	90h	00h	操作成功完成。
错误	63h	00h	操作失败。

例如：

// 将 MIFARE 1K/4K 卡中的二进制块 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 0Fh}

// 将 MIFARE Ultralight 卡中的二进制块 04 的数据更新为{00 01 02 03h}

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

3.2.2.5. Value Block Operation (INC, DEC, STORE)

Value Block 命令用于进行数值操作（例如：增加值块的值等）。

Value Block Operation 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） 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。仅适用于对值块的操作。

02h = 使值块的值减少 VB_Value。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 = {FFh, FFh, FFh, FCh}

VB_Value			
MSB		LSB	
FFh	FFh	FFh	FCh

例 2： Decimal 1 = {00h, 00h, 00h, 01h}

VB_Value			
MSB		LSB	
00h	00h	00h	01h

Value Block Operation 的响应结构（2 个字节）

响应	响应数据域	
结果	SW1	SW2

Value Block Operation 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。

3.2.2.6. Read Value Block

Read Value Block 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 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 的响应结构（4 + 2 个字节）

响应	响应数据域		
结果	Value {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 = {FFh, FFh, FFh, FCh}

值			
MSB			LSB
FFh	FFh	FFh	FCh

例 2： Decimal 1 = {00h, 00h, 00h, 01h}

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。

3.2.2.7. Copy Value Block

Copy Value Block 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：

Source Block Number (1 个字节) = 源值块中的值会被复制到目标值块。

Target Block Number (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（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 05h

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

// 读取值块 05h

APDU = {FF B1 00 05 00h}

将值块 05h 的值复制到值块 06h

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

// 使值块 05h 的值增加“5”

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

3.2.3. 访问符合 PC/SC 标准的标签（ISO 14443-4)

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。ACR1281S 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。ACR1281S 会在内部处理 ISO 14443 第 1-4 部分协议。

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。更多相关信息，请参阅“MIFARE Classic 存储标签的 PICC 命令”。

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}

Class = 80h

INS = B2h

P1 = 80h

P2 = 00h

Lc = 无

命令数据域 = 无

Le = 08h

应答：00 01 02 03 04 05 06 07h [$9000]

4.0. 外设控制

对外围设备的访问应当通过发送*PC_to_RDR_Escape*消息来实现，其中bSlot = 0。

4.1. Get Firmware Version

Get Firmware Version 命令用于获取读写器的固件信息。

Get Firmware Version 的命令结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Get Firmware Version	E0h	00h	00h	18h	00h

Get Firmware Version 的响应结构（固件信息的长度）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	待接收的字节数	固件版本号

示例响应 = E1 00 00 00 0F 41 43 52 31 32 38 31 53 5F 56 33 30 33 2E 30h

固件版本号 (HEX) = 41 43 52 31 32 38 31 53 5F 56 33 30 33 2E 30h

固件版本号(ASCII) = “ACR1281S_V303.0”

4.2. LED Control

LED Control 命令用于控制 LED 输出。

LED Control 的命令结构（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
LED Control	E0h	00h	00h	29h	01h	LED 状态

LED Control 的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	LED 状态

其中：

LED 状态（1 个字节） – LED 控制

LED 状态	模式	描述
Bit 0	红色 LED	1 = 开；0 = 关
Bit 1	绿色 LED	1 = 开；0 = 关
Bit 2 - 7	RFU	RFU

4.3. LED Status

LED Status 命令用于检查当前 LED 的状态。

LED Status 的命令结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
LED 状态	E0h	00h	00h	29h	00h

LED Status 的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	LED 状态

其中：

LED 状态（1 个字节） – LED 状态

LED 状态	模式	描述
Bit 0	红色 LED	1 = 开；0 = 关
Bit 1	绿色 LED	1 = 开；0 = 关
Bit 2 - 7	RFU	RFU

4.4. Buzzer Control

Buzzer Control 命令用于控制蜂鸣器输出。

Buzzer Control 的命令结构（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Buzzer Control	E0h	00h	00h	28h	01h	蜂鸣器持续时间

