



**Advanced Card Systems Ltd.**  
Card & Reader Technologies

# ACR39 系列 智能卡读写器

参考手册 V1.05



## 版本历史

发布日期	修订说明	版本号
2013-09-06	<ul style="list-style-type: none"><li>● 初始发布</li></ul>	1.00
2014-03-25	<ul style="list-style-type: none"><li>● 更新 2.0 节：特性</li><li>● 将标题由 ACR39U 更新为 ACR39x</li><li>● 新增品牌商标归属说明</li><li>● 将 USB 接口介绍部分移至接触式智能卡接口前</li></ul>	1.01
2015-05-13	<ul style="list-style-type: none"><li>● 更新 2.0 节：特性</li></ul>	1.02
2015-12-14	<ul style="list-style-type: none"><li>● 更新 2.0 节：特性</li><li>● 重新安排格式结构</li><li>● 更新 9.1 节：通过 PC_to_RDR_Xfrblock 访问的其它命令</li><li>● 删除附录 A – 支持的卡片类型</li></ul>	1.03
2017-10-10	<ul style="list-style-type: none"><li>● 更新文件标题</li><li>● 将所有 ACR39x 重命名为 ACR39</li><li>● 更新 2.0 节：特性</li></ul>	1.04
2020-05-27	<ul style="list-style-type: none"><li>● 更新 2.0 节：特性</li><li>● 新增 2.4 节 ACR39F</li><li>● 更新 9.0 节：通过 PC_to_RDR_XfrBlock 执行的其他命令</li><li>● 新增 10.0 节：通过 PC-to_RDR_Escape 执行的其他命令</li></ul>	1.05



## 目录

<b>1.0.</b>	<b>简介</b> .....	<b>5</b>
1.1.	参考文件 .....	5
1.2.	符号和缩写 .....	5
<b>2.0.</b>	<b>特性</b> .....	<b>6</b>
2.1.	ACR39U .....	6
2.2.	ACR39U PocketMate II .....	7
2.3.	ACR39T .....	8
2.4.	ACR39F .....	9
<b>3.0.</b>	<b>支持的智能卡</b> .....	<b>10</b>
3.1.	MCU 卡 .....	10
3.2.	存储卡 .....	10
<b>4.0.</b>	<b>USB 接口</b> .....	<b>11</b>
4.1.	通信参数 .....	11
4.2.	端点 .....	11
<b>5.0.</b>	<b>接触式智能卡接口</b> .....	<b>12</b>
5.1.	智能卡电源 VCC (C1) .....	12
5.2.	编程电压 VPP (C6) .....	12
5.3.	卡片类型选择 .....	12
5.4.	微控制器卡接口 .....	12
5.5.	卡片插拔保护 .....	12
<b>6.0.</b>	<b>电源参数</b> .....	<b>13</b>
6.1.	LED 状态指示灯 .....	13
<b>7.0.</b>	<b>USB 通信协议</b> .....	<b>14</b>
7.1.	CCID Bulk-OUT 消息 .....	16
7.1.1.	PC_to_RDR_lccPowerOn .....	16
7.1.2.	PC_to_RDR_lccPowerOff .....	16
7.1.3.	PC_to_RDR_GetSlotStatus .....	17
7.1.4.	PC_to_RDR_XfrBlock .....	17
7.1.5.	PC_to_RDR_GetParameters .....	18
7.1.6.	PC_to_RDR_ResetParameters .....	18
7.1.7.	PC_to_RDR_SetParameters .....	19
7.2.	CCID Bulk-IN 消息 .....	21
7.2.1.	RDR_to_PC_DataBlock .....	21
7.2.2.	RDR_to_PC_SlotStatus .....	21
7.2.3.	RDR_toPC_Parameters .....	22
<b>8.0.</b>	<b>存储卡命令集</b> .....	<b>23</b>
8.1.	存储卡 – 1、2、4、8 和 16 kilobit I2C 卡 .....	23
8.1.1.	SELECT_CARD_TYPE .....	23
8.1.2.	SELECT_PAGE_SIZE .....	23
8.1.3.	READ_MEMORY_CARD .....	24
8.1.4.	WRITE_MEMORY_CARD .....	24
8.2.	存储卡 – 32、64、128、256、512 和 1024 kilobit I2C 卡 .....	26
8.2.1.	SELECT_CARD_TYPE .....	26
8.2.2.	SELECT_PAGE_SIZE .....	26
8.2.3.	READ_MEMORY_CARD .....	27



