Advanced Card Systems Ltd.
Card & Reader Technologies

ACR33U-A1 SmartDuo

智能卡读写器

参考手册 V1.05

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

目录

表

1.0. 简介

ACR33U-A1 SmartDuo 连机智能卡读写器是计算机与智能卡间的通信接口。不同类型的智能卡采用不同的命令和不同的通信协议，这在大多数情况下使智能卡和计算机之间不能直接进行通信。ACR33U-A1 SmartDuo 智能卡读写器可以为多种卡片建立一个从计算机到智能卡的统一接口。它兼顾了卡片的各种特性而使得计算机软件程序员不需要负责有关智能卡操作的技术细节，在许多情况下，这些细节与智能卡系统的实施并无太大关系。

1.1. 参考文件

下列文件可以在 www.usb.org 下载。

- 通用串行总线规范 2.0（即 USB 规范），2000 年 4 月 27 日
- 通用串行总线通用类规范 1.0，1997 年 12 月 16 日
- 通用串行总线设备类：集成电路（S）卡接口设备的智能卡 CCID 规范，1.1 版，2005 年 4 月 22 日

下列文件可以在 www.ansi.org 订购。

- ISO/IEC 7816-1：识别卡 — 带触点的集成电路卡 - 第一部分：物理特性
- ISO/IEC 7816-2：识别卡 — 带触点的集成电路卡 - 第二部分：触点的尺寸和位置
- ISO/IEC 7816-3：识别卡 — 带触点的集成电路卡 - 第三部分：电信号和传输协议

1.2. 符号和缩写

缩写	描述
ATR	复位应答（Answer-to-reset）
CCID	芯片/智能卡接口设备（Chip/Smart Card Interface Device）
ICC	集成电路卡（Integrated Circuit Cards）
IFSC	T=1 的集成电路卡信息域大小（Information Field Sized for ICC for protocol T=1）
IFSD	T=1 的芯片/智能卡接口设备信息域大小（Information Field Sized for ICC for protocol T=1）
NAD	节点地址（Node Address）
PPS	协议与参数选择（Protocol and Parameters Selection）
RFU	保留为将来使用（Reserved for future use[1]）
TPDU	传输协议数据单元（Transport Protocol Data Unit）
USB	通用串行总线（Universal Serial Bus）

表 1： 符号和缩写

[1] *除非另有约定，必须设置为 0*

2.0. 特征

- USB 全速接口
- 即插即用 - 符合 CCID 标准，兼容性强
- 两个全尺寸智能卡卡槽
- 智能卡读写器：
  - 支持 ISO 7816 A 类（5 V）智能卡
  - 支持通用权限卡（CAC）
  - 支持符合 T=0 或 T=1 协议的微处理器卡
  - 支持存储卡
  - 支持 PPS（协议和参数选择）
  - 具有短路保护功能
- SAM 卡接口:
  - 三个 SAM 卡槽
- 用户可控的外设
  - 三色 LED 指示灯（绿色、红色、蓝色）
  - 蜂鸣器
- 应用程序编程接口：
  - 支持 PC/SC
  - 支持 CT-API（通过 PC/SC 上一层的封装）
- 支持 Android™ 3.1 及以上版本 [2]
- 符合下列国际标准：
  - FIPS 201
  - TAA
  - EN60950/IEC 60950
  - ISO 7816
  - CE
  - FCC
  - VCCI
  - PC/SC
  - CCID
  - Microsoft® WHQL
  - RoHS 2
  - REACH

[2] *不适用 PC/SC 和 CCID 支持*

3.0. 支持的智能卡

3.1. MCU 卡

ACR33U-A1 是一款符合 PC/SC 规范的智能卡读写器，可支持 ISO 7816 A 类（5 V）智能卡，还可以读写所有符合 T=0 或 T=1 协议的微控制器卡（MCU）。

若卡片产生的 ATR 指定了专用的操作模式（TA2 存在；TA2 中的 b5 位必须为 0），但 ACR33U-A1 不支持该特定模式，则 ACR33U-A1 会将卡片复位，使其置为协商模式。如果卡片不能被置为协商模式，ACR33U-A1 会拒绝读写该卡。

若卡片产生的 ATR 指定了协商模式（TA2 不存在时）和通信参数，而不是默认参数，则 ACR33U-A1 读卡器将执行 PPS 并尝试使用卡片在 ATR 中指定的通信参数。如果卡片不接受 PPS，读卡器会使用默认参数（F=372，D=1）。

注： *对于上述参数的含义，请参照* ***ISO 7816-3***。

3.2. 存储卡

ACR33U-A1 支持几种类型的存储卡，例如：

- 符合 I2C 总线协议且一次能写 128 字节/页的存储卡（空白存储卡），包括：
  - Atmel®：AT24C01/02/04/08/16/32/64/128/256/512/1024
- 具有安全记忆体 IC 以及密码和认证功能的存储卡，包括：
  - Atmel®：AT88SC153 和 AT88SC1608
- 具有 1 KB EEPROM 智能存储空间以及写保护功能的存储卡，包括：
  - Infineon®：SLE4418, SLE4428, SLE5518 和 SLE5528
- 具有 256 字节 EEPROM 智能存储空间以及写保护功能的存储卡，包括：
  - Infineon®：SLE4432, SLE4442, SLE5532 和 SLE5542
- ‘104’型 EEPROM 不可重置标记计数卡，包括：
  - Infineon®：SLE4406, SLE4436, SLE5536 和 SLE6636
- 包含应用区域的安全逻辑卡，包括：
  - Atmel®：AT88SC101, AT88SC102 和 AT88SC1003

