Advanced Card Systems Ltd.
Card & Reader Technologies

ACR3901T-W1

ACS 安全蓝牙™ 接触式读卡器

参考手册 V1.02

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

版本历史

发布日期	修订说明	版本号
2017-08-30	• 初始发布	1.00
2017-11-06	• 更新 5.1.2 节：电池寿命 • 更新 6.3 节：认证 • 更新 6.6 节：相互认证 • 更新 6.6.1 节： SPH_to_RDR_ReqAuth • 更新 6.6.3 节： SPH_to_RDR_AuthRsp • 更新 6.6.5 节： SPH_to_RDR_DataReq • 更新 6.6.6 节： RDR_to_SPH_DataRsp • 更新 8.1.3 节：获取固件版本命令 • 更新 8.1.14 节：默认客户主密钥重置请求	1.01
2018-12-28	• 更新 6.0 节(软件设计)的格式 • 更新 7.0 节(主机编程 API)的格式 • 更新 6.1.3 节：认证 • 更新 6.1.5 节：蓝牙通讯协议 • 更新 6.1.5.1 节：卡片上电 • 更新 6.1.5.2 节：卡片下电 • 更新 6.1.5.3 节：查看卡片是否存在 • 更新 6.1.5.4 节：APDU 命令 • 新增 6.1.5.5 节：APDU2 命令 • 更新 6.1.5.6 节：直接命令 • 更新 6.1.6.2 节：RDR_to_SPH_AuthRsp1 • 更新 6.1.6.4 节：RDR_to_SPH_AuthRsp2 • 更新 6.1.6.5 节：SPH_to_RDR_DataReq • 更新 6.1.6.6 节：RDR_to_SPH_DataRsp • 更新 6.2.1.1 节：PC_to_RDR_IccPowerOn • 更新 6.2.1.4 节：PC_to_RDR_XfrBlock • 更新 6.2.2.1 节：RDR_to_PC_DataBlock • 更新 6.2 节：USB 通信协议 • 更新 7.1.1 节：获取序列号命令 • 更新 7.1.9 节：设置卡片重置模拟选项 • 更新 7.1.10 节：获取卡片重置模拟选项 • 更新 7.1.11 节：设置卡片响应时间间隔（Set Card Response Time Interval） • 更新 7.1.12 节：获取卡片响应时间间隔（Get Card Response Time Interval） • 删除 10.0 节：通过 PC_to_RDR_XfrBlock 访问其它命令 • 更新附录 A：错误代码 • 删除附录 A：支持的卡片类型 • 更新产品推广名称	1.02

目录

图目录

表目录

1.0. 简介

ACR3901T-W1 ACS 安全蓝牙™SIM 尺寸接触式智能卡读写器是计算机/移动设备与智能卡之间的通信接口。大多数情况下彼此采用不同命令和通信协议的各类智能卡不能直接通信，但是 ACR3901T-W1 可以为各种卡片建立统一的接口。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	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）[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] *除非另有不同说明，否则必须设置为零。*

2.0. 特性

- USB 全速接口
- 蓝牙接口
- 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）SIM 尺寸卡
    - 支持符合 T=0 或 T=1 协议的微处理器卡
    - 支持各类存储卡
    - 支持协议和参数选择（PPS）
    - 具有短路保护功能
    - 支持 AES-128 加密算法
- 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）
- 内置外设：
  - LED 指示灯
  - 按键
- 具有 USB 固件升级能力 [2]
- 支持 Android™ 4.3 及更高版本 [3]
- 支持 iOS 8.0 及更高版本 [4]
- 符合下列标准：
  - EN 60950/IEC 60950
  - ISO 7816
  - 蓝牙
  - PC/SC
  - CCID
  - CE
  - FCC
  - RoHS 2
  - REACH
  - Microsoft® WHQL

[2] *适用于 PC 连机模式。*
[3] *使用 ACS 定义的安卓库*
[4] *使用 ACS 定义的 iOS 库*

3.0. 支持的智能卡

3.1. MCU 卡

ACR3901T-W1 智能卡读写器符合 PC/SC 标准，支持 ISO 7816 A 类、B 类和 C 类（5 V、3 V、1.8 V）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 ACR3901T-W1 不支持该特定模式，则 ACR3901T-W1 会将卡片复位为协商模式。如果卡片不能被置为协商模式，ACR3901T-W1 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 ACR3901T-W1 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）。

对于上述参数的含义，请参照 ISO 7816-3 的规定。

3.2. 存储卡

ACR3901T-W1 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
- ‘104’型 EEPROM 不可重置标记计数卡，包括：
  - Infineon®：SLE4406、SLE4436、SLE5536 和 SLE6636
- 具有 416 位 EEPROM 智能存储空间以及内部 PIN 检查功能的存储卡，包括：
  - Infineon®：SLE4404
- 包含应用区域的逻辑加密卡，包括：
  - Atmel®：AT88SC101、AT88SC102 和 AT88SC1003

4.0. 系统框图

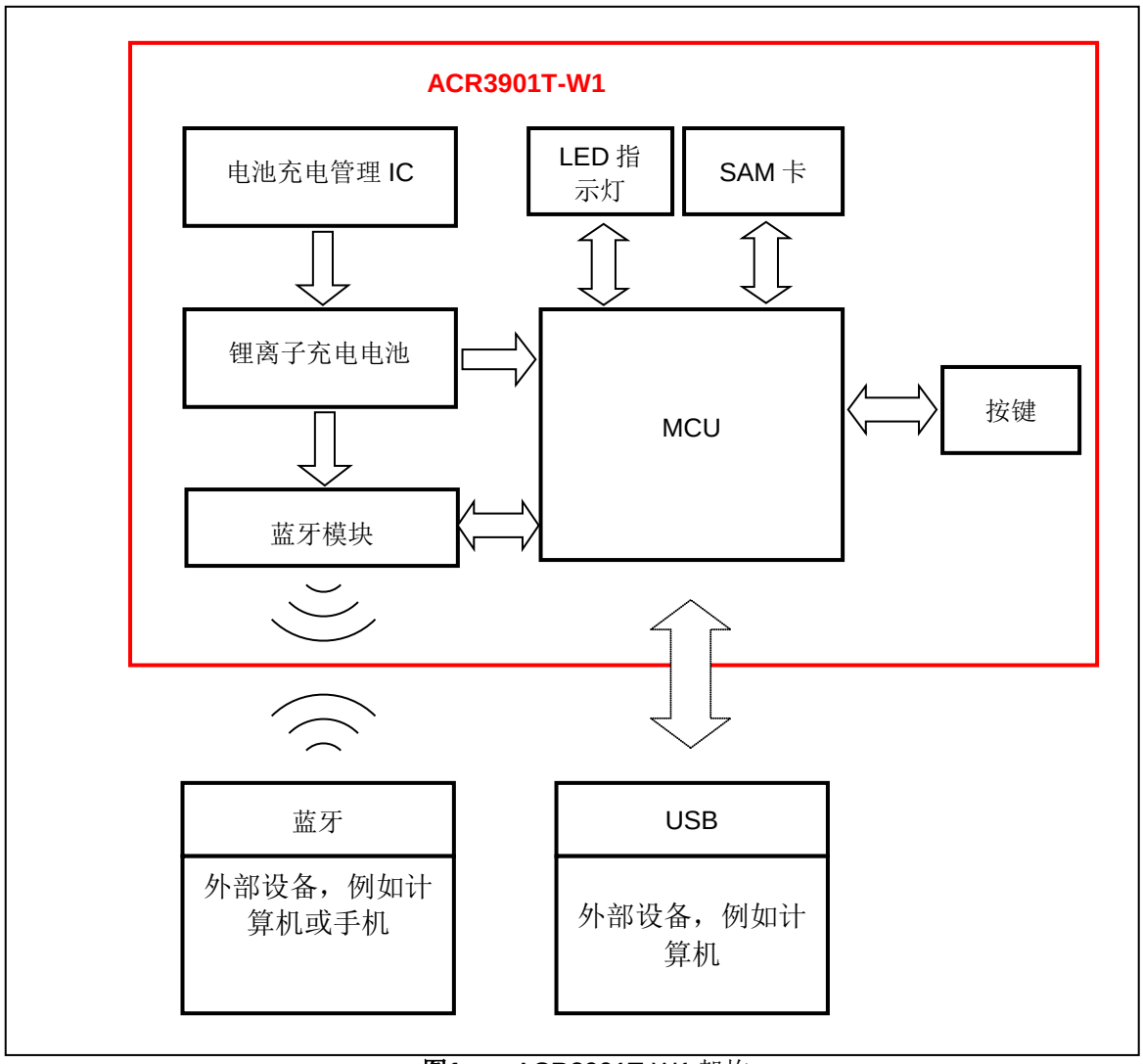

图1 ：ACR3901T-W1 架构

5.0. 硬件设计

5.1. 电池

ACR3901T-W1 使用容量 70 mAh 的锂离子充电电池。

5.1.1. 电池充电

ACR3901T-W1 电池没电时，可以在多种模式下充电：关机模式，USB 模式，蓝牙模式；前提条件是读写器连接了电源插座。

5.1.2. 电池寿命

电池寿命与设备使用情况相关。以下是各种工作条件下预估的电池寿命：

模式	预计电池寿命
工作模式	4 天* (1)
待机模式	100 天* (2)
关机模式	1 年

表2 ：预计电池寿命

注：*采用不同的智能卡也会导致结果不同。*

(1) *在蓝牙模式下，每天进行 10 次操作，每次操作一分钟。*

(2) *在蓝牙模式下，休眠时间设为 60 秒，每天唤醒一次。*

5.2. 蓝牙接口

ACR3901T-W1 使用低功耗蓝牙（LE）接口作为匹配计算机/移动设备的媒介。

5.3. USB 接口

ACR3901T-W1 可通过 Micro-USB 作为电池充电端口与计算机连接。另外利用此端口，还可以在计算机连接模式下操作 ACR3901T-W1。

5.3.1. 通信参数

ACR3901T-W1 通过符合 USB 2.0 规范的 USB 端口连接计算机。它采用 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-	ACR3901T-W1 和计算机之间以差分信号传输数据。
3	D+	ACR3901T-W1 和计算机之间以差分信号传输数据。
4	GND	参考电压等级

表3 ： USB 接口配线

5.3.2. 端点

ACR3901T-W1 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 ACR3901T-W1 的命令（数据包的大小为 64 字节）
批量输入（Bulk IN）	用于从 ACR3901T-W1 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 ACR3901T-W1 发送至主机的卡片状态消息（数据包的大小为 8 字节）

