



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

# ACM1281S-C7

## 帶 SAM 卡槽的非接觸式 读写器模块（串口）



参考手册 V1.01



## 版本历史

发布日期	修订说明	版本号
2015-04-22	<ul style="list-style-type: none"><li>初始发布</li></ul>	1.00
2017-05-30	<ul style="list-style-type: none"><li>更新 2.0 节-特性</li></ul>	1.01



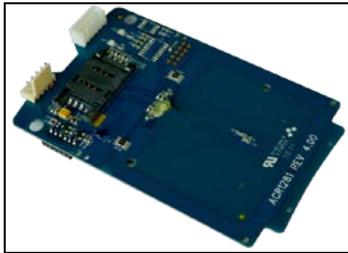
## 目录

<b>1.0.</b>	<b>简介</b> .....	<b>4</b>
<b>2.0.</b>	<b>特性</b> .....	<b>5</b>
2.1.	串行接口.....	6
2.1.1.	通信参数.....	6
2.2.	串行协议.....	6
2.3.	仿 CCID 命令.....	8
2.3.1.	Bulk-OUT 消息.....	8
2.3.2.	Bulk-IN 消息.....	11
<b>3.0.</b>	<b>非接触式智能卡协议</b> .....	<b>13</b>
3.1.	ATR 的生成.....	13
3.1.1.	ATR 信息格式（适用于 ISO 14443-3 PICC）.....	13
3.1.2.	ATR 信息格式（适用于 ISO 14443-4 PICC）.....	14
3.2.	非接触接口的私有 APDU 指令.....	15
3.2.1.	获取数据（Get Data）.....	15
3.2.2.	MIFARE 1K/4K 存储卡的 PICC 命令（T=CL 模拟）.....	16
3.2.3.	访问符合 PC/SC 标准的标签（ISO 14443-4）.....	26
<b>4.0.</b>	<b>外设控制</b> .....	<b>28</b>
4.1.	获取固件版本号（Get Firmware Version）.....	28
4.2.	LED 控制（LED Control）.....	29
4.3.	LED 状态（LED Status）.....	30
4.4.	蜂鸣器控制（Buzzer Control）.....	31
4.5.	设置 LED 和蜂鸣器默认操作（Set Default LED and Buzzer Behaviors）.....	32
4.6.	读取 LED 和蜂鸣器默认操作（Read Default LED and Buzzer Behaviors）.....	33
4.7.	初始化卡片检测计数器（Initialize Cards Insertion Counter）.....	34
4.8.	读取卡片检测计数器（Read Card's Detection Counter）.....	35
4.9.	更新卡片检测计数器（Update Card's Detection Counter）.....	36
4.10.	设置自动 PICC 轮询（Set Automatic PICC Polling）.....	37
4.11.	读取自动 PICC 轮询（Read Automatic PICC Polling）.....	39
4.12.	设置 PICC 操作参数（Set the PICC Operating Parameter）.....	40
4.13.	读取 PICC 操作参数（Read the PICC Operating Parameter）.....	41
4.14.	设置自动 PPS（Set Auto PPS）.....	42
4.15.	读取自动 PPS（Read Auto PPS）.....	43
4.16.	天线场控制（Antenna Field Control）.....	44
4.17.	读取天线场状态（Read Antenna Field Status）.....	45
4.18.	设置串行通信模式（Set Serial Communication Mode）.....	46

## 表目录

<b>表 1</b>	: RS-232 接口配线.....	<b>6</b>
<b>表 2</b>	: RS-485 接口配线.....	<b>6</b>
<b>表 3</b>	: MIFARE 1K 卡的内存结构.....	<b>18</b>
<b>表 4</b>	: MIFARE 4K 卡的内存结构.....	<b>18</b>
<b>表 5</b>	: MIFARE Ultralight 卡的内存结构.....	<b>20</b>

## 1.0. 简介



ACM1281S-C7 是一款基于 13.56 MHz 技术的非接触式读写器模块。它带有一个 SAM 卡槽，采用串行接口，支持 ISO 14443 第 1-4 部分的 A 类和 B 类卡以及 MIFARE Classic®系列卡，读取智能卡的距离可达 50 mm（视标签的类型而定）。

