Advanced Card Systems Ltd.
Card & Reader Technologies

ACR1281U-K1 (PCSC)

应用程序编程接口 V1.01

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

目录

图目录

表目录

1.0. 简介

ACR1281U-K1 是一款连机接触式和非接触式智能卡读写器。它可以访问符合 ISO 14443 1-4 标准的卡（包括 MIFARE®卡）以及符合 ISO 7816 1-3 标准的接触式智能卡，同时支持符合 ISO 7816 标准的 SAM 卡槽。

2.0. 特性

- USB 2.0全速接口
- 符合CCID标准
- 非接触式智能卡读写器：
  - 读写速度可达 848 Kbps
  - 内置天线用于读写非接触式标签，读取智能卡的距离可达 50 mm（视标签的类型而定）
  - 支持 ISO 14443 第 4 部分的 A 类卡和 B 类卡
  - 支持 MIFARE Classic 系列卡（例如 MIFARE 1K、4K、MIFARE Plus 和 DESFire）
  - 内建防冲突特性（任何时候都只能访问 1 张标签）
  - 支持扩展的 APDU（最大 64 K 字节）
- 接触式智能卡读写器：
  - 支持 ISO 7816 的 A 类、B 类和 C 类（5 V、3 V、1.8 V）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
  - 符合 ISO 7816 标准的 SAM 卡槽（最多可支持 4 个 SAM，仅支持 +5 V）
- 应用程序编程接口：
  - 支持 PC/SC
- 内置外围设备：
  - 2 个用户可控的 LED 指示灯
  - 1 个用户可控的蜂鸣器
- 具有USB固件升级能力
- 支持Android™ 3.1及以上版本
- 符合下列标准：
  - ISO 14443
  - ISO 7816
  - CE
  - FCC
  - PC/SC
  - CCID
  - PBOC（接触式接口）
  - qPBOC（非接触式接口）
  - EMVCo Level 1（接触式接口和非接触式接口）
  - Microsoft® WHQL
  - RoHS

3.0. 体系结构

ACR1281U-K1 与计算机之间的数据通讯采用 CCID 协议。使用前需要安装 ACS 的 ACR1281U-K1 驱动。PICC、ICC 和 SAM 间的通信则完全符合 PC/SC 标准。

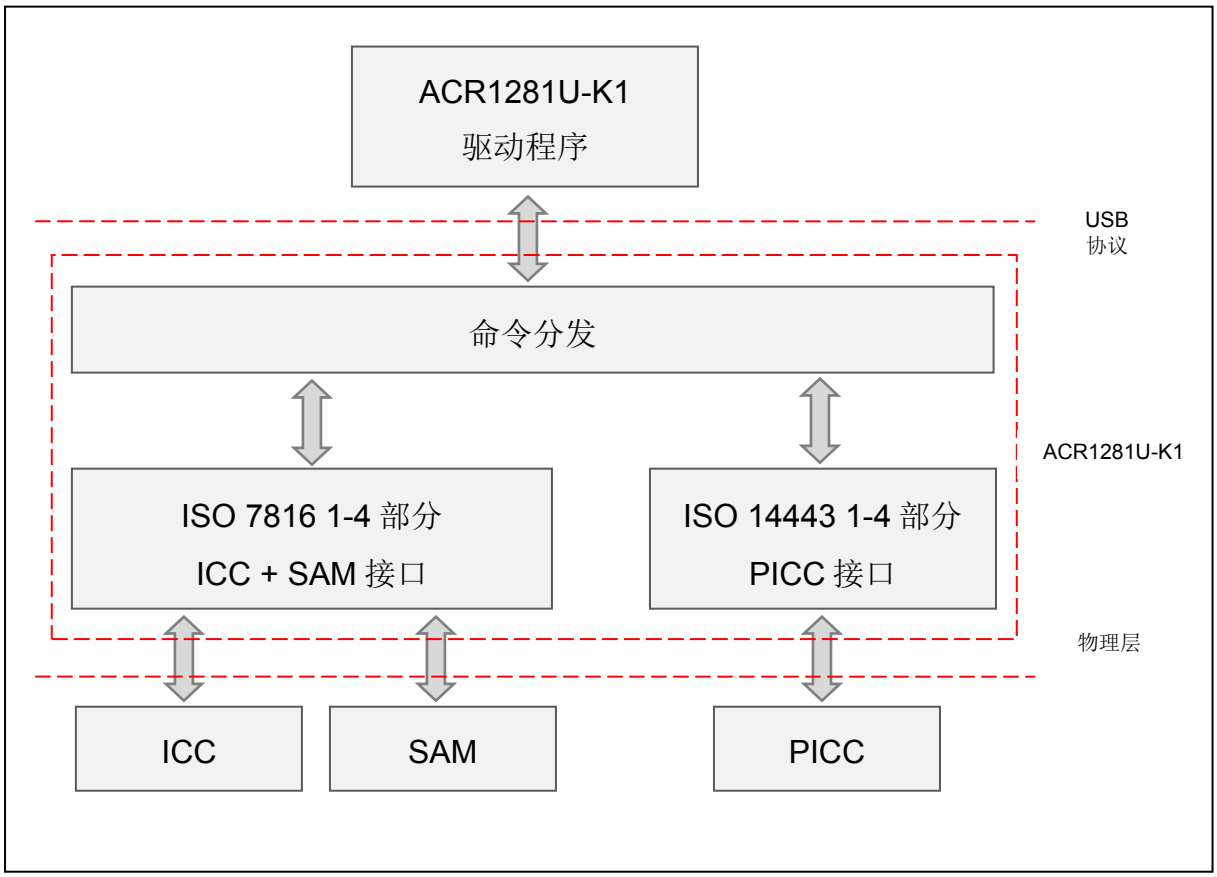

图 1：ACR1281U-K1 体系结构

4.0. 软件设计

4.1. CCID 协议

ACR1281U-K1 通过 USB 接口与主机连接。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 涵盖了操作智能卡和 PIN 所需的全部协议。

ACR1281U-K1 的 USB 端点的配置和使用应当符合 CCID 标准第 3 部分的规定。概述总结如下：

- **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。
- CCID **事件**通过中断通道发送。
- CCID **命令**经由 BULK-OUT 端点发出。每个发送给 ACR1281U-K1 的命令都有一个相应的最终响应。一些命令也有过程响应。
- CCID **响应**经由 BULK-IN 端点发出。所有发送给 ACR1281U-K1 的命令都必须同步发送。（即：对于 ACR1281U-K1 来说，*bMaxCCIDBusySlots* 等于 1）。

ACR1281U-K1 支持的 CCID 功能由其类别描述符定义：

偏移	数据域	大小	值	描述
0	*bLength*	1	36h	这个描述符的字节大小。
1	*bDescriptorType*	1	21h	CCID 功能描述符类型。
2	*bcdCCID*	2	0110h	CCID 以二进制编码的十进制指定的版本号码。
4	*bMaxSlotIndex*	1	03h	ACR1281U-K1 有 3 个卡槽。
5	*bVoltageSupport*	1	07h	ACR1281U-K1 可以支持 1.8 V、3.0 V 和 5.0 V 的 ICC 槽位电压。
6	*dwProtocols*	4	00000003h	ACR1281U-K1 支持 T=0 和 T=1 协议
10	*dwDefaultClock*	4	000012C0h	默认 ICC 时钟频率为 4 MHz。
14	*dwMaximumClock*	4	000012C0h	ICC 支持的最大时钟频率为 4 MHz。
18	*bNumClockSupported*	1	00h	不支持手动设置时钟频率。
19	*dwDataRate*	4	00003267h	默认 ICC I/O 波特率为 10752 bps。
23	*dwMaxDataRate*	4	000927C0h	ICC I/O 支持的最大波特率为 129032 bps。
27	*bNumDataRatesSupported*	1	00h	不支持手动设置波特率。
28	*dwMaxIFSD*	4	0000200h	ACR1281U-K1 T1 支持的最大 IFSD 为 512。
32	*dwSynchProtocols*	4	00000000h	ACR1281U-K1 不支持同步卡。

