



Advanced Card Systems Ltd.
Card & Reader Technologies

ACR40U

智能卡读写器（**USB** 接口）

参考手册 V1.01



版本历史

日期	修订说明	版本号
2023/9/21	<ul style="list-style-type: none">Initial version	1.00
2023/11/09	<ul style="list-style-type: none">校验文本增加附录 A 并修复 7.2 引用错误	1.01



目录

1.0.	简介	5
1.1.	参考文件.....	5
1.2.	符号和缩写.....	5
2.0.	支持的智能卡	6
2.1.	MCU 卡.....	6
2.2.	存储卡.....	6
3.0.	系统方框图	7
4.0.	USB 接口	8
4.1.	通信参数.....	8
4.2.	端点.....	8
5.0.	用户接口	9
5.1.	LED 状态指示灯.....	9
6.0.	智能卡接口	10
6.1.	智能卡电源 VCC (C1).....	10
6.2.	编程电压 VPP (C6).....	10
6.3.	卡片类型选择.....	10
6.4.	微控制器卡接口.....	10
7.0.	USB 通信协议	11
7.1.	CCID Bulk-OUT 消息.....	12
7.1.1.	PC_to_RDR_IccPowerOn.....	12
7.1.2.	PC_to_RDR_IccPowerOff.....	13
7.1.3.	PC_to_RDR_GetSlotStatus.....	13
7.1.4.	PC_to_RDR_XfrBlock.....	13
7.1.5.	PC_to_RDR_GetParameters.....	14
7.1.6.	PC_to_RDR_ResetParameters.....	14
7.1.7.	PC_to_RDR_SetParameters.....	15
7.2.	CCID Bulk-IN 消息.....	17
7.2.1.	RDR_to_PC_DataBlock.....	17
7.2.2.	RDR_to_PC_SlotStatus.....	18
7.2.3.	RDR_to_PC_Parameters.....	18
8.0.	主机编程 API	20
8.1.	外设控制 (Peripherals Control).....	20
8.1.1.	获取固件版本 (Get Firmware Version).....	20
8.1.2.	获取卡片电压选择顺序 (Get Card Voltage Selection Sequence).....	20
8.1.3.	设置卡片电压选择顺序 (Set Card Voltage Selection Sequence).....	21
8.1.4.	写入客户数据 (Write Customer Data).....	22
8.1.5.	读取客户数据 (Read Customer Data).....	22
8.1.6.	修改客户 PIN 码 (Change Customer PIN).....	22
8.1.7.	读取 LED 状态 (Read LED Status).....	22
8.1.8.	写入 LED 状态 (Write LED status).....	23
8.2.	存储卡命令集.....	24
8.2.1.	存储卡 – 1、2、4、8 和 16 Kb 的 I2C 卡.....	24
8.2.2.	存储卡 – 32、64、128、256、512 和 1024 Kb 的 I2C 卡.....	27
8.2.3.	存储卡 – ATMEL AT88SC153.....	30
8.2.4.	存储卡 – ATMEL AT88C1608.....	34



8.2.5. 存储卡 – SLE4418/SLE4428/SLE5518/SLE5528	38
8.2.6. 存储卡 – SLE4432/SLE4442/SLE5532/SLE5542	43
附录 A. 响应错误码.....	48

图目录

图 1：ACR40U 架构.....	7
--------------------	---

表目录

表 1：符号和缩写	5
表 2：USB 接口配线	8
表 3：LED 状态指示灯	9
表 4：响应错误码	48



1.0. 简介

ACR40U 智能卡读写器（USB 接口）是计算机和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。ACR40U 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 ACR40U 处理智能卡的具体细节，程序员可专注于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	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	预留给将来使用 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

ACR40U 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 ACR40U 不支持该特定模式，则 ACR40U 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，ACR40U 会拒绝读写该卡。

若卡片的 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 ACR40U 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 的规定。

2.2. 存储卡

ACR40U 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入保护功能的存储卡，包括：
 - Infineon®: SLE4418、SLE4428、SLE5518 和 SLE5528
- 具有 256 字节 EEPROM 智能存储空间以及写入保护功能的存储卡，包括：
 - Infineon®: SLE4432、SLE4442、SLE5532 和 SLE5542