5.4. 用户接口

ACR3901T-W1 带两个按键和两盏 LED 指示灯（一盏单色：红色；一盏双色：绿色和蓝色）。

5.4.1. 按键

ACR3901T-W1 提供三种模式：USB、关机和蓝牙。用户一次能以一种模式作为数据传输接口。

- 按键（正面）- 开机、关机或唤醒读写器；模拟 SAM 卡移出操作。
  - 按下按键，读写器开机，默认为蓝牙模式。连接计算机后，自动切换至 USB 模式。长按会关闭设备。
- 按键（背面） - 复位读写器

5.4.2. LED 状态指示灯

ACR3901T-W1 有两盏 LED 指示灯（一盏单色：红色；一盏双色：绿色和蓝色）用以显示不同的操作状态，包括：

- **红色 LED -** 电池状态
- **蓝色 LED -** 蓝牙连接模式下卡片和读写器的状态
- **绿色 LED -** USB 连接模式下卡片和读写器的状态

颜色	LED 操作	状态
红色	长亮	电池正在充电（电池充满后会关闭）
	缓慢闪烁 （1 秒/闪烁）	电池需要充电
蓝色	快速-缓慢闪烁 （快速：250 毫秒/闪烁，慢速：500 毫秒/闪烁）	准备就绪，可以匹配蓝牙设备
	缓慢闪烁 （2 秒/闪烁）	已连接蓝牙设备，无卡片操作
	快速闪烁	读写器和移动设备之间正在传输数据
	长亮	卡片已连接并上电
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	读写器和 PC 之间正在传输数据
	长亮	卡片已连接并上电

表4 ： LED 状态指示灯

注：*红色、蓝色和绿色 LED 关闭时，表示读写器已关机。读写器接收到蓝牙模块发送的严重错误码时，蓝色和绿色 LED 灯亮 1 秒钟，然后灯灭。*

5.5. 智能卡接口

ACR3901T-W1 与插入的智能卡之间的接口符合 ISO 7816-3 标准协议，并进行了某些限制或提升来增强 ACR3901T-W1 的实用功能。

5.5.1. 智能卡电源 VCC（C1）

插入的智能卡的电流消耗不得大于 50 mA。

5.5.2. 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，ACR3901T-W1 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

5.5.3. 卡片类型选择

每次激活插入的卡片前，处于控制地位的计算机都要向 ACR3901T-W1 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，读写器才可以通过协议与参数选择（PPS）接受并执行这样的选择。如果 MCU 卡仅支持 T=0 和 T=1 协议中的一种，则读卡器会自动采用该协议类型，而不管应用程序选择哪一种。

5.5.4. 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。

5.5.5. 卡片插拔保护

ACR3901T-W1 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 ACR3901T-W1 与卡之间的信号线路会立即取消激活。但是作为惯例，只应在断电后才从读卡器移出卡片，这样可以避免电气损伤。

注：*ACR3901T-W1 不会主动给插入的卡片上电。必须由主控计算机发送明确命令给读卡器执行此操作。*

6.0. 软件设计

6. 1. 蓝牙通信协议

6.1.1. 蓝牙连接程序流程

以下是蓝牙连接的程序流程：

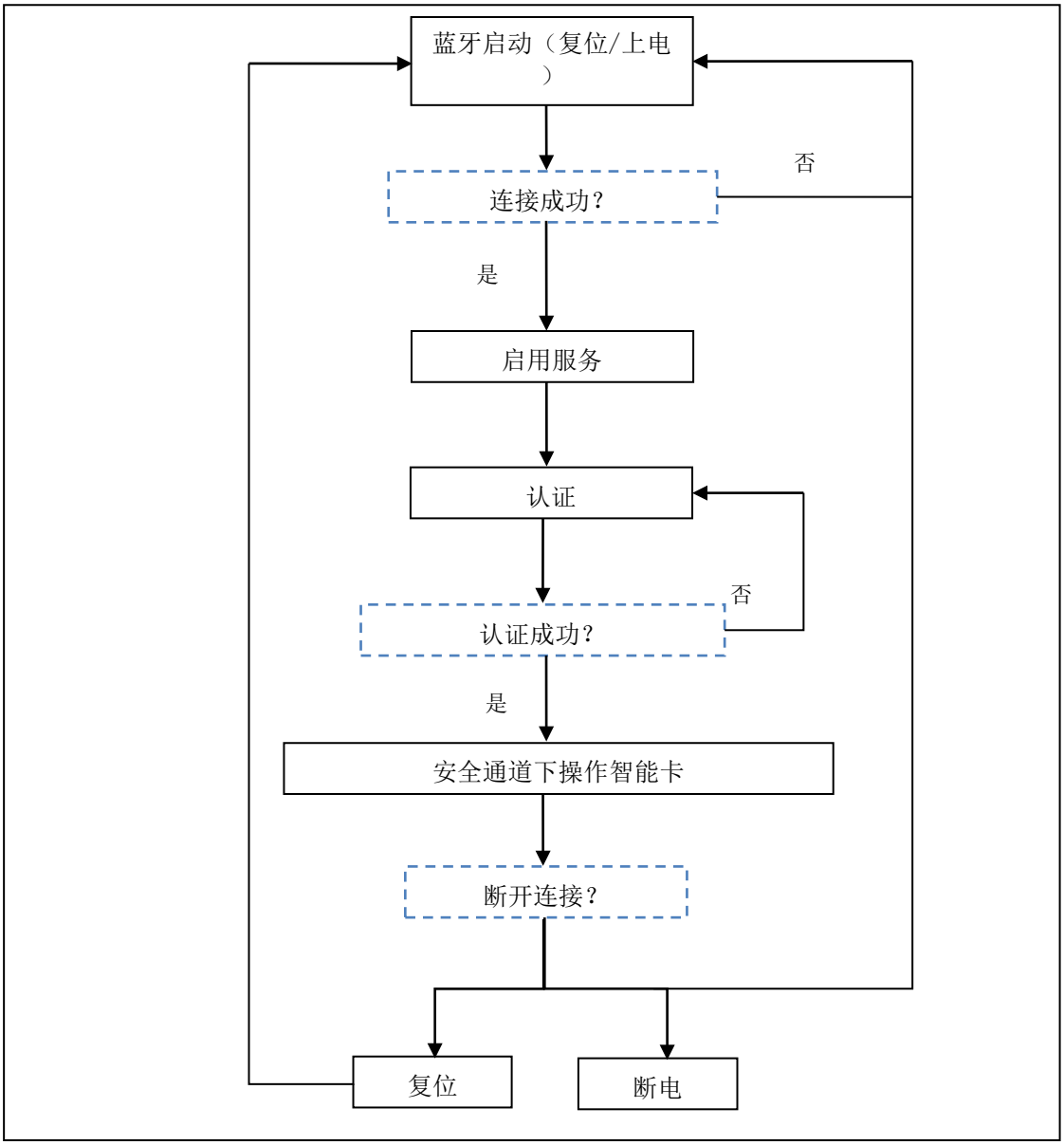

图2 ： 蓝牙连接流程

6.1.2. 配置文件选择

ACR3901T-W1 是一款使用蓝牙技术作为数据传输接口的智能卡读写器。它采用了能通过三条通道进行命令通信的定制服务：一条通道用于命令请求，一条通道用于命令响应，而第三条通道将卡片和休眠模式状态通知给配对设备。

另外，当读写器蓝在蓝牙模式下工作时，当前的电池状态很重要，因而客制化的电池服务可以将当前电池状态告知配对设备。当电池状态发生变化，读写器也将通过指定通道通知配对设备。简而言之，电池电量级别分为三类，下表列出了各个电池级别和相应的返回值：

状态	电压	返回值
电量充足	≥ 3.3 V	FEh
电量低	<3.3 V 并且 ≥ 2.9 V	FFh/FEh/00h 之外的值
电量耗尽	<2.9 V	00h
USB 模式		FFh

卡片状态发生任意变化或者读写器进入休眠模式时，卡片状态通知服务将通知配对的服务。以下是状态列表和相应返回值：

状态	返回值
没有插入卡片	50 02h
卡片已插入	50 03h
读写器已进入休眠模式	50 04h

最后，为了给用户提供更多读写器信息，增加了客制化设备信息服务。可以人工读取或者由应用请求读取，该特性包括**厂商名称**、**固件版本号**、**型号**、和**序列号**。

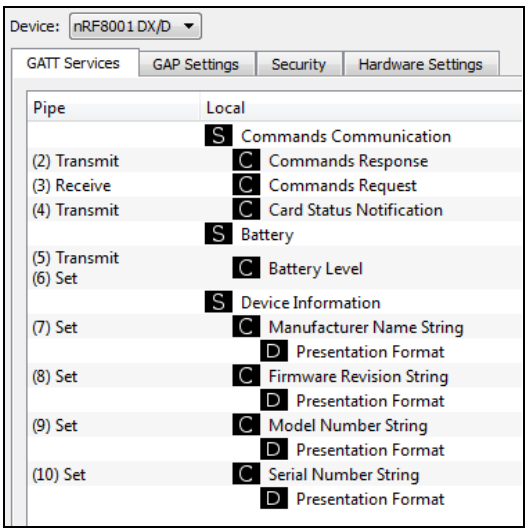

图3 ：nRFgo Studio GATT 配置页面

nRFgo-Studio 配置增加了一项服务，服务总数达到 10 项：

#define PIPE_GAP_DEVICE_NAME_SET 1

#define PIPE_COMMANDS_COMMUNICATION_COMMANDS_RESPONSE_TX 2

#define PIPE_COMMANDS_COMMUNICATION_COMMANDS_REQUEST_RX 3

#define PIPE_COMMANDS_COMMUNICATION_CARD_STATUS_NOTIFICATION_TX 4

#define PIPE_BATTERY_BATTERY_LEVEL_TX 5

#define PIPE_BATTERY_BATTERY_LEVEL_SET 6

#define PIPE_DEVICE_INFORMATION_MANUFACTURER_NAME_STRING_SET 7

#define PIPE_DEVICE_INFORMATION_FIRMWARE_REVISION_STRING_SET 8

#define PIPE_DEVICE_INFORMATION_MODEL_NUMBER_STRING_SET 9

#define PIPE_DEVICE_INFORMATION_SERIAL_NUMBER_STRING_SET 10

#define NUMBER_OF_PIPES 10

#define PIPE_GAP_DEVICE_NAME_SET 可用于控制器在运行时更改设备名称。蓝牙模式下，公告名称格式为“ACR3901T-W1XXXXXXX”，其中“XXXXXXX”是读写器序列号的最后 7 个字节。