偏移	数据域	大小	值	描述
36	*dwMechanical*	4	00000000h	ACR1281U-K1 不支持特殊机制特性
40	*dwFeatures*	4	000200BAh	ACR1281U-K1 支持以下特性： • 根据 ATR 数据自动配置参数 • 插入时自动使能 ICC • 自动选择 ICC 电压 • 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 • CCID 根据主动参数自动进行 PPS • 与 ACR1281U-K1 进行短 APDU 级交换
44	*dwMaxCCIDMessageLength*	4	0000010Fh	ACR1281U-K1 可接受的最大报文长度为 271 字节。
48	*bClassGetResponse*	1	00h	表示 CCID 回应 APDU 的类别。
49	*bClassEnvelope*	1	00h	表示 CCID 回应 APDU 的类别。
50	*wLCDLayout*	2	0000h	无 LCD。
52	*bPINSupport*	1	00h	无 PIN 校验。
53	*bMaxCCIDBusySlots*	1	01h	可以同时忙的槽位数为 1。

4.2. CCID 命令

4.2.1. CCID 命令通道 Bulk-OUT 消息

ACR1281U-K1 应当遵循 CCID 协议第四部分定义的 Bulk-OUT 消息。此外，该规范还定义了一些用于操作附加功能的扩展命令。本节将列举 ACR1281U-K1 支持的 CCID 类 Bulk-OUT 消息。

4.2.1.1. 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	描述
0	*bMessageType*	1	62h	
1	*dwLength*	4	00000000h	此消息的额外字节的大小。
2	*bSlot*	1		标识命令的插槽号。
5	*bSeq*	1		命令的序号。
6	*bPowerSelect*	1		ICC 上的电压值： 00h = 自动电压选择 01h = 5 V 02h = 3 V
7	*abRFU*	2		保留为将来使用。

此命令消息的响应是***错误！未找到引用源。*** **错误！未找到引用源。**消息，返回的数据是复位应答（ATR）。

4.2.1.2. PC_to_RDR_IccPowerOff

此命令用于取消激活卡槽。

偏移	数据域	大小	值	描述
0	*bMessageType*	1	63h	
1	*dwLength*	4	00000000h	此消息的额外字节的大小。
5	*bSlot*	1		标识命令的插槽号。
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***错误！未找到引用源。***消息。

4.2.1.3. 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	描述
0	*bMessageType*	1	65h	
1	*dwLength*	4	00000000h	此消息的额外字节的大小。
5	*bSlot*	1		标识命令的插槽号。

偏移	数据域	大小	值	描述
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***错误！未找到引用源。***消息。

4.2.1.4. PC_to_RDR_XfrBlock

此命令用于向 ICC 传输数据块。

偏移	数据域	大小	值	描述
0	*bMessageType*	1	6Fh	
1	*dwLength*	4		此消息的 *abData* 数据域的大小
5	*bSlot*	1		标识命令的插槽号。
6	*bSeq*	1		命令的序号。
7	*bBWI*	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	*wLevelParameter*	2	0000h	RFU（TPDU 交换级别）。
10	*abData*	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）。

此消息的响应是***错误！未找到引用源。***消息。

4.2.1.5. PC_to_RDR_Escape

此命令允许访问扩展特性。

偏移	数据域	大小	值	描述
0	*bMessageType*	1	6Bh	
1	*dwLength*	4		此消息的 *abData* 数据域的大小
5	*bSlot*	1		标识命令的插槽号。
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。

此命令消息的响应是 RDR_to_PC_Escape 消息。

4.2.1.6. PC_to_RDR_GetParameters

此命令用于获取卡槽的参数。

偏移	数据域	大小	值	描述
0	*bMessageType*	1	6Ch	
1	*dwLength*	4	00000000h	此消息的额外字节的大小。
5	*bSlot*	1		标识命令的插槽号。
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。

4.2.1.7. PC_to_RDR_ResetParameters

此命令用于将卡槽参数重置为默认值。

偏移	数据域	大小	值	描述
0	*bMessageType*	1	6Dh	
1	*dwLength*	4	00000000h	此消息的额外字节的大小。
5	*bSlot*	1		标识命令的插槽号。
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。

4.2.1.8. PC_to_RDR_SetParameters

此命令用于设置卡槽的参数。

偏移	数据域	大小	值	描述
0	*bMessageType*	1	61h	
1	*dwLength*	4		此消息的额外字节的大小。
5	*bSlot*	1		标识命令的插槽号。
6	*bSeq*	1		命令的序号。
7	*bProtocolNum*	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	*abRFU*	2		保留为将来使用。

偏移	数据域	大小	值	描述
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	描述
10	*bmFindexDindex*	1		B7-4 – FI – ISO/IEC 7816-3:1997 中表 7 的索引，选择一个时钟频率转换因子。 B3-0 – DI - ISO/IEC 7816-3:1997 中表 8 的索引，选择一个波特率转换因子。
11	*bmTCCKST0*	1		B0 – 0b, B7-2 – 000000b B1 – 使用的约定（b1=0：正向约定；b1=1：反向约定 ***注：****CCID 忽略该位。*
12	*bGuardTimeT0*	1		两个字符间的额外保护时间。在通常保护时间（12etu）的基础上增加 0-254 个 etu。FFh 与 00h 相同。
13	*bWaitingIntegerT0*	1		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	描述
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 至 254 个 etu）。若值为 FFh，则保护时间减少 1 个 etu。

偏移	数据域	大小	值	描述
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。

4.2.2. CCID Bulk-IN 消息

Bulk-IN 消息用于对 Bulk-OUT 消息做出响应。ACR1281U-K1 应当遵循 CCID 协议第四部分定义的 Bulk-IN 消息。本节将列举 ACR1281U-K1 支持的 CCID 类 Bulk-IN 消息。

4.2.2.1. RDR_to_PC_DataBlock

此命令由 ACR1281U-K1 发出，是对 *PC_to_RDR_IccPowerOn*、*PC_to_RDR_XfrBlock* 和 *PC_to_RDR_Secure* 消息的响应。

偏移	数据域	大小	值	描述
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 规范 4.2.1 节定义的插槽状态寄存器。
8	*bError*	1		CCID 规范 4.2.1 节定义的插槽错误寄存器。
9	*bChainParameter*	1	00h	RFU（TPDU 交换级别）。
10	*abData*	字节型数组		本数据域包含由 CCID 返还的数据。

4.2.2.2. RDR_to_PC_Escape

此消息由 ACR1281U-K1 发出，是对 *PC_to_RDR_Escape* 消息的响应。

偏移	数据域	大小	值	描述
0	*bMessageType*	1	83h	

偏移	数据域	大小	值	描述
1	*dwLength*	4		此消息的 *abData* 数据域的大小
5	*bSlot*	1		与 Bulk-OUT 消息中的值相同。
6	*bSeq*	1		与 Bulk-OUT 消息中的值相同。
7	*bStatus*	1		CCID 规范 4.2.1 节定义的插槽状态寄存器。
8	*bError*	1		CCID 规范 4.2.1 节定义的插槽错误寄存器。
9	*bRFU*	1	00h	RFU.
10	*abData*	字节型数组		本数据域包含由 CCID 返还的数据。

4.2.2.3. RDR_to_PC_SlotStatus

此消息由 ACR1281U-K1 发出，是对 *PC_to_RDR_IccPowerOff*、*PC_to_RDR_GetSlotStatus* 和 *PC_to_RDR_Abort* 消息，以及类特定 ABORT 请求的响应。

偏移	数据域	大小	值	描述
0	*bMessageType*	1	81h	
1	*dwLength*	4	00000000h	此消息的额外字节的大小。
5	*bSlot*	1		与 Bulk-OUT 消息中的值相同。
6	*bSeq*	1		与 Bulk-OUT 消息中的值相同。
7	*bStatus*	1		CCID 规范 4.2.1 节定义的插槽状态寄存器。
8	*bError*	1		CCID 规范 4.2.1 节定义的插槽错误寄存器。
9	*bClockStatus*	1		值： 00h = 时钟运行 01h = 时钟停于 L 状态 02h = 时钟停于 H 状态 03h = 时钟停止于未知状态 所有其他值保留为将来使用。

4.2.2.4. RDR_to_PC_Parameters

此消息由 ACR1281U-K1 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	描述
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。

偏移	数据域	大小	值	描述
7	*bStatus*	1		CCID 规范 4.2.1 节定义的插槽状态寄存器。
8	*bError*	1		CCID 规范 4.2.1 节定义的插槽错误寄存器。
9	*bProtocolNum*	1		指定后面的协议数据结构。 00h = T=0 协议的结构 01h = T=1 协议的结构 以下值保留为将来使用 80h = 2 线协议结构 81h = 3 线协议结构 82h = I2C 协议结构
10	*abProtocolDataStructure*	字节型数组		协议数据结构。