3.0. 系统方框图

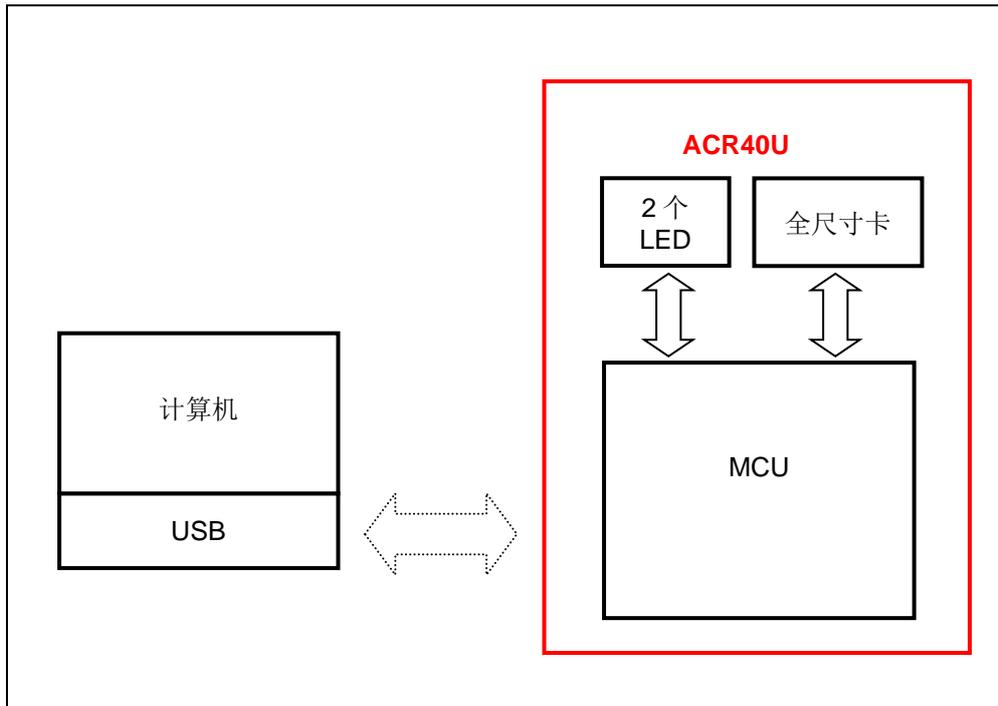


图 1: ACR40U 架构



4.0. USB 接口

4.1. 通信参数

ACR40U 通过符合 USB2.0 规范的 USB 端口与计算机建立连接，它支持 USB 全速模式，速率为 12Mbps。

引脚	信号	功能
1	V _{BUS}	为读写器提供+5 V 的电源
2	D-	ACR40U 和计算机之间以差分信号传输数据。
3	D+	ACR40U 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

ACR40U 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 ACR40U 的命令 (数据包的大小为 64 字节)
批量输入 (Bulk IN)	用于从 ACR40U 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 ACR40U 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

ACR40U 有两个 LED 指示灯，用于显示各种操作状态，其中：

- **蓝色 LED** - USB 连接模式下卡片和读写器的状态
- **绿色 LED** - 智能卡读写器电源状态

颜色	LED 操作	状态
蓝色	关	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
	开	卡片已连接并上电

颜色	LED 操作	状态
绿色	关	读写器下电
	开	读写器上电

表 3: LED 状态指示灯



6.0. 智能卡接口

ACR40U 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 ACR40U 的实用功能。

6.1. 智能卡电源 VCC (C1)

插入的智能卡的电流消耗不得大于 50 mA。

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，ACR40U 的触点 C6 (VPP) 没有用于做为普通的控制信号。

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 ACR40U 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

ACR40U 通过 USB 与主机 (host) 端建立交互。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。

ACR40U 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**经由 BULK-OUT 端点发送。发送至 ACR40U 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 ACR40U 的命令都必须同步发送 (例如: 对于 ACR40U 来说, *bMaxCCIDBusySlots* 等同于 01h)。

ACR40U 支持的 CCID 特性见其类别描述符:

偏移	数据域	大小	值	说明
0	<i>bLength</i>	1	-	描述符的字节数
1	<i>bDescriptorType</i>	1	-	CCID 功能描述符的类别
2	<i>bcdCCID</i>	2	-	二进制编码的十进制 CCID 规范版本号
4	<i>bMaxSlotIndex</i>	1	-	ACR40U 有一个卡槽
5	<i>bVoltageSupport</i>	1	-	ACR40U 可支持 1.8V、3V 和 5V 的槽位电压。
6	<i>dwProtocols</i>	4	-	ACR40U 支持 T=0 和 T=1 协议
10	<i>dwDefaultClock</i>	4	-	默认 ICC 时钟频率为 4.8 MHz
14	<i>dwMaximumClock</i>	4	-	ICC 支持的最大时钟频率为 4.8 MHz
18	<i>bNumClockSupported</i>	1	-	不支持手动设置时钟频率
19	<i>dwDataRate</i>	4	-	默认 ICC I/O 数据传输速率为 12903 bps
23	<i>dwMaxDataRate</i>	4	-	支持的最大 ICC I/O 数据传输速率为 600 Kbps
27	<i>bNumDataRatesSupported</i>	1	-	不支持手动设置数据传输速率
28	<i>dwMaxIFSD</i>	4	-	ACR40U 在 T=1 协议下支持的最大 IFSD 为 247。
32	<i>dwSynchProtocols</i>	4	-	ACR40U 不支持同步卡
36	<i>dwMechanical</i>	4	-	ACR40U 不支持特殊机制特性
40	<i>dwFeatures</i>	4	-	ACR40U 支持以下特性: <ul style="list-style-type: none"> • 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 ACR40U 进行 TPDU 级交换
44	<i>dwMaxCCIDMessageLength</i>	4	-	ACR40U 可接受的最大报文长度为 271 字节。



偏移	数据域	大小	值	说明
48	<i>bClassGetResponse</i>	1	-	对 TPDU 级别的交换没有影响
49	<i>bClassEnvelope</i>	1	-	对 TPDU 级别的交换没有影响
50	<i>wLCDLayout</i>	2	-	无 LCD
52	<i>bPINSupport</i>	1	-	支持 PIN 校验
53	<i>bMaxCCIDBusySlots</i>	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	<i>bMessageType</i>	1	62h	-
1	<i>dwLength</i>	4	00000000h	此消息的额外字节的大小
5	<i>bSlot</i>	1	-	标识命令的卡槽号
6	<i>bSeq</i>	1	-	命令的序号
7	<i>bPowerSelect</i>	1	-	ICC 上的电压值： 00h = 自动电压选择 01h = 5 V 02h = 3 V 03h = 1.8V
8	<i>abRFU</i>	2	-	预留为将来使用

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	<i>bMessageType</i>	1	63h	-
1	<i>dwLength</i>	4	00000000h	此消息的额外字节的大小
5	<i>bSlot</i>	1	-	标识命令的卡槽号
6	<i>bSeq</i>	1	-	命令的序号
7	<i>abRFU</i>	3	-	预留为将来使用

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	<i>bMessageType</i>	1	65h	-
1	<i>dwLength</i>	4	00000000h	此消息的额外字节的大小
5	<i>bSlot</i>	1	-	标识命令的卡槽号
6	<i>bSeq</i>	1	-	命令的序号
7	<i>abRFU</i>	3	-	预留为将来使用

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	<i>bMessageType</i>	1	6Fh	-
1	<i>dwLength</i>	4	-	此消息的 <i>abData</i> 数据域的大小
5	<i>bSlot</i>	1	-	标识命令的卡槽号。
6	<i>bSeq</i>	1	-	命令的序号。
7	<i>bBWI</i>	1	-	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。

偏移	数据域	大小	值	说明
8	<i>wLevelParameter</i>	2	-	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 APDU 命令并结束该 APDU 命令。 0003h - abData 字段继续传递 APDU 命令, 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
10	<i>abData</i>	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_GetParameters

