Advanced Card Systems Ltd.
Card & Reader Technologies

ACR40T

USB SIM 尺寸智能卡读写器

参考手册 V1.04

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

目录

图目录

表目录

1.0. 简介

ACR40T USB SIM 尺寸智能卡读写器是计算机与智能卡之间的通信媒介。由于不同类型的智能卡采用不同的通信协议和命令，智能卡和计算机之间难以直接进行通信。ACR40T USB SIM 尺寸智能卡读写器可以为多种智能卡提供标准化接口，使得软件开发人员摆脱复杂的智能卡操作环节。通过 ACR40T 处理智能卡的具体细节，程序员可以专注实现智能卡系统的功能，无需关注底层技术详情。

1.1. 参考文件

下列相关文件可以在 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：智能卡 CCID 规范（集成电路卡接口设备），1.1 版，2005 年 4 月 22 日

下列文件可以在 www.ansi.org 订阅。

- 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	说明
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 CCID 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[1]（Reserved for future use）
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] *除非另有说明，否则必须设置为零。*

2.0. 支持的智能卡

2.1. MCU 卡

ACR40T 是一款符合 PC/SC 标准的智能卡读写器。它支持 ISO 7816 A 类、B 类和 C 类（5V、3V 和 1.8V）智能卡，还可以读写所有符合 T=0 或 T=1 协议的 MCU 卡。

若卡片的 ATR 指定了专用的操作模式（TA2 存在；TA2 中的 b5 位必须为 0），但 ACR40T 不支持该特定模式，则 ACR40T 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，读写器会拒绝读写该卡。

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 ACR40T 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）。

关于上述参数的含义，请参照 ISO 7816-3 标准。

2.2. 存储卡

ACR40T 支持多种类型的存储卡，例如：

- 符合 I2C 总线协议（空白存储卡）、且每页最大容量为 128 字节的存储卡，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 的 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

3.0. 系统框图

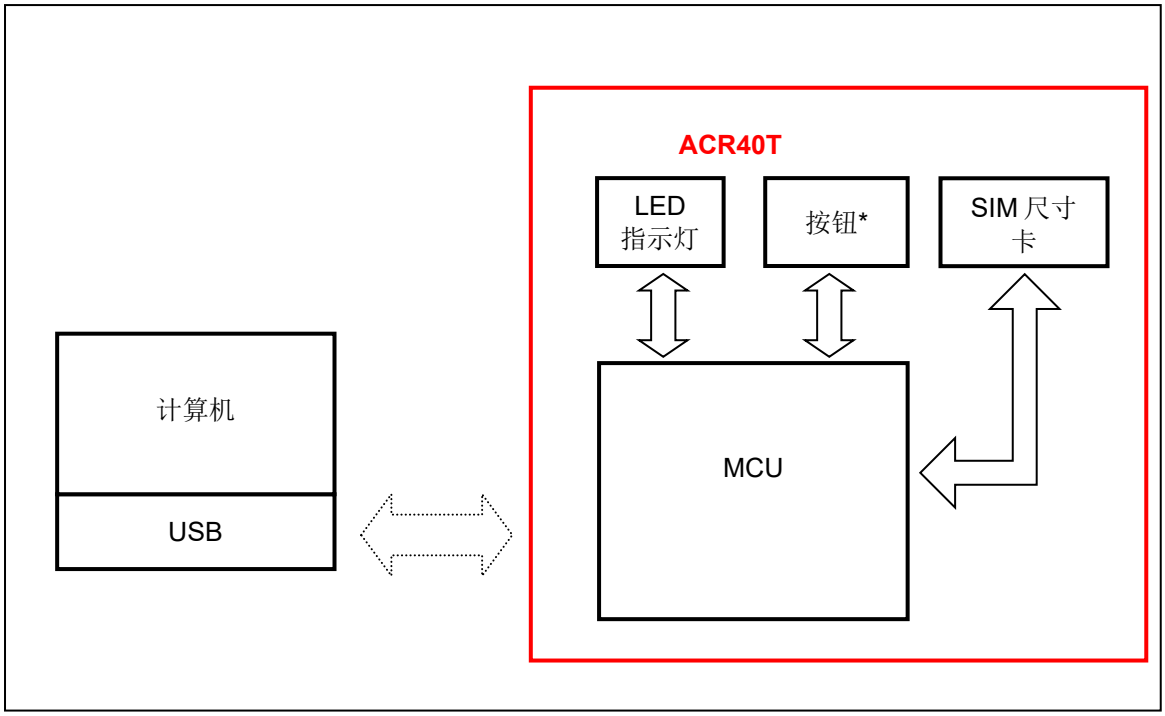

图1： ACR40T 结构

*按钮仅适用于 ACR40T-A6/7 型号。

4.0. USB 接口

4.1. 通信参数

ACR40T 按照 USB 2.0 规范通过 USB 端口连接计算机。它支持 USB 全速模式，速率为 12Mbps。

引脚	信号	功能
1	V_{BUS}	为读写器提供+5 V 的电源
2	D-	ACR40T 和计算机之间以差分信号传输数据。
3	D+	ACR40T 和计算机之间以差分信号传输数据。
4	GND	参考电压等级

表2：USB 接口配线

4.2. 端点

ACR40T 通过下列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 ACR40T 的命令 （数据包的大小为 64 字节）
批量输入（Bulk IN）	用于从 ACR40T 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 ACR40T 发送至主机的卡片状态消息 （数据包的大小为 8 字节）