8.2.4.	WRITE_MEMORY_CARD .....	27
8.3.	存储卡 – SLE4418/SLE4428/SLE5518/ SLE5528 .....	29
8.3.1.	SELECT_CARD_TYPE .....	29
8.3.2.	READ_MEMORY_CARD.....	29
8.3.3.	READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (仅限 SLE4428 和 SLE5528) 30	
8.3.4.	READ_PROTECTION_BIT .....	30
8.3.5.	WRITE_MEMORY_CARD .....	31
8.3.6.	WRITE_PROTECTION_MEMORY_CARD .....	32
8.3.7.	PRESENT_CODE_MEMORY_CARD (仅限 SLE4428 和 SLE5528) .....	32
8.4.	存储卡 – SLE4432/SLE4442/SLE5532/SLE5542 .....	34
8.4.1.	SELECT_CARD_TYPE .....	34
8.4.2.	READ_MEMORY_CARD.....	34
8.4.3.	READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (仅限 SLE4442 和 SLE5542) 35	
8.4.4.	READ_PROTECTION_BITS .....	35
8.4.5.	WRITE_MEMORY_CARD .....	36
8.4.6.	WRITE_PROTECTION_MEMORY_CARD .....	37
8.4.7.	PRESENT_CODE_MEMORY_CARD (仅限 SLE4442 和 SLE5542) .....	37
8.4.8.	CHANGE_CODE_MEMORY_CARD (仅限 SLE4442 和 SLE5542) .....	38
<b>9.0.</b>	<b>通过 PC_to_RDR_XfrBlock 执行的其他命令 .....</b>	<b>39</b>
9.1.	GET_READER_INFORMATION .....	39
<b>10.0.</b>	<b>通过 PC-to_RDR_Escape 执行的其它命令 .....</b>	<b>40</b>
10.1.	GET_READER_INFORMATION .....	40
<b>附录 A.响应错误代码.....</b>		<b>41</b>

## 表目录

<b>表 1</b>	<b>: 符号和缩写 .....</b>	<b>5</b>
<b>表 2</b>	<b>: USB 接口配线 .....</b>	<b>11</b>
<b>表 3</b>	<b>: 响应错误代码 .....</b>	<b>41</b>



## 1.0. 简介

ACR39 联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。ACR39 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](http://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](http://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	保留为将来使用（Reserved for future use <sup>1</sup> ）
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

<sup>1</sup> 除非另有说明，否则必须设置为零。



## 2.0. 特性

### 2.1. ACR39U

- USB 全速接口\*
- 即插即用 – 支持 CCID 标准，具有高度的灵活性
- 智能卡读写器：
  - 接触式接口：
    - 支持 ISO 7816 A 类、B 类和 C 类（5 V、3 V、1.8 V）卡
    - 支持 CAC（通用权限卡）
    - 支持 SIPRNET 卡
    - 支持 J-LIS 卡
    - 支持符合 T=0 和 T=1 协议的微处理器卡
    - 支持存储卡
    - 支持 PPS（协议与参数选择）
    - 具有短路保护功能
- 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<sup>2</sup>
- 符合下列标准：
  - EN60950/IEC 60950
  - ISO 7816
  - EMV™ Level 1（接触式）
  - PC/SC
  - CCID
  - CE
  - FCC
  - WEEE
  - RoHS
  - REACH
  - TAA (美国)
  - J-LIS (日本)
  - VCCI (日本)
  - PBOC (中国)
  - Microsoft® WHQL

---

<sup>2</sup> 使用 ACS 自定义的 Android 库

\*可选择 USB Type A 和 USB Type C 接头



## 2.2. ACR39U PocketMate II

- USB 全速接口\*
- 即插即用—符合 CCID 标准，具有高度的灵活性
- 旋转式设计
- 智能卡读写器：
  - 接触式接口：
    - 支持 ISO 7816 A 类、B 类和 C 类（5 V、3 V、1.8 V）卡
    - 支持 CAC（通用权限卡）
    - 支持 SIPRNET 卡
    - 支持 J-LIS 卡
    - 支持符合 T=0 和 T=1 协议的微处理器卡
    - 支持存储卡
    - 支持 PPS（协议与参数选择）
    - 具有短路保护功能
  - 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<sup>3</sup>
- 符合下列标准：
  - EN60950/IEC 60950
  - ISO 7816
  - EMV™ Level 1（接触式）
  - PC/SC
  - CCID
  - CE
  - FCC
  - WEEE
  - RoHS
  - REACH
  - TAA (美国)
  - J-LIS (日本)
  - VCCI (日本)
  - PBOC (中国)
  - Microsoft® WHQL