为了使公告名称成为“ACR3901T-W1XXXXXXX”，首先应执行蓝牙模式启动操作。

蓝牙模式启动：

1. 设置（06h）将配置上传到蓝牙模块。
2. 使用通道 1 将设备名称格式设置为“ACR3901T-W1XXXXXXX”。
   (PIPE_GAP_DEVICE_NAME_SET)
3. 连接（0Fh）。
4. 公告。

属性名称	UUID	句柄
DeviceName	2A00	03h
Send (Reader → Paired device)	8002	0Bh
Receive (Paired device →Reader)	8003	0Eh
CardStatus	8004	10h
BatteryLevel	2A19	14h
Manufacturer	2A29	18h
FW_Version	2A26	1Bh
ModelNumber	2A24	1Eh
SerialNumber	2A25	21h

表5 ： ACR3901T-W1 服务句柄和 UUID 消息列表

6.1.3. 认证

向 ACR3901T-W1 加载敏感数据之前，数据处理服务器必须通过 ACR3901T-W1 的认证，然后才有权修改读写器内部的安全数据。ACR3901T-W1 采用相互认证的方法。

为了更好的进行说明，请参考下图（图中省去了桥接设备，以便更简单明了地进行说明）：

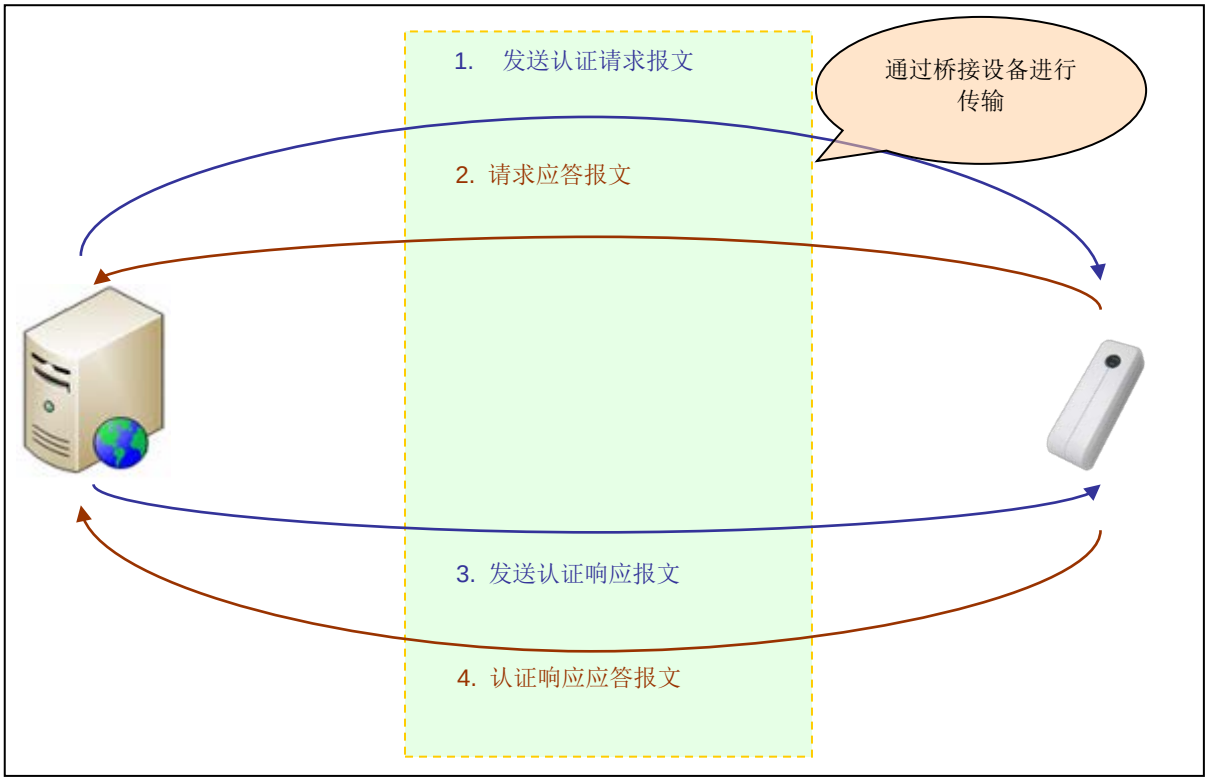

图4 ： 认证步骤

默认的客户主密钥（十六进制）： **FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF**

注意：*如果认证密钥输入错误超过六次，读写器将被锁定，无法使用。*

有关更多详细信息，您可以联系 ACS 销售代表

6.1.4. 帧格式

6.1.4.1. 蓝牙帧格式

HID 帧	长度（字节）	说明
标识符	1	命令
长度	2	长度 {数据包+校验和}
数据包	0-N	数据
校验和	1	XOR {标识符,长度,数据包}

表6 ：蓝牙帧格式

帧格式应为：

标识符 + LEN1 + LEN2 + N-字节数据包 + 校验和

如果命令总长度（包括标识符、长度和数据包）大于 20 字节，读写器或匹配的设备将自动将其划分为多个帧。

数据校验和用于检测数据无线传输过程中可能引发的错误。如需计算数据校验和：XOR {标识符,长度,数据包}。

例如：62010063 => 校验和 = 63h

6.1.4.2. 相互认证后的蓝牙帧格式

引入相互认证是为了避免中间人通过蓝牙通信通道进行攻击。相互认证成功后，**表 6** 内的蓝牙帧格式将被加密并封装为带 1 字节头部， 2 字节 Len，和 1 个校验字节。相互认证后蓝牙帧格式的结构如下：

头部 + Len + (标识符 + 长度 + 数据包 + 校验和)* + 校验字节

注意：*采用 AES-128 CBC 加密模式通过过程密钥对每 16 字节数据进行加密。AES-128 CBC 加密模式下，初始向量是 16 字节（00h）.*

<table>
<tr><th>HID 帧</th><th>长度（字节）</th><th colspan="2">说明</th></tr>
<tr><td>头部字节</td><td>1</td><td colspan="2">值：72h / 22h</td></tr>
<tr><td>Len</td><td>2</td><td colspan="2">长度 {标识符 + 长度 + 数据包 + 校验和 + 校验 + 终止字节}</td></tr>
<tr><td>标识符</td><td>1</td><td>命令</td><td rowspan="4">蓝牙帧格式的加密数据；
该部分的最终数据长度是 16*N 字节 (N>0)</td></tr>
<tr><td>长度</td><td>2</td><td>长度 {数据包+校验和}</td></tr>
<tr><td>数据包</td><td>0-N</td><td>数据</td></tr>
<tr><td>校验和</td><td>1</td><td>XOR {标识符,长度,数据包}。</td></tr>
<tr><td>校验字节</td><td>1</td><td colspan="2">XOR {头部, Len, 加密的(标识符, 长度, 数据包, 校验和)}</td></tr>
</table>

表7 ：相互认证后的加密蓝牙帧格式

6.1.5. 蓝牙通信协议

ACR3901T-W1 采用预定义协议，通过蓝牙接口与匹配的设备进行通信。协议类似于 CCID 命令通道和响应通道的格式。

命令	支持模式	发送方	说明
62h	已认证	配对设备	ICC 上电
63h	已认证	配对设备	ICC 下电（ICC Power Off）
65h	已认证	配对设备	查看卡片是否存在（Get Card Presence）
6Fh	已认证	配对设备	交换 APDU（Exchange APDU）
67h	已认证	配对设备	交换 APDU2（Exchange APDU2）
61h	已认证	配对设备	设置参数
6Bh	已认证	配对设备	外部命令
70h	已连接/已认证	配对设备	SPH_to_RDR_ReqAuth*
71h	已连接/已认证	配对设备	SPH_to_RDR_AuthRsp*

表8 ：命令码总表

命令	支持模式	发送方	说明
12h	已认证	读写器	ICC Power On 命令的响应
13h	已认证	读写器	ICC Power Off 命令的响应
14h	已认证	读写器	Get Card Presence 命令的响应
11h	已认证	读写器	Exchange APDU 命令的响应
17h	已认证	读写器	Exchange APDU2 命令的响应
16h	已认证	读写器	对设置参数做出应答
15h	已认证	读写器	外部命令的响应
20h	已连接/已认证	读写器	RDR_to_SPH_AuthRsp1*
21h	已连接/已认证	读写器	RDR_to_SPH_AuthRsp2*

表9 ：命令码总表

****注：*** *这些命令/响应码是相互认证中使用的通信代码。*

6.1.5.1. 卡片上电（Card Power On）

此命令发送上电请求给读写器。

命令格式

偏移	数据域	大小	值	说明
0	*bMessageType*	1	**62h**	-
1	*LEN1 LEN2 (wLength)*	2	0100h	表示此消息中从下一字段开始的其它字节的数量，并且表示为两个字节，LEN1 是 LSB，而 LEN2 是 MSB。
3	*CSUM (wChecksum)*	1	63h	CSUM 表示命令中所有字节的 XOR 值。

响应数据格式

偏移	数据域	大小	值	说明
0	*bMessageType*	1	12h	-
1	*LEN1 LEN2 (wLength)*	2		表示此消息中从下一字段开始的其它字节的数量，并且表示为两个字节长，LEN1 是 LSB，而 LEN2 是 MSB。
3	*N 字节的 ATR*	N	-	卡片复位应答
3+N	*CSUM (wChecksum)*	1	-	CSUM 表示命令中所有字节的 XOR 值。

响应数据格式（错误）

偏移	数据域	大小	值	说明
0	*bMessageType*	1	**92h**	-
1	*LEN1 LEN2 (wLength)*	2	0200h	**此消息中从下一字段开始的□外字□的数量。用两个字□表示。**LEN1 是 LSB，LEN2 是 MSB。
3	*Error Code (bErrorCode)*	1	-	错误代码。请参考附录 A。
4	*CSUM (wChecksum)*	1	-	CSUM 表示命令中所有字节的 XOR 值。

例如：

请求 = 62 01 00 63

响应 = 12 14 00 3B BE 11 00 00 41 01 38 00 00 00 00 12 34 56 78 01
90 00 73

ATR = 3B BE 11 00 00 41 01 38 00 00 00 00 12 34 56 78 01

6.1.5.2. 卡片下电（Card Power Off）

此命令发送下电请求给读写器。

命令格式

偏移	数据域	大小	值	说明
0	*bMessageType*	1	**63h**	-
1	*LEN1 LEN2 (wLength)*	2	0100h	表示此消息中从下一字段开始的其它字节的数量，表示为两个字节长，LEN1 是 LSB，而 LEN2 是 MSB。
3	*CSUM (wChecksum)*	1	62h	CSUM 表示命令中所有字节的 XOR 值。