5.0. 用户接口

5.1. LED 状态指示灯

ACR40T 有三种 LED 闪烁模式，用于显示各种工作状态，其中：

- **绿色 LED -** USB 模式下卡片和读写器的状态

颜色	LED 操作	状态
绿色	缓慢闪烁 （2 秒/闪烁）	无卡片操作，读写器正在等待 PC 端指令
	快速闪烁	读写器正在与 PC 进行数据传输
	长亮	卡片已连接并上电

表3：LED 状态指示灯

5.2. 可配置按钮（仅限 ACR40T-A6/7）

ACR40T 带有一个按钮，可以在多种模式下操作，其中：

模式	说明
0	发送 Escape 命令读取按钮的状态
1（默认模式）	模拟卡片移出事件
2	禁用按钮特性

表4：按钮的操作模式

5.2.1. 按钮工作流程

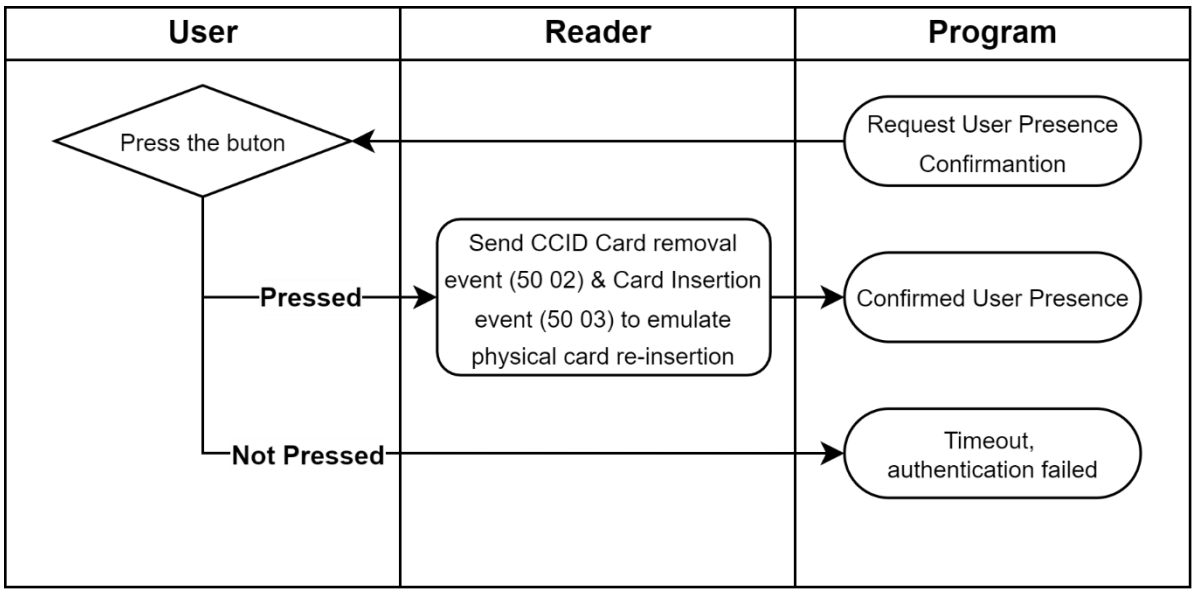

图2：ACR40T 按钮工作流程图

6.0. 智能卡接口

ACR40T 与插入的智能卡之间的接口符合 ISO 7816-3 标准，并进行了某些限制或提升来增强 ACR40T 的实用功能。

6.1. 智能卡电源 VCC（C1）

插入的智能卡电流消耗不得大于 50 mA。

6.2. 编程电压 VPP（C6）

根据 ISO7816-3 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，ACR40T 的触点 C6（VPP）已被实现为普通的控制信号。

6.3. 卡片类型选择

每次激活插入的卡片前，处于控制地位的计算机必须向 ACR40T 发送正确的命令来选择卡片类型。这些卡片包括存储卡和 MCU 卡。

对于 MCU 卡来说，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 MCU 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。

6.4. 微控制器卡接口

基于微控制器的智能卡只使用触点 C1（VCC）、C2（RST)、C3（CLK）、C5（GND）和 C7（I/O）。时钟信号（C3）的频率为 4.8 MHz。

7.0. USB 通信协议

ACR40T 通过 USB 与主机端（host）建立接口。现在的行业内规范—CCID 标准，已经为 USB 芯片-智能卡接口设备定义了与此相关的协议。CCID 涵盖了操作智能卡所需的全部协议。

ACR40T 的 USB 端点的配置和使用应当符合 CCID 标准（版本 1.0）第 3 部分的规定。

概述如下：

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**经由 BULK-OUT 端点发送。发送至 ACR40T 的每个命令都有一个相关的最终响应，一些命令也可以有过程响应。
4. **CCID 响应**经由 BULK-IN 端点发送。所有发送至 ACR40T 的命令都必须同步发送（例如：对于 ACR40T 来说，*bMaxCCIDBusySlots* 等同于 01h）。