此命令用于获取卡槽的参数。

偏移	数据域	大小	值	说明
0	<i>bMessageType</i>	1	6Ch	-
1	<i>DwLength</i>	4	00000000h	此消息的额外字节的大小
5	<i>BSlot</i>	1	-	标识命令的卡槽号
6	<i>BSeq</i>	1	-	命令的序号
7	<i>AbRFU</i>	3	-	预留为将来使用

此消息的响应是 *RDR_to_PC_Parameters* 消息。

7.1.6. 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	<i>bMessageType</i>	1	6Dh	-
1	<i>DwLength</i>	4	00000000h	此消息的额外字节的大小
5	<i>BSlot</i>	1	-	标识命令的卡槽号
6	<i>BSeq</i>	1	-	命令的序号
7	<i>AbRFU</i>	3	-	预留为将来使用

此消息的响应是 *RDR_to_PC_Parameters* 消息。



7.1.7. PC_to_RDR_SetParameters

此命令用于设置卡槽的参数。

偏移	数据域	大小	值	说明
0	<i>bMessageType</i>	1	61h	-
1	<i>dwLength</i>	4	-	此消息的额外字节的大小
5	<i>bSlot</i>	1	-	标识命令的卡槽号
6	<i>bSeq</i>	1	-	命令的序号
7	<i>bProtocolNum</i>	1	-	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 以下值预留为将来使用： 80h: 2线协议结构 81h: 3线协议结构 82h: I2C 协议结构
8	<i>abRFU</i>	2	-	预留为将来使用
10	<i>abProtocolDataStructure</i>	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	<i>bmFindexDindex</i>	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	<i>bmTCCKST0</i>	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	<i>bGuardTimeT0</i>	1	-	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	<i>bWaitingIntegerT0</i>	1	-	T=0 时 WI 用于定义 WWT
14	<i>bClockStop</i>	1	-	支持 ICC 时钟停止 00h = 不允许停止时钟 01h = 时钟信号为低时停止 02h = 时钟信号为高时停止 03h = 时钟信号为高或为低时停止

T=1 的协议数据结构(dwLength=00000007h)

偏移	数据域	大小	值	说明
10	<i>bmFindexDindex</i>	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	<i>BmTCCKST1</i>	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	<i>BGuardTimeT1</i>	1	-	额外保护时间 (两个字符间为 0-254 个 etu)。若值为 FFh, 则保护时间减少 1 个 etu。
13	<i>BwaitingIntegerT1</i>	1	-	B7-4 = BWI 值, 0-9 有效 B3-0 = CWI 值, 0-Fh 有效

偏移	数据域	大小	值	说明
14	<i>bClockStop</i>	1	-	支持 ICC 时钟停止 00h = 不允许停止时钟 01h = 时钟信号为低时停止 02h = 时钟信号为高时停止 03h = 时钟信号为高或为低时停止
15	<i>bIFSC</i>	1	-	商定的 IFSC 的大小
16	<i>bNadValue</i>	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 ACR40U 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	<i>bMessageType</i>	1	80h	表示 CCID 正在发送一个数据块。
1	<i>dwLength</i>	4	-	此消息的额外字节的大小
5	<i>bSlot</i>	1	-	与 Bulk-OUT 消息中的值相同
6	<i>bSeq</i>	1	-	与 Bulk-OUT 消息中的值相同
7	<i>bStatus</i>	1	-	CCID 标准（1.1 版本）6.2.1 节定义的插槽状态寄存器
8	<i>bError</i>	1	-	响应错误码和 CCID 标准（1.1 版本）6.2.6 节定义的插槽错误寄存器
9	<i>bChainParameter</i>	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 <i>abData</i> 字段继续传递响应 APDU 并结束该响应 APDU。 03h - 此 <i>abData</i> 字段继续传递响应 APDU，后面跟随另外一个数据块。 10h - 空的 <i>abData</i> 字段，下一个 <i>PC_to_RDR_XfrBlock</i> 命令会继续传递命令 APDU
10	<i>abData</i>	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 ACR40U 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	<i>bMessageType</i>	1	81h	-
1	<i>dwLength</i>	4	00000000h	此消息的额外字节的大小
5	<i>bSlot</i>	1	-	与 Bulk-OUT 消息中的值相同
6	<i>bSeq</i>	1	-	与 Bulk-OUT 消息中的值相同
7	<i>bStatus</i>	1	-	CCID 标准（1.0 版本）4.2.1 节定义的插槽状态寄存器
8	<i>bError</i>	1	-	响应错误码和 CCID 标准（1.0 版本）4.2.1 节定义的插槽错误寄存器
9	<i>bClockStatus</i>	1	-	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 所有其他值预留为将来使用