4.3. 非接触式智能卡协议

4.3.1. ATR 的生成

读写器检测到 PICC 后，一个 ATR 会被发送至 ACR1281U-K1 驱动来识别 PICC。

4.3.1.1. ATR 信息格式（适用于 ISO 14443-3 PICC）

字节	值	标记	描述
0	3Bh	初始字符	
1	8Nh	T0	高半字节 8 表示：后续不存在 TA1、TB1 和 TC1，只存在 TD1。 低半字节 N 指出历史字符的个数（HistByte 0 - HistByte N-1）
2	80h	TD1	高半字节 8 表示：后续不存在 TA2、TB2 和 TC2，只存在 TD2。 低半字节 0 表示协议类型为 T=0
3	01h	TD2	高半字节 0 表示后续不存在 TA3、TB3、TC3 和 TD3。 低半字节 1 表示协议类型为 T=1
4	80h	T1	类别指示字节，80 表示在可选的 COMPACT-TLV 数据对象中或许存在一个状态标识符
	4Fh	Tk	应用标识符存在标识
	0Ch		长度
至	RID		注册的应用提供商标识(RID) # A0 00 00 03 06h
3+N	SS		标准字节
	C0 ..C1h		卡片名称字节
	00 00 00 00h	RFU	RFU # 00 00 00 00
4+N	UUh	TCK	T0 至 Tk 的所有字符按位异或

例如：

MIFARE 1K 卡的 ATR = {3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 01 00 00 00 00 6Ah}

其中：

长度 (YY) = 0Ch

RID = A0 00 00 03 06h (PC/SC 工作组)

标准 (SS) = 03h (ISO 14443A，第 3 部分)

卡片名称 (C0 ..C1) = [00 01h] (MIFARE 1K)

标准 (SS) = 03h：ISO 14443A，第 3 部分

= 11h：FeliCa

卡片名称 (C0 ..C1)	00 01：MIFARE 1K	00 30：Topaz 和 Jewel
	00 02：MIFARE 4K	00 3B：FeliCa
	00 03：MIFARE Ultralight	FF 28：JCOP 30
	00 26：MIFARE Mini	FF [SAK]：未定义的标签

4.3.1.2. ATR 信息格式（适用于 ISO 14443-4 PICC）

<table>
<tr><th>字节</th><th>值</th><th>标记</th><th>描述</th></tr>
<tr><td>0</td><td>3Bh</td><td>初始字符</td><td></td></tr>
<tr><td>1</td><td>8Nh</td><td>T0</td><td>高半字节 8 表示：后续不存在 TA1、TB1 和 TC1，只存在 TD1。
低半字节 N 指出历史字符的个数（HistByte 0 - HistByte N-1)</td></tr>
<tr><td>2</td><td>80h</td><td>TD1</td><td>高半字节 8 表示：后续不存在 TA2、TB2 和 TC2，只存在 TD2。
低半字节 0 表示协议类型为 T=0</td></tr>
<tr><td>3</td><td>01h</td><td>TD2</td><td>高半字节 0 表示后续不存在 TA3、TB3、TC3 和 TD3。
低半字节 1 表示协议类型为 T=1</td></tr>
<tr><td rowspan="2">4
至
3 + N</td><td>XXh</td><td>T1</td><td rowspan="2">历史字节：

ISO 14443A：
来自 ATS 响应的历史字节。请参阅 ISO14443-4 的规定。

ISO 14443B：<table><tr><th>Byte1-4</th><th>Byte5-7</th><th>Byte8</th></tr><tr><td>ATQB 的应用数据</td><td>ATQB 的协议信息字符</td><td>高半字节 =ATTRIB 命令的 MBLI；低半字节 (RFU)=0</td></tr></table></td></tr>
<tr><td>XX
XX
XXh</td><td>Tk</td></tr>
<tr><td>4+N</td><td>UU</td><td>TCK</td><td>T0 至 Tk 的所有字符按位异或</td></tr>
</table>

例 1：MIFARE DESFire 的 ATR = { 3B 81 80 01 80 80 } // 6 个字节的 ATR

注：使用 APDU"FF CA 01 00 00h"来区分是符合 ISO 14443A-4 的 PICC 还是符合 ISO 14443B-4 的 PICC，并且如果有的话，取回完整的 ATS。符合 ISO 14443A-3 或 ISO 14443B-3/4 的 PICC 会返回 ATS。

APDU 命令 = FF CA 01 00 00h

APDU 响应 = 06 75 77 81 02 80 90 00h

ATS = {06 75 77 81 02 80h}

例 2：EZ-Link 的 ATR = {3B 88 80 01 1C 2D 94 11 F7 71 85 00 BEh}

ATQB 的应用数据 = 1C 2D 94 11h

ATQB 的协议信息 = F7 71 85h

ATTRIB 的 MBLI = 00h

5.0. PCSC API

这一章节将会描述一些用于应用程序编程的 PCSC API。关于这些 API 的更多细节，请参考 Microsoft MSDN 库或 PCSC 工作组。

5.1. SCardEstablishContext

SCardEstablishContext 函数用于建立进行设备数据库操作的资源管理器上下文。

请参考：http://msdn.microsoft.com/en-us/library/windows/desktop/aa379479%28v=vs.85%29.aspx

5.2. SCardListReaders

SCardListReaders 函数可以给出系统中在指定读卡器组集合中的读卡器名字列表（去掉重复的）。

调用者提供一个读卡器组列表，函数返回这些指定组里面的读卡器名字列表。无法识别的组名会被忽略。这个函数只会返回当前系统中可以使用的组里面的读卡器。

请参考：http://msdn.microsoft.com/en-us/library/windows/desktop/aa379793%28v=vs.85%29.aspx

5.3. SCardConnect

SCardConnect 函数利用特定资源管理器上下文，在应用程序与包含在特定读卡器中的智能卡之间建立一条连接。如果特定读卡器中没有卡片，会返回一条错误信息。

请参考：http://msdn.microsoft.com/en-us/library/windows/desktop/aa379473%28v=vs.85%29.aspx

5.4. SCardControl

SCardControl 函数提供对读卡器的直接控制。你可以在 **SCardConnect** 函数成功调用后，但 **SCardDisconnect** 函数成功调用前随时调用此函数。它对读卡器状态的影响取决于控制码。

请参考：http://msdn.microsoft.com/en-us/library/windows/desktop/aa379474%28v=vs.85%29.aspx

注：*6.4 节的命令要使用此 API 进行发送。*

5.5. ScardTransmit

SCardTransmit 函数用来发送服务请求给智能卡，并接收从智能卡返回的数据。

请参考：http://msdn.microsoft.com/en-us/library/windows/desktop/aa379804%28v=vs.85%29.aspx

注：*APDU 命令（即：发送给已建立连接的卡片的命令，6.1 节）使用此 API 进行发送。*

5.6. ScardDisconnect

SCardDisconnect 函数用来断开先前在应用程序和目标读卡器中的智能卡之间建立的连接。

请参考：http://msdn.microsoft.com/en-us/library/windows/desktop/aa379475%28v=vs.85%29.aspx