ACR40T 支持的 CCID 特性在其类别描述符中进行了说明：

偏移	数据域	大小	值	说明
0	*bLength*	1	-	描述符的字节数
1	*bDescriptorType*	1	-	CCID 功能描述符的类别
2	*bcdCCID*	2	-	二进制编码的十进制 CCID 规范版本号
4	*bMaxSlotIndex*	1	-	ACR40T 有一个卡槽
5	*bVoltageSupport*	1	-	ACR40T 可支持 1.8V、3V 和 5V 的槽位电压。
6	*dwProtocols*	4	-	ACR40T 支持 T=0 和 T=1 协议
10	*dwDefaultClock*	4	-	默认 ICC 时钟频率为 4.8 MHz
14	*dwMaximumClock*	4	-	ICC 支持的最大时钟频率为 4.8 MHz
18	*bNumClockSupported*	1	-	不支持手动设置时钟频率
19	*dwDataRate*	4	-	默认 ICC I/O 数据传输速率为 12903 bps
23	*dwMaxDataRate*	4	-	支持的最大 ICC I/O 数据传输速率为 600 Kbps
27	*bNumDataRatesSupported*	1	-	不支持手动设置数据传输速率
28	*dwMaxIFSD*	4	-	T=1 协议下，ACR40T 支持的最大 IFSD 为 247。
32	*dwSynchProtocols*	4	-	ACR40T 不支持同步卡
36	*dwMechanical*	4	-	ACR40T 不支持特殊机制特性
40	*dwFeatures*	4	-	ACR40T 支持以下特性： • 根据参数自动改变 ICC 时钟频率 • 根据频率和 FI、DI 参数自动改变波特率 • 与 ACR40T 进行 TPDU 级交换
44	*dwMaxCCIDMessageLength*	4	-	ACR40T 可接受的最大信息长度为 271 字节。

偏移	数据域	大小	值	说明
48	*bClassGetResponse*	1	-	对 TPDU 级别的交换没有影响
49	*bClassEnvelope*	1	-	对 TPDU 级别的交换没有影响
50	*wLCDLayout*	2	-	没有 LCD
52	*bPINSupport*	1	-	支持 PIN 校验
53	*bMaxCCIDBusySlots*	1	-	同一时间只能有 1 个槽位处于工作状态

7.1. CCID Bulk-OUT 消息

7.1.1. 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	说明
0	*bMessageType*	1	62h	-
1	*dwLength*	4	00000000h	此消息的额外字节的大小
5	*bSlot*	1	00h	标识命令的卡槽号 00h：ICC
6	*bSeq*	1	00-FFh	命令的序号
7	*bPowerSelect*	1	00h-03h	ICC 上的电压值： 00h = 自动电压选择 01h = 5 V 02h = 3 V 03h = 1.8V
8	*abRFU*	2	0000h	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。

7.1.2. PC_to_RDR_IccPowerOff

此命令用于取消激活卡槽。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	63h	-
1	*dwLength*	4	00000000h	此消息的额外字节的大小
5	*bSlot*	1	00h	标识命令的卡槽号 00h：ICC
6	*bSeq*	1	00-FFh	命令的序号
7	*abRFU*	3	000000h	保留为将来使用

此消息的响应是 *RDR_to_PC_SlotStatus* 消息。

7.1.3. PC_to_RDR_GetSlotStatus

此命令用于获取卡槽的当前状态。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	65h	-
1	*dwLength*	4	00000000h	此消息的额外字节的大小
5	*bSlot*	1	00h	标识命令的卡槽号 00h：ICC
6	*bSeq*	1	00-FFh	命令的序号
7	*abRFU*	3	000000h	保留为将来使用

此消息的响应是 *RDR_to_PC_SlotStatus* 消息。

7.1.4. PC_to_RDR_XfrBlock

此命令用于向 ICC 传输数据块。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	6Fh	-
1	*dwLength*	4	00000000-0000FFFFh	此消息的 *abData* 数据域的大小。数据域以小端格式存储。
5	*bSlot*	1	00h	标识命令的卡槽号。 00h：ICC
6	*bSeq*	1	00-FFh	命令的序号。
7	*bBWI*	1	00-FFh	用于延长当前传输的 CCID 块超时等待时间。“该数值乘以块等待时间”的时间段过去后，CCID 将设置该块超时。
8	*wLevelParameter*	2	-	数据域以小端格式存储。 短 APDU 级, RFU = 0000h 扩展 APDU 级： 0000h – 命令 APDU 在此命令中开始和结束。 0001h – 命令 APDU 在此命令中开始，并在下一个 PC_to_RDR_XfrBlock 中继续。 0002h – abData 字段继续传递命令 APDU 并结束该 APDU 命令。 0003h – abData 字段继续传递命令 APDU，后面还跟随另外一个数据块。 0010h – 空的 abData 字段，下一个 RDR_to_PC_DataBlock 会继续传递响应 APDU

偏移	数据域	大小	值	说明
10	*abData*	字节型数组	-	发送给 CCID 的数据块。

此消息的响应是 *RDR_to_PC_DataBlock* 消息。

7.1.5. PC_to_RDR_Escape

此命令用于访问扩展功能。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	6Bh	-
1	*dwLength*	4	00000000-000000FFh	此消息的 *abData* 数据域的大小。数据域以小端格式存储。
5	*bSlot*	1	00h	标识命令的卡槽号。 00h：ICC
6	*bSeq*	1	00-FFh	命令的序号
7	*abRFU*	3	000000h	保留为将来使用
10	*abData*	字节型数组	-	发送至 CCID 的数据块

