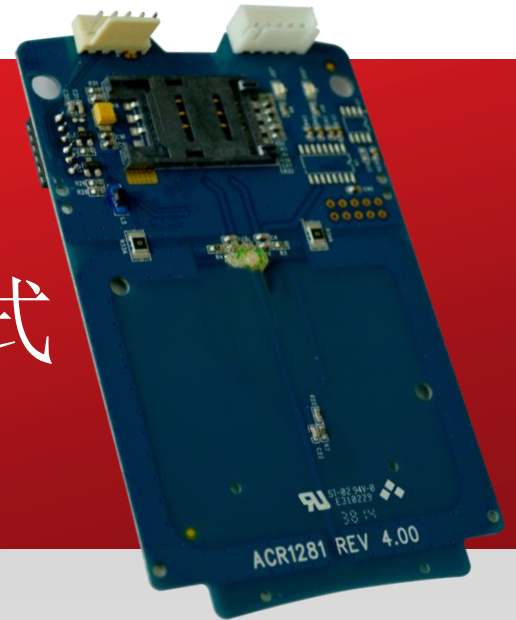




Advanced Card Systems Ltd.
Card & Reader Technologies

ACM1281U-C7

帶 **SAM** 卡槽的非接觸式
读写器模块 (USB 接口)



参考手册 V1.03



版本历史

发布日期	修订说明	版本号
2015-04-22	<ul style="list-style-type: none">初始发布	1.00
2017-05-31	<ul style="list-style-type: none">更新 2.0 节-特性	1.01
2020-06-01	<ul style="list-style-type: none">更新 2.0 节-特性更新 5.0 节-主机编程（联机）API	1.02
2020-06-16	<ul style="list-style-type: none">更新 5.2.5 节-设置 LED 和蜂鸣器默认操作（Set Default LED and Buzzer Behaviors）更新 5.2.6 节-读取 LED 和蜂鸣器默认操作（Read Default LED and Buzzer Behaviors）	1.03



目录

1.0.	简介	5
2.0.	特性	6
3.0.	ACM1281U-C7 体系架构	7
3.1.	读写器功能框图	7
3.2.	PC/SC 驱动、PICC 及 SAM 间的通信	7
4.0.	硬件设计	8
4.1.	USB	8
4.1.1.	通信参数	8
4.1.2.	端点	8
4.2.	非接触式智能卡接口	8
4.2.1.	载波频率	8
4.2.2.	卡片轮询	8
4.3.	用户接口	9
4.3.1.	蜂鸣器	9
4.3.2.	LED 指示灯	9
5.0.	主机编程 (联机) API	10
5.1.	非接触式智能卡协议	10
5.1.1.	ATR 的生成	10
5.1.2.	非接触接口的私有 APDU 指令	13
5.1.3.	MIFARE 1K/4K 存储卡的 PICC 命令 (T=CL 模拟)	14
5.1.4.	访问符合 PC/SC 标准的标签 (ISO 14443-4)	24
5.1.5.	访问 MIFARE DESFire 标签 (ISO 14443-4)	26
5.2.	外设控制	28
5.2.1.	获取固件版本 (Get Firmware Version)	28
5.2.2.	LED 控制 (LED Control)	29
5.2.3.	LED 状态 (LED Status)	30
5.2.4.	蜂鸣器控制 (Buzzer Control)	31
5.2.5.	设置 LED 和蜂鸣器默认操作 (Set Default LED and Buzzer Behaviors)	32
5.2.6.	读取 LED 和蜂鸣器默认操作 (Read Default LED and Buzzer Behaviors)	33
5.2.7.	设置自动 PICC 轮询 (Set Automatic PICC Polling)	34
5.2.8.	读取自动 PICC 轮询 (Read Automatic PICC Polling)	36
5.2.9.	手动 PICC 轮询 (Manual PICC Polling)	37
5.2.10.	设置 PICC 操作参数 (Set PICC Operating Parameter)	38
5.2.11.	读取 PICC 操作参数 (Read PICC Operating Parameter)	39
附录 A.	非接触式应用基本流程	40
附录 B.	扩展的 APDU 示例	41
附录 C.	直接命令 (Escape Command) 示例	43
附录 D.	ACR128 兼容性	44



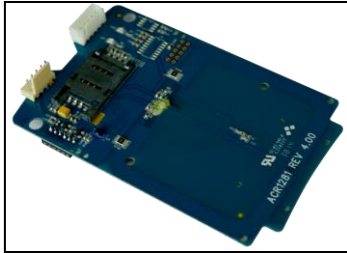
图目录

图 1	: ACM1281U-C7 读写器功能框图	7
图 2	: ACM1281U-C7 体系架构	7

表目录

表 1	: USB 接口配线	8
表 2	: 蜂鸣器事件说明	9
表 3	: LED 指示灯	9
表 4	: ISO 14443 第 3 部分规定的 ATR 格式	10
表 5	: ISO 14443 第 4 部分规定的 ATR 格式	12
表 6	: MIFARE 1K 卡的内存结构	17
表 7	: MIFARE 4K 卡的内存结构	17
表 8	: MIFARE Ultralight 卡的内存结构	18

1.0. 简介



ACM1281U-C7 是一款工作频率为 13.56 MHz 的非接触式读写器模块，带有一个 SAM 卡槽，采用 USB 接口，能够快速而轻松地集成到嵌入式系统。它采用 USB CCID 类驱动程序，接受来自计算机应用程序的卡片命令。