注： *ACR33U-A1 SmartDuo 智能卡读写器只在* ***ICC 卡槽 0*** *支持存储卡。*

4.0. 智能卡界面

ACR33U-A1 与插入的智能卡之间的界面遵循 ISO 7816-3 标准，并进行了某些限制或提升来增强 ACR33U-A1 的实用功能。

4.1. 智能卡电源 VCC（C1）

插入的智能卡电流消耗不得大于 50 mA。

4.2. 编程电压 VPP（C6）

根据 ISO 7816-3 的规定，由智能卡上的触点 C6（VPP）为智能卡提供编程电压。但由于市面上的智能卡大多数基于 EEPROM，不需要为其提供外部编程电压，ACR33U-A1 的触点 C6（VPP）已被实现为普通的控制信号。此触点的电气规格与 RST 信号（触点 C2）的规格相同

4.3. 卡片类型选择

每次激活插入的卡片前，处于控制地位的电脑都要向 ACR33U-A1 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。当基于 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哪一种。

4.4. 微控制器卡界面

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 MHz。

4.5. 卡片插拔保护

ACR33U-A1 提供了一种机制来保护在上电状态下被突然拔出的卡片。当卡片被移出时，卡片的电源以及 ACR33U-A1 与卡之间的信号线路会立即取消激活。但是作为惯例，只应在断电后才应从读卡器移出卡片，这样可以避免电气损伤。

注： *ACR33U-A1 本身不会接通向卡片的供电。此操作由处于控制地位的电脑向读卡器发送适当的命令来进行。*

5.0. 电源

ACR33U-A1 需要 5 V，100 mA 的直流稳压电源，由计算机供电（通过与各类型读卡器一起提供的电缆）。

5.1. LED 状态指示灯

LED 可以指示智能卡界面的激活状态：

- **缓慢闪烁（每 2 秒钟开启 200 毫秒）**
  表示 ACR33U-A1 已上电并处于待机状态。智能卡尚未被插入，或者智能卡尚未上电（如果卡片已经插入）
- **长亮**
  表示已经开启向智能卡的供电，即智能卡处于激活状态。
- **快速闪烁**
  表示 ACR33U-A1 和智能卡间存在通信。

不同颜色的 LED 代表了 ACR33U-A1 的不同状态，其中：

- **红色 LED**
  供电状态
- **绿色 LED**
  主卡槽状态（ICC 卡槽 0）
- **蓝色 LED**
  副卡槽状态（ICC 卡槽 1）

6.0. USB 接口

6.1. 通信参数

ACR33U-A1 按照 USB2.0 规范的要求通过 USB 接口与计算机连接，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-	ACR33U-A1 和 PC 间以微分信号传输数据
3	D+	ACR33U-A1 和 PC 间以微分信号传输数据
4	GND	参考电压等级

表 2：USB 接口配线

注： *要使 ACR33U-A1 能够通过 USB 接口正常工作，必须先安装 ACS 专有设备驱动程序或 ACS PC/SC 设备驱动程序。*

6.2. 端点

ACR33U-A1 通过以下端点与主计算机进行通信。

控制端点（Control Endpoint）	用于进行设置和控制
批量输出（Bulk OUT）	用于从主机发送至 ACR33U-A1 的命令 （数据包的大小为 64 字节）
批量输入（Bulk IN）	用于从 ACR33U-A1 发送至主机的响应 （数据包的大小为 64 字节）
中断输入（Interrupt IN）	用于从 ACR33U-A1 发送至主机的卡片状态消息 （数据包的大小为 8 字节）

7.0. 通信协议

ACR33U-A1 应当通过 USB 连接与主机(host)端建立接口。现在的行业内部规范 -- 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 涵盖了操作智能卡所需的全部协议。

ACR33U-A1 上 USB 端点的配置和使用应当符合 CCID 标准的第三部分。

概述总结如下：

1. **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。
2. **CCID 事件**通过中断通道发送。
3. **CCID 命令**由 BULK-OUT 端点发出。发送至 ACR33U-A1 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。
4. **CCID 响应**由 BULK-IN 端点发出。所有发送至 ACR33U-A1 的命令都必须同步发送（例如：对于 ACR33U-A1 来说，*bMaxCCIDBusySlots* 等同于 01h）。

ACR33U-A1 支持的 CCID 特性见其类别描述符：

Offset	Field	Size	Value	Description
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	0100h	CCID 以二进制编码的十进制指定版本号码
4	*bMaxSlotIndex*	1	05h	ACR33U-A1 有 2 个大卡槽和 3 个 SAM 卡槽
5	*bVoltageSupport*	1	01h	ACR33U-A1 可支持 5V 的槽位电压
6	*dwProtocols*	4	00000003h	ACR33U-A1 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	00054024h	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	00000FEh	ACR33U-A1 T1 支持的最大 IFSD 为 254
32	*dwSynchProtocols*	4	00000000h	ACR33U-A1 不支持同步卡
36	*dwMechanical*	4	00000000h	ACR33U-A1 不支持特殊机制特性

Offset	Field	Size	Value	Description
40	*dwFeatures*	4	000204B0h	ACR33U-A1 支持以下特性： • ICC 时钟频率根据参数自动更改 • 波特率根据频率和 FI、DI 参数自动更改 • CCID 根据活动参数自动进行 PPS • 自动 IFSD 交换作为第一次交换（T=1 协议下） • 与 CCID 进行短 APDU 级交换
44	*dwMaxCCIDMessageLength*	4	0000010Fh	ACR33U-A1 可接受的最大信息长度为 271 字节。
48	*bClassGetResponse*	1	00h	对 TPDU 级别的交换没有影响
49	*bClassEnvelope*	1	00h	对 TPDU 级别的交换没有影响
50	*wLCDLayout*	2	0000h	没有 LCD
52	*bPINSupport*	1	03h	支持 PIN 验证和修改
53	*bMaxCCIDBusySlots*	1	01h	可以同时忙的槽位数为 1

8.0. 选择存储卡类型

对存储卡执行其他命令前，必须先执行 SELECT_CARD_TYPE 命令。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。只有使用 *SCardConnect()* API 建立逻辑智能卡读写器通信之后才可以使用此命令。