其中：

蜂鸣器持续时间（1 个字节） 00h = 关闭

01 - FFh = 持续时间（单位：10ms）

Buzzer Control 的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	00

4.5. Set Default LED and Buzzer Behaviors

Set Default LED and Buzzer Behaviors 命令用于设置 LED 和蜂鸣器的默认操作属性。

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	E0h	00h	00h	21h	01h	默认操作

其中：

默认操作（1 个字节）

默认操作	模式	描述
Bit 0	ICC 激活状态 LED	显示 ICC 界面的激活状态 1 = 启用；0 = 禁用
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 接口的激活状态 1 = 启用；0 = 禁用
Bit 3	RFU	RFU
Bit 4	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。（包括 ICC 和 PICC） 1 = 启用；0 = 禁用
Bit 5	RC531 复位指示蜂鸣器	RC531 复位时发出哔的一声。 1 = 启用；0 = 禁用
Bit 6	独享模式状态蜂鸣器 ICC 或 PICC 界面只有一个可以被激活。	独享模式被激活时会发出哔的一声。 1 = 启用；0 = 禁用
Bit 7	卡片操作闪烁 LED	LED 在卡片（PICC 或 ICC）被访问时会闪烁。

注：默认操作的默认值 = FBh*

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	E1h	00h	00h	00h	01h	默认操作

4.6. Read Default LED and Buzzer Behaviors

Read Default LED and Buzzer Behaviors 命令用于读取 LED 和蜂鸣器的当前默认操作属性。。

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	E0h	00h	00h	21h	00h

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	E1h	00h	00h	00h	01h	默认操作

其中：

默认操作（1 个字节）

默认操作	模式	描述
Bit 0	ICC 激活状态 LED	显示 ICC 界面的激活状态 1 = 启用；0 = 禁用
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 接口的激活状态 1 = 启用；0 = 禁用
Bit 3	RFU	RFU
Bit 4	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。（包括 ICC 和 PICC） 1 = 启用；0 = 禁用
Bit 5	RC531 复位指示蜂鸣器	RC531 复位时发出哔的一声。 1 = 启用；0 = 禁用
Bit 6	独享模式状态蜂鸣器 ICC 或 PICC 界面只有一个可以被激活。	独享模式被激活时会发出哔的一声。 1 = 启用；0 = 禁用
Bit 7	卡片操作闪烁 LED	LED 在卡片（PICC 或 ICC）被访问时会闪烁。

注：*默认操作的默认值 = FBh*

4.7. Initialize Cards Insertion Counter

Initialize Cards Insertion Counter 命令用于初始化卡片插入/检测计数器。

Initialize Cards Insertion Counter 的命令结构（9 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域			
Initialize Cards Insertion Counter	E0h	00h	00h	09h	04h	ICC Cnt (LSB)	ICC Cnt (MSB)	PICC Cnt (LSB)	PICC Cnt (MSB)

Initialize Cards Insertion Counter 的响应结构（9 个字节）

响应	CLA	INS	P1	P2	Lc	响应数据域			
结果	E1h	00h	00h	00h	04h	ICC Cnt (LSB)	ICC Cnt (MSB)	PICC Cnt (LSB)	PICC Cnt (MSB)

其中：

ICC Cnt (LSB) (1 个字节) = ICC 插入计数器(LSB)

ICC Cnt (MSB) (1 个字节) = ICC 插入计数器(MSB)

PICC Cnt (LSB) (1 个字节) = PICC 插入计数器(LSB)

PICC Cnt (MSB) (1 个字节) = PICC 插入计数器(MSB)

4.8. Read Cards Insertion Counter

Read Cards Insertion Counter 命令用于查看卡片插入/检测计数器的值。

Read Cards Insertion Counter 的命令结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read Cards Insertion Counter	E0h	00h	00h	09h	00h

Read Cards Insertion Counter 的响应结构（9 个字节）

响应	CLA	INS	P1	P2	Lc	响应数据域			
结果	E1h	00h	00h	00h	04h	ICC Cnt (LSB)	ICC Cnt (MSB)	PICC Cnt (LSB)	PICC Cnt (MSB)