5.7. APDU 流程图

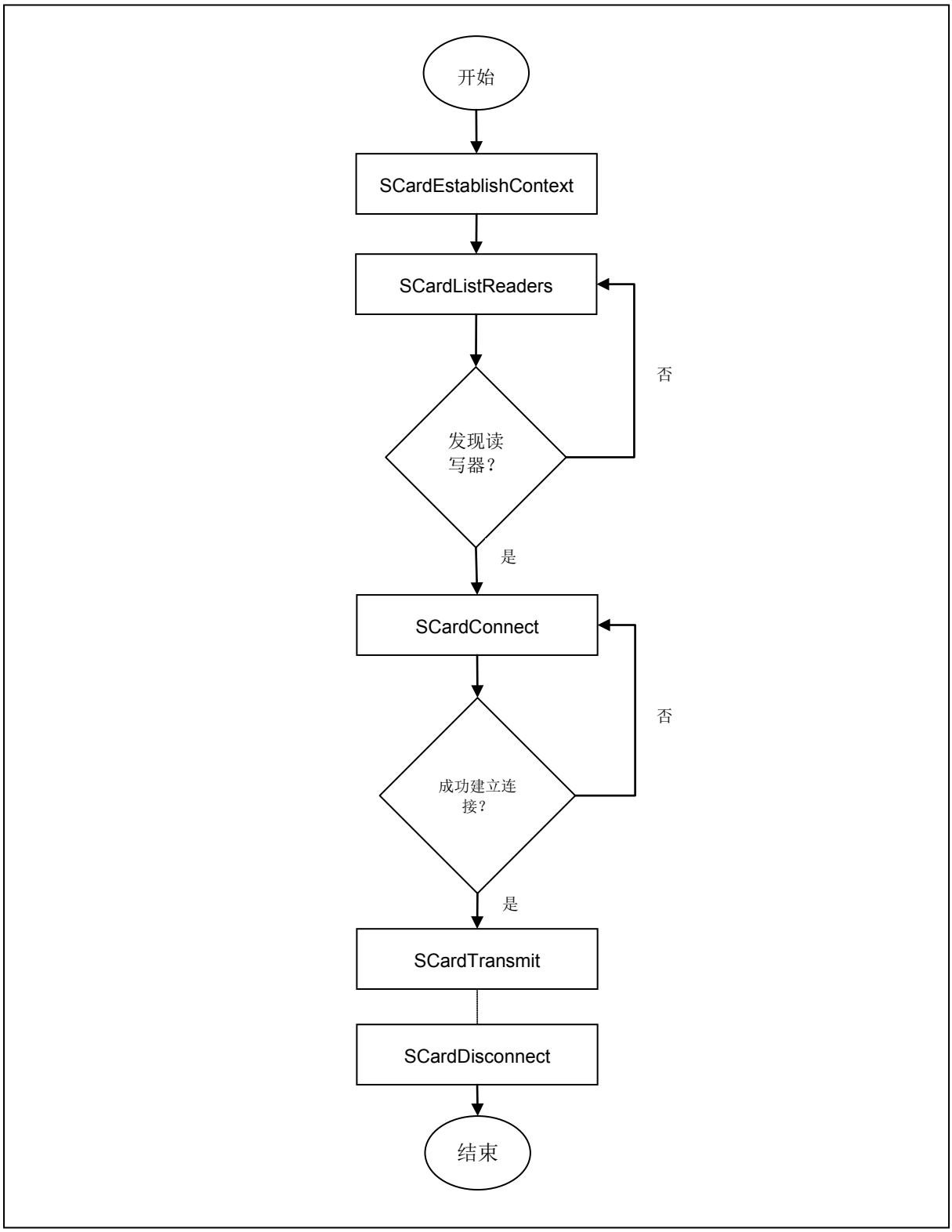

图 2:APDU 流程图

5.8. 直接命令（Escape Command）流程图

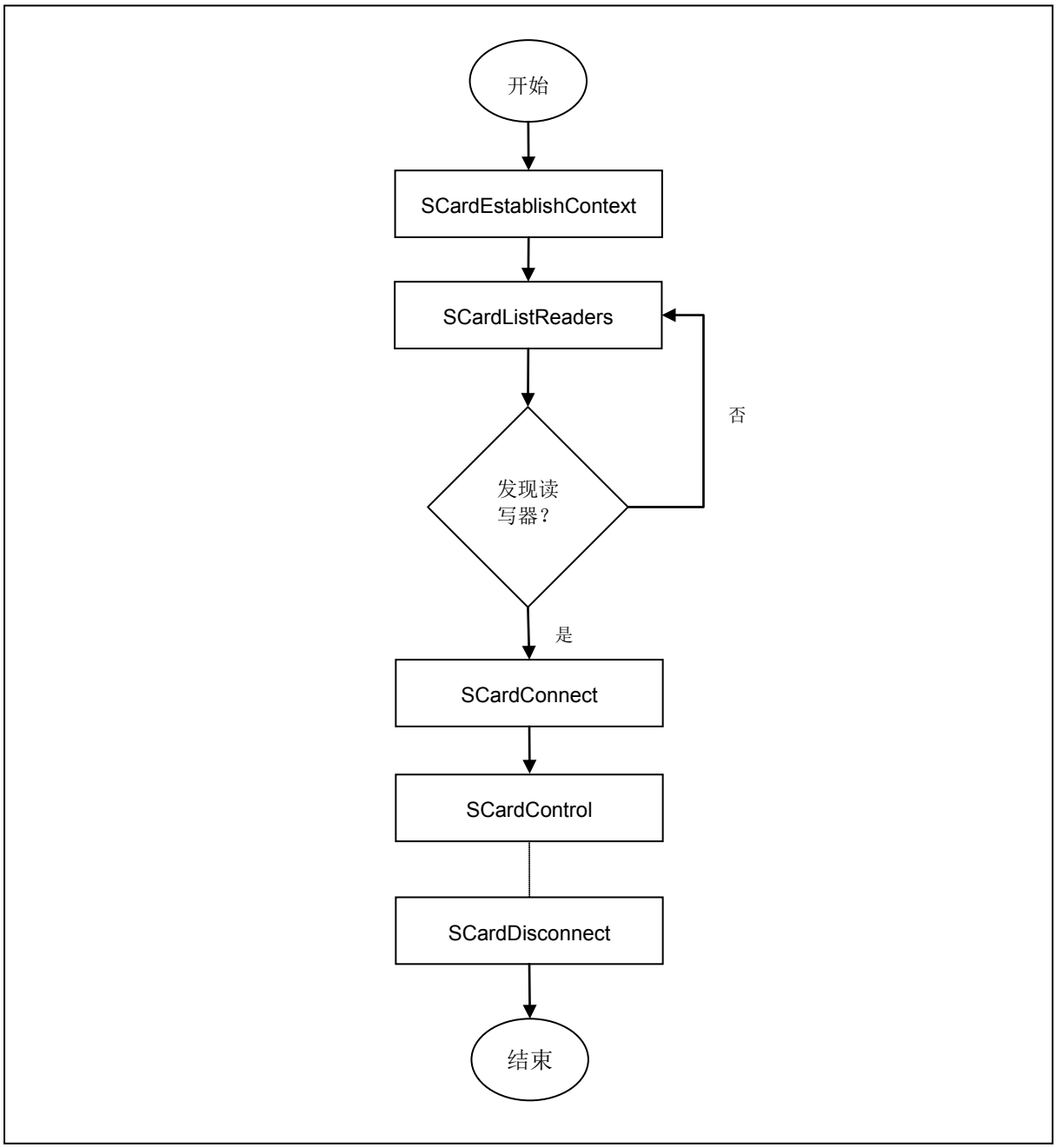

图 3: 直接命令（Escape Command）流程图

6.0. 命令集

6.1. MIFARE 1K/4K 存储卡的 PICC 命令（T=CL 模拟）

6.1.1. 加载认证密钥（Load Authentication Keys）

此命令用于向读写器加载认证密钥。该认证密钥用于验证 MIFARE 1K/4K 存储卡的特定扇区。读写器提供了两种认证密钥位置，分别是易失密钥位置和非易失密钥位置。

Load Authentication Keys 命令的 APDU 结构（11 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Load Authentication Keys	FFh	82h	密钥结构	密钥号	06h	密钥（6 个字节）

其中：

密钥结构 1 个字节。

00h = 密钥被载入读写器的易失存储器。

其它 = 保留。

密钥号 1 个字节。

00h ~ 01h = 密钥位置。一旦读写器与电脑断开连接，密钥就会消失。

密钥 6 个字节。

载入读写器的密钥值，例如：{FF FF FF FF FF FFh}

Load Authentication Keys 的响应结构（2 个字节）

响应	响应数据域	
结果	SW1	SW2

响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

例如：

// 向密钥位置 00h 加载密钥{FF FF FF FF FF FFh}。

APDU = {FF 82 00 00 06 FF FF FF FF FF FFh}

6.1.2. MIFARE 1K/4K 卡认证（Authentication for MIFARE 1K/4K）

此命令使用存储在读写器内的密钥来验证 MIFARE 1K/4K 卡（PICC）。其中会用到两种认证密钥：TYPE_A 和 TYPE_B。

Load Authentication Keys 命令的 APDU 结构（6 个字节）[弃用]

命令	CLA	INS	P1	P2	P3	命令数据域
Authentication	FFh	88h	00h	块号	密钥类型	密钥号

Load Authentication Keys 命令的 APDU 结构（10 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Authentication	FFh	86h	00h	00h	05h	认证数据字节