响应数据格式

偏移	数据域	大小	值	说明
0	*bMessageType*	1	**13h**	-
1	*LEN1 LEN2 (wLength)*	2	0100h	表示此消息中从下一字段开始的其它字节的数量，表示为两个字节长，LEN1 是 LSB，而 LEN2 是 MSB。
3	*CSUM (wChecksum)*	1	12h	CSUM 表示命令中所有字节的 XOR 值。

响应数据格式（错误）

偏移	数据域	大小	值	说明
0	*bMessageType*	1	**93h**	-
1	*LEN1 LEN2 (wLength)*	2	0200h	此消息中从下一字段开始的□外字□的数量。用两个字□表示。LEN1 是 LSB，LEN2 是 MSB。
3	*Error Code (bErrorCode)*	1	-	错误代码。请参考附录 A。
4	*CSUM (wChecksum)*	1	-	CSUM 表示命令中所有字节的 XOR 值。

例如：

请求 = 62 01 00 62

响应 = 13 01 00 12

6.1.5.3. 查看卡片是否存在（Get Card Presence）

此命令查看是否插入卡片。

命令格式

偏移	数据域	大小	值	说明
0	*bMessageType*	1	**65h**	-
1	*LEN1 LEN2 (wLength)*	2	0100h	表示此消息中从下一字段开始的其它字节的数量，表示为两个字节长，LEN1 是 LSB，而 LEN2 是 MSB。
3	*CSUM (wChecksum)*	1	64h	CSUM 表示命令中所有字节的 XOR 值。

响应数据格式

偏移	数据域	大小	值	说明
0	*bMessageType*	1	14h	-
1	*LEN1 LEN2 (wLength)*	2	0200h	表示此消息中从下一字段开始的其它字节的数量，表示为两个字节长，LEN1 是 LSB，而 LEN2 是 MSB。
3	*STA*	1	-	卡片状态： 00 = 未知状态 01 = 没有卡片 02 = 有卡但是未激活 03 = 有卡并且已激活
4	*CSUM (wChecksum)*	1	-	CSUM 表示命令中所有字节的 XOR 值。

响应数据格式（错误）

偏移	数据域	大小	值	说明
0	*bMessageType*	1	**94h**	-
1	*LEN1 LEN2 (wLength)*	2	0200h	**此消息中从下一字段开始的□外字□的数量。用两个字□表示。LEN1 是 LSB，LEN2 是 MSB。**
3	*Error Code (bErrorCode)*	1	-	错误代码。请参考附录 A。
4	*CSUM (wChecksum)*	1	-	CSUM 表示命令中所有字节的 XOR 值。

例如：

请求 = 65 01 00 64

响应 = 14 02 00 03 15

6.1.5.4. APDU 命令（APDU COMMAND）

此命令用于发送 APDU 命令给读写器。

命令格式

偏移	数据域	大小	值	说明
0	*bMessageType*	1	**6Fh**	-
1	*LEN1 LEN2 (wLength)*	2	-	表示此消息中从下一字段开始的其它字节的数量，表示为两个字节长，LEN1 是 LSB，而 LEN2 是 MSB。
3	*APDU CMD*	N	-	APDU 命令
3+N	*CSUM (wChecksum)*	1	-	CSUM 表示命令中所有字节的 XOR 值。

响应数据格式

偏移	数据域	大小	值	说明
0	*bMessageType*	1	**11h**	-
1	*LEN1 LEN2 (wLength)*	2	-	表示此消息中从下一字段开始的其它字节的数量，表示为两个字节长，LEN1 是 LSB，而 LEN2 是 MSB。
3	*APDU 响应*	N	-	APDU 结构数据
3+N	*CSUM (wChecksum)*	1	-	CSUM 表示命令中所有字节的 XOR 值。

响应数据格式（错误）

偏移	数据域	大小	值	说明
0	*bMessageType*	1	**91h**	-
1	*LEN1 LEN2 (wLength)*	2	0200h	此消息中从下一字段开始的□外字□的数量。用两个字□表示。LEN1 是 LSB，LEN2 是 MSB。
3	*Error Code (bErrorCode)*	1	-	错误代码。请参考附录 A。
4	*CSUM (wChecksum)*	1	-	CSUM 表示命令中所有字节的 XOR 值。

例如：

请求 = 6F 06 00 80 84 00 00 08 65

响应 = 11 0B 00 C1 7A 3B AA D6 5A FA CE 90 00 18

6.1.5.5. APDU2 命令（固件 v6.01 及以上版本）

此命令用于向读写器发送一条 APDU 命令，该命令支持拓展的 APDU。

命令格式

偏移	数据域		大小	值	说明
0	*bMessageType*		1	**67h**	-
1	*LEN1 LEN2 (wLength)*		2	-	此消息中从下一字段开始的□外字□的数量。用两个字□表示。LEN1 是 LSB，LEN2 是 MSB。（最大长度为 263）
3	*数据*	*Param*	1	-	参数： 短 APDU 级 00h –默认 扩展 APDU 级 00h – 命令 APDU 在本命令中开始和结束。 01h – 命令 APDU 在本命令中开始，并在下一个 APDU 命令中继续。 02h – 数据字段继续传递命令 APDU 并结束该 APDU 命令。 03h – 数据字段继续传递命令 APDU，后面跟随另外一个数据块。 10h – 空的数据字段，下一个响应会继续传递响应 APDU
4		*APDU CMD*	N	-	APDU 命令 （最大长度为 261）
4+N	*CSUM (wChecksum)*		1	-	CSUM 表示数据中所有字节的 XOR 值。

响应数据格式

偏移	数据域		大小	值	说明
0	*bMessageType*		1	**17h**	-
1	*LEN1 LEN2 (wLength)*		2	-	此消息中从下一字段开始的□外字□的数量。用两个字□表示。LEN1 是 LSB，LEN2 是 MSB。
3	*数据*	*Param*	1	-	参数： 短 APDU 级 00h – 默认 扩展 APDU 级 00h – 响应 APDU 在本命令中开始和结束。 01h – 响应 APDU 在本命令中开始，并会继续。 02h – 数据字段继续传递响应 APDU 并结束该响应 APDU。 03h – 数据字段继续传递响应 APDU，后面跟随另外一个数据块。 10h – 空的数据字段，下一个命令会继续传递命令 APDU
4		*APDU RSP*	N	-	APDU 响应
4+N	*CSUM (wChecksum)*		1	-	CSUM 表示数据中所有字节的 XOR 值。

响应数据格式（WTX）

偏移	数据域	大小	值	说明
0	*bMessageType*	1	**18h**	-
1	*LEN1 LEN2 (wLength)*	2	0300h	此消息中从下一字段开始的□外字□的数量。用两个字□表示。LEN1 是 LSB，LEN2 是 MSB。
3	*STA*	1	-	卡片状态： 00 = 未知状态 01 = 卡片不存在 02 = 卡片存在，未激活 03 = 卡片存在，已激活
4	*WTXM*	1	-	等待时间延迟乘数
5	*CSUM (wChecksum)*	1	-	CSUM 表示命令中所有字节的 XOR 值

响应数据格式（错误）

偏移	数据域	大小	值	说明
0	*bMessageType*	1	**97h**	-
1	*LEN1 LEN2 (wLength)*	2	0200h	此消息中从下一字段开始的□外字□的数量。用两个字□表示。LEN1 是 LSB，LEN2 是 MSB。
3	*Error Code (bErrorCode)*	1	-	错误代码，请参考附录 A
4	*CSUM (wChecksum)*	1	-	CSUM 表示命令中所有字节的 XOR 值

例如：

向卡片发送 600 字节的数据

1. Command = 67 07 01 01 (261 字节的数据) 校验和
   Response = 17 02 00 10 校验和
2. Command = 67 07 01 03 (261 字节的数据) 校验和
   Response = 17 02 00 10 校验和
3. Command = 67 50 00 02 (78 字节的数据) 校验和
   Response = 17 04 00 00 90 00 校验和

Receives 600 bytes data from the card

1. Command = 67 09 00 00 00 B0 87 00 00 02 58 校验和
   Response = 17 02 01 01 (256 字节的数据) 校验和
2. Command = 67 02 00 10 校验和
   Response = 17 02 01 03 (256 字节的数据) 校验和
3. Command = 67 02 00 10 checksum
   Response = 17 5C 00 02 (88 字节的数据) 90 00 校验和

6.1.5.6. 直接命令（Escape Commands）

此命令允许访问读写器的扩展功能。

命令格式

偏移	数据域		大小	值	说明
0	*bMessageType*		1	**6Bh**	直接命令的头部
1	*LEN1 LEN2 (wLength)*		2	-	表示此消息中从下一字段开始的其它字节的数量，表示为两个字节长，LEN1 是 LSB，而 LEN2 是 MSB。
3	*abData1*	*CommandCode*	1	-	命令头
4		*Len (CommandLength)*	1	-	表示此消息中从下一字段开始的其它字节的数量，并且表示为一个字节长。
5		*数据*	N	-	0 =< N <= 255
5+N	*CSUM (wChecksum)*		1	-	CSUM 表示命令中所有字节的 XOR 值。

响应数据格式

偏移	数据域		大小	值	说明
0	*bMessageType*		1	**15h**	直接命令响应的头部
1	*LEN1 LEN2 (wLength)*		2	-	表示此消息中从下一字段开始的其它字节的数量，表示为两个字节长，LEN1 是 LSB，而 LEN2 是 MSB。
3	*abData2*	*ResponseCode*	1	-	响应的头部
4		*Len (CommandLength)*	1	-	表示此消息中从下一字段开始的其它字节的数量，并且表示为一个字节长。
5		*数据*	N	-	0 =< N <= 255
5+N	*CSUM (wChecksum)*		1	-	CSUM 表示命令中所有字节的 XOR 值。

响应数据格式（错误）

偏移	数据域	大小	值	说明
0	*bMessageType*	1	**95h**	-
1	*LEN1 LEN2 (wLength)*	2	0200h	此消息中从下一字段开始的□外字□的数量。用两个字□表示。LEN1 是 LSB，LEN2 是 MSB。

偏移	数据域	大小	值	说明
3	*Error Code (bErrorCode)*	1	-	错误代码。请参考附录 A。
4	*CSUM (wChecksum)*	1	-	CSUM 表示命令中所有字节的 XOR 值。