ACM1281U-C7 采用板线一体设计，可选配 USB 线，支持固件升级和扩展的 APDU 等特性。

它支持 ISO 14443 第 1-4 部分的 A 类和 B 类卡以及 MIFARE Classic® 系列卡片。另外它内置一个符合 ISO 7816 标准的 A 类 SAM（安全存取模块）卡槽，能够与 SAM 卡配合使用，实现更高级别的安全性。ACM1281U-C7 能够以高达 848 Kbps 的速度与非接触式卡片进行通信，非常适合高要求的智能卡应用，例如自动售卖机的支付系统、自助服务终端、游戏机以及其他集成系统。

本参考手册将详细介绍如何执行 PC/SC APDU 命令来支持非接触式接口和 SAM 卡,以及如何控制 ACM1281U-C7 的外设。



2.0. 特性

- USB 全速接口
- 智能卡读写器
 - 非接触式接口：
 - 读写速率高达 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 类和 B 类卡，以及 MIFARE Classic 系列卡
 - 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 KB）
 - SAM接口：
 - 1个SAM卡槽
 - ISO 7816 A类（5 V）SAM卡槽
- 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¹
- 符合下列标准：
 - ISO 14443
 - ISO 7816
 - PC/SC
 - CCID
 - CE
 - FCC
 - RoHS
 - REACH
 - Microsoft® WHQL