认证数据字节（5 个字节）

字节 1	字节 2	字节 3	字节 4	字节 5
版本号 01h	00h	块号	密钥类型	密钥号

其中：

块号 1 个字节。指出待验证的存储块。

密钥类型 1 个字节。

60h = 该密钥被用作 TYPE A 密钥进行验证

61h = 该密钥被用作 TYPE B 密钥进行验证

密钥号 1 个字节。

00h ~ 01h = 密钥位置

注：*一张 MIFARE 1K 卡分为 16 个扇区，每个扇区包含 4 个连续的块。例如：扇区 00h 包含块{00h、01h、02h 和 03h}；扇区 01h 包含块{04h、05h、06h 和 07h}；最后一个扇区 0Fh 包含块{3Ch、3Dh、3Eh 和 3Fh}。*

验证通过后，读取同一扇区内的其他块不需要再次进行验证。详情请参考 MIFARE 1K/4K 卡标准。

Load Authentication Keys 命令的响应结构（2 个字节）

响应	响应数据域	
结果	SW1	SW2

响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

扇区 (共 16 个扇区，每个扇区包含 4 个连续的块)	数据块 (3 个块，每块 16 个字节)	尾部块 (1 个块，16 个字节)
扇区 0	00h ~ 02h	03h
扇区 1	04h ~ 06h	07h
..		
..		
扇区 14	38h ~ 0Ah	3Bh
扇区 15	3Ch ~ 3Eh	3Fh

1K 字节

表 1：MIFARE 1K 卡的内存结构

扇区 (共 32 个扇区，每个扇区包含 4 个连续的块)	数据块 (3 个块，每块 16 个字节)	尾部块 (1 个块，16 个字节)
扇区 0	00 ~ 02h	03h
扇区 1	04 ~ 06h	07h
..		
..		
扇区 30	78 ~ 7Ah	7Bh
扇区 31	7C ~ 7Eh	7Fh

2K 字节

扇区 (共 8 个扇区，每个扇区包含 16 个连续的块)	数据块 (15 个块，每块 16 个字节)	尾部块 (1 个块，16 个字节)
扇区 32	80 ~ 8Eh	8Fh
扇区 33	90 ~ 9Eh	9Fh
..		
..		
扇区 38	E0 ~ EEh	EFh
扇区 39	F0 ~ FEh	FFh

2K 字节

表 2：MIFARE 4K 卡的内存结构

字节号	0	1	2	3	页
序列号	SN0	SN1	SN2	BCC0	0
序列号	SN3	SN4	SN5	SN6	1
内部 / 锁	BCC1	Internal	Lock0	Lock1	2
OTP	OPT0	OPT1	OTP2	OTP3	3
数据读/写	Data0	Data1	Data2	Data3	4
数据读/写	Data4	Data5	Data6	Data7	5
数据读/写	Data8	Data9	Data10	Data11	6
数据读/写	Data12	Data13	Data14	Data15	7
数据读/写	Data16	Data17	Data18	Data19	8
数据读/写	Data20	Data21	Data22	Data23	9
数据读/写	Data24	Data25	Data26	Data27	10
数据读/写	Data28	Data29	Data30	Data31	11
数据读/写	Data32	Data33	Data34	Data35	12
数据读/写	Data36	Data37	Data38	Data39	13
数据读/写	Data40	Data41	Data42	Data43	14
数据读/写	Data44	Data45	Data46	Data47	15

512 位
或
64 字节

表 3：MIFARE Ultralight 卡的内存结构

例如：

// 要使用{TYPE A，密钥号 00h}验证块 04h。PC/SC V2.01，弃用

APDU = {FF 88 00 04 60 00h};

// 要使用{TYPE A，密钥号 00h}验证块 04h。PC/SC V2.07

APDU = {FF 86 00 00 05 01 00 04 60 00h}

注：MIFARE Ultralight 不需要进行验证，其内存可以自由访问。

6.1.3. 读二进制块（Read Binary Blocks）

此命令用于从 PICC 卡片中取回多个“数据块”。执行 Read Binary Blocks 命令前，必须先对数据块/尾部块进行验证。

Read Binary 命令的 APDU 结构（5 个字节）

命令	CLA	INS	P1	P2	Le
Read Binary Blocks	FFh	B0h	00h	块号	待读取的字节数

其中：

块号 1 个字节。起始块。

待读取的字节数 1 个字节。MIFARE 1K/4K 卡的待读字节的长度应该是 16 字节的倍数；MIFARE Ultralight 卡应该是 4 字节的倍数。

MIFARE Ultralight 卡的待读字节数最大为 16。

MIFARE 1K 卡的待读字节数最大为 48。（多块模式：3 个连续的块）

MIFARE 4K 卡的待读字节数最大为 240。（多块模式：15 个连续的块）

例 1：10h (16 个字节)。仅起始块。（单块模式）

例 2：40h (64 个字节)。从起始块至起始 + 3 块。（多块模式）

注：*出于安全因素考虑，多块模式仅用于访问数据块。尾部块不能在多块模式下访问，请使用单块模式对其进行访问。*

Read Binary Block 命令的响应结构（4/16 的倍数 + 2 个字节）

响应	响应数据域		
结果	数据（4/16 字节的倍数）	SW1	SW2

Read Binary Block 命令的响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

例如：

// 从二进制块 04h 中读取 16 个字节（MIFARE 1K/4K）

APDU = {FF B0 00 04 10h}

从二进制块 80h 开始读取 240 个字节（MIFARE 4K）

// 块 80h 至块 8Eh（15 个块）

APDU = {FF B0 00 80 F0h}

6.1.4. 更新二进制块（Update Binary Blocks)

此命令用于向 PICC 卡写入多个“数据块”。执行 Update Binary Blocks 命令前，必须先对数据块/尾部块进行验证。

Update Binary 命令的 APDU 结构（16 的倍数 + 5 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Update Binary Blocks	FFh	D6h	00h	块号	待更新的字节数	块数据 (16 字节的倍数)

其中：

块号 1 个字节。待更新的起始块

待更新的字节数 1 个字节。MIFARE 1K/4K 卡的待更新字节的长度应该是 16 字节的倍数；MIFARE Ultralight 卡是 4 字节的倍数。

MIFARE 1K 卡的待读字节数最大为 48。（多块模式；3 个连续的块）

MIFARE 4K 卡的待读字节数最大为 240。（多块模式；15 个连续的块）

块数据 16 字节的倍数 + 2 个字节，或 6 个字节。待写入二进制块的数据。

例 1：10h (16 个字节)。仅起始块。（单块模式）

例 2：30h (48 个字节)。从起始块至起始+2 块。（多块模式）

注：*出于安全因素考虑，多块模式仅用于访问数据块。尾部块不能在多块模式下访问，请使用单块模式对其进行访问。*

Update Binary Block 命令的响应状态码（2 个字节）

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

例如：

// 将 MIFARE 1K/4K 卡中的二进制块 04h 的数据更新为{00 01 ..0Fh}

APDU = {FF D6 00 04 10 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0Fh}

// 将 MIFARE Ultralight 卡中的二进制块 04h 的数据更新为{00 01 02 03h}

APDU = {FF D6 00 04 04 00 01 02 03h}

6.1.5. 值块操作（Value Block Operation）（INC, DEC, STORE）

此命令用于处理基于数值的交易，例如：增加值块的值。

Value Block Operation 命令的 APDU 结构（10 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域	
Value Block Operation	FFh	D7h	00h	块号	05h	VB_OP	VB_Value (4 个字节) {MSB ..LSB}