6.1.5.7. 卡片设置参数（Card Set Parameters）

此命令用于上电后修改所插入卡片的参数。

命令格式

偏移	数据域		大小	值	说明
0	*bMessageType*		1	**61h**	-
1	*LEN1 LEN2 (wLength)*		2	-	表示此消息中从下一字段开始的其它字节的数量，表示为两个字节长，LEN1 是 LSB，而 LEN2 是 MSB。
3	*abData1*	*ProtocolNum*	1	-	卡片协议数据结构： 00h = T=0 协议的结构 01h = T=1 协议的结构
4		*ProtocolDataStructure*	N	-	协议数据结构
4+N	*CSUM (wChecksum)*		1	-	CSUM 表示命令中所有字节的 XOR 值。

响应数据格式

偏移	数据域		大小	值	说明
0	*bMessageType*		1	**16h**	直接命令响应的头部
1	*LEN1 LEN2 (wLength)*		2	-	表示此消息中从下一字段开始的其它字节的数量，表示为两个字节长，LEN1 是 LSB，而 LEN2 是 MSB。
3	*abData2*	*ProtocolNum*	1	-	卡片协议数据结构： 00h = T=0 协议的结构 01h = T=1 协议的结构
4		*ProtocolDataStructure*	N	-	协议数据结构
4+N	*CSUM (wChecksum)*		1	-	CSUM 表示命令中所有字节的 XOR 值。

T=0 协议的协议数据结构（ProtocolNum = 0，wLength = 0700h）

偏移	数据域	大小	值	说明
4	*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
5	*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。*
6	*bGuardTimeT0*	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。
7	*bWaitingIntegerT0*	1	-	T=0 时 WI 用于定义 WWT
8	*bClockStop*	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（ProtocolNum = 1，wLength = 0900h）

偏移	数据域	大小	值	说明
4	*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
5	*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。*
6	*BGuardTimeT1*	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。
7	*BwaitingIntegerT1*	1	-	B7-4 = BWI 值 0-9 有效 B3-0 = CWI 值 0-Fh 有效
8	*bClockStop*	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
9	*bIFSC*	1	-	商定的 IFSC 的大小
10	*bNadValue*	1	00h	只支持 NAD = 00h

例如：（T0 协议）

请求 = 61 07 00 00 11 00 00 0A 00 7D

响应 = 16 07 00 00 11 00 00 0A 00 0A

例如：（T1 协议）

请求 = 61 09 00 01 96 10 00 45 00 FE 00 54

响应 = 16 09 00 01 96 10 00 45 00 FE 00 23

6.1.6. 相互认证和加密协议

蓝牙模式下，相互认证成功后将对通信协议进行加密和传输。

命令	支持模式	发送方	说明
70h	已连接	配对设备	SPH_to_RDR_ReqAuth
71h	已连接	配对设备	SPH_to_RDR_AuthRsp
72h	已认证	配对设备	SPH_to_RDR_DataReq
20h	已连接	读写器	RDR_to_SPH_AuthRsp1
21h	已连接	读写器	RDR_to_SPH_AuthRsp2
22h	已认证	读写器	RDR_to_SPH_DataRsp

表10 ：相互认证命令概要

6.1.6.1. SPH_to_RDR_ReqAuth

此命令请求 ACR3901T-W1 对匹配的密钥生成设备进行认证。认证成功后，可以通过匹配的密钥生成设备修改客户主密钥。

有关认证流程的更多信息，请参考**认证**。

偏移	数据域	大小	值	说明	加密
0	*bMessageType*	1	**70h**	-	否
1	*LEN1 LEN2 (wLength)*	2	0100h	表示此消息中从下一字段开始的其它字节的数量，表示为两个字节长，LEN1 是 LSB，而 LEN2 是 MSB。	
3	*wChecksum*	1	71h	CSUM 表示命令中所有字节的 XOR 值。	

如果接收的命令消息无误，将收到响应 *RDR_to_SPH_AuthRsp1*。

6.1.6.2. RDR_to_SPH_AuthRsp1

此命令是由配对设备发送给 *SPH_to_RDR_ReqAuth* 的响应。

更多信息请参阅**认证**。

偏移	数据域	大小	值	说明	加密
0	*bMessageType*	1	**20h**	-	否
1	*LEN1 LEN2 (wLength)*	2	1100h	表示此消息中从下一字段开始的其它字节的数量，表示为两个字节长，LEN1 是 LSB，而 LEN2 是 MSB。	否
3	*abRndNum*	16	-	abRndNum[0:15] – 16 字节随机数 所有的 16 字节数据必须使用当前存储在 ACR3901T-W1 内的客户主密钥进行加密。	是
19	*wChecksum*	1	-	CSUM 表示命令中所有字节的 XOR 值。	否

6.1.6.3. SPH_to_RDR_AuthRsp

此命令属于认证流程的第二阶段。设备发送 *SPH_to_RDR_ReqAuth* 命令给 ACR3901T-W1 之后，如果命令无误，读写器将返回 *RDR_to_SPH_AuthRsp1* 消息。

RDR_to_SPH_AuthRsp1 包含一系列通过客户主密钥加密的 16 字节随机数。匹配的密钥生成设备应当使用正确的客户主密钥进行解密，并将其添加到 16 字节随机数的末尾。然后使用客户主密钥解密全部 32 字节的随机数，使用此命令将结果返回给 ACR3901T-W1，以成功完成认证。

有关认证流程的更多信息，请参考**认证**。

偏移	数据域	大小	值	说明	加密
0	*bMessageType*	1	**71h**	-	否
1	*LEN1 LEN2 (wLength)*	2	2100h	表示此消息中从下一字段开始的其它字节的数量，表示为两个字节长，LEN1 是 LSB，而 LEN2 是 MSB。	否
3	*abAuthData*	32	-	abAuthData[0:15] – 数据□理服□器生成的 16 字□随机数。 abAuthData[16:31] – 接收自 ACR3901T-W1 的 16 字□解密随机数 全部 32 字□数据将在 AES128 CBC 加密模式下通□客□主密□□行解密□理。	是
35	*wChecksum*	1	-	CSUM 表示命令中所有字节的 XOR 值。	否

如果命令消息无误并且 ACR3901T-W1 返回的随机数也是正确的，响应为 *RDR_to_SPH_AuthRsp2*。

6.1.6.4. RDR_to_SPH_AuthRsp2

此命令是由配对设备发送给 *SPH_to_RDR_AuthRsp* 的响应。

更多信息请参阅**认证**。

偏移	数据域	大小	值	说明	加密
0	*bMessageType*	1	**21h**	-	否
1	*LEN1 LEN2 (wLength)*	2	1100h	表示此消息中从下一字段开始的其它字节的数量，表示为两个字节长，LEN1 是 LSB，而 LEN2 是 MSB。	否
3	*abRndNum*	16		abRndNum[0:15] – 接收自数据处理服务器的 16 字节随机数。 所有的 16 字节数据都必须使用当前存储在 ACR3901T-W1 内的客户主密钥进行加密。	是
19	*wCheckSum*	1		CSUM 表示命令中所有字节的 XOR 值。	否

6.1.6.5. SPH_to_RDR_DataReq

此命令是由配对设备在相互认证流程后发送给 ACR3901T-W1 的。

蓝牙模式下，相互认证成功后将对**卡片上电（Card Power On）**至**卡片设置参数(Card Set Parameters)**中的通信协议进行加密和传输。

偏移	数据域	大小	值	说明	加密
0	*bMessageType*	1	**72h**	-	否
1	*LEN1 LEN2 (wLength)*	2	-	表示此消息中从下一字段开始的其它字节的数量，表示为两个字节长，LEN1 是 LSB，而 LEN2 是 MSB。	否
3	*abEncryptedData*	N*16	-	每 16 字节数据都将在 AES128 CBC 加密模式下通过过程密钥进行加密操作。	是
3 + N*16	*wChecksum*	1	-	CSUM 表示命令中所有字节的 XOR 值。	否

abEncryptedData 是加密后的（□□符 + □度 + 数据包 + 校□和）数据，□度□ N*16 字□，数据中的每个字□都会在 AES128 CBC 加密模式下使用相互□□生成的□程密□□行加密□理。

AES-128 CBC 加密模式下，初始向量为 16 字节 00h

如果原始数据长度小于 N*16，加密前直接在末尾添加 FFh，使其长度变成 16*N 字节。

HID 帧	长度（字节）	说明	
标识符	1	命令	真实数据是用 abEncryptedData 解密的，并删除虚拟数据。
长度	2	长度 {数据包+校验和}	
数据包	0-N	数据	
校验和	1	XOR {标识符,长度,数据包}	

例如：

相互认证成功后，配对设备发送上电命令给读写器，此命令如下：

72 11 00 XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX

其中：

命令头：72

上电命令的加密数据（16 字节）：XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX

如果接收的命令消息无误，将收到响应 RDR_to_SPH_DataRsp。

abEncryptedData 是通信□□的加密数据。每 16 字□数据在 AES-128 CBC 加密模式下通□□程密□□行加密。

6.1.6.6. RDR_to_SPH_DataRsp

相互认证成功后，读写器发送此命令给配对设备。

蓝牙模式下，相互认证成功后将对**卡片上电（Card Power On）**至**卡片设置参数(Card Set Parameters)**中的通信协议进行加密和传输。

偏移	数据域	大小	值	说明	加密
0	*bMessageType*	1	**22h**	-	否
1	*LEN1 LEN2 (wLength)*	2	-	表示此消息中从下一字段开始的其它字节的数量，并且表示为两个字节长，LEN1 是 LSB，而 LEN2 是 MSB	否
3	*abEncryptedData*	N*16	-	每 16 字□数据都将在 AES128 CBC 加密模式下通□□程密□□行加密□理。	是
3 + N*16	*wChecksum*	1	-	CSUM 表示命令中所有字节的 XOR 值。	否

6.2. USB 通信协议

ACR3901T-W1 应当通过 USB 连接与主机(host)端对接。 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。

ACR3901T-W1 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 端点发出。发送至 ACR3901T-W1 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**经由 BULK-IN 端点发出。所有发送至 ACR3901T-W1 的命令都必须同步发送（例如：对于 ACR3901T-W1 来说，*bMaxCCIDBusySlots* 等同于 01h）。