注： *关于 SCardConnect() API 的详细说明参见 PC/SC 规范。*

下列程序代码段演示了如何选择存储卡类型：

```
SCARDCONTEXT hContext;
SCARDHANDLE hCard;
unsigned long dwActProtocol;
SCARD_IO_REQUEST ioRequest;
DWORD size = 64, SendLen = 6, RecvLen = 255, retCode;
byte cardType;
//Establish PC/SC Connection
retCode   =   SCardEstablishContext   (SCARD_SCOPE_USER,   NULL,   NULL,
&hContext);
//List all readers in the system
retCode = SCardListReaders (hContext, NULL, readerName, &size);
//Connect to the reader
retCode   =   SCardConnect(hContext,   readerName,   SCARD_SHARE_SHARED,
SCARD_PROTOCOL_T0, &hCard, &dwActProtocol);
//Select Card Type
unsigned char SendBuff[] = {0xFF,0xA4,0x00,0x00,0x01,cardType};
retCode  =  SCardTransmit(  hCard,  &ioRequest,  SendBuff,  SendLen,  NULL,
RecvBuff, &RecvLen);
//Disconnect from the reader
retCode = SCardDisconnect(hCard, SCARD_UNPOWER_CARD);
//End the established context
retCode = SCardReleaseContext(hContext);
```

9.0. 命令

9.1. CCID 命令通道，Bulk-OUT 消息

ACR33U-A1 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 ACR33U-A1 支持的 CCID 类 Bulk-OUT 消息。

9.1.1. PC_to_RDR_IccPowerOn

此命令用于激活卡槽并返回卡片的 ATR。

Offset	Field	Size	Value	Description
0	*bMessageType*	1	62h	
1	*dwLength*	4	00000000h	此消息的额外字节的大小。
2	*bSlot*	1	00-05h	标识命令的插槽号。
5	*bSeq*	1	00-FFh	命令的序号。
6	*bPowerSelect*	1	01h	ICC 上的可选电压值： 01h = 5 volts
7	*abRFU*	2		保留为将来使用。

此命令消息的响应是 *RDR_to_PC_DataBlock* 响应消息，返回的是复位应答（ATR）数据。

9.1.2. PC_to_RDR_IccPowerOff

此命令用于取消激活卡槽。

Offset	Field	Size	Value	Description
0	*bMessageType*	1	63h	
1	*dwLength*	4	00000000h	此消息的额外字节的大小。
5	*bSlot*	1	00-05h	标识命令的插槽号。
6	*bSeq*	1	00-FFh	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。

9.1.3. PC_to_RDR_GetSlotStatus

此命令用于获取卡槽的当前状态。

Offset	Field	Size	Value	Description
0	*bMessageType*	1	65h	
1	*dwLength*	4	00000000h	此消息的额外字节的大小。
5	*bSlot*	1	00-05h	标识命令的插槽号。
6	*bSeq*	1	00-FFh	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。

9.1.4. PC_to_RDR_XfrBlock

此命令用于向 ICC 传输数据块。

Offset	Field	Size	Value	Description
0	*bMessageType*	1	6Fh	
1	*dwLength*	4		此消息的 abData 数据域的大小。
5	*bSlot*	1	00-05h	标识命令的插槽号。
6	*bSeq*	1	00-FFh	命令的序号。
7	*bBWI*	1	00-FFh	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。

9.1.5. PC_to_RDR_GetParameters

此命令用于获取卡槽参数。

Offset	Field	Size	Value	Description
0	*bMessageType*	1	6Ch	
1	*dwLength*	4	00000000h	此消息的额外字节的大小。
5	*bSlot*	1	00-05h	标识命令的插槽号。
6	*bSeq*	1	00-FFh	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。

9.1.6. PC_to_RDR_SetParameters

此命令用于设置卡槽参数。

Offset	Field	Size	Value	Description
0	*bMessageType*	1	61h	
1	*dwLength*	4		此消息的额外字节的大小。
5	*bSlot*	1	00-05h	标识命令的插槽号。
6	*bSeq*	1	00-FFh	命令的序号。
7	*bProtocolNum*	1	00h, 01h	指定后面的协议数据结构。 00h = T=0 协议结构 01h = T=1 协议结构 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。

此消息的响应是 *RDR_to_PC_Parameters* 消息。

T=0 协议的协议数据结构（*dwLength*=00000005h）

Offset	Field	Size	Value	Description
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	00-FFh	两个字符间的额外保护时间。在通常保护时间（12etu）的基础上增加 0-254 个 etu。FFh 与 00h 相同。
13	*bWaitingIntegerT0*	1	00-FFh	WI for T=0，用于定义 WWT
14	*bClockStop*	1	00-03h	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

此消息的响应是 *RDR_to_PC_Parameters* 消息。

T=1 协议的协议数据结构（*dwLength*=00000007h）

Offset	Field	Size	Value	Description
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	00-FFh	额外保护时间（两个字符间为 0 至 254etu）。如果值为 FFh，则保护时间减少 1etu。
13	*BwaitingIntegerT1*	1	00-9Fh	B7-4 = BWI 值 0-9 有效 B3-0 = CWI 值 0-Fh 有效
14	*bClockStop*	1	00-03h	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	00-FFh	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。

9.1.7. PC_to_RDR_Escape

此命令用于定义及访问扩展特性。

Offset	Field	Size	Value	Description
0	*bMessageType*	1	6Bh	
1	*dwLength*	4		此消息的 *abData* 数据域的大小。
5	*bSlot*	1	00-05h	标识命令的插槽号。
6	*bSeq*	1	00-FFh	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。

9.1.7.1. LED

Offset	Field	Size	Value	Description
10	*bcmdCode*	1	01h	
11	*wcmdLength*	2	0001h	
13	*abRFU*	2		保留为将来使用。
15	*abData*	1	00000 XYZb	00000xxx：3 个 LED， XYZ: 000 => 3 个 LED 全部关闭 XYZ: 001 => 第 1 个 LED 亮，绿色，一秒钟 XYZ: 010 => 第 2 个 LED 亮，红色，一秒钟 XYZ: 100 => 第 3 个 LED 亮，蓝色，一秒钟