ACM1281S-C7 采用板线一体设计，可选配串行连接线。另外它还支持 USB 固件升级和扩展的 APDU 等功能。

ACM1281S-C7 可以即插即用，使用时无需安装任何驱动，能够快速而轻松地集成进嵌入式系统。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，为非接触式交易提供更高级别的安全性。ACM1281S-C7 能够以高达 848 Kbps 的速度与非接触式卡片进行通信，非常适合高要求的应用，例如自动售货机的支付系统、自助服务终端、游戏机、以及其他具有不同串行端口的集成系统。

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 ACM1281S-C7 的外设。



## 2.0. 特性

- RS-232串行接口：波特率 = 9.6 Kbps（默认）、19.2 Kbps、38.4 Kbps、57.6 Kbps、115.2 Kbps、230.4 Kbps
- USB接口取电
- 仿CCID架构
- 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
    - 支持ISO 7816 A类（5 V）SAM卡
- 内置外围设备：
  - 2 个用户可控的 LED 指示灯
  - 用户可控的蜂鸣器
- 具有USB固件升级能力
- 符合下列标准：
  - ISO 14443
  - ISO 7816
  - CE
  - FCC
  - PC/SC
  - RoHS 2
  - REACH

## 2.1. 串行接口

ACM1281S-C7 通过一个串行接口（RS-232 或 RS-485）与计算机建立连接。

### 2.1.1. 通信参数

ACM1281S-C7 通过串行接口（RS-232 或 RS-485）与主机建立连接，支持以下通讯波特率：9,600 bps（默认）、19,200 bps、38,400 bps、57,600 bps、115,200 bps 和 230,400 bps。

引脚	信号	功能
1	VCC	为读写器提供+5 V 的电源
2	TXD	主机向读写器发送的信号
3	RXD	读写器向主机发送的信号
4	GND	参考电压等级

表1 : RS-232 接口配线

引脚	信号	功能
1	VCC	为读写器提供+5 V 的电源
2	A	读写器和主机间以差分信号传输数据
3	B	读写器和主机间以差分信号传输数据
4	GND	参考电压等级

表2 : RS-485 接口配线

## 2.2. 串行协议

ACM1281S-C7 通过串行接口与主机连接，采用仿 CCID 架构用于通信。

命令格式如下：

STX (02h)	Bulk-OUT 头	APDU 命令或参数	校验和	ETX (03h)
1 个字节	10 个字节	M 个字节 (如有)	1 个字节	1 个字节

其中：

**STX** – 起始字节，通知读写器开始接收命令，必须为 02h

**ETX** – 结束字节，通知读写器命令结束，必须为 03h

**Bulk-OUT 头** – 10 个字节的仿 CCID 头

**APDU 命令或参数** – 用于访问读写器或卡片的 APDU 命令或参数

**校验和** – 错误检查，等于 XOR {Bulk-OUT 头, APDU 命令或参数}



## 2.3. 仿 CCID 命令

### 2.3.1. Bulk-OUT 消息

ACM1281S 应接受 CCID 协议第 6.1 部分定义的 CCID 类 Bulk-OUT 消息。本规范还定义了一些用于操作附加功能的扩展命令。此节将列举 ACM1281S 支持的 CCID 类 Bulk-OUT 消息。

#### 2.3.1.1. PC\_to\_RDR\_IccPowerOn

此命令用于激活卡槽并返回卡片的 ATR。

偏移	数据域	大小	值	说明
0	<i>bMessageType</i>	1	62h	
1	<i>dwLength</i>	4	00000000h	此消息的额外字节的大小。
2	<i>bSlot</i>	1		标识命令的插槽号。 SAM 接口: <i>bSlot</i> = 2。 ICC 接口: <i>bSlot</i> = 1。 PICC 接口: <i>bSlot</i> = 0。
5	<i>bSeq</i>	1		命令的序号。
6	<i>bPowerSelect</i>	1		ICC 上的电压值 00h – 自动电压选择 01h – 5 V 02h – 3 V
7	<i>abRFU</i>	2		保留为将来使用。