7.2.3. RDR_to_PC_Parameters

此消息由 ACR40U 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	<i>bMessageType</i>	1	82h	-
1	<i>dwLength</i>	4	-	此消息的额外字节的大小
5	<i>bSlot</i>	1	-	与 Bulk-OUT 消息中的值相同
6	<i>bSeq</i>	1	-	与 Bulk-OUT 消息中的值相同
7	<i>bStatus</i>	1	-	CCID 标准（1.0 版本）4.2.1 节定义的插槽状态寄存器
8	<i>bError</i>	1	-	响应错误码和 CCID 标准（1.0 版本）4.2.1 节定义的插槽错误寄存器
9	<i>bProtocolNum</i>	1	-	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 以下值预留为将来使用： 80h: 2 线协议结构 81h: 3 线协议结构 82h: I2C 协议结构



偏移	数据域	大小	值	说明
10	<i>abProtocolDataStructure</i>	字节 型数 组	-	协议数据结构如 CCID 标准（1.0 版本） 5.2.3 节汇总。

8.0. 主机编程 API

8.1. 外设控制 (Peripherals Control)

读写器的外设控制命令在 USB 模式下通过 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 命令的格式 (5 字节)

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

Get Firmware Version 的响应格式 (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	待接收字节数	固件版本号

例如: 响应 = E1h 00h 00h 00h 0Bh 41h 43h 52h 34h 30h 55h 2Dh 50h 32h 30h 30h

固件版本 (HEX) = 41h 43h 52h 34h 30h 55h 2Dh 50h 32h 30h 30h

固件版本 (ASCII) = "ACR40U-P200"

8.1.2. 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 Sequence 的响应格式 (5 字节 + 卡片电压选择顺序)

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	待接收字节数	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

8.1.3. 设置卡片电压选择顺序 (Set Card Voltage Selection Sequence)

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

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

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

Set Card Voltage Selection Sequence 的响应格式 (5 字节 + 卡片电压选择顺序)

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	待接收字节数	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. 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。

8.1.5. 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。

8.1.6. 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。

8.1.7. 读取 LED 状态（Read LED Status）

此命令用于读取 LED 的状态。

Read the Power LED' s status 命令的格式（5 字节）

命令	CLA	INS	P1	P2	Lc
Read button status	E0h	00h	00h	29h	00h

Read power LED status 的响应格式（5 字节 + 电源 LED 状态 + 2 字节返回码）

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	E1h	00h	00h	00h	待接收字节数	电源 LED 状态 (1 字节)	返回码 (2 字节)

其中：

电源 LED 状态（1 字节）：

LED 状态	描述	说明
Bit 0	当前电源 LED 的状态	1 = 开；0 = 关
Bit 1	默认电源 LED 设置（在启动时激活）	1 = 开；0 = 关
Bit 2 - 7	RFU	RFU

返回码

结果	SW1 SW2	含义
成功	90 00h	命令执行成功



结果	SW1 SW2	含义
失败	6D 00h	命令执行失败

8.1.8. 写入 LED 状态 (Write LED status)

此命令用于写入电源 LED 的状态。

Write LED status 命令的格式 (5 字节)

命令	CLA	INS	P1	P2	Lc	数据	
Write LED status	E0h	00h	00h	29h	02h	存储	LED 状态

Write LED status 的响应格式 (5 字节 + 电源 LED 状态 + 2 字节返回码)

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	E1h	00h	00h	00h	待接收字节数	电源 LED 状态	返回码

其中:

存储 (1 字节):

存储	说明
1	将设置存储到 EEPROM
0	不将设置存储到 EEPROM

电源 LED 状态 (1 字节):

LED 状态	描述	说明
Bit 0	当前电源 LED 的状态	1 = 开; 0 = 关
Bit 1	默认电源 LED 的设置 (在启动时激活)	1 = 开; 0 = 关
Bit 2 - 7	RFU	RFU