此命令消息的响应是 *RDR_to_PC_Escape* 消息。

9.1.7.2. 蜂鸣器

Offset	Field	Size	Value	Description
10	*bcmdCode*	1	08h	
11	*wcmdLength*	2	0001h	
13	*abRFU*	2		保留为将来使用。
15	*abData*	1	XXh	XX 表示蜂鸣器打开或关闭， YZ：5A = 蜂鸣器开启 1 秒钟 YZ：A5 = 蜂鸣器关闭

此命令消息的响应是 *RDR_to_PC_Escape* 消息。

9.1.7.3. 获取固件版本

Offset	Field	Size	Value	Description
10	*bcmdCode*	1	04h	
11	*wcmdLength*	2	0000h	
13	*abRFU*	2		保留为将来使用。

此命令消息的响应是 *RDR_to_PC_Escape* 消息。

9.2. CCID 响应通道，Bulk-IN 消息

Bulk-IN 消息用于对 Bulk-OUT 消息做出响应。ACR33U-A1 应当遵循 CCID 协议第四部分定义的 Bulk-IN 消息。

此节列举了 ACR33U-A1 支持的 CCID 类 Bulk-IN 消息。

注： *槽号（bSlot）和序列号（bSeq）的值与 Bulk-OUT 消息中的值相同。*

9.2.1. RDR_to_PC_DataBlock

此消息由 ACR33U-A1 发出，是对命令 *PC_to_RDR_IccPowerOn* 和 *PC_to_RDR_XfrBlock* 的响应。

Offset	Field	Size	Value	Description
0	*bMessageType*	1	80h	表示 CCID 正在发送一个数据块。
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 节定义的插槽错误寄存器。
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。

9.2.2. RDR_to_PC_SlotStatus

此消息由 ACR33U-A1 发出，是对 *PC_to_RDR_IccPowerOff* 和 *PC_to_RDR_GetSlotStatus* 的响应。

Offset	Field	Size	Value	Description
0	*bMessageType*	1	81	
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。

9.2.3. RDR_to_PC_Parameters

此消息由 ACR33U-A1 发出，是对 *PC_to_RDR_GetParameters* 和 *PC_to_RDR_SetParameters* 消息的响应。

Offset	Field	Size	Value	Description
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 CCID 10.1.6 节汇总。

T=0 协议的协议数据结构（*bProtocolNum*=0，*dwLength*=00000005h）

Offset	Field	Size	Value	Description
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	00h, 02h	For T=0, B0 – 0b, B7-2 – 000000b B1 – 使用的约定（b1=0：正向约定；b1=1：反向约定）
12	*bGuardTimeT0*	1	00-FFh	两个字符间的额外保护时间。在通常保护时间（12etu）的基础上增加 0-254 个 etu。FFh 与 00h 相同。
13	*bWaitingIntegerT0*	1	00-FFh	WI for T=0，用于定义 WWT
14	*bClockStop*	1	00-03h	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（*bProtocolNum*=1，*dwLength*=00000007h）

Offset	Field	Size	Value	Description
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	10h, 11h, 12h, 13h	For T-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）
12	*bGuardTimeT1*	1	00-FFh	额外保护时间（两个字符间为 0 至 254etu）。如果值为 FFh，则保护时间减少 1h。
13	*bwaitingIntegerT1*	1	00-9Fh	B7-4 = BWI B3-0 = CWI
14	*bClockStop*	1	00-03h	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	00-FFh	商定的 IFSC 的大小
16	*bNadValue*	1	00h	只支持 NAD = 00h

9.2.4. RDR_to_PC_Escape

此消息由 ACR33U-A1 发出，是对 *PC_to_RDR_Escape* 消息的响应。

Offset	Field	Size	Value	Description
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 节定义的插槽错误寄存器。
9	*bRFU*	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。

9.2.4.1. LED

此消息由 ACR33U-A1 发出，是对 *PC_to_RDR_Escape* LED 消息的响应。

Offset	Field	Size	Value	Description
10	*bcmdCode*	1	81	
11	*wcmdLength*	2	0000	
13	*abStatus*	2	00XX	XXh for SW2： 00h：成功 01h：错误的参数

9.2.4.2. 蜂鸣器

此消息由 ACR33U-A1 发出，是对 *PC_to_RDR_Escape* 蜂鸣器消息的响应。

Offset	Field	Size	Value	Description
10	*bcmdCode*	1	88h	
11	*wcmdLength*	2	0000h	
13	*abStatus*	2	00XXh	XXh for SW2： 00h：成功 01h：错误的参数

9.2.4.3. 获取固件版本

此消息由 ACR33U-A1 发出，是对 *PC_to_RDR_Escape* 获取固件版本消息的响应。

Offset	Field	Size	Value	Description
10	*bcmdCode*	1	84h	
11	*wcmdLength*	2	0004h	
13	*abStatus*	2	00XXh	XXh： 00h：成功 01h：错误的参数
15	*abData*	13	41h 43h 52h 33h 33h 2Dh 41h 31h 20h XXh XXh XXh XXh	XX XX XX XX：固件版本

9.3. 存储卡命令集

存储卡可以通过 *PC_to_RDR_XfrBlock* 命令被访问。所有的存储卡功能均被映射进了 pseudo-APDU。

9.3.1. 存储卡 – 1、2、4、8、16 kilobit I2C 卡

9.3.1.1. 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。

注： *只有使用 SCardConnect( ) API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 SCardConnect( ) API 的详细说明参见 PC/SC 规范。*

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.1.2. 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（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