其中：

块号 1 个字节。待处理的值块

VB_OP 1 个字节。

00h = 将 VB_Value 存入该块，然后该块变为一个值块。

01h = 使值块的值增加 VB_Value。此命令仅适用于对值块的操作。

02h = 使值块的值减少 VB_Value。此命令仅适用于对值块的操作。

VB_Value 4 个字节。用于算数运算的数值，是一个有符号长整数（4 个字节）。

例 1：Decimal –4 = {FFh, FFh, FFh, FCh}

VB_Value			
MSB			LSB
FFh	FFh	FFh	FCh

例 2：Decimal 1 = {00h, 00h, 00h, 01h}

VB_Value			
MSB			LSB
00h	00h	00h	01h

Value Block Operation 命令的响应结构（2 个字节）

响应	响应数据域	
结果	SW1	SW2

Value Block Operation 响应码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

6.1.6. 读值块（Read Value Block）

此命令用于获取值块中的数值，仅适用于对值块的操作。

Read Value Block 命令的 APDU 结构（5 个字节）

命令	CLA	INS	P1	P2	Le
Read Value Block	FFh	B1h	00h	块号	04h

其中：

块号 1 个字节。待访问的值块

Read Value Block 的响应报文结构（4+2 个字节）

响应	响应数据域		
结果	值 {MSB ..LSB}	SW1	SW2

其中：

值 4 个字节。卡片返回的数值，是一个有符号长整数（4 个字节）。

例 1：Decimal –4 = {FFh, FFh, FFh, FCh}

值			
MSB			LSB
FFh	FFh	FFh	FCh

例 2：Decimal 1 = {00h, 00h, 00h, 01h}

值			
MSB			LSB
00h	00h	00h	01h

Read Value Block 命令的响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

6.1.7. 复制值块（Copy Value Block）

此命令用于将一个值块中的数值复制到另外一个值块。

Copy Value Block 命令的 APDU 结构（7 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域	
Value Block Operation	FFh	D7h	00h	源块号	02h	03h	目标块号

其中：

源块号 1 个字节。源值块中的值会被复制到目标值块。

目标块号 1 个字节。要恢复的值块。源值块和目标值块必须位于同一个扇区。

Copy Value Block 的响应报文结构（4+2 个字节）

响应	响应数据域	
结果	SW1	SW2

Copy Value Block 命令的响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

例如：

// 将数值“1”存入块 05h

APDU = {FF D7 00 05 05 00 00 00 00 01h}

// 读取值块 05h

APDU = {FF B1 00 05 04h}

将值块 05h 的值复制到值块 06h

APDU = {FF D7 00 05 02 03 06h}

// 使值块 05h 的值增加“5”

APDU = {FF D7 00 05 05 01 00 00 00 05h}

6.2. 访问符合 PCSC 标准的标签（ISO 14443-4）

基本上，所有符合 ISO 14443-4 标准的卡片（PICC）都可以理解符合 ISO 7816-4 规定的 APDU。ACR1281U-K1 读写器与符合 ISO 14443-4 标准的卡片进行通信时，需要对 ISO 7816-4 规定的 APDU 和响应进行转换。ACR1281U-K1 会在内部处理 ISO 14443 第 1-4 部分协议。

另外 MIFARE 1K, 4K, MINI 和 Ultralight 标签是通过 T=CL 模拟进行支持的，只要将 MIFARE 标签视作标准的 ISO 14443-4 标签即可。更多相关信息，请参阅“MIFARE Classic 存储标签的 PICC 命令”。

ISO 7816-4 规定的 APDU 报文的结构

命令	CLA	INS	P1	P2	Lc	命令数据域	Le
ISO 7816 第 4 部分规定的命令					命令数据域的长度		期望返回的响应数据的长度

ISO 7816-4 规定的响应报文的结构（数据 + 2 个字节）

响应	响应数据域		
结果	响应数据	SW1	SW2

通用的 ISO 7816-4 命令的响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

典型的操作顺序为：

1. 出示标签，并连接 PICC 界面。
2. 读取/更新标签的存储内容。

要实现这些：

1. **与标签建立连接。**

   标签的 ATR 为 3B 88 80 01 00 00 00 00 33 81 81 00 3Ah.

   其中，

   ATQB 应用数据 = 00 00 00 00，ATQB 协议信息 = 33 81 81。这是一个 ISO 14443-4 Type B 标签。

2. **发送 APDU，取随机数**

   << 00 84 00 00 08h

   >> 1A F7 F3 1B CD 2B A9 58h [90 00h]

   注：*对于 ISO 14443-4 Type A 标签来说，可以通过 APDU“FF CA 01 00 00h”来获取 ATS。*

例如：

// 从 ISO 14443-4 Type B PICC（ST19XR08E）中读取 8 个字节

APDU = {80 B2 80 00 08h}

CLA = 80h

INS = B2h

P1 = 80h

P2 = 00h

Lc = 无

命令数据域 = 无

Le = 08h

应答：00 01 02 03 04 05 06 07h [$9000h]

6.3. 访问 MIFARE DESFire 标签（ISO 14443-4）

MIFARE DESFire 支持 ISO7816-4 APDU 包模式和本地模式。一旦 MIFARE DESFire 标签被激活，发送至 MIFARE DESFire 标签的第一个 APDU 就会确定“命令的模式”。如果第一个 APDU 采用“本地模式”，则其余的 APDU 都必须是“本地模式”。同样地，如果第一个 APDU 采用“ISO 7816-4 APDU 包模式”，则其余的 APUDU 都必须是“ISO 7816-4 APDU 包模式”。

例 1：MIFARE DESFire ISO 7816-4 APDU 包

从 ISO 14443-4 Type A PICC（MIFARE DESFire）中读取 8 个字节的随机数

APDU = {90 0A 00 00 01 00 00h}

CLA = 90h; INS = 0Ah (MIFARE DESFire Instruction); P1 = 00h; P2 = 00h

Lc = 01h; Data In = 00h; Le = 00h (Le = 00h for maximum length)

应答：1A 29 06 2D 0D C4 00 A6h [$91AFh]

状态码[91 AFh]由 MIFARE DESFire 标准定义，详情请参阅 MIFARE DESFire 标准。

例 2：MIFARE DESFire 分页链接（ISO 7816 APDU 包模式）

// 在本例中，应用涉及到"分页链接"。

// 要获得 MIFARE DESFire 卡的版本号。

步骤 1：发送 APDU {90 60 00 00 00h}来获取第一个数据页。INS=60h

应答：04 01 01 00 02 18 05 91 AFh [$91AFh]

步骤 2：发送 APDU {90 AF 00 00 00}来获取第二个数据页。INS=AFh

应答：04 01 01 00 06 18 05 91 AFh [$91AFh]

步骤 3：发送 APDU {90 AF 00 00 00h}来获取最后一个数据页。INS=AFh

应答：04 52 5A 19 B2 1B 80 8E 36 54 4D 40 26 04 91 00h [$9100h]

例 3：MIFARE DESFire 本地命令

// 若本地 DESFire 命令更易于操作，则可以向读写器发送不带 ISO 7816 包的本地 MIFARE DESFire 命令。

从 ISO 14443-4 Type A PICC（MIFARE DESFire）中读取 8 个字节的随机数

APDU = {0A 00h}

应答：AF DC E3 4F 26 1D 2D 51 49h [$5149h]

其中，第一个字节“AFh”是 MIFARE DESFire 卡片返回的状态码。

应用程序可以对[$5149h]中的数据予以忽略。

例 4：MIFARE DESFire 分页链接 (本地模式)

// 在本例中，应用涉及到"分页链接"。

// 要获得 MIFARE DESFire 卡的版本号。

步骤 1：发送 APDU {60h} 来获取第一个数据页。INS=60h

应答：AF 04 01 01 00 02 18 05h [$1805h]

步骤 2：发送 APDU {AFh} 来获取第二个数据页。INS=AFh

应答：AF 04 01 01 00 06 18 05h [$1805h]

步骤 3：发送 APDU {AFh} 来获取最后一个数据页。INS=AFh

应答：00 04 52 5A 19 B2 1B 80 8E 36 54 4D 40 26 04h [$2604h]

注：*在 MIFARE DESFire 本地模式下，如果响应的长度大于 1，则不会在响应中出现状态码[90 00h]。但是如果响应的长度小于 2，则会根据 PCSC 的要求在响应中增加状态码[90 00h]。最短的响应长度为 2。*

6.4. 外设控制

读写器的外设控制命令通过 *PC_to_RDR_Escape* 函数来实现。

6.4.1. 获取固件版本号（Get Firmware Version）

此命令用于获取读写器的固件信息。

Get Firmware Version 命令的结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Get Firmware Version	E0h	00h	00h	18h	00h

Get Firmware Version 的响应结构（5 个字节 + 固件信息的长度）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	待接收的字节数	固件版本

例如：

响应 = E1 00 00 00 0F 41 43 52 31 32 35 31 55 5F 56 32 30 34 2E 30

固件版本号 (HEX) = 41 43 52 31 32 35 31 55 5F 56 32 30 34 2E 30

固件版本号 (ASCII) = “ACR1281U-K1_V204.0”

6.4.2. LED 控制（LED Control）

此命令用于控制 LED 的输出。

LED Control 命令的结构（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
LED Control	E0h	00h	00h	29h	01h	LED 状态