此消息的响应是 *RDR_to_PC_Escape* 消息。

7.1.6. PC_to_RDR_GetParameters

此命令用于获取卡槽的参数。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	6Ch	-
1	*DwLength*	4	00000000h	此消息的额外字节的大小
5	*BSlot*	1	00h	标识命令的卡槽号。 00h：ICC
6	*BSeq*	1	00-FFh	命令的序号
7	*AbRFU*	3	000000h	保留为将来使用

此消息的响应是 *RDR_to_PC_Parameters* 消息。

7.1.7. PC_to_RDR_ResetParameters

此命令用于将卡槽参数重置为默认值。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	6Dh	-
1	*DwLength*	4	00000000h	此消息的额外字节的大小

偏移	数据域	大小	值	说明
5	*BSlot*	1	00h	标识命令的卡槽号。 00h：ICC
6	*BSeq*	1	00-FFh	命令的序号
7	*AbRFU*	3	000000h	保留为将来使用

此消息的响应是 *RDR_to_PC_Parameters* 消息。

7.1.8. PC_to_RDR_SetParameters

此命令用于设置卡槽的参数。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	61h	-
1	*dwLength*	4	00000005h or 00000007h	此消息的额外字节的大小。数据域以小端格式存储
5	*bSlot*	1	00h	标识命令的卡槽号 00h：ICC
6	*bSeq*	1	00-FFh	命令的序号
7	*bProtocolNum*	1	00-01h	指定采用的协议数据结构： 00h：T=0 协议的结构 01h：T=1 协议的结构 以下值保留为将来使用： 80h：2 线协议结构 81h：3 线协议结构 82h：I2C 协议结构
8	*abRFU*	2	0000h	保留为将来使用
10	*abProtocolDataStructure*	字节型数组	-	协议数据结构

T=0 的协议数据结构（*dwLength*=00000005h）

偏移	数据域	大小	值	说明
10	*bmFindexDindex*	1		B7-4 – FI – ISO/IEC 7816-3:1997 中表 7 的索引，选择一个时钟频率转换因子。 B3-0 – DI – ISO/IEC 7816-3:1997 中表 8 的索引，选择一个波特率转换因子。

偏移	数据域	大小	值	说明
11	*bmTCCKST0*	1	00h, 02h	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。在通常的保护时间（12 etu）基础上增加 0-254 个 etu。FFh 与 00h 相同。
13	*bWaitingIntegerT0*	1	00-FFh	T=0 时 WI 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 的协议数据结构(dwLength=00000007h)

偏移	数据域	大小	值	说明
10	*bmFindexDindex*	1		B7-4 – FI – ISO/IEC 7816-3:1997 中表 7 的索引，选择一个时钟频率转换因子。 B3-0 – DI – ISO/IEC 7816-3:1997 中表 8 的索引，选择一个波特率转换因子。
11	*BmTCCKST1*	1	10h, 11h, 12h, 13h	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 0-254 个 etu）。若值为 FFh，则保护时间减少 1 个 etu。
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-FEh	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。

7.2. CCID Bulk-IN 消息

7.2.1. RDR_to_PC_DataBlock

此消息由 ACR40T 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	说明
0	*bMessageType*	1	80h	表示 CCID 正在发送一个数据块。
1	*dwLength*	4	00000000-000001E7h	此消息的额外字节的大小。 数据域以小端格式存储
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	-	插槽状态寄存器，定义见**附录 A**
8	*bError*	1	-	插槽错误寄存器，定义见**附录 A**
9	*bChainParameter*	1	-	短 APDU 级，RFU = 00h 扩展 APDU 级： 00h – 响应 APDU 在此命令中开始和结束。 01h – 响应 APDU 在此命令中开始，并会继续。 02h – 此 abData 字段继续传递响应 APDU 并结束该响应 APDU。 03h – 此 abData 字段继续传递响应 APDU，后面跟随另外一个数据块。 10h – 空的 abData 字段，下一个 PC_to_RDR_XfrBlock 命令会继续传递命令 APDU
10	*abData*	字节型数组	-	本字段包含由 CCID 返还的数据

7.2.2. RDR_to_PC_SlotStatus

此消息由 ACR40T 发出，是对 *PC_to_RDR_IccPowerOff* 和 *PC_to_RDR_GetSlotStatus* 消息的响应。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	81h	-
1	*dwLength*	4	00000000h	此消息的额外字节的大小
5	*bSlot*	1	-	与 Bulk-OUT 消息中的值相同
6	*bSeq*	1	-	与 Bulk-OUT 消息中的值相同
7	*bStatus*	1	-	插槽状态寄存器，定义见**附录 A**
8	*bError*	1	-	插槽错误寄存器，定义见**附录 A**

偏移	数据域	大小	值	说明
9	*bClockStatus*	1	00-03h	值： 00h = 时钟运行 01h = 时钟停于低状态 02h = 时钟停于高状态 03h = 时钟停止于未知状态 所有其他值保留为将来使用

7.2.3. RDR_to_PC_Parameters

此消息由 ACR40T 发出，是对 *PC_to_RDR_GetParameters*、*PC_to_RDR_ResetParameters* 和 *PC_to_RDR_SetParameters* 消息的响应。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	82h	-
1	*dwLength*	4	00000005h or 00000007h	此消息的 abProtocolDataStructure 数据域的大小。数据域以小端格式存储。
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	-	插槽状态寄存器，定义见**附录 A**
8	*bError*	1	-	插槽错误寄存器，定义见**附录 A**
9	*bProtocolNum*	1	00-01h	指定采用的协议数据结构： 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*	字节型数组	-	协议数据结构，如 **7.1.8** 所述。