Page size = 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

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	**SW2**

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.1.3. READ_MEMORY_CARD

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

Byte Address 存储卡的内存地址位置。

MEM_L 待从存储卡读取的数据的长度。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

BYTE 1	…	…	BYTE N	SW1	SW2

其中：

BYTE x 从存储卡中读取的数据。

如果没有错误，**SW1 SW2** = 90 00h。

9.3.1.4. WRITE_MEMORY_CARD

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

Byte Address 存储卡的内存地址位置。

MEM_L 待写入存储卡的数据的长度。

Byte x 待写入存储卡的数据。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.2. 存储卡 – 32、64、128、256、512、1024 kilobit I2C 卡

9.3.2.1. 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。

注：*只有使用 SCardConnect( ) API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 SCardConnect( ) API 的详细说明参见 PC/SC 规范。*

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.2.2. 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（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

Data 待发送给卡片的 TPDU

Page size = 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

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.2.3. READ_MEMORY_CARD

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FFh				

其中：

INS = B0h：32、64、128、256、512 kilobit iic 卡

= 1011 000*b：1024 kilobit iic 卡，其中*为 17 位寻址的 MSB

Byte Address 存储卡的内存地址位置。

MEM_L 待从存储卡读取的数据的长度。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

BYTE 1	…	…	BYTE N	SW1	SW2

其中：

BYTE x 从存储卡中读取的数据。

如果没有错误，**SW1 SW2** = 90 00h。

9.3.2.4. WRITE_MEMORY_CARD

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

INS = D0h：32、64、128、256、512 kilobit iic 卡

= 1101 000*b：1024 kilobit iic 卡，其中*为 17 位寻址的 MSB。

Byte Address 存储卡的内存地址位置。

MEM_L 待写入存储卡的数据的长度。

Byte x 待写入存储卡的数据。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.3. 存储卡 – ATMEL AT88SC153

9.3.3.1. 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。还将选择页面大小为 8 字节页写。

注：*只有使用 SCardConnect( ) API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 ScardConnect( ) API 的详细说明参见 PC/SC 规范。*

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	**SW2**

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.3.2. READ_MEMORY_CARD

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

INS = B0h：读取 00b
= B1h：读取 01b
= B2h：读取 10b
= B3h：读取 11b
= B4h：读取标识位

Byte Address 存储卡的内存地址位置。

MEM_L 待从存储卡读取的数据的长度。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

BYTE 1	…	…	BYTE N	SW1	SW2

其中：

BYTE x 从存储卡中读取的数据。

如果没有错误，**SW1 SW2** = 90 00h。

9.3.3.3. WRITE_MEMORY_CARD

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

Pseudo-APDU								
CLA	**INS**	**P1**	**Bye Address**	**MEM_L**	**Byte 1**	**....**	**....**	**Byte n**
FFh		00h						

其中：

INS = D0h：写 00b
= D1h：写 01b
= D2h：写 10b
= D3h：写 11b
= D4h：写标识位

Byte Address 存储卡的内存地址位置。

MEM_L 待写入存储卡的数据的长度。

MEM_D 待写入存储卡的数据。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.3.4. VERIFY_PASSWORD

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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) 要发送给存储卡的密码

P2 = 0000 00rpb

其中的“rp”位指明待比较的密码

r = 0：写密码，

r = 1：读密码，

p = 密码集数，

rp = 01：安全密码

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2 ErrorCnt
90h	

其中：

SW1 = 90h。

SW2 (ErrorCnt) = 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。

9.3.3.5. INITIALIZE_AUTHENTICATION

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

Q(0),Q(1)…Q(7) 主机随机数，8 个字节。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	**SW2**

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.3.6. VERIFY_AUTHENTICATION

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

Ch(0),Ch(1)…Ch(7) 主机挑战数，8 个字节。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.4. 存储卡 – ATMEL AT88SC1608

9.3.4.1. 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。还将选择页面大小为 16 字节页写。

注：*只有使用 SCardConnect( ) API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 SCardConnect( ) API 的详细说明参见 PC/SC 规范。*

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	**SW2**

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.4.2. READ_MEMORY_CARD

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

Pseudo-APDU				
CLA	**INS**	**Zone Address**	**Byte Address**	**MEM_L**
FFh				

其中：

INS = B0h：读取用户区

= B1h：读取配置区或读取标识位

Zone Address = 0000 $0A_{10}A_9A_8$ b，其中 A_{10} 是区域地址的 MSB

= 并不一定要读取标识位

Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$ b 是存储卡的内存地址位置

= 1000 0000b ：读取标识位

MEM_L 待从存储卡读取的数据的长度。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

BYTE 1	…	…	BYTE N	SW1	SW2

其中：

BYTE x 从存储卡中读取的数据。

如果没有错误，**SW1 SW2** = 90 00h。

9.3.4.3. WRITE_MEMORY_CARD

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

INS = D0h：写用户区

= D1h：写配置区或写标识位

Zone Address = 0000 $0A_{10}A_9A_8$ b，其中 A_{10} 是区域地址的 MSB

= 并不一定要写标识位

Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$ b 是存储卡的内存地址位置

= 1000 0000_b ：写标识位

MEM_L 待写入存储卡的数据的长度。

Byte x 待写入存储卡的数据。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.4.4. VERIFY_PASSWORD

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

Pw(0),Pw(1),Pw(2)：待发送给存储卡的密码

RP = 0000 $rp_2p_1p_0b$

其中“$rp_2p_1p_0$”位指明待比较的密码

r = 0：写密码，

r = 1：读密码，

$p_2p_1p_0$：密码集数，

($rp_2p_1p_0$ = 0111：安全密码）

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2 ErrorCnt
90h	

其中：

SW1 = 90h。

SW2 (ErrorCnt) = 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。

9.3.4.5. INITIALIZE_AUTHENTICATION

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

Byte Address 存储卡的内存地址位置。

Q(0),Q(1)…Q(7) 主机随机数，8 个字节

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.4.6. VERIFY_AUTHENTICATION

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

Pseudo-APDU								
CLA	**INS**	**P1**	**P2**	**Lc**	**Q1(0)**	**Q1(1)**	**…**	**Q1(7)**
FFh	82h	00h	00h	08h				