¹ 使用 ACS 定义的 Android 库

3.0. ACM1281U-C7 体系架构

3.1. 读写器功能框图

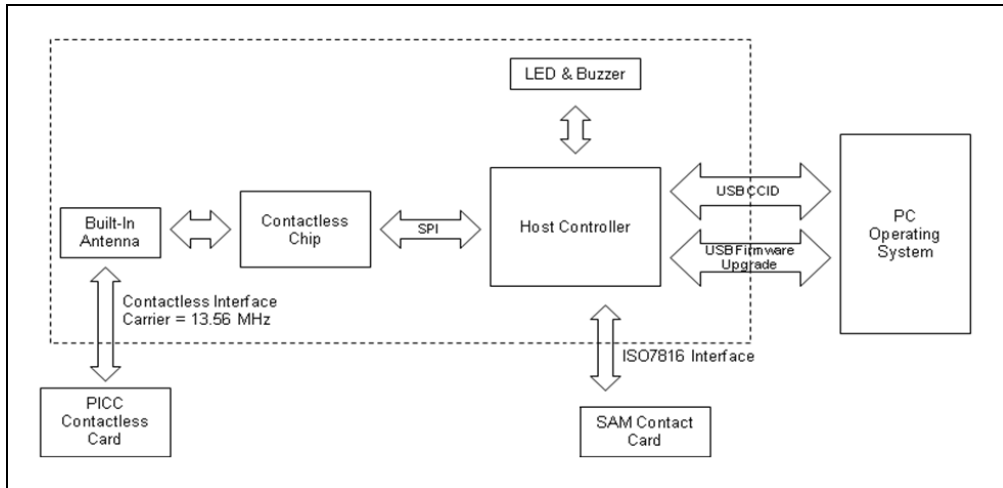


图1：ACM1281U-C7 读写器功能框图

3.2. PC/SC 驱动、PICC 及 SAM 间的通信

ACM1281U-C7 与计算机之间采用 CCID 协议。而 PICC 和 SAM 间的通信则符合 PC/SC 标准。

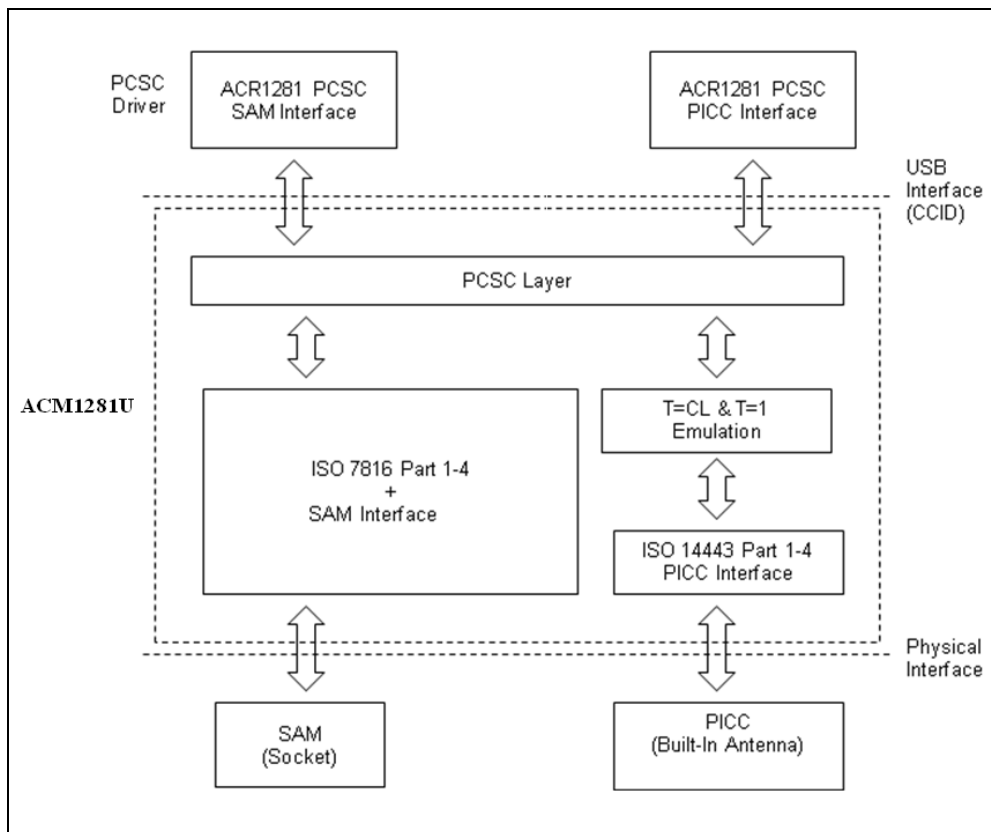


图2：ACM1281U-C7 体系架构

4.0. 硬件设计

4.1. USB

ACM1281U-C7 通过符合 USB 标准的 USB 接口与计算机连接。

4.1.1. 通信参数

ACM1281U-C7 按照 USB2.0 规范的要求通过 USB 接口与计算机连接，支持 USB 全速模式，速率为 12Mbps。

引脚	信号	功能
1	V _{BUS}	为读写器提供+5 V 的电源
2	D-	ACM1281U-C7 和 PC 间以差分信号传输数据
3	D+	ACM1281U-C7 和 PC 间以差分信号传输数据
4	GND	参考电压等级

表1：USB 接口配线

注：为了使 ACM1281U-C7 通过 USB 接口正常运行，应该先安装设备驱动程序。

4.1.2. 端点

ACM1281U-C7 通过如下的端点与主计算机进行通信。

Control Endpoint – 用于设置和控制

Bulk-OUT – 用于从主计算机发送至 ACM1281U-C7 的命令（数据包大小为 64 字节）

Bulk-IN – 用于从 ACM1281U-C7 发送至主计算机的响应（数据包大小为 64 字节）

Interrupt-IN – 用于从 ACM1281U-C7 发送至主计算机的卡片状态报文（数据包大小为 8 字节）

4.2. 非接触式智能卡接口

ACM1281U-C7 与非接触卡之间的接口符合 ISO 14443 标准协议，并进行了某些限制或提升来增强 ACM1281U-C7 的实用功能。

4.2.1. 载波频率

ACM1281U-C7 的载波频率为 13.56MHz。

4.2.2. 卡片轮询

ACM1281U-C7 会自动检测进入工作场内的非接触卡。此功能支持 ISO 14443-4 的 A 类卡和 B 类卡，以及 MIFARE 卡。



4.3. 用户接口

4.3.1. 蜂鸣器

ACM1281U-C7 配有一个单音蜂鸣器，用于指示“卡插入”和“卡移出”事件。

事件	蜂鸣器
1. 读写器上电，初始化成功。	鸣响 1 次
2. 卡片插入(PICC)	鸣响 1 次
3. 卡片移出(PICC)	鸣响 1 次

表2：蜂鸣器事件说明

4.3.2. LED 指示灯

ACM1281U-C7 配有 LED 指示灯，用于显示接触式和非接触式接口的状态。红色 LED 是 PICC 状态指示灯，绿色 LED 是 ICC 状态指示灯。

读写器状态	红色 LED PICC 指示器	绿色 LED ICC 指示器
1. 无法找到 PICC，或者 PICC 存在但无激活。	5 秒闪烁一次	N/A
2. PICC 存在并激活。	开	N/A
3. PICC 操作中。	闪烁	N/A

表3：LED 指示灯

5.0. 主机编程（联机）API

5.1. 非接触式智能卡协议

5.1.1. ATR 的生成

读写器检测到 PICC 后，一个 ATR 会被发送至 PC/SC 驱动来识别 PICC。

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

字节	值（十六进制）	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 至 3+N	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
	RID		注册应用供应商标识(RID) # A0 00 00 03 06h
	SS		标准字节
	C0h..C1h		卡片名称字节
	00 00 00 00h		RFU
4+N	UU	TCK	T0 至 Tk 的所有字节按位异或

表4：ISO 14443 第 3 部分规定的 ATR 格式

例:

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}

ATR											
初始字符	T0	TD1	TD2	T1	Tk	长度	RID	标准	卡片名称	RFU	TCK
3Bh	8Fh	80h	01h	80h	4Fh	0Ch	A0 00 00 03 06h	03h	00h 01h	00 00 00 00h	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)

[00 02h] (MIFARE 4K)

[00 03h] (MIFARE Ultralight)

[00 26h] (MIFARE Mini)

[00 36h] (MIFARE PLUS SL1_2K)

[00 37h] (MIFARE PLUS SL1_4K)

[00 38h] (MIFARE PLUS SL2_2K)

[00 39h] (MIFARE PLUS SL2_4K)

[00 3Ah] (MIFARE Ultralight C)

[FF 28h] JCOP 30

FF SAK 未定义标签

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

字节	值 (十六进制)	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 至 3 + N	XX	T1	历史字节: ISO 14443A:
	XX	Tk	

字节	值 (十六进制)	标记	描述						
	XX XX		来自 ATS 响应的历史字节。参考 ISO 14443-4 标准。 ISO 14443B: <table border="1"> <thead> <tr> <th>Byte1-4</th> <th>Byte5-7</th> <th>Byte8</th> </tr> </thead> <tbody> <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> </tbody> </table>	Byte1-4	Byte5-7	Byte8	ATQB 的应用数据	ATQB 的协议信息字符	高半字节 =ATTRIB 命令的 MBLI; 低半字节 (RFU)=0
Byte1-4	Byte5-7	Byte8							
ATQB 的应用数据	ATQB 的协议信息字符	高半字节 =ATTRIB 命令的 MBLI; 低半字节 (RFU)=0							
4+N	UU	TCK	T0 至 Tk 的所有字节按位异或						