---

<sup>3</sup> 使用 ACS 自定义的 Android 库

\*可选择 USB Type A、USB Micro-B 或 USB Type C 接头



## 2.3. ACR39T

- USB 全速接口\*
- 即插即用 – 符合 CCID 标准，具有高度的灵活性
- 带有 USB 保护帽
- 智能卡读写器：
  - 接触式接口：
    - 支持 ISO 7816 A 类、B 类和 C 类（5 V、3 V、1.8 V）卡
    - 支持符合 T=0 和 T=1 协议的微处理器卡
    - 支持存储卡
    - 支持 PPS（协议与参数选择）
    - 具有短路保护功能
- 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<sup>4</sup>
- 符合下列标准：
  - EN 60950/IEC 60950
  - ISO 7816
  - PC/SC
  - CCID
  - CE
  - FCC
  - WEEE
  - RoHS
  - REACH
  - VCCI (日本)
  - Microsoft® WHQL

---

<sup>4</sup> 使用 ACS 自定义的 Android 库

\*可选择 USB Micro-B 和 Type C 接头





## 2.4. ACR39F

- USB 2.0 全速接口（通过可拆卸的连接线）
- 即插即用 –支持 CCID 标准，具有高度灵活性
- 智能卡读写器：
  - 接触式接口：
    - 支持 ISO 7816 A 类、B 类和 C 类（5 V、3 V、1.8 V）卡
    - 支持 CAC（通用权限卡）
    - 支持 SIPRNET 卡
    - 支持 J-LIS 卡
    - 支持符合 T=0 和 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
  - 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<sup>5</sup>
- 符合下列标准：
  - EN 60950/IEC 60950
  - ISO 7816
  - EMV™ Level 1（接触式）
  - PC/SC
  - CCID
  - CE
  - FCC
  - WEEE
  - RoHS
  - REACH
  - Microsoft® WHQL

---

<sup>5</sup> 不适用 PC/SC 和 CCID 支持



## 3.0. 支持的智能卡

### 3.1. MCU 卡

ACR39 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，读写速率最高达 600Kbps，还支持下述 T=0 和 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 ACR39 不支持该特定模式，则 ACR39 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，ACR39 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 ACR39 读写器将执行 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 的规定。*

### 3.2. 存储卡

ACR39 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
  - SGS-Thomson: ST14C02C、ST14C04C
  - Gemplus: GFM1K、GFM2K、GFM4K、GFM8K
- 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



## 4.0. USB 接口

### 4.1. 通信参数

ACR39 通过符合 USB 2.0 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 12 Mbps。

引脚	信号	功能
1	V <sub>BUS</sub>	为读写器提供+5 V 的电源
2	D-	ACR39 和 PC 间以差分信号传输数据
3	D+	ACR39 和 PC 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

ACR39 通过如下端点与主计算机进行通信：

- Control Endpoint** 用于设置和控制。
- Bulk-OUT** 用于从主机发送至 ACR39 的命令。  
(数据包的大小为 64 字节)
- Bulk-IN** 用于从 ACR39 发送至主机的响应。  
(数据包的大小为 64 字节)
- Interrupt-IN** 用于从 ACR39 发送至主机的卡片状态消息。  
(数据包的大小为 8 字节)



## 5.0. 接触式智能卡接口

ACR39 与智能卡之间的接口符合 ISO 7816-3 规范，并进行了某些限制或提升来增强 ACR39 的实用功能。

### 5.1. 智能卡电源 VCC (C1)

插入的智能卡的电流消耗不得大于 50 mA。

### 5.2. 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，ACR39 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。

### 5.3. 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 ACR39 发送适当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。

### 5.4. 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。

### 5.5. 卡片插拔保护

ACR39 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 ACR39 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：**ACR39 本身不会接通向卡片的供电。此操作由处于控制地位的计算机向读写器发送适当的命令来明确进行。



## 6.0. 电源参数

ACR39 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连接线）。

### 6.1. LED 状态指示灯

LED 可以指示智能卡接口的激活状态：

- **缓慢闪烁（每 2 秒钟开启 200 毫秒）**

表示 ACR39 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 ACR39 和智能卡间存在通信。

## 7.0. USB 通信协议

ACR39 应当通过 USB 连接与主机 (host) 端建立接口。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。