其中：

Byte Address： 存储卡的内存地址位置。

Q1(0),Q1(1)…Q1(7)： 主机挑战数，8 个字节。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	**SW2**

其中：

如果没有错误，**SW1 SW2** = 90 00h。

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

9.3.5.1. 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。

注： *只有使用 SCardConnect( ) API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 SCardConnect( ) API 的详细说明参见 PC/SC 规范。*

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.5.2. READ_MEMORY_CARD

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

MSB Byte Address = 0000 00A_9A_8 b 是存储卡的内存地址位置。

LSB Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$ b 是存储卡的内存地址位置。

MEM_L 待从存储卡读取的数据的长度。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

BYTE 1	…	…	BYTE N	SW1	SW2

其中：

BYTE x 从存储卡中读取的数据。

如果没有错误，**SW1 SW2** = 90 00h。

9.3.5.3. READ_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（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

ERRCNT	DUMMY 1	DUMMY 2	SW1	SW2

其中：

ERRCNT 密码输入错误计数器的值。FFh 表示最后的验证正确。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。

DUMMY 从卡片读取的 2 个字节的虚拟数据。

如果没有错误，**SW1 SW2** = 90 00h。

9.3.5.4. READ_PROTECTION_BIT

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

MSB Byte Address = 0000 00A_9A_8b 是存储卡的内存地址位置。

LSB Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$b 是存储卡的内存地址位置。

MEM_L： 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（number of bits-1)/8）。

例如，要读取始于内存 0010h 的 8 个保护位，应当运行下面的 pseudo-APDU：

FF B1 00 10 01h

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

PROT 1	…	…	PROT L	SW1	SW2

其中：

PROT y 含有保护位的字节

如果没有错误，**SW1 SW2** = 90 00h。

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

9.3.5.5. WRITE_MEMORY_CARD

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

MSB Byte Address= 0000 00A_9A_8b 是存储卡的内存地址位置。

LSB Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$b 是存储卡的内存地址位置。

MEM_L： 待写入存储卡的数据的长度。

Byte x： 待写入存储卡的数据。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.5.6. WRITE_PROTECTION_MEMORY_CARD

命令指定的每一个字节均在卡片内部与存储在特定地址中的字节进行对比，若数据相符，则相应的保护位就会被不可逆转的设定为'0'。

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

MSB Byte Address = 0000 00A_9A_8b 是存储卡的内存地址位置

LSB Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$b 是存储卡的内存地址位置

MEM_L 待写入存储卡的数据的长度

Byte x 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。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.5.7. PRESENT_CODE_MEMORY_CARD（仅限 SLE 4428 和 SLE5528）

此命令用于向存储卡提交密码，使其能够对 SLE4428 和 SLE5528 进行写操作。执行以下步骤：

- 搜索密码输入错误计数器中值为‘1’的一个位，然后将该位写为‘0’
- 向卡片提交指定的密码
- 擦除密码错误计数器

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

CODE 2 个字节的密码（PIN）

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2 ErrorCnt
90h	

其中：

SW1 = 90h。

SW2 (ErrorCnt) = 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。

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

9.3.6.1. 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。

注： *只有使用 SCardConnect( ) API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 SCardConnect( ) API 的详细说明参见 PC/SC 规范。*

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.6.2. READ_MEMORY_CARD

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$b 是存储卡的内存地址位置。

MEM_L 待从存储卡读取的数据的长度。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

BYTE 1	…	…	BYTE N	SW1	SW2

其中：

BYTE x 从存储卡中读取的数据。

如果没有错误，**SW1 SW2** = 90 00h。

9.3.6.3. READ_PRESENTATION_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（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

ERRCNT	DUMMY 1	DUMMY 2	SW1	SW2

其中：

ERRCNT 密码输入错误计数器的值。07h 表示最后的验证正确。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。

DUMMY 从卡片读取的 3 个字节的虚拟数据。

如果没有错误，**SW1 SW2** = 90 00h。

9.3.6.4. READ_PROTECTION_BITS

此命令用于读取前 32 字节的保护位。

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

PROT 1	PROT 2	PROT3	PROT 4	SW1	SW2

其中：

PROT y 含有保护位的字节

如果没有错误，**SW1 SW2** = 90 00h。

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

9.3.6.5. WRITE_MEMORY_CARD

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$b 是存储卡的内存地址位置。

MEM_L 待写入存储卡的数据的长度。

Byte x 待写入存储卡的数据。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.6.6. WRITE_PROTECTION_MEMORY_CARD

命令指定的每一个字节均在卡片内部与存储在特定地址中的字节进行对比，若数据相符，则相应的保护位就会被不可逆转的设定为'0'。

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

Byte Address = 000A_4 $A_3A_2A_1A_0$b (00 to 1F)是存储卡的保护内存地址位置。

MEM_L 待写入存储卡的数据的长度。

Byte x 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。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.6.7. PRESENT_CODE_MEMORY_CARD（仅限 SLE4442 和 SLE5542）

此命令用于向存储卡提交密码，使其能够对 SLE4442 和 SLE5542 进行写操作。执行以下步骤：

1. 搜索密码输入错误计数器中值为‘1’的一个位，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（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

CODE 3 个字节的密码（PIN）。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2 ErrorCnt
90h	

其中：

SW1 = 90h。

SW2 (ErrorCnt) = 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。

9.3.6.8. 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（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

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

9.3.7.1. 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。

注： *只有使用 SCardConnect() API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 SCardConnect() API 的详细说明参见 PC/SC 规范。*