此消息的响应是 *RDR\_to\_PC\_DataBlock* 消息，返回的数据是复位应答（ATR）。

注：访问接触式卡之前必须先激活 ICC 接口和 SAM 接口。

#### 2.3.1.2. PC\_to\_RDR\_IccPowerOff

此命令用于取消激活卡槽。

偏移	数据域	大小	值	说明
0	<i>bMessageType</i>	1	63h	
1	<i>dwLength</i>	4	00000000h	此消息的额外字节的大小。
5	<i>bSlot</i>	1		标识命令的插槽号 SAM 接口: <i>bSlot</i> = 2。 ICC 接口: <i>bSlot</i> = 1。 PICC 接口: <i>bSlot</i> = 0。
6	<i>bSeq</i>	1		命令的序号。
7	<i>abRFU</i>	3		保留为将来使用。

此消息的响应是 *RDR\_to\_PC\_SlotStatus* 消息。

### 2.3.1.3. PC\_to\_RDR\_GetSlotStatus

此命令用于获取当前的卡槽状态。

偏移	数据域	大小	值	说明
0	<i>bMessageType</i>	1	65h	
1	<i>dwLength</i>	4	00000000h	此消息的额外字节的大小。
5	<i>bSlot</i>	1		标识命令的插槽号。 SAM 接口: <i>bSlot</i> = 2。 ICC 接口: <i>bSlot</i> = 1。 PICC 接口: <i>bSlot</i> = 0。
6	<i>bSeq</i>	1		命令的序号。
7	<i>abRFU</i>	3		保留为将来使用。

此消息的响应是 *RDR\_to\_PC\_SlotStatus* 消息。

### 2.3.1.4. PC\_to\_RDR\_XfrBlock

此命令用于向 ICC 传输数据块。

偏移	数据域	大小	值	说明
0	<i>bMessageType</i>	1	6Fh	
1	<i>dwLength</i>	4		此消息的 <i>abData</i> 数据域的大小。
5	<i>bSlot</i>	1		标识命令的插槽号。 SAM 接口: <i>bSlot</i> = 2。 ICC 接口: <i>bSlot</i> = 1。 PICC 接口: <i>bSlot</i> = 0。
6	<i>bSeq</i>	1		命令的序号。
7	<i>bBWI</i>	1		用于为当前传输延长 CCID 块的超时等待时间。“该数值乘以块等待时间”的时间段过去后, CCID 将超时该块。
8	<i>wLevelParameter</i>	2	0000h	RFU (TPDU 交换级别)。
10	<i>abData</i>	字节型数组		发送给 CCID 的数据块。信息是“按原样”发送至 ICC (TPDU 交换级别) 的。

此消息的响应是 *RDR\_to\_PC\_DataBlock* 消息。

### 2.3.1.5. PC\_to\_RDR\_Escape

此命令用于访问扩展特性。

偏移	数据域	大小	值	说明
0	<i>bMessageType</i>	1	6Bh	
1	<i>dwLength</i>	4		此消息的 <i>abData</i> 数据域的大小



偏移	数据域	大小	值	说明
5	<i>bSlot</i>	1		标识命令的插槽号。 SAM 接口: <i>bSlot</i> = 2。 ICC 接口: <i>bSlot</i> = 1。 PICC 接口: <i>bSlot</i> = 0。
6	<i>bSeq</i>	1		命令的序号。
7	<i>abRFU</i>	3		保留为将来使用。
10	<i>abData</i>	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。

### 2.3.2. Bulk-IN 消息

Bulk-IN 消息用于对 Bulk-OUT 消息做出响应。ACM1281S 应接受 CCID 协议第四部分定义的 Bulk-IN 消息。此节列举了 ACM1281S 支持的 CCID 类 Bulk-IN 消息。