ACR39 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 (默认通道) 发送。其中包括类特定请求和 USB 标准请求。由默认通道发送的命令会通过默认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 端点发出。发送至 ACR39 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 端点发出。所有发送至 ACR39 的命令都必须同步发送 (例如: 对于 ACR39 来说, bMaxCCIDBusySlots 等同于 01h)。

ACR39 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 功能描述符的类型。
2	<i>bcdCCID</i>	2		CCID 以二进制编码的十进制指定的版本号码。
4	<i>bMaxSlotIndex</i>	1		ACR39 有一个卡槽
5	<i>bVoltageSupport</i>	1		ACR39 可支持 1.8 V、3 V 和 5 V 的槽位电压。
6	<i>dwProtocols</i>	4		ACR39 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 数据率为 12918 bps。
23	<i>dwMaxDataRate</i>	4		ICC I/O 支持的最大数据率为 826 Kbps。
27	<i>bNumDataRatesSupported</i>	1		不支持手动设置数据率。
28	<i>dwMaxIFSD</i>	4		ACR39 T1 支持的最大 IFSD 为 247。
32	<i>dwSynchProtocols</i>	4		ACR39 不支持同步卡。
36	<i>dwMechanical</i>	4		ACR39 不支持特殊机制特性。
40	<i>dwFeatures</i>	4		ACR39 支持以下特性: <ul style="list-style-type: none"> <li>• 自动根据参数改变 ICC 时钟频率</li> <li>• 自动根据频率和 FI、DI 参数更改波特率</li> <li>• 与 ACR39 进行 TPDU 级交换</li> </ul>
44	<i>dwMaxCCIDMessageLength</i>	4		ACR39 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 验证。
53	<i>bMaxCCIDBusySlots</i>	1		可以同时忙的槽位数为 1。

## 7.1. CCID Bulk-OUT 消息

ACR39 应当遵循 CCID 协议（版本 1.0）第 4.1 部分定义的 Bulk-OUT 消息。该规范还定义了一些用于操作附加功能的扩展命令。

本节将列举 ACR39 支持的 CCID 类 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。
2	<i>bSlot</i>	1		标识命令的插槽号。
5	<i>bSeq</i>	1		命令的序号。
6	<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 注：自动电压选择的顺序首先是 1.8V，然后是 3V，最后是 5V。
7	<i>abRFU</i>	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。

### 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		标识命令的插槽号。
6	<i>bSeq</i>	1		命令的序号。
7	<i>abRFU</i>	3		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

此命令用于获取当前的卡槽状态。

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		标识命令的插槽号。
6	<i>bSeq</i>	1		命令的序号。
7	<i>abRFU</i>	3		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	<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		标识命令的插槽号。
6	<i>bSeq</i>	1		命令的序号。
7	<i>bBWI</i>	1		用于为当前传输延长 CCID 块的超时等待时间。“该数值乘以块等待时间”的时间段过去后，CCID 将超时该块。
8	<i>wLevelParameter</i>	2	0000h	RFU (TPDU 交换级别)。
10	<i>abData</i>	字节型数组		发送给 CCID 的数据块。信息是“按原样”发送至 ICC (TPDU 交换级别) 的。

此消息的响应是 *RDR\_to\_PC\_DataBlock* 消息。



### 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		标识命令的插槽号。
6	<i>bSeq</i>	1		命令的序号。
7	<i>abRFU</i>	3		保留为将来使用。

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		标识命令的插槽号。
6	<i>bSeq</i>	1		命令的序号。
7	<i>abRFU</i>	3		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\_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		标识命令的插槽号。
6	<i>bSeq</i>	1		命令的序号。
7	<i>bProtocolNum</i>	1		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	<i>abRFU</i>	2		保留为将来使用。
10	<i>abProtocolDataStructure</i>	字节型 数组		协议数据结构。

T=0 协议的协议数据结构 (*dwLength*=00000005h)

偏移	数据域	大小	值	说明
10	<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>bmTCKKST0</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) <b>注: CCID 忽略该位。</b>
12	<i>bGuardTimeT0</i>	1		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	<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 协议的协议数据结构(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) <b>注: CCID 忽略该位。</b>
12	<i>bGuardTimeT1</i>	1		额外保护时间 (两个字符间为 0 至 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 0-9h 有效 B3-0 – CWI 值 0-Fh 有效
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