其中：

ICC Cnt (LSB) (1 个字节) = ICC 插入计数器(LSB)

ICC Cnt (MSB) (1 个字节) = ICC 插入计数器(MSB)

PICC Cnt (LSB) (1 个字节) = PICC 插入计数器(LSB)

PICC Cnt (MSB) (1 个字节) = PICC 插入计数器(MSB)

4.9. Update Cards Insertion Counter

Update Cards Insertion Counter 命令用于更新卡片插入/检测计数器的值。

Update Cards Insertion Counter 的命令结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Update Cards Insertion Counter	E0h	00h	00h	0Ah	00h

Update Cards Insertion Counter 的响应结构（9 个字节）

响应	CLA	INS	P1	P2	Lc	响应数据域			
结果	E1h	00h	00h	00h	04h	ICC Cnt (LSB)	ICC Cnt (MSB)	PICC Cnt (LSB)	PICC Cnt (MSB)

其中：

ICC Cnt (LSB) (1 个字节) = ICC 插入计数器(LSB)

ICC Cnt (MSB) (1 个字节) = ICC 插入计数器(MSB)

PICC Cnt (LSB) (1 个字节) = PICC 插入计数器(LSB)

PICC Cnt (MSB) (1 个字节) = PICC 插入计数器(MSB)

4.10. Set Automatic PICC Polling

Set Automatic PICC Polling 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 PCSC Escape 命令接口发送。

注：*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 的命令结构（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Set Automatic PICC Polling	E0h	00h	00h	23h	01h	轮询设置

Set Automatic PICC Polling 的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	轮询设置

其中：

轮询设置（1 个字节）

轮询设置	参数	说明
Bit 0	自动 PICC 轮询	1 = 启用；0 = 禁用
Bit 1	如果没有找到 PICC，关闭天线场	1 = 启用；0 = 禁用
Bit 2	如果 PICC 闲置，关闭天线场。	1 = 启用；0 = 禁用
Bit 3	检测到 PICC 后将其激活	1 = 启用；0 = 禁用
Bit 5 ..4	PICC 轮询间隔	<Bit 5 – Bit 4> <0 – 0> = 250 ms <0 – 1> = 500 ms <1 – 0> = 1000 ms <1 – 1> = 2500 ms
Bit 6	RFU	-
Bit 7	强制执行 ISO 14443A 第 4 部分	1 = 启用；0 = 禁用。

注：*轮询设置的默认值 = 8Fh*

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 尚未激活。*
3. *读写器会自动激活"ISO 14443A-4 PICC"的 ISO 14443A-4 模式。B 类 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，应用就必须要选定一种模式。*

4.11. Read Automatic PICC Polling

Read the Automatic PICC Polling 命令用于检查当前的自动 PICC 轮询设置。

Read Automatic PICC Polling 的命令结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read Automatic PICC Polling	E0h	00h	00h	23h	00h

Read the Configure mode 的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	轮询设置

其中：

轮询设置（1 个字节）

轮询设置	参数	说明
Bit 0	自动 PICC 轮询	1 = 启用；0 = 禁用
Bit 1	如果没有找到 PICC，关闭天线场。	1 = 启用；0 = 禁用
Bit 2	如果 PICC 闲置，关闭天线场。	1 = 启用；0 = 禁用
Bit 3	检测到 PICC 后将其激活	1 = 启用；0 = 禁用
Bit 5 ..4	PICC 轮询间隔	<Bit 5 – Bit 4> <0 – 0> = 250 ms <0 – 1> = 500 ms <1 – 0> = 1000 ms <1 – 1> = 2500 ms
Bit 6	RFU	-
Bit 7	强制执行 ISO 14443A 第 4 部分	1 = 启用；0 = 禁用。

注：*轮询设置的默认值 = 8Fh*

4.12. Set the PICC Operating Parameter

Set the PICC Operating Parameter 命令用于设置 PICC 的操作参数。

Set the PICC Operating Parameter 的命令结构（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Set the PICC Operating Parameter	E0h	00h	00h	20h	01h	操作参数