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.7.2. READ_MEMORY_CARD

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

Byte Address = 存储卡的内存地址位置。

MEM_L 待从存储卡读取的数据的长度。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

BYTE 1	…	…	BYTE N	SW1	SW2

其中：

BYTE x 从存储卡中读取的数据

如果没有错误，**SW1 SW2** = 90 00h。

9.3.7.3. 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（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

Byte Address = 存储卡的内存地址位置

MODE 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 待写入卡片的字节值

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.7.4. PRESENT_CODE_MEMORY_CARD

此命令用于向存储卡提交密码来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’的一个位，然后将该位写为‘0’。
2. 向卡片提交指定的密码。

密码提交后，ACR33U-A1 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（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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 输入错误计数器的字节地址。

CODE 3 个字节的密码（PIN）。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.7.5. AUTHENTICATE_MEMORY_CARD（仅限 SLE4436、SLE5536 和 SLE6636）

此命令用于从 SLE5536 卡或 SLE6636 卡读取卡片认证证书，ACR33U-A1 执行的操作如下：

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（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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： 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： 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。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2
61h	02h

其中：

如果没有错误，**SW1 SW2** = 61 02h，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（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

CERT		SW1	SW2

其中：

CERT 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。

如果没有错误，**SW1 SW2** = 90 00h。

9.3.8. 存储卡 – SLE4404

9.3.8.1. 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。

注：*只有使用 SCardConnect( ) API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 ScardConnect( ) API 的详细说明参见 PC/SC 规范。*

命令格式(*PC_to_RDR_XfrBlock* 中的 abData 数据域)

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

响应数据格式 (RDR_to_PC_DataBlock 中的 *abData* 数据域)

SW1	**SW2**

其中：

SW1 SW2 = 90 00h（未发生错误）。

9.3.8.2. READ_MEMORY_CARD

命令格式(*PC_to_RDR_XfrBlock* 中的 *abData* 数据域)

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

其中：

Byte Address = 存储卡的内存地址位置。

MEM_L 待从存储卡读取的数据的长度。

响应数据格式 *(RDR_to_PC_DataBlock* 中的 *abData* 数据域*)*

BYTE 1	…	…	BYTE N	SW1	SW2

其中：

BYTE x 从存储卡读取的数据。

SW1 SW2 = 90 00h（未发生错误）。

9.3.8.3. 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 *(PC_to_RDR_XfrBlock* 中的 *abData* 数据域*)*

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

其中：

Byte Address = 存储卡的内存地址位置。

MEM_L 待写入存储卡的数据的长度。

BYTE 待写入卡片的字节值。

响应数据格式 *(RDR_to_PC_DataBlock* 中的 *abData* 数据域*)*

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）。

9.3.8.4. 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”。

要擦除错误计数器或用户区的内容，请执行 **4.8.5 节**介绍的 VERIFY_USER_CODE 命令。

命令格式 *(PC_to_RDR_XfrBlock* 中的 *abData* 数据域*)*

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

其中：

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

响应数据格式 *(RDR_to_PC_DataBlock* 中的 *abData* 数据域*)*

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）。

9.3.8.5. VERIFY_USER_CODE

此命令用于向所插入的卡片提交用户密码（2 个字节）。用户密码提交正确即可访问卡片内存。

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 *(PC_to_RDR_XfrBlock* 中的 *abData* 数据域*)*

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 密码输入错误计数器的长度，单位为比特。

Byte Address 卡片中密钥的字节地址。

CODE 2 字节用户密码

响应数据格式 *(RDR_to_PC_DataBlock* 中的 *abData* 数据域*)*

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 = 9000h 后，再次读取用户错误计数器可检查 VERIFY_USER_CODE 是否正确。如果用户错误计数器被擦除并且等于 FFh，说明先前验证成功。*

9.3.8.6. 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 *(PC_to_RDR_XfrBlock* 中的 *abData* 数据域*)*

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 密码输入错误计数器的长度，单位为比特。

Byte Address 卡片中密钥的字节地址。

CODE 4 字节存储密码。

响应数据格式 *(RDR_to_PC_DataBlock* 中的 *abData* 数据域*)*

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，再次读取应用区可以检查 VERIFY_MEMORY_CODE 是否正确。如果应用区的全部数据都被擦除并且等于 FFh，证明先前的验证成功。*

9.3.9. 存储卡 – AT88SC101/AT88SC102/AT88SC1003

9.3.9.1. 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。

注：*只有使用 SCardConnect( ) API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 SCardConnect( ) API 的详细说明参见 PC/SC 规范。*

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.9.2. READ_MEMORY_CARD

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

Byte Address = 存储卡的内存地址位置。

MEM_L 待从存储卡读取的数据的长度。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

BYTE 1	…	…	BYTE N	SW1	SW2

其中：

BYTE x 从存储卡中读取的数据

如果没有错误，**SW1 SW2** = 90 00h。

9.3.9.3. WRITE_MEMORY_CARD

此命令用于向所插入卡片的特定地址写入数据。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（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

Byte Address = 存储卡的内存地址位置。

MEM_L 待写入存储卡的数据的长度。

BYTE 待写入卡片的字节值。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.9.4. 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 操作，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：

- **8.3.8.5 节**定义的 ERASE_APPLICATION_ZONE_WITH_ERASE 命令
- **8.3.8.6 节**定义的 ERASE_APPLICATION_ZONE_WITH_WRITE_AND_ERASE 命令
- **8.3.8.7 节**定义的 VERIFY_SECURITY_CODE 命令

命令格式（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

Byte Address = 待擦除的字的内存字节地址位置。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.3.9.5. 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（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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：　密码输入错误计数器的长度，单位为比特。值始终是 00h。

Byte Address：　卡片中应用区域密钥的字节地址。正确值请参阅下表：

Case	Byte Address	LEN
AT88SC101: Erase Application Zone with EC function disabled	96h	04h
AT88SC102: Erase Application Zone 1	56h	06h
AT88SC102: Erase Application Zone 2 with EC2 function disabled	9Ch	04h
AT88SC1003: Erase Application Zone 1	36h	06h
AT88SC1003: Erase Application Zone 2 with EC2 function disabled	5Ch	04h
AT88SC1003: Erase Application Zone 3	C0h	06h