Bulk-IN 消息用于对 Bulk-OUT 消息做出响应。ACR39 应当遵循 CCID 协议（版本 1.0）第 4 部分定义的 Bulk-IN 消息。

此节将列举 ACR39 支持的 CCID 类 Bulk-IN 消息。

### 7.2.1. RDR\_to\_PC\_DataBlock

RDR\_to\_PC\_DataBlock 消息由 ACR39 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	表示正在从 CCID 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 1.0 规范 4.2.1 节定义的插槽状态寄存器
8	<i>bError</i>	1		CCID 1.0 规范 4.2.1 节定义的插槽错误寄存器
9	<i>bChainParameter</i>	1	00h	RFU（TPDU 交换级别）。
10	<i>abData</i>	字节型数组		本数据域包含由 CCID 返回的数据。

### 7.2.2. RDR\_to\_PC\_SlotStatus

RDR\_to\_PC\_DataBlock 消息由 ACR39 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	00 00 00 00h	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 1.0 规范 4.2.1 节定义的插槽状态寄存器
8	<i>bError</i>	1		CCID 1.0 规范 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 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。



### 7.2.3. RDR\_toPC\_Parameters

此消息由 ACR39 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 1.0 规范 4.2.1 节定义的插槽状态寄存器
8	<i>bError</i>	1		CCID 规范 4.2.1 节定义的插槽错误寄存器
9	<i>bProtocolNum</i>	1		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
10	<i>abProtocolDataStructure</i>	字节型数组		协议数据结构。

## 8.0. 存储卡命令集

存储卡可以通过 *PC\_to\_RDR\_XfrBlock* 命令被访问。存储卡的所有功能均被映射进了 pseudo-APDU。

### 8.1. 存储卡 – 1、2、4、8 和 16 kilobit I2C 卡

#### 8.1.1. SELECT\_CARD\_TYPE

此命令用于对选定的卡片进行上电/下电，之后进行卡片复位操作。

**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：

**SW1, SW2** = 90 00h (未发生错误)。

#### 8.1.2. SELECT\_PAGE\_SIZE

此命令用来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：

**SW1, SW2** = 90 00h（未发生错误）。

### 8.1.3. 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**              要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）。

### 8.1.4. 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**              要从存储卡读取的数据的长度。

**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：

**SW1, SW2 = 90 00h**（未发生错误）。



## 8.2. 存储卡 – 32、64、128、256、512 和 1024 kilobit I2C 卡

### 8.2.1. SELECT\_CARD\_TYPE

此命令用于对选定的卡片进行上电/下电，之后进行卡片复位操作。

**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：

**SW1, SW2** = 90 00h（未发生错误）。

### 8.2.2. SELECT\_PAGE\_SIZE

此命令用来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：

**SW1, SW2** = 90 00h（未发生错误）。

### 8.2.3. 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 和 512 kilobit iic 卡  
 = 1011 000\*b: 1024 kilobit iic 卡  
 其中 \* 表示 17 位地址的 MSB。

**Byte Address** 存储卡的内存地址位置。

**MEM\_L** 要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）。

### 8.2.4. 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 和 512 kilobit iic 卡  
 = 1101 000\*b: 1024 kilobit iic 卡  
 其中 \* 表示 17 位地址的 MSB。

**Byte Address** 存储卡的内存地址位置。



**MEM\_L**                    要从存储卡读取的数据的长度。  
**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：

**SW1, SW2 = 90 00h**（未发生错误）。

### 8.3. 存储卡 – SLE4418/SLE4428/SLE5518/ SLE5528

#### 8.3.1. SELECT\_CARD\_TYPE

此命令用于对选定的卡片进行上电/下电，之后进行卡片复位操作。

**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：

**SW1, SW2** = 90 00h（未发生错误）。

#### 8.3.2. 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<sub>9</sub>A<sub>8</sub>b 是存储卡的内存地址位置

**LSB Byte Address** = A<sub>7</sub>A<sub>6</sub>A<sub>5</sub>A<sub>4</sub> A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub>b 是存储卡的内存地址位置

**MEM\_L** 要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）。

### 8.3.3. 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 个字节的虚拟数据

**SW1 SW2** = 90 00h (未发生错误)。

### 8.3.4. 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<sub>9</sub>A<sub>8</sub>b 是存储卡的内存地址位置

**LSB Byte Address** = A<sub>7</sub>A<sub>6</sub>A<sub>5</sub>A<sub>4</sub> A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub>b 是存储卡的内存地址位置

**MEM\_L** 要从存储卡中读取的数据的长度 (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, 应当发送下面的 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。
- SW1 SW2**        = 90 00h (未发生错误)。

在 PROT 字节中, 保护位的排列如下:

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

其中:

- Px** 是响应数据中 BYTE x 的保护位。
- '0' 字节被写保护。
- '1' 字节可以被写入

### 8.3.5. 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 00A9A8b 是存储卡的内存地址位置。
- LSB Byte Address**    = A7A6A5A4 A3A2A1A0b 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:

- SW1 SW2** = 90 00h (未发生错误)。

### 8.3.6. WRITE\_PROTECTION\_MEMORY\_CARD

命令中指定的每一个字节与存储在特定地址位置中的字节在卡片中对比。如果数据相符，则相应的保护位就会不可逆地被设定为“0”。

命令格式（*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<sub>9</sub>A<sub>8</sub>b 是存储卡的内存地址位置。
- LSB Byte Address** = A<sub>7</sub>A<sub>6</sub>A<sub>5</sub>A<sub>4</sub> A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub>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：

- SW1 SW2** = 90 00h（未发生错误）。

### 8.3.7. PRESENT\_CODE\_MEMORY\_CARD (仅限 SLE4428 和 SLE5528)

此命令用于向存储卡提交密码，使能够对 SLE4428 和 SLE5528 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	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。

## 8.4. 存储卡 – SLE4432/SLE4442/SLE5532/SLE5542

### 8.4.1. SELECT\_CARD\_TYPE

此命令用于对选定的卡片进行上电/下电，再进行卡片复位操作。

**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该命令。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：

**SW1 SW2** = 90 00h（未发生错误）。

### 8.4.2. 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<sub>7</sub>A<sub>6</sub>A<sub>5</sub>A<sub>4</sub> A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub>b 是存储卡的内存地址位置。

**MEM\_L** 要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	PROT 1	PROT 2	PROT 3	PROT 4	SW1	SW2

其中：

**BYTE x** 从存储卡读取的数据。

**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 BYTE x 的保护位。

‘0’ 字节被写保护。

‘1’ 字节可以被写入。

### 8.4.3. 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	DUMMY 3	SW1	SW2

其中：

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

**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)。

### 8.4.4. 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 BYTE x 的保护位。
- ‘0’字节被写保护
- ‘1’字节可以被写入。

### 8.4.5. 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**    = A7A6A5A4 A3A2A1A0b 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：

- SW1 SW2** = 90 00h（未发生错误）。

### 8.4.6. 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<sub>4</sub> A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub>b (00h 至 1Fh) 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：

**SW1 SW2** = 90 00h（未发生错误）。

### 8.4.7. PRESENT\_CODE\_MEMORY\_CARD (仅限 SLE4442 和 SLE5542)

此命令用于向存储卡提交密码，使能够对 SLE4442 和 SLE5542 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	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。

#### 8.4.8. 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：

**SW1 SW2** = 90 00h（未发生错误）。



## 9.0. 通过 PC\_to\_RDR\_XfrBlock 执行的其他命令

### 9.1. GET\_READER\_INFORMATION

此命令用于返回 ACR39 读写器的固件版本号。

**注：**只有通过 SCardConnect() API 建立 T=0 协议的逻辑智能卡读写器通信后才可以使用此命令。关于 SCardConnect() API 的详细说明请参见 PC/SC 规范。

命令格式（PC\_to\_RDR\_XfrBlock 中的 abData 数据域）

Pseudo-APDU				
CLA	INS	P1	P2	Le
FFh	09h	00h	00h	11h

响应数据格式（RDR\_to\_PC\_DataBlock 中的 abData 数据域）

FIRMWARE										

其中：

**FIRMWARE** 11 字节的固件版本数据。



## 10.0.通过 PC-to\_RDR\_Escape 执行的其它命令

### 10.1. GET\_READER\_INFORMATION

此命令用于返回 ACR39 读写器的固件版本号。

*注：该命令仅用于固件版本为 003R 及以上的 ACM39U 产品，使用前必须先安装好 ACS 驱动。如使用 Microsoft CCID 驱动，则需通过另外的应用来启动 CCID Escape 函数。*

Get Reader Information 的命令格式（5 个字节）

命令	CLA	INS	P1	P2	Lc
获取固件版本	E0h	00h	00h	19h	00h

Get Reader Information 的响应格式（5 个字节 + 固件信息长度）

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：

响应 = E1 00 00 00 0C 41 43 52 33 39 55 2D 30 2E 30 33 52

固件版本 (HEX) = 41 43 52 33 39 55 2D 30 2E 30 33 52

固件版本 (ASCII) = "ACR39U-0.03R"





## 附录A. 响应错误代码

下表汇总了 ACR39 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3 : 响应错误代码