7.2.4. RDR_to_PC_Escape

此消息由 ACR40T 发出，是对 *PC_to_RDR_Escape* 消息的响应。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	83h	-
1	*dwLength*	4	00000000- 000000FFh	此消息的 *abData* 数据域的大小。数据域以小端格式存储。
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

偏移	数据域	大小	值	说明
7	*bStatus*	1	-	插槽状态寄存器，定义见**附录 A**
8	*bError*	1	-	插槽错误寄存器，定义见**附录 A**
9	*bChainParameter*	1	00h	RFU
10	*abData*	字节型数组	-	从 CCID 发送的数据

8.0. 主机编程 API

8.1. 外设控制

读写器的外设控制命令在 USB 模式下通过直接（Escape）命令（0x6B）以 PC_to_RDR_Escape 来实现。

8.1.1. 获取固件版本（Get Firmware Version）

此命令用于获取读写器的固件版本。

Get Firmware Version 命令的结构（5 字节）

命令	CLA	INS	P1	P2	Lc
Get Firmware Version	E0h	00h	00h	19h	00h

Get Firmware Version 的响应结构（5 字节 + 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

例如：响应 = E1h 00h 00h 00h 0Bh 41h 43h 52h 34h 30h 54h 2Dh 50h 31h 30h 30h

固件版本（HEX） = 41h 43h 52h 34h 30h 54h 2Dh 50h 31h 30h 30h

固件版本（ASCII） = “ACR40T-P100 ”

8.1.2. 获取序列号（Get Serial Number）

此命令用于获取序列号。

Get Serial Number 命令的结构（5 个字节）

命令	CLA	INS	P1	P2	Le
Get Serial Number	E0h	00h	00h	33h	00h

Get Serial Number 的响应结构（5 字节 + 序列号的长度）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	序列号的长度	序列号

8.1.3. 获取卡片电压选择顺序（Get Card Voltage Selection Sequence）

Get Card Voltage Selection Sequence 命令用于获取卡片电压的上电顺序。

Get Card Voltage Selection Sequence 命令的格式（5 字节）

命令	CLA	INS	P1	P2	Lc
Get Card Voltage Selection Sequence	E0h	00h	00h	0Bh	00h

Get Card Voltage Selection 的响应结构（5 字节 + 获取卡片电压选择)

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	卡片电压上电顺序

其中

卡片电压上电顺序　　卡片电压的上电顺序（1 字节）

00h = Class C => Class B => Class A

01h = 仅 Class A

02h = 仅 Class B

03h = 仅 Class C

04h = Class A => Class B => Class C

8.1.4. 设置卡片电压选择（Set Card Voltage Selection）

Set Card Voltage Selection Sequence 命令用于设置卡片电压的上电顺序。

Set Card Voltage Selection 命令的格式（5 字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Set Card Voltage Selection	E0h	00h	00h	0Bh	01h	卡片电压上电顺序

Set Card Voltage Selection 的响应结构（5 字节 + 获取卡片电压选择)

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	卡片电压上电顺序

其中

卡片电压上电顺序　　卡片电压的上电顺序（1 字节）

00h = Class C => Class B => Class A

01h = 仅 Class A

02h = 仅 Class B

03h = 仅 Class C

04h = Class A => Class B => Class C

8.1.5. 写入客户数据（Write Customer Data）

此命令用于写入用户自定义数据（客户数据最大长度：64 字节）

注：*关于此命令的详细信息，请发送电邮至 info@acs.com.hk 或联系龙杰智能卡有限公司的销售代表。*

8.1.6. 读取客户数据（Read Customer Data）

此命令用于读取客户数据：

注：*关于此命令的详细信息，请发送电邮至 info@acs.com.hk 或联系龙杰智能卡有限公司的销售代表。*

8.1.7. 设置 USB 描述符中的 S/N（Set S/N in USB Descriptor）

此命令用于设置 USB 描述符中的 S/N。

注：*关于此命令的详细信息，请发送电邮至 info@acs.com.hk 或联系龙杰智能卡有限公司的销售代表。*

8.1.8. 修改客户 PIN 码（Change Customer PIN）

此命令用于修改客户 PIN 码和只读 PIN 码。

注：*关于此命令的详细信息，请发送电邮至 info@acs.com.hk 或联系龙杰智能卡有限公司的销售代表。*

8.1.9. 读取按钮模式（Read Button Mode）

此命令用于读取按钮的操作模式。

Read Button Mode（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Read Button Mode	E0h	00h	00h	E3h	01h	FFh

Read Button Mode 的响应结构（5 字节 + 按钮模式 + 2 字节响应状态码）

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	按钮模式	响应状态码

其中

按钮模式 1 个字节

00h = 通过发送 Escape 命令读取按钮的状态

01h = 按动按钮移除卡片（卡片移除事件）

02h = 禁用按钮功能

默认值 ---- 01h

响应状态码

结果	SW1 SW2	含义
成功	90 00h	按钮模式变更成功
错误	67 00h	按钮模式变更失败

8.1.10. 设置按钮模式（Set Button Mode）

此命令用于配置按钮的操作模式。

Set Button Mode（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Set Button Mode	E0h	00h	00h	E3h	01h	按钮模式