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

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	操作参数

其中：

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 = 03h。*

4.13. Read the PICC Operating Parameter

Read the PICC Operating Parameter 命令用于检查 PICC 的操作参数。

Read the PICC Operating Parameter 的命令结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read the PICC Operating Parameter	E0h	00h	00h	20h	00h

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

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	操作参数

其中：

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

4.14. Set the Exclusive Mode

Set the Exclusive Mode 命令用于设置读写器进入/离开独享模式。

Set the Exclusive Mode 的命令结构（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Set the Exclusive Mode	E0h	00h	00h	2Bh	01h	独享模式

Set the Exclusive Mode 的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	独享模式

其中：

独享模式(1 个字节)　　00h = 共享模式，ICC 和 PICC 同时工作

01h = 独享模式，插入 ICC 后，PICC 禁用自动轮询功能，天线关闭（默认）

4.15. Read the Exclusive Mode

Read the Exclusive Mode 命令用于检查当前独享模式的设置。

Read the Exclusive Mode 的命令结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read the Exclusive Mode	E0h	00h	00h	2Bh	00h

Read the Exclusive Mode 的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	独享模式

其中：

独享模式(1 个字节)

00 = 共享模式，ICC 和 PICC 同时工作

01 = 独享模式，插入 ICC 后，PICC 禁用自动轮询功能，天线关闭（默认）

4.16. Set Auto PPS

每次识别出 PICC，读写器都会尝试改变由**最快连接速度**定义的 PCD 和 PICC 之间的通信数据速率。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 的命令结构（7 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Set Auto PPS	E0h	00h	00h	24h	01h	Max Speed

Set Auto PPS 的响应结构（9 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	E1h	00h	00h	00h	02h	Max Speed	Current Speed

其中：

Max Speed (1 个字节) = 最大速度

Current Speed (1 个字节) = 当前速度

值可以为 106k bps = 00h -> 等于无自动 PPS（默认设置）

212k bps = 01h

424k bps = 02h

848k bps = 03h

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。*

4.17. Read Auto PPS

Read Auto PPS 命令用于检查当前的自动 PPS 设置。

Read Auto PPS 的命令结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read Auto PPS	E0h	00h	00h	24h	00h

Set Auto PPS 的响应结构（9 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	E1h	00h	00h	00h	02h	Max Speed	Current Speed

其中：

Max Speed (1 个字节) = 最大速度

Current Speed (1 个字节) = 当前速度

值可以为 106k bps = 00h -> 等于无自动 PPS（默认设置）

212k bps = 01h

424k bps = 02h

848k bps = 03h

4.18. Antenna Field Control

Antennal Field Control 命令用于打开/关闭天线场。

Antenna Field Control 的命令结构（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Antenna Field Control	E0h	00h	00h	25h	01h	状态

Antenna Field Control 的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	状态

其中：

状态（1 个字节）： 01h = 启用天线场

00h = 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。*

4.19. Read Antenna Field Status

Read Antenna Field Status 命令用于检查当前的天线场状态。

Read Antenna Field Status 的命令结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read Antenna Field Status	E0h	00h	00h	25h	00h

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

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	状态

其中：

状态（1 个字节） 01h = 启用天线场

00h = 停用天线场

4.20. User Extra Guard Time Setting

User Extra Guard Time Setting 命令用于设置 ICC 和 SAM 通信的额外保护时间。

注：*用户额外保护时间值会被存储至 EEPROM 中。*

User Extra Guard Time Setting 的命令格式（7 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域	
User Extra Guard Time Setting	E0h	00h	00h	2Eh	02h	ICC UserGuardTime	SAM UserGuardTime

User Extra Guard Time Setting 的响应格式（7 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	E1h	00h	00h	00h	02h	ICC UserGuardTime	SAM UserGuardTime