LED Control 命令的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	LED 状态

其中：

LED 状态（1 个字节）

LED 状态	描述	描述
Bit 0	红色 LED	1 = 开 0 = 关
Bit 1	绿色 LED	1 = 开 0 = 关
Bit 2 – 7	RFU	RFU

6.4.3. LED 状态（LED Status）

此命令用于检查当前 LED 的状态。

LED Status 命令的结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
LED Status	E0h	00h	00h	29h	00h

LED Status 命令的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	LED 状态

其中：

LED 状态（1 个字节）

LED 状态	描述	描述
Bit 0	红色 LED	1 = 开；0 = 关
Bit 1	绿色 LED	1 = 开；0 = 关
Bit 2 - 7	RFU	RFU

6.4.4. 蜂鸣器控制（Buzzer Control）

此命令用于控制蜂鸣器的输出。

Buzzer Control 命令的结构（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Buzzer Control	E0h	00h	00h	28h	01h	蜂鸣器持续时间

其中：

蜂鸣器持续时间 1 个字节。

00h = 关闭

01h - FFh = 持续时间（单位：10ms）

Buzzer Control 命令的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	00h

6.4.5. 蜂鸣器状态（Buzzer Status）

此命令用于检查当前蜂鸣器的状态。

Buzzer Status 命令的结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Buzzer Status	E0h	00h	00h	28h	00h

Buzzer Status 命令的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	00h

6.4.6. 设置 LED 和蜂鸣器状态指示器（Set LED and Buzzer Status Indicator Behavior）

此命令用于设置 LED 和蜂鸣器作为状态指示器的各种操作。

注：*该设置将保存在非易失存储器中。*

Set LED and Buzzer status Indicator Behaviors 的命令结构（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Set LED and Buzzer Status Indicator Behaviors	E0h	00h	00h	21h	01h	操作

其中：

操作（1 个字节）

操作	模式	描述
Bit 0	ICC 激活状态 LED	显示 ICC 接口的激活状态。 1 =启用；0 =禁用
Bit 1	PICC 轮询状态 LED	显示 PICC 轮询状态。 1 =启用；0 =禁用
Bit 2	PICC 激活状态蜂鸣器	每秒钟发出哔的一声，表示 PICC 处于激活状态 1 =启用；0 =禁用
Bit 3	RFU	RFU
Bit 4	卡片插入和卡片移出事件蜂鸣器	每次检测到卡片插入或者卡片移出就会发出哔的一声。（ICC 和 PICC） 1 =启用；0 =禁用
Bit 5	RC531 复位指示蜂鸣器	RC531 复位时发出哔的一声. 1 =启用；0 =禁用
Bit 6	独享模式状态蜂鸣器。 # ICC 或 PICC 接口只能激活一个	独享模式被激活时发出哔的一声. 1 =启用；0 =禁用
Bit 7	卡片操作闪烁 LED	LED 在卡片（PICC 或 ICC）被访问时会闪烁.

注：*操作的默认值 = F3h*

Set LED and Buzzer Status Indicator Behaviors 的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	默认操作

6.4.7. 读取 LED 和蜂鸣器状态指示器（Read LED and Buzzer Status Indicator Behavior）

此命令用于读取 LED 和蜂鸣器作为状态指示器的当前默认操作。

Read LED and Buzzer Status Indicator Behaviors 的命令结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read LED and Buzzer Status Indicator Behaviors	E0h	00h	00h	21h	00h

Read LED and Buzzer Status Indicator Behaviors 的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	操作

其中：

操作（1 个字节）

操作	模式	描述
Bit 0	ICC 激活状态 LED	显示 ICC 接口的激活状态。 1 =启用；0 =禁用
Bit 1	PICC 轮询状态 LED	显示 PICC 轮询状态。 1 =启用；0 =禁用
Bit 2	PICC 激活状态蜂鸣器	每秒钟发出哔的一声，表示 PICC 处于激活状态 1 =启用；0 =禁用
Bit 3	RFU	RFU
Bit 4	卡片插入和卡片移出事件蜂鸣器	每次检测到卡片插入或者卡片移出就会发出哔的一声。（ICC 和 PICC） 1 =启用；0 =禁用
Bit 5	RC531 复位指示蜂鸣器	RC531 复位时发出哔的一声. 1 =启用；0 =禁用
Bit 6	独享模式状态蜂鸣器。 # ICC 或 PICC 接口只能激活一个	独享模式被激活时发出哔的一声. 1 =启用；0 =禁用
Bit 7	卡片操作闪烁 LED	LED 在卡片（PICC 或 ICC）被访问时会闪烁.

***注：** 操作的默认值 = F3h*

6.4.8. 设置自动 PICC 轮询（Set Automatic PICC Polling）

此命令用于设置读写器的轮询模式。

每当读写器连接到计算机上，读写器的 PICC 轮询功能就会启动 PICC 扫描，以确定是否有 PICC 被放置于/移出了内置天线的范围。

您可以发送一条命令来停用 PICC 轮询功能。该命令通过 PCSC Escape Command 接口发送。为了满足节能要求，PICC 闲置，或者找不到 PICC 的时候，我们提供了几种关闭天线场的特殊模式。在省电模式下，读写器会消耗更低的电能。

注：*该设置将保存在非易失存储器中。*

Set Automatic PICC Polling 的命令结构（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Set Automatic PICC Polling	E0h	00h	00h	23h	01h	轮询设置

Set Automatic PICC Polling 的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	轮询设置

其中：

轮询设置（1 个字节）

轮询设置	描述	描述
Bit 0	自动 PICC 轮询	1 = 启用；0 = 禁用
Bit 1	如果没有找到 PICC，关闭天线场	1 = 启用；0 = 禁用
Bit 2	如果 PICC 闲置，关闭天线场。	1 = 启用；0 = 禁用
Bit 3	检测到 PICC 后将其激活	1 = 启用；0 = 禁用
Bit 5 ..4	PICC 轮询间隔	<Bit 5 – Bit 4> <0 – 0> = 250 ms <0 – 1> = 500 ms <1 – 0> = 1000 ms <1 – 1> = 2500 ms
Bit 6	RFU	
Bit 7	强制执行 ISO 14443A 第 4 部分	1 = 启用；0 = 禁用。

注：*操作的默认值 = 8Fh*

提示：

1. *建议启用“如果 PICC 闲置，关闭天线场”选项，这样闲置的 PICC 就不会一直暴露在天线场中，可以防止 PICC“发热”。*
2. *PICC 轮询间隔时间越长，节能效果越好。然而，PICC 轮询的响应时间也会增加。在节能状态下，空闲时的电流消耗约为 60 mA；而在非节能状态下，空闲时的电流消耗约为 130 mA。**注***

：空闲时的电流消耗=PICC 尚未激活。

3. *读写器会自动激活“ISO 14443A-4 PICC”的 ISO 14443A-4 模式。B 类 PICC 不会受此选项影响。*
4. *JCOP30 卡片有两种模式：ISO 14443A-3（MIFARE 1K）和 ISO 14443A-4 模式。一旦 PICC 被激活，应用就必须要选定一种模式。*

6.4.9. 读取自动 PICC 轮询（Read Automatic PICC Polling）

此命令用于检查当前的 PICC 轮询设置。

Read Automatic PICC Polling 的命令结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read Automatic PICC Polling	E0h	00h	00h	23h	00h

Read the Configure mode 的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	轮询设置

其中：

轮询设置（1 个字节）

轮询设置	描述	描述
Bit 0	自动 PICC 轮询	1 = 启用；0 = 禁用
Bit 1	如果没有找到 PICC，关闭天线场	1 = 启用；0 = 禁用
Bit 2	如果 PICC 闲置，关闭天线场。	1 = 启用；0 = 禁用
Bit 3	检测到 PICC 后将其激活	1 = 启用；0 = 禁用
Bit 5 ..4	PICC 轮询间隔	<Bit 5 – Bit 4> <0 – 0> = 250 ms <0 – 1> = 500 ms <1 – 0> = 1000 ms <1 – 1> = 2500 ms
Bit 6	RFU	
Bit 7	强制执行 ISO 14443A 第 4 部分	1 = 启用；0 = 禁用。