Set Button Mode 的响应结构（5 个字节 + 按钮模式 + 2 字节响应状态码）

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	按钮模式	响应状态码

其中

按钮模式 1 个字节

00h = 通过发送 Escape 命令读取按钮的状态

01h = 按动按钮移除卡片（卡片移除事件）

02h = 禁用按钮功能

默认值 ---- 01h

响应状态码

结果	SW1 SW2	含义
成功	90 00h	按钮模式更改成功
错误	67 00h	按钮模式更改失败

8.1.11. 获取按钮状态（Get Status of Button）

当按钮模式设置为 00h 时，此命令用于读取按钮的状态。

Get the status of button（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read button status	E0h	00h	00h	E3h	02h

Read button status 的响应格式（5 字节 + 按钮状态 + 2 字节响应状态码）

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	按钮状态	00h	响应状态码

其中：

按钮状态 1 个字节

00h = 开启（未按下）

01h = 闭合（按下）

返回状态码

结果	SW1 SW2	含义
成功	90 00h	命令成功执行完毕
错误	67 00h	命令执行错误

8.2. 存储卡命令集

8.2.1. 存储卡 – 1、2、4、8 和 16 Kb 的 I2C 卡

8.2.1.1. SELECT_CARD_TYPE

此命令对插入读写器的已选定卡片进行上/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**	**卡片类型**
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：

SW1 SW2 = 90 00h（未发生错误）

8.2.1.2. SELECT_PAGE_SIZE

此命令会选择读取智能卡的页面大小。默认值是 8 字节页写。卡片移出或读写器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**	**页面大小**
FFh	01h	00h	00h	01h	

其中：

页面大小 = 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：

SW1 SW2 = 90 00h（未发生错误）

8.2.1.3. READ_MEMORY_CARD

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

Pseudo-APDU				
CLA	INS	字节地址		MEM_L
		MSB	LSB	
FFh	B0h			

其中：

字节地址 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）

字节 1	…	…	字节 N	SW1	SW2

其中：

字节 x 从存储卡读取的数据。

SW1 SW2 = 90 00h（未发生错误）

8.2.1.4. WRITE_MEMORY_CARD

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

Pseudo-APDU								
CLA	INS	字节地址		MEM_L	字节 1	....	....	字节 n
		MSB	LSB					
FFh	D0h							

其中：

字节地址 存储卡的内存地址位置

MEM_L 要写入存储卡的数据的长度

字节 x 要写入存储卡的数据

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

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.2.2. 存储卡 – 32、64、128、256、512 和 1024 Kb 的 I2C 卡

8.2.2.1. SELECT_CARD_TYPE

此命令用于对插入读写器的已选定卡片进行上/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	卡片类型
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：

SW1 SW2 = 90 00h（未发生错误）

8.2.2.2. SELECT_PAGE_SIZE

此命令会选择读取智能卡的页面大小。默认值是 8 字节页写。卡片移出或读写器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	页面大小
FFh	01h	00h	00h	01h	

其中：

数据 待发送给卡片的 TPDU

页面大小 = 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：

SW1 SW2 = 90 00h（未发生错误）

8.2.2.3. READ_MEMORY_CARD

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

Pseudo-APDU				
CLA	INS	字节地址		MEM_L
		MSB	LSB	
FFh				

其中：

INS = B0h：32 kilobit、64 kilobit、128 kilobit、256 kilobit 和 512 kilobit 的 IIC 卡

= 1011 000*b：1024 kilobit IIC 卡，

其中 * 表示 17 位地址的 MSB。

字节地址 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）

字节 1	…	…	字节 N	SW1	SW2

其中：

字节 x 从存储卡中读取的数据

SW1 SW2 = 90 00h（未发生错误）

8.2.2.4. WRITE_MEMORY_CARD

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

Pseudo-APDU								
CLA	INS	字节地址		MEM_L	字节 1	….	….	字节 n
		MSB	LSB					
FFh								

其中：

INS = D0h：32 kilobit、64 kilobit、128 kilobit、256 kilobit 和 512 kilobit 的 IIC 卡
= 1101 000*b：1024 kilobit IIC 卡，
其中 * 表示 17 位地址的 MSB。

字节地址 存储卡的内存地址位置

MEM_L 要写入存储卡的数据的长度

字节 x 要写入存储卡的数据

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

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.2.3. 存储卡 – ATMEL AT88SC153

8.2.3.1. SELECT_CARD_TYPE

此命令用于对插入读写器的已选定卡片进行上/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**	**卡片类型**
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：

SW1 SW2 = 90 00h（未发生错误）

8.2.3.2. READ_MEMORY_CARD

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

Pseudo-APDU				
CLA	**INS**	**P1**	**字节地址**	**MEM_L**

Pseudo-APDU				
FFh		00h		

其中：

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

字节地址 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）

字节 1	…	…	字节 N	SW1	SW2

其中：

字节 x 从存储卡中读取的数据

SW1 SW2 = 90 00h（未发生错误）

8.2.3.3. WRITE_MEMORY_CARD

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

Pseudo-APDU								
CLA	INS	P1	字节地址	MEM_L	字节 1	....	....	字节 n
FFh		00h						

其中：

INS = D0h：写入 00b 区
= D1h：写入 01b 区
= D2h：写入 10b 区
= D3h：写入 11b 区
= D4h：写入标识位

字节地址 存储卡的内存地址位置

MEM_L 要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：

SW1 SW2 = 90 00h（未发生错误）

8.2.3.4. 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) 待发送给存储卡的密码

P2 = 0000 00rp$_b$

其中的“rp”位指明待比较的密码

r = 0：“写”密码

r = 1：“读”密码

p：密码集编号

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（或超过最大重试次数）。其它值表示当前验证失败。

8.2.3.5. 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：

SW1 SW2 = 90 00h（未发生错误）

8.2.3.6. 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：

SW1 SW2 = 90 00h（未发生错误）

8.2.4. 存储卡 – ATMEL AT88C1608

8.2.4.1. SELECT_CARD_TYPE

此命令用于对插入读写器的已选定卡片进行上/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	卡片类型
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：