其中：

MEM_L　　擦除密钥的长度。正确值请参阅下表。

CODE　　擦除密钥的 N 个字节。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

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"，证明先前的验证成功。*

9.3.9.6. 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 标识位没有被破坏并处于“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’的一个位，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（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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 密码输入错误计数器的长度，单位为比特。该值始终是 08h。

Byte Address 卡片中应用区域密钥的字节地址。

	Byte Address
AT88SC101	96h
AT88SC102	9Ch
AT88SC1003	5Ch

CODE 四个字节的擦除密钥。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

如果不再有重试的机会，则该状态字=63 00h

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”，证明先前的验证成功。*

9.3.9.7. 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’的一个位，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（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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 密码输入错误计数器的长度，单位为比特。

Byte Address 卡片中密钥的字节地址。

CODE 2 个字节的安全密码。

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

如果不再有重试的机会，则该状态字=63 00h。

注： *收到状态字 SW1SW2 = 90 00h 后，可重新读取安全密码尝试计数器（SCAC）来检查 VERIFY_USER_CODE 命令是否正确。如果 SCAC 已经被擦除并且等于“FFh”，证明先前的验证成功。*

9.3.9.8. 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（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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 or 01h

其中：

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

State of FUS Pin FUS pin 的状态，始终应该是 01h。

State of RST Pin RST pin 的状态，正确值请参阅下表。

		Fuse Bit Addr (High)	Fuse Bit Addr (Low)	State of RST Pin
AT88SC101	Manufacturer Fuse	05h	80h	01h
	EC_EN Fuse	05h	C9h	01h
	Issuer Fuse	05h	E0h	01h
AT88SC102	Manufacturer Fuse	05h	B0h	01h
	EC2EN Fuse	05h	F9h	01h
	Issuer Fuse	06h	10h	01h
AT88SC1003	Manufacturer Fuse	03h	F8h	00h
	EC2EN Fuse	03h	FCh	00h
	Issuer Fuse	03h	E0h	00h

响应数据格式（*RDR_to_PC_DataBlock* 中的 *abData* 数据域）

SW1	SW2

其中：

如果没有错误，**SW1 SW2** = 90 00h。

9.4. 通过 PC_to_RDR_XfrBlock 执行的其它命令

9.4.1. GET_READER_INFORMATION

此命令用于取回关于 ACR33U-A1 SmartDuo 的特定型号以及当前工作状态的信息，例如固件版本号、命令和响应的最大数据长度、支持的卡片类型、是否插入卡片并上电等。

注：*只有使用 SCardConnect( ) API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 ScardConnect( ) API 的详细说明参见 PC/SC 规范。*

命令格式*(PC_to_RDR_XfrBlock* 中的 *abData* 数据域*)*

Pseudo-APDU				
CLA	**INS**	**P1**	**P2**	**Lc**
FFh	09h	00h	00h	10h

响应数据格式 *(RDR_to_PC_DataBlock* 中的 *abData* 数据域*)*

FIRMWARE										MAX_C	MAX_R	C_TYPE		C_SEL	C_STAT

其中：

FIRMWARE 10 字节硬件版本数据。

MAX_C 最大命令数据字节数。

MAX_R 最多可以请求在响应中传输的数据字节数

C_TYPE ACR33U-A1 SmartDuo 支持的卡片类型。此数据字段是一个位图，每个位表示一种卡片类型。元素为“1”表示卡片支持相应的卡片类型，可以使用 SELECT_CARD_TYPE 命令来选择此种类型的卡片。各个位的分配如下：

Byte	1								2							
card type	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

关于各个位与卡片类型的对应关系，请参考下一节内容。

C_SEL 当前选定的卡片类型。00h 表示未选择卡片类型。

C_STAT 标识卡片是否已经插入读写器，以及卡片是否已经上电：

00h：未插入卡片

01h：卡片已插入，但未上电

03h：卡片已经上电

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。

字节	卡片类型
00h	自动选择 T=0 或 T=1 通信协议
01h	I2C 存储卡（1、2、4、8 和 16 kilobits)
02h	I2C 存储卡（32、64、128、256、512 和 1024 kilobits)
03h	Atmel AT88SC153 安全存储卡
04h	Atmel AT88SC1608 安全存储卡
05h	Infineon SLE 4418 和 SLE 4428
06h	Infineon SLE 4432 和 SLE 4442
07h	Infineon SLE 4406、SLE 4436 和 SLE 5536
08h	Infineon SLE 4404
09h	Atmel AT88SC101、AT88SC102 和 AT88SC1003
0Ch	MCU 卡（T=0 通信协议）
0Dh	MCU 卡（T=1 通信协议）

表 3：支持的卡片类型

附录B. CCID 响应错误代码

下表汇总了 ACR33U-A1 可能返回的错误代码：

错误代码	状态
FFh	SLOTERROR_CMD_ABORTED
FEh	SLOTERROR_ICC_MUTE
FDh	SLOTERROR_XFR_PARITY_ERROR
FCh	SLOTERROR_XFR_OVERRUN
FBh	SLOTERROR_HW_ERROR
F8h	SLOTERROR_BAD_ATR_TS
F7h	SLOTERROR_BAD_ATR_TCK
F6h	SLOTERROR_ICC_PROTOCOL_NOT_SUPPORTED
F5h	SLOTERROR_ICC_CLASS_NOT_SUPPORTED
F4h	SLOTERROR_PROCEDURE_BYTE_CONFLICE
F3h	SLOTERROR_DEACTIVATED_PROTOCOL
F2h	SLOTERROR_BUSY_WITH_AUTO_SEQUENCE
E0h	SLOTERROR_CMD_SLOT_BUSY

表 4： 可能返回的错误代码