#### 2.3.2.1. RDR\_to\_PC\_DataBlock

此消息由 ACM1281S 发出，是对 *PC\_to\_RDR\_IccPowerOn* 和 *PC\_to\_RDR\_XfrBlock* 消息的响应。

偏移	数据域	大小	值	说明
0	<i>bMessageType</i>	1	80h	表示正在从 CCID 发送一个数据块。
1	<i>dwLength</i>	4		此消息的额外字节的大小。
5	<i>bSlot</i>	1		与 Bulk-OUT 消息中的值相同。 SAM 接口: <i>bSlot</i> = 2。 ICC 接口: <i>bSlot</i> = 1。 PICC 接口: <i>bSlot</i> = 0。
6	<i>bSeq</i>	1		与 Bulk-OUT 消息中的值相同。
7	<i>bStatus</i>	1		CCID 规范 6.2.6 节定义的插槽状态寄存器
8	<i>bError</i>	1		CCID 规范 6.2.6 节定义的插槽错误寄存器
9	<i>bChainParameter</i>	1	00h	RFU (TPDU 交换级别)。
10	<i>abData</i>	字节 型数 组		本数据域包含由 CCID 返回的 数据。

#### 2.3.2.2. RDR\_to\_PC\_Escape

此消息由 ACM1281S 发出，是对 *PC\_to\_RDR\_Escape* 消息的响应。

偏移	数据域	大小	值	说明
0	<i>bMessageType</i>	1	83h	
1	<i>dwLength</i>	4		此消息的 <i>abData</i> 数据域的大小。
5	<i>bSlot</i>	1		与 Bulk-OUT 消息中的值相同。 SAM 接口: <i>bSlot</i> = 2。 ICC 接口: <i>bSlot</i> = 1。 PICC 接口: <i>bSlot</i> = 0。
6	<i>bSeq</i>	1		与 Bulk-OUT 消息中的值相同。
7	<i>bStatus</i>	1		CCID 规范 6.2.6 节定义的插槽状态寄存器
8	<i>bError</i>	1		CCID 规范 6.2.6 节定义的插槽错误寄存器
9	<i>bRFU</i>	1	00h	RFU.
10	<i>abData</i>	字节 型数 组		本数据域包含由 CCID 返回的 数据。



### 2.3.2.3. RDR\_to\_PC\_SlotStatus

此消息由 ACM1281S 发出，是对 *PC\_to\_RDR\_IccPowerOff* 和 *PC\_to\_RDR\_GetSlotStatus* 消息，以及类特定 ABORT 请求的响应。

偏移	数据域	大小	值	说明
0	<i>bMessageType</i>	1	81h	
1	<i>dwLength</i>	4	00000000h	此消息的额外字节的大小。
5	<i>bSlot</i>	1		与 Bulk-OUT 消息中的值相同。 SAM 接口: <i>bSlot</i> = 2。 ICC 接口: <i>bSlot</i> = 1。 PICC 接口: <i>bSlot</i> = 0。
6	<i>bSeq</i>	1		与 Bulk-OUT 消息中的值相同。
7	<i>bStatus</i>	1		CCID 规范 6.2.6 节定义的插槽状态寄存器
8	<i>bError</i>	1		CCID 规范 6.2.6 节定义的插槽错误寄存器
9	<i>bClockStatus</i>	1		值: 00h = 时钟运行 01h = 时钟停于 L 状态 02h = 时钟停于 H 状态 03h = 时钟停止于未知状态 所有其他值保留为将来使用。

### 3.0. 非接触式智能卡协议

#### 3.1. 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。

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

字节	值 (Hex)	标记	说明
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		注册的应用提供商标识(RID) A0 00 00 03 06h
	SS		标准字节
	C0h ..C1h		卡片名称字节
	00 00 00 00h	RFU	RFU 00 00 00 00h
4+N	UUh	TCK	T0 至 Tk 的所有字节按位异或

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}

长度 (YY) = 0Ch

RID = {A0 00 00 03 06h} (PC/SC 工作组)

标准 (SS) = 03 (ISO 14443A, 第 3 部分)

卡片名称(C0 ..C1) = {00 01h} (MIFARE 1K)

00 01h: MIFARE 1K

00 36h: MIFARE PLUS SL1\_2K

00 3Ah: MIFARE Ultralight C

00 02h: MIFARE 4K

00 37h: MIFARE PLUS SL1\_4K

FF 28h: JCOP 30

00 03h: MIFARE Ultralight

00 38h: MIFARE PLUS SL2\_2K

FF [SAK]h: 未定义的标签

00 26h: MIFARE Mini

00 39h: MIFARE PLUS SL2\_4K

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

字节	值 (Hex)	标记	说明						
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	XXh	T1	历史字节:  ISO 14443A: 来自 ATS 响应的历史字节。参考 ISO 14443-4 标准。  ISO 14443B: <table border="1" data-bbox="730 1099 1311 1335"> <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							
XXh XXh XXh	Tk								
4+N	UUh	TCK	T0 至 Tk 的所有字节按位异或						