ACR3901T-W1 支持的 CCID 特性见其类别描述符：

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	-	CCID 以二进制编码的十进制指定的版本号码
4	*bMaxSlotIndex*	1	-	ACR3901T-W1 有一个卡槽。
5	*bVoltageSupport*	1	-	ACR3901T-W1 可支持 1.8V、3V 和 5V 的槽位电压。
6	*dwProtocols*	4	-	ACR3901T-W1 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	-	最大 ICC 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，ACR3901T-W1 支持的最大 IFSD 为 254。
32	*dwSynchProtocols*	4	-	ACR3901T-W1 不支持同步卡。
36	*dwMechanical*	4	-	ACR3901T-W1 不支持特殊机制特性。
40	*dwFeatures*	4	-	ACR3901T-W1 有以下特性： • 自动根据参数改变 ICC 时钟频率 • 自动根据频率和 FI、DI 参数改变波特率 • 短 APDU 和扩展 APDU 级交换

偏移	数据域	大小	值	说明
44	*dwMaxCCIDMessageLength*	4	-	ACR3901T-W1 可接受的最大报文长度为 271 字节。
48	*bClassGetResponse*	1	-	默认值为 00h
49	*bClassEnvelope*	1	-	默认值为 00h
50	*wLCDLayout*	2	-	无 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。

6.2.1. CCID Bulk-OUT 消息

6.2.1.1. 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	-	标识命令的卡槽号
6	*bSeq*	1	-	命令的序号
7	*bPowerSelect*	1	-	ICC 上的电压值： 00h = 自动电压选择 01h = 5 V 02h = 3 V
8	*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。

6.2.1.2. 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	-	标识命令的卡槽号
6	*bSeq*	1	-	命令的序号
7	*abRFU*	3	-	保留为将来使用

此消息的响应是 *RDR_to_PC_SlotStatus* 消息。

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

此消息的响应是 *RDR_to_PC_SlotStatus* 消息。

6.2.1.4. 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	-	此消息的 *abData* 数据域的大小
5	*bSlot*	1	-	标识命令的卡槽号。
6	*bSeq*	1	-	命令的序号。
7	*bBWI*	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。
8	*wLevelParameter*	2		短 APDU □, RFU = 0000h □展 APDU □： 0000h – 命令 APDU 在本命令中开始和□束。 0001h – 命令 APDU 在本命令中开始，并在下一个 PC_to_RDR_XfrBlock 中□□。 0002h – abData 字段□□□□命令 APDU 并□束□ APDU 命令。 0003h – abData 字段□□□□命令 APDU，后面□跟随另外一个数据□。 0010h – 空的 abData 字段，下一个 RDR_to_PC_DataBlock 会□□□□响□ APDU
10	*abData*	字节型数组	-	发送给 CCID 的数据块。信息“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。

6.2.1.5. 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	*bMessageType*	1	**6Ch**	-
1	*DwLength*	4	00000000h	此消息的额外字节的大小
5	*BSlot*	1	-	标识命令的卡槽号
6	*BSeq*	1	-	命令的序号
7	*AbRFU*	3	-	保留为将来使用

此消息的响应是 *RDR_to_PC_Parameters* 消息。

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

此消息的响应是 *RDR_to_PC_Parameters* 消息。

6.2.1.7. 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	*bMessageType*	1	**61h**	-
1	*dwLength*	4	-	此消息的额外字节的大小
5	*bSlot*	1	-	标识命令的卡槽号
6	*bSeq*	1	-	命令的序号
7	*bProtocolNum*	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*	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	*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	-	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	-	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	-	T=0 时 WI 用于定义 WWT
14	*bClockStop*	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	*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	-	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	-	B7-4 = BWI 值 0-9 有效 B3-0 = CWI 值 0-Fh 有效
14	*bClockStop*	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	*bIFSC*	1	-	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。

6.2.1.8. PC_to_RDR_Escape

此命令访问扩展特性。

偏移	数据域	大小	值	说明
0	*bMessageType*	1	**6Bh**	-
1	*dwLength*	4	-	此消息的 *abData* 数据域的大小
5	*bSlot*	1	-	标识命令的卡槽号。
6	*bSeq*	1	-	命令的序号。
7	*abRFU*	3	-	保留为将来使用
10	*abData*	字节型数组	-	发送给 CCID 的数据块。

此消息的响应是 *RDR_to_PC_Escape* 消息。

6.2.2. CCID Bulk-IN 消息

6.2.2.1. RDR_to_PC_DataBlock

此消息由 ACR3901T-W1 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**	表示正在从 CCID 发送一个数据块。
1	*dwLength*	4	-	此消息的额外字节的大小
5	*bSlot*	1	-	与 Bulk-OUT 消息中的值相同
6	*bSeq*	1	-	与 Bulk-OUT 消息中的值相同
7	*bStatus*	1	-	CCID 标准（1.0 版本）4.2.1 节定义的插槽状态寄存器
8	*bError*	1	-	错误代码 和 CCID 标准（1.0 版本）4.2.1 节定义的插槽错误寄存器
9	*bChainParameter*	1		短 APDU □, RFU = 00h □展 APDU □： 00h – 响□ APDU 在此命令中开始和□束。 01h – 响□ APDU 在此命令开始，并会□□。 02h – abData 字段□□□□响□ APDU 并□束□响□ APDU。 03h – abData 字段□□□□响□ APDU，后面跟随另外一个数据□。 10h – 空的 abData 字段，下一个 PC_to_RDR_XfrBlock 命令会□□□□命令 APDU
10	*abData*	字节型数组	-	本字段包含由 CCID 返还的数据

6.2.2.2. RDR_to_PC_SlotStatus

此消息由 ACR3901T-W1 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

偏移	数据域	大小	值	说明
7	*bStatus*	1	-	CCID 标准（1.0 版本）4.2.1 节定义的插槽状态寄存器
8	*bError*	1	-	错误代码 和 CCID 标准（1.0 版本）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

6.2.2.3. RDR_to_PC_Parameters

此消息由 ACR3901T-W1 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	-	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 标准（1.0 版本）4.2.1 节定义的插槽状态寄存器
8	*bError*	1	-	错误代码 和 CCID 标准（1.0 版本）4.2.1 节定义的插槽错误寄存器
9	*bProtocolNum*	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 CCID 标准（1.0 版本）5.2.3 节汇总。

6.2.2.4. RDR_to_PC_Escape

此消息是 ACR3901T-W1 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	-	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 标准（1.0 版本）4.2.1 节定义的插槽状态寄存器
8	*bError*	1	-	错误代码 和 CCID 标准（1.0 版本）4.2.1 节定义的插槽错误寄存器
9	*bRFU*	1	-	保留为将来使用 Reserved for Future Use
10	*abData*	字节型数组	-	由 CCID 发送的数据

7.0. 主机编程 API

7.1. 外设控制（Peripherals Control）

读写器的外设控制命令在蓝牙模式下通过 Escape 命令（0x6B），在 USB 模式下通过 PC_to_RDR_Escape 命令来实现。

7.1.1. 获取序列号（Get Serial Number）命令

该命令用于读取读写器的唯一序列号。

命令格式

偏移	数据域		大小	值	说明
0	*abData1*	*CommandCode*	1	**02h**	Get Serial Number 的命令码
1		*Len (CommandLength)*	1	00h	数据中额外字节的数量
2		*数据*	0	-	-

应答格式

偏移	数据域		大小	值	说明
0	*abData2*	*ResponseCode*	1	**82h**	Get Serial Number 的响应码
1		*Len (CommandLength)*	1	-	数据中额外字节的数量
2		*数据*	N	-	序列号的字节数

例如：

请求 = 02 00

响应 = 82 0A FF FF FF FF FF FF FF FF FF FF

Serial Number = FF FF FF FF FF FF FF FF FF FF

7.1.2. 获取随机数（Get Random Number）命令

此命令从读写器获取随机数（通过 AES 加密算法和主密钥对随机数进行加密以便用于认证）。*（仅限于蓝牙模式）*

命令格式

偏移	数据域		大小	值	说明
0	*abData1*	*CommandCode*	1	**03h**	Get Random Number 的命令码。
1		*Len (CommandLength)*	1	00h	数据中额外字节的数量
2		*数据*	0	-	-

应答格式

偏移	数据域		大小	值	说明
0	*abData2*	*ResponseCode*	1	**83h**	Get Random Number 的响应码
1		*Len (CommandLength)*	1	10h	数据中额外字节的数量
2		*数据*	16	-	16 字节的随机数

例如：

请求 = 03 00

响应 = 83 10 F2 8F B7 EF BA 43 C4 6B 85 D8 51 7B 84 08 C3 25

7.1.3. 获取固件版本（Get Firmware Version）命令

此命令用于获取读写器的固件版本。

命令格式

偏移	数据域		大小	值	说明
0	*abData1*	*CommandCode*	1	**04h**	Get Firmware Version 的命令码
1		*Len (CommandLength)*	1	00h	数据中额外字节的数量
2		*数据*	0	-	-

应答格式

偏移	数据域		大小	值	说明
0	*abData2*	*ResponseCode*	1	**84h**	Get Firmware Version 的响应码
1		*Len (CommandLength)*	1	-	数据中额外字节的数量
1		*数据*	N	-	固件版本的字节数，格式为 “Vx.xx”

例如：

请求 = 04 00

响应 = 84 08 56 36 2E 30 30 2E 30 30

固件版本 (HEX) = 56 36 2E 30 30 2E 30 30

固件版本 (ASCII) = “V6.00.00”

7.1.4. 重写主密钥（Rewrite Master Key）命令

此命令用于向读写器重写主密钥。需要使用旧密钥通过 AES 加密算法进行加密。

命令格式

偏移	数据域		大小	值	说明
0	*abData1*	*CommandCode*	1	**07h**	Rewrite Master Key 的命令码
1		*Len (CommandLength)*	1	20h	数据中额外字节的数量
2		*数据*	32	-	使用原始客口主密口加密的随机数(KeyRstRnd[0:15]) + 使用原始客口主密口加密的 16 字口新客口主密口

应答格式

偏移	数据域		大小	值	说明
0	*abData2*	*ResponseCode*	1	**87h**	Rewrite Master Key 的响应码
1		*Len (CommandLength)*	1	01h	数据中额外字节的数量
2		*数据*	1	-	00h = 成功 01h = 失败

例如：

更多信息请参考**客户主密钥重置请求**。

7.1.5. 休眠模式选项（Sleep Mode Option）

此命令用于设置设备进入休眠模式前的空闲时长。默认情况下，如果 60 秒内没有操作，读写器会进入休眠模式。

命令格式

偏移	数据域		大小	值	说明
0	*abData1*	*CommandCode*	1	**0Dh**	Sleep Mode Option 的命令码
1		*Len (CommandLength)*	1	01h	数据中额外字节的数量
2		*数据*	1	-	00h = 60 秒（默认） 01h = 90 秒 02h = 120 秒 03h = 180 秒 04h = 禁用