表5：ISO 14443 第 4 部分规定的 ATR 格式

例 1：来自 MIFARE® DESFire®的 ATR 如下：

MIFARE DESFire (ATR) = 3B 81 80 01 80 80h (6 字节 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 类的 PICC 会返回 ATS。

APDU 命令 = FF CA 01 00 00h

APDU 响应 = 06 75 77 81 02 90 00h

ATS = {06 75 77 81 02 80h}

例 2：来自 EZ-Link 的 ATR 如下：

EZ-Link (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

5.1.2. 非接触接口的私有 APDU 指令

5.1.2.1. 获取数据 (Get Data)

此命令用于返回“已经建立连接的卡片”的序列号或 ATS。

命令

命令	CLA	INS	P1	P2	Le
Get Data	FFh	CAh	00h 01h	00h	00h (全长)

如果 P1 = 00h, 获取 UID

响应	UID	UID	SW1	SW2
结果	LSB			MSB		

如果 P1 = 01H, 获取 ATS (仅适用于 ISO 14443 的 A 类卡)

响应	响应数据域		
结果	ATS	SW1	SW2

响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
警告	62 82h	UID/ATS 的末尾先于 Le 字节到达 (Le 大于 UID 的长度)。
错误	6C XXh	长度错误 (错误的 Le: 'XX'表示确切的数字), 如果 Le 小于 UID 的长度。
错误	63 00h	操作失败。
错误	6A 81h	不支持此功能

例 1: 获取“已经建立连接的 PICC”的序列号

```
UINT8 GET_UID[5] = {FF CA 00 00 00h};
```

例 2: 获取“已经建立连接的 ISO14443-A PICC”的 ATS

```
UINT8 GET_ATS[5] = {FF CA 01 00 00h};
```

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

5.1.3.1. 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：易失密钥位置和非易失密钥位置。

命令

命令	CLA	INS	P1	P2	Le	命令数据域
Load Authentication Keys	FFh	82h	密钥结构	密钥号	06h	密钥

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

20h = 密钥被载入读写器的非易失存储器

其它 = 保留

密钥号 (1 个字节)

00h – 1Fh = 用于存储密钥的非易失存储器。密钥被永久地存在读写器中，即使读写器与电脑断开连接也不会被擦除。读写器的非易失存储器内可以存储最多 32 组密钥。

20h (过程密钥) = 用于临时存储密钥的易失存储器。读写器与电脑断开连接的时候，密钥会被擦除。易失存储器只有一个。易失密钥可以用作不同会话的过程密钥。默认值 = FF FF FF FF FF FFh.

密钥 (6 个字节)

载入读写器的密钥值。

例如：{FF FF FF FF FF FFh}

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h (表示操作成功完成)

= 63 00h (表示操作失败)



例 1:

向非易失性存储器位置 **05h** 加载密钥 {FF FF FF FF FF FFh}。

APDU = {FF 82 20 **05** 06 FF FF FF FF FF FFh}

向易失性存储器位置 **20h** 加载密钥 {FF FF FF FF FF FFh}。

APDU = {FF 82 00 **20** 06 FF FF FF FF FF FFh}

注:

1. 应用程序需要了解所有正在使用的密钥。出于安全方面的考虑，建议将所有需要的密钥存储在非易失性存储器内。任何应用都无法读取易失性和非易失性存储器的内容。
2. 直到读写器复位或下电，易失性存储器的内容“过程密钥 20h”才会失效。过程密钥适于存储经常变化的密钥值。被存储在“内部 RAM”中，而非易失密钥存储在“EEPROM”中。EEPROM 相对于内部 RAM 速度稍慢。
3. 不建议使用“非易失密钥位置 00-1Fh”来存储任何经常变化的“临时密钥”。“非易失密钥”主要是用于存储不经常变化的“密钥值”。如果“密钥值”会不时变化，则可以将其存储在“易失密钥位置 20h”。

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

此命令用于使用存储在读写器内的密钥来验证 MIFARE 1K/4K 卡(PICC)。其中用到两种认证密钥: Type_A 和 Type_B。

命令

命令	CLA	INS	P1	P2	P3	命令数据域
Authentication 6 字节 (弃用)	FFh	88h	00h	块号	密钥类型	密钥号

命令	CLA	INS	P1	P2	Lc	命令数据域
Authentication 10 字节	FFh	86h	00h	00h	05h	认证数据字节

其中:

认证数据字节 (5 个字节)

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

其中:

块号 (1 个字节)

待验证的存储块。

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, 读取同一个扇区内的其他块不再需要进行验证。详情请参考 MIFARE 1K/4K 卡标准。

密钥类型 (1 个字节)

60h = 密钥被用作 Key A 密钥进行验证

61h = 密钥被用作 Key B 密钥进行验证

密钥号 (1 个字节)

00h – 1Fh = 用于存储密钥的非易失存储器。密钥被永久地存在读写器中, 即使读写器与电脑断开连接也不会被擦除。读写器的非易失存储器内可以存储最多 32 组密钥。

20h (过程密钥) = 用于临时存储密钥的易失存储器。读写器与电脑断开连接的时候, 密钥会被擦除。易失存储器只有一个。易失密钥可以用作不同会话的过程密钥。默认值 = FF FF FF FF FF FFh。



响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h (表示操作成功完成)
= 63 00h (表示操作失败)

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)	} 1 KB
扇区 0	00h ~ 02h	03h	
扇区 1	04h ~ 06h	07h	
..			
..			
扇区 14	38h ~ 0Ah	3Bh	
扇区 15	3Ch ~ 3Eh	3Fh	

表6：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)	} 2 KB
扇区 0	00h ~ 02h	03h	
扇区 1	04h ~ 06h	07h	
...			
...			
扇区 30	78h ~ 7Ah	7Bh	
扇区 31	7Ch ~ 7Eh	7Fh	

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)	} 2 KB
扇区 32	80h ~ 8Eh	8Fh	
扇区 33	90h ~ 9Eh	9Fh	
...			
...			
扇区 38	E0h ~ EEh	EFh	
扇区 39	F0h ~ FEh	FFh	

表7：MIFARE 4K 卡的内存结构



例 1:

通过下列特征验证 Block 04h: Key A, key number 00h, from PC/SC V2.01 (Obsolete).