**例 1:** MIFARE DESFire 的 ATR = {3B 81 80 01 80 80h} // 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

### 3.2. 非接触接口的私有 APDU 指令

私有 APDU 用于访问非接触标签和外围设备。

私有 APDU 通过 *PC\_to\_RDR\_XfrBlock* 消息发送，其中 *bSlot = 0*。

#### 3.2.1. 获取数据 (Get Data)

此命令用于获取“已建立连接的 PICC”的序列号或 ATS。

GET UID 的 APDU 结构 (5 个字节)

命令	CLA	INS	P1	P2	Le
Get Data	FFh	CAh	00h 01h	00h	00h (最大长度)

若 P1 = 00h, Get UID 的响应报文结构 (UID + 2 个字节)

响应	响应数据域				
结果	UID (LSB)	...	...	UID (MSB)	SW1 SW2

如果 P1 = 01h, 获取 ISO 14443 A 类卡的 ATS (ATS + 2 个字节)

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	90h 00h	操作成功完成。
警告	62h 82h	UID/ATS 的末尾先于 Le 字节到达 (Le 大于 UID 的长度)。
错误	6Ch 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	63h 00h	操作失败。
错误	6Ah 81h	不支持此功能。

例如:

// 获取“已经建立连接的 PICC”的序列号

```
UINT8 GET_UID[5]={FFh, CAh, 00h, 00h, 00h};
```

// 获取“已经建立连接的 ISO 14443-A PICC”的 ATS

UINT8 GET\_ATS[5]={FFh, CAh, 01h, 00h, 00h};

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

#### 3.2.2.1. 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。

Load Authentication Keys 的 APDU 结构 (11 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
	20h	= 密钥被载入读写器的非易失性存储器
	其它	= 保留
密钥号 (1 个字节)	00h ~ 1Fh	= 使用非失性存储器存储密钥。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消失。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)	= 使用易失性存储器来存储临时密钥。一旦读写器与电脑断开连接, 密钥就会消失。易失密钥只有一 (1) 个, 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}	

Load Authentication Keys 的响应结构 (2 字节)

响应	响应数据域	
结果	SW1	SW2

Load Authentication Keys 命令的响应状态码

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

- 基本上, 应用程序需要了解所有正在被使用的密钥。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。易失性存储器和非易失性存储器的内容都无法从外部读取。
- 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。它们被存储在“内部 RAM”中。而非易失密钥被存储在“EEPROM”中。EEPROM 相对于内部 RAM 存储速度稍慢。
- 我们不建议使用“非易失密钥位置 00h ~ 1Fh”来存储任何经常变化的“临时密钥值”。“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”会不时的变化, 请将其存储在“易失密钥位置 020h”。

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

Authentication 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 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 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。一张 MIFARE 1K 卡分为 16 个扇区, 每个扇区包含四 (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。

注: 一旦该块被成功验证, 即可访问属于同一扇区的所有块。

**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 ~ 1Fh =使用非易失性存储器存储密钥。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消失。读写器的非易失存储器可以存储 32 个密钥。

20h (过程密钥) = 使用易失性存储器来存储密钥。一旦读写器与电脑断开连接，密钥就会消失。易失密钥只有一 (1) 个，可以用作不同会话的过程密钥。

Load Authentication Keys 的响应结构 (2 字节)

响应	响应数据域	
结果	SW1	SW2

Load Authentication Keys 命令的响应状态码

结果	SW1 SW2	含义
成功	90h 00h	操作成功完成。
错误	63h 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)	} 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3 : 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	

表4 : MIFARE 4K 卡的内存结构



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	80h ~ 8Eh	8Fh
扇区 33	90h ~ 9Eh	9Fh
..		
..		
扇区 38	E0h ~ EEh	EFh
扇区 39	F0h ~ FEh	FFh

} 2 KB

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};