返回码

结果	SW1 SW2	含义
成功	90 00h	命令执行成功
失败	6D 00h	命令执行失败

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)

私有 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)

私有 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）

私有 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）

私有 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)

私有 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)

私有 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）

私有 APDU				
CLA	INS	字节地址		MEM_L
		MSB	LSB	
FFh				

其中：

INS = B0h: 32 Kb、64 Kb、128 Kb、256 Kb 和 512 Kb 的 IIC 卡
 = 1011 000*b: 1024 Kb 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）

私有 APDU								
CLA	INS	字节地址		MEM_L	字节 1	字节 n
		MSB	LSB					
FFh								

其中：

- INS** = D0h: 32 Kb、64 Kb、128 Kb、256 Kb 和 512 Kb 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)

私有 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)

私有 APDU				
CLA	INS	P1	字节地址	MEM_L
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）

私有 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 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）

8.2.3.4. VERIFY_PASSWORD

命令格式（PC_to_RDR_XfrBlock 中的 abData 数据域）

私有 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）

私有 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）

私有 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)

私有 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)

私有 APDU				
CLA	INS	区域地址	字节地址	MEM_L
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 0A₁₀A₉A₈b, 其中 A₁₀ 是区域地址的 MSB
= 读标识位时无关

字节地址 = A₇A₆A₅A₄ A₃A₂A₁A₀b 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）

私有 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 0A₁₀A₉A₈b, 其中 A₁₀ 是区域地址的 MSB
 = 写标识位时无关
字节地址 = A₇A₆A₅A₄ A₃A₂A₁A₀b 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）

私有 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 rp₂p₁p₀b

其中“rp2p1p0”位指明待比较的密码

r = 0: “写”密码

r = 1: “读”密码

p2p1p0: 密码集编号。

(rp2p1p0 = 0111: 安全码)

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)

私有 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)

私有 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）

私有 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）

私有 APDU				
CLA	INS	字节地址		MEM_L
		MSB	LSB	
FFh	B0h			

其中：

MSB 字节地址 = 0000 00A₉A₈b 是存储卡的内存地址位置

LSB 字节地址 = A₇A₆A₅A₄ A₃A₂A₁A₀b 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)

私有 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)

私有 APDU				
CLA	INS	字节地址		MEM_L
		MSB	LSB	
FFh	B2h			

其中:

- MSB 字节地址** = 0000 00A₉A₈b 是存储卡的内存地址位置
- LSB 字节地址** = A₇A₆A₅A₄ A₃A₂A₁A₀b 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								...									
P 8	P 7	P 6	P 5	P 4	P 3	P 2	P 1	P1 6	P1 5	P1 4	P1 3	P1 2	P1 1	P1 0	P 9	P1 8	P1 7

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）

私有 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 00A9A8b 是存储卡的内存地址位置
- LSB 字节地址** = A7A6A5A4 A3A2A1A0b 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）

私有 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 00A₉A₈b 是存储卡的内存地址位置
- LSB 字节地址** = A₇A₆A₅A₄ A₃A₂A₁A₀b 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。
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. 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)

私有 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)

私有 APDU				
CLA	INS	P1	字节地址	MEM_L
FFh	B0h	00h		

其中：

字节地址 = A₇A₆A₅A₄ A₃A₂A₁A₀b 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)

私有 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)

私有 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）

私有 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 = A7A6A5A4 A3A2A1A0b 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）

私有 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 = 000A4 A3A2A1A0b (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)

私有 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）

私有 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. 响应错误码

以下表格总结了 ACR40U 可能返回的错误码：

错误码	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：响应错误码

Android 是 Google LLC. 的商标

Atmel 是 Atmel Corporation 或其子公司在美国及其他国家的注册商标。

蓝牙™ 字样、标记和标识是蓝牙技术联盟拥有的注册商标，龙杰智能卡有限公司对上述标记的使用都具有合法授权。其他商标或商品名称均为各自所有者的财产。

Infineon 是 Infineon Technologies AG 的注册商标。

Microsoft 是 Microsoft Corporation 在美国及/或其他国家的注册商标。