SW1 SW2 = 90 00h（未发生错误）

8.2.4.2. READ_MEMORY_CARD

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

Pseudo-APDU				
CLA	INS	区域地址	字节地址	MEM_L

Pseudo-APDU				
FFh				

其中：

INS = B0h：读取用户区。
= B1h：读取配置区或读取标识位

区域地址 = 0000 $0A_{10}A_9A_8$b，其中 A_{10} 是区域地址的 MSB
= 读标识位时无关

字节地址 = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$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）

字节 1	…	…	字节 N	SW1	SW2

其中：

字节 x 从存储卡中读取的数据

SW1 SW2 = 90 00h（未发生错误）

8.2.4.3. WRITE_MEMORY_CARD

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

Pseudo-APDU								
CLA	INS	区域地址	字节地址	MEM_L	字节 1	…	…	字节 n
FFh								

其中：

INS = D0h：写用户区。
= D1h：写配置区或写标识位

区域地址 = 0000 $0A_{10}A_9A_8$b，其中 A_{10} 是区域地址的 MSB
= 写标识位时无关

字节地址 = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$b 是存储卡的内存地址位置
= 1000 0000b：写标识位

MEM_L 要写入存储卡的数据的长度

字节 x 要写入存储卡的数据

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

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.2.4.4. VERIFY_PASSWORD

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

Pseudo-APDU								
CLA	**INS**	**P1**	**P2**	**Lc**	**数据**			
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

$p_2p_1p_0$：密码集编号。

($rp_2p_1p_0$ = 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（或超过最大重试次数）。其它值表示当前验证失败。

8.2.4.5. 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：

字节地址 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：

SW1 SW2 = 90 00h（未发生错误）

8.2.4.6. 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：

字节地址 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：

SW1 SW2 = 90 00h（未发生错误）

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

8.2.5.1. SELECT_CARD_TYPE

此命令用于对插入读写器的已选定卡片进行上/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**	**卡片类型**
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：

SW1 SW2 = 90 00h（未发生错误）

8.2.5.2. READ_MEMORY_CARD

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

Pseudo-APDU				
CLA	INS	字节地址		MEM_L
		MSB	LSB	
FFh	B0h			

其中：

MSB 字节地址 = 0000 00A_9A_8b 是存储卡的内存地址位置

LSB 字节地址 = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$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）

字节 1	…	…	字节 N	SW1	SW2

其中：

字节 x 从存储卡读取的数据。

SW1, SW2 = 90 00h（未发生错误）

8.2.5.3. READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (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 错误计数器。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。

DUMMY 从卡片读取的 2 字节虚拟数据

SW1 SW2 = 90 00h（未发生错误）

8.2.5.4. READ_PROTECTION_BIT

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

Pseudo-APDU				
CLA	INS	字节地址		MEM_L
		MSB	LSB	
FFh	B2h			

其中：

MSB 字节地址 = 0000 00A_9A_8b 是存储卡的内存地址位置

LSB 字节地址 = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$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((位数 - 1)/8)

例如，要读取始于内存 0010h 的 8 个保护位，应当发送下面的私有 APDU：

FF B2 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

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 是响应数据中字节 x 的保护位

‘0’ 字节被写保护

‘1’ 字节可以被写入

8.2.5.5. WRITE_MEMORY_CARD

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

Pseudo-APDU								
CLA	INS	字节地址		MEM_L	字节 1	....	....	字节 N
		MSB	LSB					
FFh	D0h							

其中：

MSB 字节地址 = 0000 00A_9A_8b 是存储卡的内存地址位置

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

MEM_L 要写入存储卡的数据的长度

字节 x 要写入存储卡的数据

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

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.2.5.6. WRITE_PROTECTION_MEMORY_CARD

命令指定字节在卡片中与存储在特定地址位置的字节逐个对比。若数据相符，相应的保护位会不可逆地设定为“0”。

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

Pseudo-APDU								
CLA	INS	字节地址		MEM_L	字节 1	....	....	字节 N
		MSB	LSB					
FFh	D1h							

其中：

MSB 字节地址 = 0000 00A_9A_8b 是存储卡的内存地址位置

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

MEM_L 要写入存储卡的数据的长度

字节 x 要与卡片内始于**字节地址**（*Byte Address*）的数据做比较的字节值。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：

SW1 SW2 = 90 00h（未发生错误）

8.2.5.7. PRESENT_CODE_MEMORY_CARD (SLE4428 和 SLE5528)

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

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（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

Pseudo-APDU						
CLA	INS	P1	P2	MEM_L	密码	
					字节 1	字节 2
FFh	20h	00h	00h	02h		

其中：

密码 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 表示校验成功。00h 表示密码被锁定（或超过最大重试次数）。其它值表示当前验证失败。

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

8.2.6.1. SELECT_CARD_TYPE

此命令用于对插入读写器的已选定卡片进行上/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后才可以使用此命令。关于 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**	**卡片类型**
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：