其中：

ICC UserGuardTime (1 个字节) = ICC 卡槽的用户额外保护时间值

SAM UserGuardTime (1 个字节) = SAM 卡槽的用户额外保护时间值

4.21. Read User Extra Guard Time

Read User Extra Guard Time 命令用于读取为 ICC 和 SAM 通信设置的额外保护时间。

Read User Extra Guard Time 的命令格式（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read User Extra Guard Time	E0h	00h	00h	2Eh	00h

Read User Extra Guard Time 的响应格式（7 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	E1h	00h	00h	00h	02h	ICC UserGuardTime	SAM UserGuardTime

其中：

ICC UserGuardTime (1 个字节) = ICC 卡槽的用户额外保护时间值

SAM UserGuardTime (1 个字节) = SAM 卡槽的用户额外保护时间值

4.22. “616C” Auto Handle Option Setting

“616C” Auto Handle Option Setting 命令用于设置“616C”自动操作选项。

*T=0 ACOS5 的可选命令

“616C” Auto Handle Option Setting 的命令格式（7 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域	
“616C” Auto Handle Option Setting	E0h	00h	00h	32h	02h	ICC Option	SAM Option

“616C” Auto Handle Option Setting 的响应格式（7 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	E1h	00h	00h	00h	02h	ICC Option	SAM Option

其中：

ICC Option (1 个字节): ICC 卡槽的用户保护时间值

FFh = 启用“616C”自动操作

00h = 停用“616C”自动操作（默认）

SAM Option (1 个字节): SAM 卡槽的用户保护时间值

FFh = 启用“616C”自动操作

00h = 停用“616C”自动操作（默认）

4.23. Read “616C” Auto Handle Option

Read “616C” Auto Handle Option 命令用于读取“616C”自动操作选项。

Read “616C” Auto Handle Option 的命令格式（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read “616C” Auto Handle Option	E0h	00h	00h	32h	00h

Read “616C” Auto Handle Option 的响应格式（7 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	E1h	00h	00h	00h	02h	ICC Option	SAM Option

其中：

ICC Option (1 个字节): ICC 卡槽的用户保护时间值

FFh = 启用“616C”自动操作

00h = 停用“616C”自动操作（默认）

SAM Option (1 个字节): SAM 卡槽的用户保护时间值

FFh = 启用“616C”自动操作

00h = 停用“616C”自动操作（默认）

4.24. Set Serial Communication Mode

Set Serial Communication Mode 命令用于设置通信速度和通信模式。

Set Serial Communication Mode 的命令结构（2 个字节）

命令	Byte 0	Byte 1
Set Serial Communication Mode	44h	Mode Select

Set Serial Communication Mode 的响应结构（2 个字节）

响应	Byte 0	Byte 1
结果	90h	Mode Select

偏移	参数	说明
Bit 0-3	串行通信速度	000b= 9600bps(默认) 001b= 19200bps 010b= 38400bps 011b= 57600bps 100b= 115200bps 101b= 128000bps 110b= 230400bps 其他值保留为将来使用。
Bit 4 - 6	RFU	RFU
Bit 7	Interrupt-In 消息（仿 CCID 架构）	1 = 报告 Interrupt-In 消息。 0 = 无报告（默认）

表 6：模式选择（1 个字节） – 通信速度和模式选择

注：*成功修改通信速度后，程序必须对通信速度进行调整，以便继续剩下的数据交换。*

附录 A. 支持的卡片类型

下表总结了 GET_READER_INFORMATION 命令返回的卡片类型数据以及相对应的卡片。

卡片类型代码	卡片类型
00h	自动选择 T=0 或 T=1 通信协议
01h	I2C 存储卡 (1k, 2k, 4k, 8k 和 16k bits)
02h	I2C 存储卡 (32k, 64k, 128k, 256k, 512k 和 1024k bits)
03h	Atmel AT88SC153 安全存储卡
04h	Atmel AT88SC1608 安全存储卡
05h	Infineon SLE4418 和 SLE4428
06h	Infineon SLE4432 和 SLE4442
07h	Infineon SLE4406, SLE4436 和 SLE5536
08h	Infineon SLE4404
09h	Atmel AT88SC101、AT88SC102 和 AT88SC1003

表 7：支持的卡片类型