APDU = { FF 88 00 04 60 00h }

例 2:

类似于上面的例子，通过下列特征验证 Block 04h: Key A, key number 00h, from PC/SC V2.07.

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

注：由于 MIFARE® Ultralight 的用户数据区域可以自由访问，所以 MIFARE Ultralight 不需要通过验证。

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

512 位
或者
64 字节

表8：MIFARE Ultralight 卡的内存结构

5.1.3.3. 读取二进制块 (Read Binary Blocks)

此命令用于从 PICC 卡片中获取多个“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。

命令

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	MIFARE 1K/4K 卡的待读字节的长度可以是 16 字节的倍数；MIFARE Ultralight 卡的待读字节的长度可以是 4 字节的倍数（1 个字节） 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被读写，请使用单块模式对其进行读写。

响应

响应	响应数据域		
结果	数据 (4 或 16 字节的倍数)	SW1	SW2

其中：

SW1 SW2	= 90 00h（表示操作成功完成） = 63 00h（表示操作失败）
----------------	--

例 1：从二进制块 04h 中读取 16 个字节（MIFARE 1K 或 4K）。

APDU = { FF B0 00 04 10h }

例 2：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 F0 }

5.1.3.4. 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。

命令

命令	CLA	INS	P1	P2	Le	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)
起始块

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

待读取的字节数 MIFARE 1K/4K 卡的待读字节的长度可以是 16 字节的倍数; MIFARE Ultralight 卡的待读字节的长度可以是 4 字节的倍数 (1 个字节)

MIFARE Ultralight 卡的待更新字节数最大为 16。

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

MIFARE 4K 卡的待更新字节数最大为 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: 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被读写, 请使用单块模式对其进行读写。

响应

响应	响应数据域	
结果	SW1	SW2

其中:

SW1 SW2 = 90 00h (表示操作成功完成)
= 63 00h (表示操作失败)

例 1: 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 }

例 2: 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 }

5.1.3.5. 值块操作 (Value Block Operation) (Increment, Decrement, Store)

此命令用于对基于数值的交易进行操作（例如增加值块的值等）。

命令

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 Bytes) {MSB...LSB}

其中:

块号 (1 个字节)

待操作的数值块

VB_OP (1 个字节)

值块操作

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*。仅适用于对值块的操作。

02h = 使值块的值减少 *VB_Value*。仅适用于对值块的操作。

VB_Value (4 个字节)

进行操作的数值，是一个有符号长整数。

例 1: Decimal - 4 = { FF FF FF FCh }

VB_Value			
MSB			LSB
FFh	FFh	FFh	FCh

例 2: Decimal 1 = { 00 00 00 01h }

VB_Value			
MSB			LSB
00h	00h	00h	01h

响应

响应	响应数据域	
结果	SW1	SW2

其中:

SW1 SW2 = 90 00h (表示操作成功完成)

= 63 00h (表示操作失败)

5.1.3.6. 读取值块 (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。

命令

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

其中：

块号 (1 个字节)
待访问的值块

响应

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。

例 1: Decimal - 4 = { FF FF FF FCh }

VB_Value			
MSB			LSB
FFh	FFh	FFh	FCh

例 2: Decimal 1 = { 00 00 00 01h }

VB_Value			
MSB			LSB
00h	00h	00h	01h

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h (表示操作成功完成)
= 63 00h (表示操作失败)

5.1.3.7. 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。

命令

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

其中:

源块号 (1 个字节)

数值所在的块的编号, 该数值会被复制到目标值块。

目标块号 (1 个字节)

目标值块的编号, 来自源值块的数值会被复制到此值块。源值块和目标值块必须位于同一个扇区。

响应

响应	响应数据域	
结果	SW1	SW2

其中:

SW1 SW2 = 90 00h (表示操作成功完成)

= 63 00h (表示操作失败)

例 1: 将数值“1”存入块 05h

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

例 2: 读取值块 05h

APDU = {FF B1 00 05 00h}

例 3: 将数值从值块 05h 复制到值块 06h

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

例 4: 使值块 05h 的值增加“5”

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



5.1.4. 访问符合 PC/SC 标准的标签 (ISO 14443-4)

基本上，所有符合 ISO14443-4 标准的卡片 (PICC 卡) 都可以理解 ISO 7816-4 规定的 APDU。ACM1281U-C7 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 进行转换和响应。ACM1281U-C7 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，请参见 5.1.3 节 – MIFARE 1K/4K 存储卡的 PICC 命令。

命令

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

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h (表示操作成功完成)
= 63 00h (表示操作失败)

典型的操作顺序是：

1. 出示标签，与 PICC 接口建立连接。
2. 读取/更新标签的存储内容。

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

步骤 2: 发送 APDU，取随机数

<< 00 84 00 00 08h

>> 1A F7 F3 1B CD 2B A9 58 [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：ISO 7816-4 APDU