应答格式

偏移	数据域		大小	值	说明
0	*abData2*	*ResponseCode*	1	**8Dh**	Sleep Mode Option 的响应码
1		*Len (CommandLength)*	1	01h	数据中额外字节的数量
2		*数据*	1	-	00h = 成功 01h = 失败

例如：

请求设置为 90s = 0D 01 01

响应 = 8D 01 00

7.1.6. 获取设备地址（Get Device Address）

此命令用于获取设备的蓝牙地址。（仅限于 USB 模式）

命令格式

偏移	数据域		大小	值	说明
0	*abData1*	*CommandCode*	1	**0Eh**	Get Device Address 的命令码
1		*Len (CommandLength)*	1	00h	数据中额外字节的数量
2		*数据*	0	-	-

应答格式

偏移	数据域		大小	值	说明
0	*abData2*	*ResponseCode*	1	**8Eh**	Get Device Address 的响应码
1		*Len (CommandLength)*	1	06h	数据中额外字节的数量
2		*数据*	6	-	6 字节的蓝牙地址

例如：

请求 = 0E 00

响应 = 8E 06 AA BB CC DD EE FF

设备地址：AA BB CC DD EE FF

7.1.7. 设置 Tx 功率（Set Tx Power）

此命令用于设置读写器的蓝牙传输功率。

命令格式

偏移	数据域		大小	值	说明
0	*abData1*	*CommandCode*	1	**08h**	Set Tx Power 的指令代码
1		*Len (CommandLength)*	1	01h	数据中额外字节的数量
2		*数据*	1	-	00h = -18 dBm (默认)， 距离：~4 米 01h = -12 dBm 距离：~7 米 02h = -6 dBm 距离：~16 米 03h = 0 dBm 距离：~25 米

应答格式

偏移	数据域		大小	值	说明
0	*abData2*	*ResponseCode*	1	**88h**	Set Tx Power 的应答代码
1		*Len (CommandLength)*	1	01h	数据中额外字节的数量
2		*数据*	1	-	00h = 成功 01h = 失败

例如：

请求 = 08 01 00

响应 = 88 01 00

7.1.8. 读取 Tx 功率值（Read Tx Power value）

此命令用于查看读写器的蓝牙传输功率。

命令格式

偏移	数据域		大小	值	说明
0	*abData1*	*CommandCode*	1	**09h**	Read Tx Power 的指令代码
1		*Len (CommandLength)*	1	00h	数据中额外字节的数量
2		*数据*	0	-	-

应答格式

偏移	数据域		大小	值	说明
0	*abData2*	*ResponseCode*	1	**89h**	Read Tx Power 的应答代码
1		*Len (CommandLength)*	1	01h	数据中额外字节的数量
2		*数据*	1	-	00h = -18 dBm（默认） 距离：~4 米 01h = -12 dBm 距离：~7 米 02h = -6 dBm 距离：~16 米 03h = 0 dBm 距离：~25 米

例如：

请求 = 09 00

响应 = 89 01 00

7.1.9. 口置卡片重置模口口口（Set Card Reset Simulation Option）

此命令用于口置卡片重置事件，按下按口口无需移除 SAM 卡。

命令格式

偏移	数据域		大小	值	说明
0	*abData1*	*CommandCode*	1	**1Ah**	Set Card Simulation Option 的命令码
1		*Len (CommandLength)*	1	01h	数据中额外字节的数量
2		*数据*	1	-	00h = 禁用 01h = 启用

应答格式

偏移	数据域		大小	值	说明
0	*abData2*	*ResponseCode*	1	**9Ah**	Set Card Simulation Option 的响应码
1		*Len (CommandLength)*	1	01h	数据中额外字节的数量
2		*数据*	1	-	00h = 禁用 01h = 启用

例如：

请求 = 1A 01 01

响应 = 9A 01 01

7.1.10. □取卡片重置模□□□（Get Card Reset Simulation Option）

此命令用于□取卡片重置模□功能的状□。

命令格式

偏移	数据域		大小	值	说明
0	*abData1*	*CommandCode*	1	**1Ah**	Get Card Simulation Option 的命令码
1		*Len (CommandLength)*	1	00h	数据中额外字节的数量
2		*数据*	0	-	

应答格式

偏移	数据域		大小	值	说明
0	*abData2*	*ResponseCode*	1	**9Ah**	Get Card Simulation Option 的响应码
1		*Len (CommandLength)*	1	01h	数据中额外字节的数量
2		*数据*	1	-	00h = 禁用 01h = 启用

例如：

请求 = 1A 00

响应 = 9A 01 01

7.1.11. 设置卡片响应时间间隔（Set Card Response Time Interval）

此命令在卡片重置模□功能已启用的情况下□置 SAM 卡的响□□□□隔。

命令格式

偏移	数据域		大小	值	说明
0	*abData1*	*CommandCode*	1	**18h**	Set Card Response Time Interval 的命令码
1		*Len (CommandLength)*	1	01h	数据中额外字节的数量
2		*数据*	1	-	00h = 0s 01h = 500 ms 02h = 1000 ms 03h = 1500 ms (默认) 04h = 2000 ms 05h = 2500 ms 06h = 3000 ms

应答格式

偏移	数据域		大小	值	说明
0	*abData2*	*ResponseCode*	1	**98h**	Set Card Response Time Interval 的响应码
1		*Len (CommandLength)*	1	01h	数据中额外字节的数量
2		*数据*	1	-	00h = 成功 01h = 失败

例如：

请求 = 18 01 00

响应 = 98 01 00

7.1.12. 获取卡片响应时间间隔（Get Card Response Time Interval）

此命令在卡片重置模□功能已启用的情况下获取 SAM 卡的响□□□□隔。

命令格式

偏移	数据域		大小	值	说明
0	*abData1*	*CommandCode*	1	**19h**	Get Card Response Time Interval 的命令码
1		*Len (CommandLength)*	1	00h	数据中额外字节的数量
2		*数据*	0	-	

应答格式

偏移	数据域		大小	值	说明
0	*abData2*	*ResponseCode*	1	**99h**	Get Card Response Time Interval 的响应码
1		*Len (CommandLength)*	1	01h	数据中额外字节的数量
2		*数据*	1	-	00h = 0s 01h = 500 ms 02h = 1000 ms 03h = 1500 ms (默认) 04h = 2000 ms 05h = 2500 ms 06h = 3000 ms

例如：

请求 = 19 00

响应 = 99 01 03

7.1.13. 查看按键状态（Check button status）

此命令用于查看按键的当前状态。

命令格式

偏移	数据域		大小	值	说明
0	*abData1*	*CommandCode*	1	**1Bh**	Check button status 的命令码
1		*Len (CommandLength)*	1	00h	数据中额外字节的数量
2		*数据*	0	-	

应答格式

偏移	数据域		大小	值	说明
0	*abData2*	*ResponseCode*	1	**9Bh**	Check button status 的响应码
1		*Len (CommandLength)*	1	01h	数据中额外字节的数量
2		*数据*	1	-	00h = 释放 01h = 按下

例如：

请求 = 1B 00

响应 (按键被按下) = 9B 01 01

7.1.14. 客户主密钥重置请求（Customer Master Key Reset Request）

此命令请求读写器生成一组随机数，用于客户主密钥重置认证。

命令格式

偏移	数据域		大小	值	说明
0	*abData1*	*CommandCode*	1	**0Fh**	Customer Master Key Reset Request 的命令码
1		*Len (CommandLength)*	1	00h	数据中额外字节的数量
2		*数据*	0	-	-

应答格式

偏移	数据域		大小	值	说明
0	*abData2*	*ResponseCode*	1	**8Fh**	Rewrite Master Key 的响应码
1		*Len (CommandLength)*	1	10h	数据中额外字节的数量
2		*数据*	16	-	读写器生成的 16 字节随机数 (KeyRSTRnd[0:15])

例如：

1. 生成随机数。

   客户主密钥重置请求 = 0F 00

   客户主密钥重置命令响应 = 8F 10 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11

2. 在 AES-128 CBC 模式下，使用原来的客户主密钥加密随机数和新的客户主密钥。上述操作是由应用的加密引擎完成的，加密结果将被存储起来以备后用。

   随机数：11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11

   加密的随机数：F1 9F D2 D2 BA 1C 22 E1 6D C1 FE 1B 4B 43 D5 30

   新的客户主密钥：11 22 33 44 55 66 77 88 11 22 33 44 55 66 77 88

   加密后的新的客户主密钥：27 E7 DA BE A6 1E 4B CD 29 F6 9B 36 25 05 8E 41

3. 重写主密钥（参见重**写主密钥（Rewrite Master Key）命令**节).

   重置主密钥命令请求 = 07 20 F1 9F D2 D2 BA 1C 22 E1 6D C1 FE 1B 4B 43 D5 30 27 E7 DA BE A6 1E 4B CD 29 F6 9B 36 25 05 8E 41

   重置主密钥命令响应 = 87 01 00

7.2. 存储卡命令集

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

7.2.1.1. 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。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	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）

7.2.1.2. SELECT_PAGE_SIZE

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

命令格式（*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）

7.2.1.3. 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 要从存储卡中读取的数据的长度

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

SW1 SW2 = 90 00h（未发生错误）

7.2.1.4. 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 要写入存储卡的数据的长度

Byte 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）

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

7.2.2.1. SELECT_CARD_TYPE

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。

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**	**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）

7.2.2.2. SELECT_PAGE_SIZE

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

命令格式（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）

7.2.2.3. 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 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。

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 从存储卡中读取的数据

SW1 SW2 = 90 00h（未发生错误）

7.2.2.4. 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 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。

Byte Address 存储卡的内存地址位置

MEM_L 要写入存储卡的数据的长度

Byte 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）

7.2.3. 存储卡 – ATMEL AT88SC153

7.2.3.1. SELECT_CARD_TYPE

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

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**	**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）

7.2.3.2. READ_MEMORY_CARD

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

Pseudo-APDU				
CLA	**INS**	**P1**	**Byte Address**	**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

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 从存储卡中读取的数据

SW1 SW2 = 90 00h（未发生错误）

7.2.3.3. WRITE_MEMORY_CARD

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

Pseudo-APDU								
CLA	INS	P1	Byte Address	MEM_L	Byte 1	....	....	Byte 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