<同样>

// 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。*





**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 的响应结构 (4/16 的倍数 + 2 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	90h 00h	操作成功完成。
错误	63h 00h	操作失败。

例如:

// 从二进制块 04h 中读取 16 个字节 (MIFARE 1K 或 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}

### 3.2.2.4. 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 的 APDU 结构 (16 的倍数 + 5 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)

**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。*

**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。

Update Binary Block 的响应状态码 (2 字节)

结果	SW1 SW2	含义
成功	90h 00h	操作成功完成。
错误	63h 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 04 的数据更新为{00 01 02 03h}

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

### 3.2.2.5. 值块操作 (Value Block Operation) (INC, DEC, 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)。

Value Block Operation 的 APDU 结构 (10 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。待操作的值块。

**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。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。用于算数运算的数值, 是一个有符号长整数。

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 的响应结构 (2 个字节)

响应	响应数据域	
结果	SW1	SW2

Value Block Operation 命令的响应状态码

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

### 3.2.2.6. 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。

Read Value Block 的 APDU 结构 (5 个字节)

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

其中:

块号      1 字节。待读取的值块

Read Value Block 的响应结构 (4 + 2 字节)

响应	响应数据域		
结果	Value {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 00h}

//将数值从值块 05h 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}

### 3.2.3. 访问符合 PC/SC 标准的标签 (ISO 14443-4)

所有符合 ISO 14443-4 标准的卡片 (PICC) 都可以理解 ISO 7816-4 规定的 APDU。ACM1281S 读写器与符合 ISO 14443-4 标准的卡片进行通信时, 只需要对 ISO 7816-4 规定的 APDU 和响应进行转换。ACM1281S 会在内部处理 ISO 14443 第 1-4 部分协议。

另外 MIFARE 1K、MIFARE 4K、MIFARE Mini 和 MIFARE 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	90h 00h	操作成功完成。
错误	63h 00h	操作失败。

典型的操作顺序为:

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 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}

Class = 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 [\$9000]



## 4.0. 外设控制

对外围设备的访问应当通过发送 *PC\_to\_RDR\_Escape* 消息来实现，其中 *bSlot = 0*。

### 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 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 (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	固件版本号

示例响应 = E1 00 00 00 0F 41 43 52 31 32 38 31 53 5F 56 33 30 38 2E 30h

固件版本号 (HEX) = 41 43 52 31 32 38 31 53 5F 56 33 30 38 2E 30h

固件版本号 (ASCII) = "ACR1281S\_V308.0"



## 4.2. LED 控制 (LED Control)

此命令用于控制 LED 输出。

LED Control 命令的结构 (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

LED Control 命令的响应结构 (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

LED 状态 (1 个字节) – LED 控制

LED 状态	模式	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

### 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 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 Status

LED 状态 (1 个字节) – LED 状态

LED 状态	模式	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



## 4.4. 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
- 1 - 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

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

此命令用于设置读写器 LED 指示灯和蜂鸣器的默认操作。

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

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

默认操作 (1 个字节)

默认操作	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 LED	显示 PICC 接口的激活状态 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。(包括 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 界面只有一个可以被激活。	独享模式被激活时会发出哔的一声。 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:** 默认操作的默认值 = FBh。

Set Default LED and Buzzer Behaviors 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	默认操作

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

此命令用于读取读写器 LED 指示灯和蜂鸣器的当前默认操作。

Read Default LED and Buzzer Behaviors 命令的结构（5 个字节）

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

Read Default LED and Buzzer Behaviors 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	默认操作

默认操作（1 个字节）

默认操作	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 LED	显示 PICC 接口的激活状态 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。（包括 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 界面只有一个可以被激活。	独享模式被激活时会发出哔的一声。 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：**默认操作的默认值 = FBh。



## 4.7. 初始化卡片检测计数器 (Initialize Cards Insertion Counter)

此命令用于初始化卡片检测计数器。

Initialize Card's Detection Counter 的命令结构 (9 个字节)

命令	CLA	INS	P1	P2	Lc	命令数据域			
Initialize Card's Detection Counter	E0h	00h	00h	09h	04h	RFU	RFU	PICC Cnt (LSB)	PICC Cnt (MSB)

Initialize Card's Detection Counter 的响应结构 (9 个字节)