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 = None; Data In = None; Le = 08h

应答： 00 01 02 03 04 05 06 07 [\$90 00h]



5.1.5. 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 DESFire 标签被激活，发送至 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，如果第一个 APDU 采用“ISO 7816-4 APDU 包模式”，则其余的 APDU 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 (DESFire) 中读取 8 个字节的随机数:

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

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

Lc = 01h; Data In = 00h; Le = 00h (Le = 00h, 表示最大长度)

应答: 7B 18 92 9D 9A 25 05 21h [\$91AFh]

注: 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 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 00h}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 MIFARE 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 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 25 9C 65 0C 87 65 1D D7h [\$1DD7h]

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

应用程序可以对[\$1DD7h]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, 则会在响应中增加状态码[90 00h]以满足 PC/SC 的要求。最短的响应长度为 2。



5.2. 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。

注：驱动程序会自动增加 Class、INS 和 P1。

5.2.1. 获取固件版本 (Get Firmware Version)

此命令用于获取读写器的固件信息。

Get Firmware Version 的命令结构 (5 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 (6 字节)

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	固件版本号

例如：

响应 = E1 00 00 00 0F 41 43 52 31 32 38 31 55 5F 56 37 30 32 2E 32

固件版本号 (HEX) = 41 43 52 31 32 38 31 55 5F 56 37 30 32 2E 32

固件版本号 (ASCII) = "ACR1281U_V702.2"



5.2.2. LED 控制 (LED Control)

此命令用于控制 LED 输出。

LED Control 的命令结构 (6 字节)

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

LED Control 的响应结构 (6 字节)

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



5.2.3. LED 状态 (LED Status)

此命令用于检查当前 LED 的状态。

LED Status 的命令结构 (5 字节)

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

LED Status 的响应结构 (6 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	说明
Bit 0	红色 LED	1 = 开 0 = 关
Bit 1	绿色 LED	1 = 开 0 = 关
Bit 2 – 7	RFU	RFU



5.2.4. 蜂鸣器控制 (Buzzer Control)

此命令用于控制蜂鸣器的输出。

Buzzer Control 的命令结构 (6 字节)

命令	CLA	INS	P1	P2	Lc	命令数据域
Buzzer Control	E0h	00h	00h	28h	01h	蜂鸣器鸣响时间

其中:

- 蜂鸣器鸣响时间 (1 个字节)
- 00h = 关闭
- 01 – FFh = 持续时间 (单位: 10 ms)

Buzzer Control 的响应结构 (6 字节)

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

5.2.5. 设置 LED 和蜂鸣器默认操作 (Set Default LED and Buzzer Behaviors)

此命令用于设置 LED 和蜂鸣器的默认操作。

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

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

其中:

默认操作 (1 个字节)

状态	描述	说明
Bit 0	RFU	RFU
Bit 1	PICC 轮询状态 LED	显示 PICC 轮询状态 1 = 启用 0 = 停用
Bit 2	RFU	RFU
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。 1 = 启用 0 = 停用
Bit 5	非接触芯片复位指示蜂鸣器	非接触芯片复位时发出哔的一声。 1 = 启用 0 = 停用
Bit 6	RFU	RFU
Bit 7	卡片操作闪烁 LED	LED 在卡片 (PICC) 被访问时会闪烁。

注: 默认的操作值 = FBh

Set LED and Buzzer Behaviors 的响应结构 (6 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

5.2.6. 读取 LED 和蜂鸣器默认操作(Read Default LED and Buzzer Behaviors)

此命令用于读取 LED 和蜂鸣器的当前默认操作。

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

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

Read LED and Buzzer Behaviors 的响应结构 (6 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

其中:

默认操作 (1 个字节)

状态	描述	说明
Bit 0	RFU	RFU
Bit 1	PICC 轮询状态 LED	显示 PICC 轮询状态 1 = 启用 0 = 停用
Bit 2	RFU	RFU
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。 1 = 启用 0 = 停用
Bit 5	非接触芯片复位指示蜂鸣器	非接触芯片复位时发出哔的一声。 1 = 启用 0 = 停用
Bit 6	RFU	RFU
Bit 7	卡片操作闪烁 LED	LED 在卡片 (PICC) 被访问时会闪烁。

注: 默认的操作值 = FBh

5.2.7. 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。

每当读写器连接到电脑上，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。

您可以通过 PC/SC Escape 命令接口发送一条命令来停用 PICC 轮询功能。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，读写器的电能消耗更低。

Set Automatic PICC Polling 的命令结构 (6 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

其中：

轮询设置 (1 个字节)

轮询设置	描述	说明
Bit 0	自动 PICC 轮询	1 = 启用 0 = 停用
Bit 1	如果没有找到 PICC，关闭天线场	1 = 启用 0 = 停用
Bit 2	如果 PICC 闲置，关闭天线场	1 = 启用 0 = 停用
Bit 3	RFU	RFU
Bit 5 – 4	PICC 轮询间隔	Bit 5 – Bit 4: 0 – 0 = 250 毫秒 0 – 1 = 500 毫秒 1 – 0 = 1000 毫秒 1 – 1 = 2500 毫秒
Bit 6	RFU	RFU
Bit 7	强制执行 ISO 14443A 第 4 部分	1 = 启用 0 = 停用

注：轮询设置的默认值 = 8Fh。

Set Automatic PICC Polling 的响应结构 (6 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

注：



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。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，应用就必须选定一种模式。