Byte Address 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）

7.2.3.4. 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。

7.2.3.5. 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）

7.2.3.6. 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）

7.2.4. 存储卡 – ATMEL AT88C1608

7.2.4.1. SELECT_CARD_TYPE

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

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**	**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）

7.2.4.2. 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 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 从存储卡中读取的数据

SW1 SW2 = 90 00h（未发生错误）

7.2.4.3. 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

= 写标识位时无关

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 ：写标识位

MEM_L 要写入存储卡的数据的长度

Byte 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）

7.2.4.4. 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。

7.2.4.5. 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：

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：

SW1 SW2 = 90 00h（未发生错误）

7.2.4.6. 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：

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：

SW1 SW2 = 90 00h（未发生错误）

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

7.2.5.1. SELECT_CARD_TYPE

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。

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**	**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）

7.2.5.2. 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_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 存储卡中待读取数据的长度

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）

7.2.5.3. 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 个字节的虚拟数据

SW1 SW2 = 90 00h（未发生错误）

7.2.5.4. 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((位数 - 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 BYTE x 的保护位

‘0’字节被写保护

‘1’字节可以被写入

7.2.5.5. 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 要写入存储卡的数据的长度

Byte 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）

7.2.5.6. WRITE_PROTECTION_MEMORY_CARD

命令中指定的每一个字节与存储在特定地址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被不可逆地设定为“0”。

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

Pseudo-APDU								
CLA	INS	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 要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）

7.2.5.7. 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. 尝试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。

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

7.2.6.1. SELECT_CARD_TYPE

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。

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。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）

7.2.6.2. READ_MEMORY_CARD

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

Pseudo-APDU				
CLA	**INS**	**P1**	**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 要从存储卡中读取的数据的长度

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

SW1, SW2 = 90 00h（未发生错误）

7.2.6.3. 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）

7.2.6.4. 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	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

7.2.6.5. WRITE_MEMORY_CARD

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

Pseudo-APDU								
CLA	**INS**	**P1**	**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 要写入存储卡的数据的长度

Byte 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）

7.2.6.6. 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 = 000$A_4\ A_3A_2A_1A_0$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

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）

7.2.6.7. 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. 尝试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。

7.2.6.8. 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	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）

7.2.7. 存储卡 – SLE 4406/SLE 4436/SLE 5536/SLE 6636

7.2.7.1. SELECT_CARD_TYPE

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。

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。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	07h

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）

7.2.7.2. 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 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 从存储卡中读取的数据

SW1, SW2 = 90 00h（未发生错误）

7.2.7.3. WRITE_ONE_BYTE_MEMORY_CARD

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

此类卡片有四种不同的写入模式，通过命令数据域内的标志加以区分。

a. **写入**

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。

b. **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。此模式仅适用于卡内计数器值的更新。

c. 写入时启动备份功能（只针对 SLE 4436、SLE 5536 和 SLE 6636）

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。

d. **启动备份功能的‘Write with carry’命令**（只针对 SLE 4436、SLE 5536 和 SLE 6636）

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。此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”。

SLE 4436 和 SLE 5536 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：写入

01h：Write with carry

02h：写入时启动备份功能（只针对 SLE 4436、SLE 5536 和 SLE 6636）

03h：启动备份的“Write with carry”命令（只针对 SLE 4436、SLE 5536 和 SLE 6636）

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）

7.2.7.4. 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，ACR3901T-W1 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 卡片中输入计数器的字节地址

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：

SW1 SW2 = 90 00h（未发生错误）

7.2.7.5. AUTHENTICATE_MEMORY_CARD (SLE 4436、SLE 5536 和 SLE 6636)

要从 SLE 5536 或 SLE 6636 卡片中读取卡片认证证书，ACR3901T-W1 要执行以下操作：

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

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

步骤 1：向卡片发送认证证书。

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

Pseudo-APDU												
CLA	INS	P1	P2	MEM_L	CODE							
					KEY	CLK_CNT	Byte 1	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（只针对 SLE 5536 和 SLE 6636）

81h:Key 2 带密码块链接（只针对 SLE 5536 和 SLE 6636）

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：

SW1 SW2 = 61 02h（未发生错误），表示两个字节的认证数据准备就绪。可以通过 *Get_Response* 命令获取认证数据。

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。

SW1 SW2 = 90 00h（未发生错误）

7.2.8. 存储卡 – SLE 4404

7.2.8.1. SELECT_CARD_TYPE

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。

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。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	01	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）

7.2.8.2. 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 要从存储卡中读取的数据的长度

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

SW1 SW2 = 90 00h（未发生错误）

7.2.8.3. WRITE_MEMORY_CARD

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

指定地址上的字节在写操作前不会被擦除，所以存储位只能由‘1’设为‘0’。

命令格式（*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 要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）

7.2.8.4. ERASE_SCRATCH_PAD_MEMORY_CARD

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

要擦除错误计数器或者用户区，请使用 VERIFY_USER_CODE 命令，如 **VERIFY_USER_CODE** 所示。

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）

7.2.8.5. VERIFY_USER_CODE

此命令将用户密码（2 字节）提交给插入的卡片。用户密码用于启用卡片存储的访问权限。

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 字节的用户密码

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）

= 63 00h（如果不再有重试的机会）

注： *收到响应 SW1SW2 = 9000h 后，应当再次读取用户错误计数器，检查 VERIFY_USER_CODE 是否正确。如果用户错误计数器被擦除并且等于‘FFh’，证明先前的验证成功。*

7.2.8.6. VERIFY_MEMORY_CODE

此命令用于向插入的卡片提交存储密码（4 个字节）。该存储密码用于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				

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）

= 63 00h（如果不再有重试的机会）

注：*收到响应 SW1SW2 = 9000h 后，应当再次读取应用区，检查 VERIFY_MEMORY_CODE 是否正确。如果应用区域的全部数据都被擦除并且等于‘FFh’，证明先前的验证成功。*

7.2.9. 存储卡 – AT88SC101/AT88SC102/AT88SC1003

7.2.9.1. SELECT_CARD_TYPE

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。

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。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	09h

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）

7.2.9.2. 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

SW1 SW2 = 90 00h（未发生错误）

7.2.9.3. WRITE_MEMORY_CARD

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

指定地址上的字节在写操作前不会被擦除，所以存储位只能由‘1’设为‘0’。

命令格式（*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 要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）

7.2.9.4. ERASE_NON_APPLICATION_ZONE

此命令用于擦除存储在非应用区的数据。EEPROM 内存由 16 位字构成。即使只擦除单独的一个位，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，请参考以下命令：

1. *ERASE_APPLICATION_ZONE_WITH_ERASE 命令*，定义见：
   ERASE_APPLICATION_ZONE_WITH_ERASE
2. *ERASE_APPLICATION_ZONE_WITH_WRITE_AND_ERASE* 命令，定义见：
   ERASE_APPLICATION_ZONE_WITH_WRITE_AND_ERASE
3. *VERIFY_SECURITY_CODE* 命令，定义见：
   VERIFY_SECURITY_CODE

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：

SW1 SW2 = 90 00h（未发生错误）

7.2.9.5. ERASE_APPLICATION_ZONE_WITH_ERASE

此命令可用于下列情况：

1. AT88SC101：擦除应用区域中的数据，EC 功能禁用。
2. AT88SC102：擦除应用区域 1 中的数据。
3. AT88SC102：擦除应用区域 2 中的数据，EC2 功能禁用。
4. AT88SC1003：擦除应用区域 1 中的数据。
5. AT88SC1003：擦除应用区域 2 中的数据，EC2 功能禁用。
6. 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。
   a. 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 卡片中的应用区密钥的字节地址。正确值请参阅下表：

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

	Byte Address	LEN
AT88SC1003：擦除应用区域 1	36h	06h
AT88SC1003：擦除应用区域 2，EC2 功能禁用	5Ch	04h
AT88SC1003：擦除应用区域 3	C0h	06h

MEM_L 擦除密钥的长度。正确值请参阅上表。

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：

SW1 SW2 = 90 00h（未发生错误）

注： *收到状态字 SW1SW2 = 9000h 后，可重新读取应用区域内的数据来检查 ERASE_APPLICATION_ZONE_WITH_ERASE 命令是否正确。如果应用区域的全部数据都被擦除并且等于'FFh'，则说明先前的验证成功。*

7.2.9.6. ERASE_APPLICATION_ZONE_WITH_WRITE_AND_ERASE

此命令可用于下列情况：

1. AT88SC101：擦除应用区域中的数据，EC 功能启用。
2. AT88SC102：擦除应用区域 2 中的数据，EC2 功能启用。
3. 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，单位为比特值始终是 80h。

Byte Address 卡片中的应用区密钥的字节地址

	Byte Address
AT88SC101	96h
AT88SC102	9Ch
AT88SC1003	5Ch

CODE 4 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：

SW1 SW2 = 90 00h（未发生错误）

= 63 00h（如果不再有重试的机会）

注： *收到状态字 SW1SW2 = 9000h 后，可重新读取应用区域内的数据来检查 ERASE_APPLICATION_ZONE_WITH_WRITE_AND_ERASE 命令是否正确。如果应用区域的全部数据都被擦除并且等于'FFh'，则说明先前的验证成功。*

7.2.9.7. VERIFY_SECURITY_CODE

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

执行的操作如下：

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

命令格式（*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：

SW1, SW2 = 90 00h（未发生错误）

= 63 00h（如果不再有重试的机会）

注：*收到状态字 SW1SW2 = 9000h 后，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 被擦除并且等于'FFh'，证明先前的验证成功。*

7.2.9.8. BLOWN_FUSE

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

注：*更改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 或 01h

其中：

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

State of FUS Pin FUS pin 的状态始终是 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	生产商标识位	05h	80h	01h
	EC_EN 标识位	05h	C9h	01h
	发行商标识位	05h	E0h	01h
AT88SC102	生产商标识位	05h	B0h	01h
	EC2EN 标识位	05h	F9h	01h
	发行商标识位	06h	10h	01h
AT88SC1003	生产商标识位	03h	F8h	00h
	EC2EN 标识位	03h	FCh	00h
	发行商标识位	03h	E0h	00h

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□A. 错误代码

下表汇总了 ACR3901T-W1 可能返回的错误代码：

错误代码	说明
01h	校验和无效
02h	数据长度无效
03h	命令结构无效
04h	命令无效/未知的命令 ID
05h	卡操作错误
06h	需要认证/认证错误
07h	电量低
08h	认证失败
09h	认证失败次数超过最大限制
0Ah	T1 卡片操作错误

表11 ： 错误代码