SW1 SW2 = 90 00h（未发生错误）

8.2.6.2. READ_MEMORY_CARD

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

Pseudo-APDU				
CLA	**INS**	**P1**	**字节地址**	**MEM_L**
FFh	B0h	00h		

其中：

字节地址 = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$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）

字节 1	…	…	字节 N	SW1	SW2

其中：

字节 x 从存储卡读取的数据。

SW1, SW2 = 90 00h（未发生错误）

8.2.6.3. READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (SLE 4442 和 SLE 5542)

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	DUMMY 3	SW1	SW2

其中：

ERRCNT 错误计数器。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）。其它值表示最后一次验证失败。

DUMMY 从卡片读取的 3 个字节的虚拟数据

SW1 SW2 = 90 00h（未发生错误）

8.2.6.4. READ_PROTECTION_BITS

此命令用于读取前 32 个字节的保护位。

命令格式（*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	PROT 3	PROT 4	SW1	SW2

其中：

PROT y 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 是响应数据中字节 x 的保护位

‘0’ 字节被写保护

‘1’ 字节可以被写入

8.2.6.5. WRITE_MEMORY_CARD

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

Pseudo-APDU								
CLA	**INS**	**P1**	**字节地址**	**MEM_L**	**字节 1**	**....**	**....**	**字节 N**
FFh	D0h	00h						

其中：

字节地址 = $A_7A_6A_5A_4\ A_3A_2A_1A_0$b 是存储卡的内存地址位置

MEM_L 要写入存储卡的数据的长度

字节 x 要写入存储卡的数据

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

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.2.6.6. WRITE_PROTECTION_MEMORY_CARD

命令指定字节在卡片中与存储在特定地址位置的字节逐个对比。若数据相符，相应的保护位会不可逆地设定为“0”。

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

Pseudo-APDU								
CLA	**INS**	**P1**	**字节地址**	**MEM_L**	**字节 1**	**....**	**....**	**字节 N**
FFh	D1h	00h						

其中：

字节地址 = $000A_4\ A_3A_2A_1A_0$b (00h - 1Fh) 是存储卡的保护内存地址位置

MEM_L 要写入存储卡的数据的长度

字节 x 要与卡片内始于**字节地址**（Byte Address）的数据做比较的字节值。字节 1 与在 Byte Address 的数据比较；字节 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：

SW1 SW2 = 90 00h（未发生错误）

8.2.6.7. PRESENT_CODE_MEMORY_CARD (SLE 4442 和 SLE 5542)

此命令用于向存储卡提交密码，使能够对 SLE 4442 和 SLE 5542 卡进行写操作。执行以下操作：

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（*PC_to_RDR_XfrBlock* 中的 *abData* 数据域）

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

其中：

密码 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。

8.2.6.8. CHANGE_CODE_MEMORY_CARD (SLE 4442 和 SLE 5542)

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	密码		
					字节 1	字节 2	字节 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：

SW1 SW2 = 90 00h（未发生错误）

附录 A. 槽位状态和槽位错误

所有 Builk-IN 消息都包含槽位错误和槽位状态寄存器的值。

偏移	数据域	大小	值	说明
0	bmICCStatus	2 位	0, 1, 2	0 - ICC 存在且处于激活状态（已开启稳定供电，RST 信号未激活） 1 - ICC 存在但未激活（由于硬件错误导致未激活或关闭) 2 - ICC 不存在 3 - RFU
2	bmRFU	4 位	RFU	智能海报数据的长度（15 字节）
6	bmCommandStatus	2 位	0, 1, 2	0 - 处理无误 1 - 失败（错误寄存器提供的错误代码） 2 - 请求延长时间 3 - RFU

表5：槽位状态寄存器

错误代码	错误名称	可能的原因
FFh	CMD_ABORTED	主机中止了当前活动
FEh	ICC_MUTE	与 ICC 通讯时，CCID 超时
FDh	XFR_PARITY_ERROR	与 ICC 通讯时，奇偶校验错误
FCh	XFR_OVERRUN	与 ICC 通讯时，超限错误
FBh	HW_ERROR	发生全面的硬件错误
F8h	BAD_ATR_TS	
F7h	BAD_ATR_TCK	
F6h	ICC_PROTOCOL_NOT_SUPPORTED	
F5h	ICC_CLASS_NOT_SUPPORTED	
F4h	PROCEDURE_BYTE_CONFLICT	
F3h	DEACTIVATED_PROTOCOL	
F2h	BUSY_WITH_AUTO_SEQUENCE	自动序列进行中
E0h	CMD_SLOT_BUSY	向已经在处理命令的插槽发送第二个命令
C0h 至 81h	用户定义	
80h 以及填补空缺的值	RFU	

错误代码	错误名称	可能的原因
7Fh 至 01h	不受支持的/错误的消息参数的索引	01h: dwLength 错误 05h: bSlot 不存在 07h: bPowerselect 错误（不支持） 08h: wLevelParameter 错误 0Ah: FI – DI 配对无效或不受支持 0Bh: TCCKTS 参数无效 0Ch: 不支持保护时间 0Dh: T = 0 WI 无效或不受支持 T = 1 BWI 或 CWI 无效或不受支持 0Eh: 请求的时钟停止支持无效或不受支持 0Fh: IFSC 大小无效或不受支持 10h: NAD 值无效或不受支持
00h	不支持此命令	

表6：槽位错误寄存器（bmCommandStatus = 1）