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

此命令用于检查当前的自动 PICC 轮询设置。

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

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

Read Automatic PICC Polling 的响应结构 (6 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	说明
Bit 0	自动 PICC 轮询	1 = 启用 0 = 停用
Bit 1	如果没有找到 PICC, 关闭天线场	1 = 启用 0 = 停用
Bit 2	如果 PICC 闲置, 关闭天线场	1 = 启用 0 = 停用
Bit 3	RFU	RFU
Bit 5 – 4	PICC 轮询间隔	Bit 5 – Bit 4: 0 – 0 = 250 毫秒 0 – 1 = 500 毫秒 1 – 0 = 1000 毫秒 1 – 1 = 2500 毫秒
Bit 6	RFU	RFU
Bit 7	强制执行 ISO 14443A 第 4 部分	1 = 启用 0 = 停用

注: 轮询设置的默认值 = 8Fh。



5.2.9. 手动 PICC 轮询 (Manual PICC Polling)

此命令用于确定是否有 PICC 处于读写器的检测范围内。在自动 PICC 轮询功能停用时，可以使用此命令。

Manual PICC Polling 的命令结构 (6 字节)

命令	CLA	INS	P1	P2	Lc	命令数据域
Manual PICC Polling	E0h	00h	00h	22h	01h	0Ah

Manual PICC Polling 的响应结构 (6 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 = 检测到 PICC
- FFh = 未检测到 PICC

5.2.10. 设置 PICC 操作参数 (Set PICC Operating Parameter)

此命令用于设置 PICC 的操作参数。

Set PICC Operating Parameter 的命令结构 (6 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

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 轮询中待检测的标签的类别	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。

Set PICC Operating Parameter 的响应结构 (6 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



5.2.11. 读取 PICC 操作参数 (Read PICC Operating Parameter)

此命令用于检查当前的 PICC 操作参数。

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

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

Read PICC Operating Parameter 的响应结构 (6 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 轮询中待检测的标签的类别	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。



附录A. 非接触式应用基本流程

步骤 0: 启动应用程序，读写器会不断地进行 PICC 轮询和标签扫描。一旦发现并检测到标签，相应的 ATR 会被发送到 PC。

步骤 1: 通过 T=1 协议连接“ACR1281U PICC 接口”。

步骤 2: 通过 APDU 交换访问 PICC。

..

步骤 N: 断开“ACR1281U PICC 接口”的连接，关闭应用。

访问 MIFARE DESFire 标签 (ISO 14443-4)



附录B. 扩展的 APDU 示例

卡片: ACOS7 (支持扩展的 APDU, 回响应)

写 CMD: **80 D2 00 00 XX XX XXh**

CLA = 80h

INS = D2h

P1 = 00h

P2 = 00h

Data Len = XX XX XXh

例 1: APDU 长度 = 263 个字节

APDU 命令:

```
80D2000000100000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F
202122232425262728292A2B2C2D2E2F303132333435363738393A3B3C3D3E3F40414243444546
4748494A4B4C4D4E4F505152535455565758595A5B5C5D5E5F606162636465666768696A6B6C6
D6E6F707172737475767778797A7B7C7D7E7F808182838485868788898A8B8C8D8E8F90919293
9495969798999A9B9C9D9E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B
8B9BABBBCBDBEBFC0C1C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9D
ADBDCDDDEDFE0E1E2E3E4E5E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFD
FEFFh
```

响应:

```
000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F20212223242526
2728292A2B2C2D2E2F303132333435363738393A3B3C3D3E3F404142434445464748494A4B4C4
D4E4F505152535455565758595A5B5C5D5E5F606162636465666768696A6B6C6D6E6F70717273
7475767778797A7B7C7D7E7F808182838485868788898A8B8C8D8E8F909192939495969798999A
9B9C9D9E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BABBBCBDB
EBFC0C1C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFE
0E1E2E3E4E5E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFF9000h
```

例 2: APDU 长度 = 775 个字节

APDU 命令:

```
80D2000000300000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F
202122232425262728292A2B2C2D2E2F303132333435363738393A3B3C3D3E3F40414243444546
4748494A4B4C4D4E4F505152535455565758595A5B5C5D5E5F606162636465666768696A6B6C6
D6E6F707172737475767778797A7B7C7D7E7F808182838485868788898A8B8C8D8E8F90919293
9495969798999A9B9C9D9E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B
8B9BABBBCBDBEBFC0C1C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9D
ADBDCDDDEDFE0E1E2E3E4E5E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFD
FEFF000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F2021222324
25262728292A2B2C2D2E2F303132333435363738393A3B3C3D3E3F404142434445464748494A4B
4C4D4E4F505152535455565758595A5B5C5D5E5F606162636465666768696A6B6C6D6E6F70717
2737475767778797A7B7C7D7E7F808182838485868788898A8B8C8D8E8F9091929394959697989
99A9B9C9D9E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BABBBC
BDBEBFC0C1C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDE
DFE0E1E2E3E4E5E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFF0001020
30405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F202122232425262728292
A2B2C2D2E2F303132333435363738393A3B3C3D3E3F404142434445464748494A4B4C4D4E4F50
```



5152535455565758595A5B5C5D5E5F606162636465666768696A6B6C6D6E6F7071727374757677
78797A7B7C7D7E7F808182838485868788898A8B8C8D8E8F909192939495969798999A9B9C9D9
E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BABBBCBDBEBFC0C1
C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFE0E1E2E3
E4E5E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFFh

响应:

000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F20212223242526
2728292A2B2C2D2E2F303132333435363738393A3B3C3D3E3F404142434445464748494A4B4C4
D4E4F505152535455565758595A5B5C5D5E5F606162636465666768696A6B6C6D6E6F70717273
7475767778797A7B7C7D7E7F808182838485868788898A8B8C8D8E8F909192939495969798999A
9B9C9D9E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BABBBCBDB
EBFC0C1C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFE
0E1E2E3E4E5E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFF00010203040
5060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F202122232425262728292A2B2
C2D2E2F303132333435363738393A3B3C3D3E3F404142434445464748494A4B4C4D4E4F505152
535455565758595A5B5C5D5E5F606162636465666768696A6B6C6D6E6F70717273747576777879
7A7B7C7D7E7F808182838485868788898A8B8C8D8E8F909192939495969798999A9B9C9D9E9FA
0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BABBBCBDBEBFC0C1C2C3
C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFE0E1E2E3E4E5
E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFF000102030405060708090A
0B0C0D0E0F101112131415161718191A1B1C1D1E1F202122232425262728292A2B2C2D2E2F303
132333435363738393A3B3C3D3E3F404142434445464748494A4B4C4D4E4F50515253545556575
8595A5B5C5D5E5F606162636465666768696A6B6C6D6E6F707172737475767778797A7B7C7D7E
7F808182838485868788898A8B8C8D8E8F909192939495969798999A9B9C9D9E9FA0A1A2A3A4A
5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BABBBCBDBEBFC0C1C2C3C4C5C6C7C
8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFE0E1E2E3E4E5E6E7E8E9EA
EBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFF9000h



附录C. 直接命令（Escape Command）示例

例如：获取固件版本号（使用 PCSCDirectCommand.exe）

步骤 1： 将 ACM1281 读写器连接到电脑上。

步骤 2： 打开 PCSCDirectCommand.exe.

步骤 3： 选择读写器以“Direct”模式连接，会显示出 ATR（如果存在卡片）或“No ATR retrieved (ATRLen = 0)”（如果没有卡片）

步骤 4： 输入命令：“3500”

输入数据：“E0 00 00 18 00” (用于获取固件版本的 APDU)

单击 Enter 将命令发送至读写器，然后检查 Response。



附录D. ACR128 兼容性

以下是 ACR128 读写器的一些功能，ACM1281U-C7 读写器对这些功能的实现方式不同或不支持这些功能。

功能	ACR128	ACM1281U-C7
1. 修改已激活的 PICC 的默认 FWI 和传输帧大小	1F 03 [数据: 3 个字节]	不支持。
2. 收发器设置	20 04 06 [数据: 3 个字节]	不支持。
3. PICC 设置	2A 0C [数据: 12 个字节]	不支持。
4. PICC T=CL 数据交换错误处理	2C 02 [数据: 1 个字节]	不支持。
5. 读取寄存器	19 01 [寄存器编号]	不支持。
6. 更新寄存器	1A 02 [寄存器编号] [值]	不支持。
7. 特定类型的 PICC 轮询	20 02 [数据: 1 个字节] FF	20 01 [数据: 1 个字节]
8. 蜂鸣器控制	28 01 [鸣响持续时间] 鸣响持续时间: 00 = 关闭 01 – FE = 鸣响持续时间 x 10 ms FF = 开启	28 01 [鸣响持续时间] 鸣响持续时间: 01 – FF = 鸣响持续时间 x 10 ms



功能	ACR128	ACM1281U-C7
9. 设置/读取默认的 LED 和蜂鸣器操作	设置: 21 01 [数据: 1 个字节] 读取: 21 00 数据: Bit 0 = ICC 激活状态 Bit 1 = PICC 轮询状态 LED Bit 2 = PICC 激活状态蜂鸣器 Bit 3 = PICC PPS 状态蜂鸣器 Bit 4 = 卡片插拔事件蜂鸣器 Bit 5 = 非接触芯片复位指示蜂鸣器 Bit 6 = 独占模式状态蜂鸣器 Bit 7 = 卡片操作闪烁 LED	设置: 21 01 [数据: 1 个字节] 读取: 21 00 数据: Bit 0 = RFU Bit 1 = PICC 轮询状态 LED Bit 2 = RFU Bit 3 = RFU Bit 4 = 卡片插拔事件蜂鸣器 Bit 5 = 非接触芯片复位指示蜂鸣器 Bit 6 = RFU Bit 7 = 卡片操作闪烁 LED
10. 设置/读取自动 PICC 轮询	设置: 23 01 [数据: 1 个字节] 读取: 23 00 数据: Bit 0 = 自动 PICC 轮询 Bit 1 = 如果没有找到 PICC, 关闭天线场 Bit 2 = 如果 PICC 闲置, 关闭天线场 Bit 3 = 检测到 PICC 后激活 PICC Bit 4..5 = PICC 轮询间隔 Bit 6 = 测试模式 Bit 7 = 执行 ISO 14443A 第 4 部分	设置: 23 01 [数据: 1 个字节] 读取: 23 00 数据: Bit 0 = 自动 PICC 轮询 Bit 1 = 如果没有找到 PICC, 关闭天线场 Bit 2 = 如果 PICC 闲置, 关闭天线场 Bit 3 = RFU Bit 4..5 = PICC 轮询间隔 Bit 6 = RFU Bit 7 = 执行 ISO 14443A 第 4 部分

Microsoft 是 Microsoft Corporation 在美国及/或其他国家的注册商标。
MIFARE、MIFARE Classic、MIFARE DESFire 和 MIFARE Ultralight 是 NXP B.V.的注册商标，根据授权使用。