注：*操作的默认值* = *8Fh*

6.4.10. 设置 PICC 操作参数（Set the PICC Operating Parameter）

此命令用于设置 PICC 操作参数。

注：*该设置将保存在非易失存储器中。*

Set the PICC Operating Parameter 命令的结构（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Set the PICC Operating Parameter	E0h	00h	00h	20h	01h	操作参数

Set the PICC Operating Parameter 命令的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	操作参数

其中：

操作参数（1 个字节）

操作参数	参数	描述	选项
Bit 0	ISO 14443 A 类	PICC 轮询要检测的标签.	1 = 检测 0 = 跳过
Bit 1	ISO 14443 B 类		1 = 检测 0 = 跳过
Bit 2 - 7	RFU	RFU	RFU

注：*操作参数的默认值 = 03h*

6.4.11. 读取 PICC 操作参数（Read the PICC Operating Parameter）

此命令用于检查当前的 PICC 操作参数。

Read the PICC Operating Parameter 的命令结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read the PICC Operating Parameter	E0h	00h	00h	20h	00h

Read the PICC Operating Parameter 命令的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	操作参数

其中：

操作参数（1 个字节）

操作参数	参数	描述	选项
Bit 0	ISO 14443 A 类	PICC 轮询要检测的标签.	1 = 检测 0 = 跳过
Bit 1	ISO 14443 B 类		1 = 检测 0 = 跳过
Bit 2 - 7	RFU	RFU	RFU

注：*操作参数的默认值 = 03h*

6.4.12. 设置自动 PPS（Set Auto PPS）

每次识别出 PICC，读写器都会尝试更改由最大连接速度定义的 PCD 和 PICC 间的通信速率。若卡片不支持建议的连接速度，读写器会尝试以较慢的速度与卡片建立连接。

注：*该设置将保存在非易失存储器中。*

Set Auto PPS 的命令结构（7 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Set Auto PPS	E0h	00h	00h	24h	01h	最大速度

Set Auto PPS 的响应结构（9 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	E1h	00h	00h	00h	02h	最大速度	当前速度

其中：

最大速度 1 个字节，最高的速度。

当前速度 1 个字节，当前的速度。

值可以为：

106k bps = 00h（默认设置）

212k bps = 01h

424k bps = 02h

848k bps = 03h

无自动 PPS = FFh

注：

1. *通常来讲，应用程序应当知道正在被使用的 PICC 的最大连接速率，周围环境也会对最大可达速率有所影响。读写器只是使用建议的通信速率来与 PICC 进行对话。如果 PICC 或周围环境不能满足建议的通信速率的要求，PICC 将变得不能访问。*
2. *读写器支持不同的数据发送速度和接收速度。*

6.4.13. 读取自动 PPS（Read Auto PPS）

此命令用于检查当前的自动 PPS 设置。

Read Auto PPS 的命令结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read Auto PPS	E0h	00h	00h	24h	00h

Set Auto PPS 命令的响应结构（7 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	E1h	00h	00h	00h	02h	最大速度	当前速度

其中：

最大速度 1 个字节，最高的速度。

当前速度 1 个字节，当前的速度。

可以为：

106k bps = 00h（默认设置）

212k bps = 01h

424k bps = 02h

848k bps = 03h

无自动 PPS = FFh

6.4.14. 天线场控制（Antenna Field Control）

此命令用于打开/关闭天线场。

Antenna Field Control 的命令结构（6 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Antenna Field Control	E0h	00h	00h	25h	01h	状态

其中：

状态 1 个字节。

01h = 启用天线场

00h = 停用天线场

Antenna Field Control 命令的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	状态

其中：

状态 1 个字节。

00h = PICC 关闭

01h = PICC 闲置[准备好进行非接触式标签轮询，但没有检测到此类标签]

02h = PICC 就绪[PICC 请求（参阅 ISO 14443）成功，也就是说检测到了非接触式标签]

03h = PICC 已选定[PICC 选择（ 参阅 ISO 14443）成功]

04h = PICC 已激活[PICC 激活（参阅 ISO 14443）成功，准备好进行 APDU 交换]

注：*关闭天线场前，要确保自动 PICC 轮询功能已经停用。*

6.4.15. 读取天线场状态（Read Antenna Field Status）

此命令用于检查当前的天线场状态。

Read Antenna Field Status 的命令结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read Antenna Field Status	E0h	00h	00h	25h	00h

Read Antenna Field Status 的响应结构（6 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	状态

其中：

状态 1 个字节。

00h = PICC 关闭

01h = PICC 闲置 [准备好进行非接触式标签轮询，但没有检测到此类标签]

02h = PICC 就绪 [PICC 请求（参阅 ISO 14443）成功，也就是说检测到了非接触式标签]

03h = PICC 已选定 [PICC 选择（参阅 ISO 14443）成功]

04h = PICC 已激活 [PICC 激活（参阅 ISO 14443）成功，准备好进行 APDU 交换]

6.4.16. 用户额外保护时间设置（User Extra Guard Time Setting）

此命令用于为 ICC 和 SAM 通信设置额外保护时间。

注：*该设置将保存在非易失存储器中。*

User Extra Guard Time Setting 的命令格式（7 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域	
User Extra Guard Time Setting	E0h	00h	00h	2Eh	02h	ICC UserGuardTime	SAM UserGuardTime

User Extra Guard Time Setting 的响应格式（7 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	E1h	00h	00h	00h	02h	ICC UserGuardTime	SAM UserGuardTime

其中：

ICC UserGuardTime 1 个字节。ICC 的用户保护时间值。

SAM UserGuardTime 1 个字节。SAM 的用户保护时间值。

6.4.17. 读取用户额外保护时间（Read User Extra Guard Time）

此命令用于读取为 ICC 和 SAM 通信设置的额外保护时间。

Read User Extra Guard Time 的命令格式（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read User Extra Guard Time	E0h	00h	00h	2Eh	00h

Read User Extra Guard Time 的响应格式（7 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	E1h	00h	00h	00h	01h	ICC UserGuardTime	SAM UserGuardTime

其中：

ICC UserGuardTime 1 个字节。ICC 的用户保护时间值。

SAM UserGuardTime 1 个字节。SAM 的用户保护时间值。

6.4.18. “616C”自动操作选项设置（“616C” Auto Handle Option Setting）

（T=0 ACOS5 的可选命令）

此命令用于设置“61 6Ch”自动操作选项，

注：*该设置将保存在非易失存储器中。*

“616C” Auto Handle Option Setting 的命令格式（7 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域	
“616C” Auto Handle Option Setting	E0h	00h	00h	32h	02h	ICC 选项	SAM 选项

“616C” Auto Handle Option Setting 的响应格式（7 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	E1h	00h	00h	00h	02h	ICC 选项	SAM 选项

其中：

ICC 选项/ SAM 选项 1 个字节。用户保护时间值。

FFh = 启用“616C”自动操作

00h = 停用“616C”自动操作（默认）

6.4.19. 读取“616C”自动操作选项（Read “616C” Auto Handle Option）

此命令用于读取“616C”自动操作选项。

Read “616C” Auto Handle Option 的命令格式（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read “616C” Auto Handle Option	E0h	00h	00h	32h	00h

Read “616C” Auto Handle Option 的响应格式（7 个字节）

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	E1h	00h	00h	00h	02h	ICC 选项	SAM 选项

其中：

ICC 选项/ SAM 选项 1 个字节。用户保护时间值。

FFh = 启用“616C”自动操作

00h = 停用“616C”自动操作（默认）

6.4.20. 刷新接口状态（Refresh the Interface Status）

此命令用于对接口的状态进行刷新。.

Refresh the Interface Status 的命令格式（5 个字节）

命令	Class	INS	P1	P2	Lc
Refresh the Interface Status	E0h	00h	00h	2Dh	01h

Refresh the Interface Status 的响应格式（6 个字节）

响应	Class	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	待刷新的接口

其中：

接口号（1 个字节）： Bit 0 = ICC 接口

Bit 1 = PICC 接口

Bit 2 = SAM 接口

6.4.21. 读取当前状态（Read the Existing Status）

此命令用于读取接口的当前状态。

Read the Existing Status 的命令格式（5 个字节)

命令	Class	INS	P1	P2	Lc
Read the existing status	E0h	00h	00h	2Dh	00h

Read the Existing Status 的响应结构（6 个字节）

响应	Class	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	待刷新的接口

其中：

接口号（1 个字节）： Bit 0 = ICC 接口

Bit 1 = PICC 接口

Bit 2 = SAM 接口