响应	CLA	INS	P1	P2	Lc	响应数据域			
结果	E1h	00h	00h	00h	04h	RFU	RFU	PICC Cnt (LSB)	PICC Cnt (MSB)

其中:

**PICC Cnt (LSB)** 1 个字节。PICC 检测计数器 (LSB)

**PICC Cnt (MSB)** 1 个字节。PICC 检测计数器 (MSB)



## 4.8. 读取卡片检测计数器（Read Card's Detection Counter）

此命令用于查看卡片检测计数器的值。

Read Card's Detection Counter 的命令结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Read Card's Detection Counter	E0h	00h	00h	09h	00h

Read Card's Detection Counter 的响应结构（9 个字节）

响应	CLA	INS	P1	P2	Lc	响应数据域			
结果	E1h	00h	00h	00h	04h	RFU	RFU	PICC Cnt (LSB)	PICC Cnt (MSB)

其中：

**PICC Cnt (LSB)** 1 个字节。PICC 检测计数器（LSB）

**PICC Cnt (MSB)** 1 个字节。PICC 检测计数器（MSB）

## 4.9. 更新卡片检测计数器（Update Card's Detection Counter）

此命令用于更新卡片检测计数器的值。

Update Card's Detection Counter 的命令结构（5 个字节）

命令	CLA	INS	P1	P2	Lc
Update Card's Detection Counter	E0h	00h	00h	0Ah	00h

Update Card's Detection Counter 的响应结构（9 个字节）

响应	CLA	INS	P1	P2	Lc	响应数据域			
结果	E1h	00h	00h	00h	04h	RFU	RFU	PICC Cnt (LSB)	PICC Cnt (MSB)

其中：

**PICC Cnt (LSB)** 1 个字节。PICC 检测计数器（LSB）

**PICC Cnt (MSB)** 1 个字节。PICC 检测计数器（MSB）

## 4.10. 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。

我们可以发送一个命令来停用 PICC 轮询功能。该命令通过 PCSC Escape 命令接口发送。

**注：** 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

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

轮询设置 (1 字节)

轮询设置	参数	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：** 轮询设置的默认值 = 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。空闲时的电流消耗= PICC 尚未激活。
3. 读写器会自动激活“ISO14443A-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，应用就必须选定一种模式。

## 4.11. 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 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

轮询设置 (1 字节)

轮询设置	参数	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:** 轮询设置的默认值 = 8Fh

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

此命令用于配置 PICC 操作参数。

Set the 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

Set the 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

操作参数 (1 字节)

操作参数	参数	描述	选项
Bit0	ISO 14443 A 类	PICC 轮询要检测的标签类别	1 = 检测 0 = 跳过
Bit1	ISO 14443 B 类		1 = 检测 0 = 跳过
Bit2 - 7	RFU	RFU	RFU

**注:** 操作参数的默认值 = 03h

### 4.13. 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 字节)

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

Read the 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

操作参数 (1 字节)

操作参数	参数	描述	选项
Bit0	ISO 14443 A 类	PICC 轮询要检测的标签类别	1 = 检测 0 = 跳过
Bit1	ISO 14443 B 类		1 = 检测 0 = 跳过
Bit2 - 7	RFU	RFU	RFU

#### 4.14. 设置自动 PPS (Set Auto PPS)

每次识别出 PICC，读写器都会尝试改变由**最快连接速度**定义的 PCD 和 PICC 之间的通信数据速率。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。

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 命令的响应结构 (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。最大速度。

**当前速度**      1 个字节。当前速度。

可以为：

- 106 Kbps = 00h (无自动 PPS; 默认设置)
- 212 Kbps = 01h
- 424 Kbps = 02h
- 848 Kbps = 03h

**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。



## 4.15. 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 (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。最大速度。

**当前速度**      1 个字节。当前速度。

可以为:

- 106 Kbps = 00h (无自动 PPS; 默认设置)
- 212 Kbps = 01h
- 424 Kbps = 02h
- 848 Kbps = 03h



## 4.16. 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

Antenna Field Control 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。
- 01h = 启用天线场
  - 00h = 停用天线场

**注：** 关闭天线场前要确保自动 PICC 轮询功能已经停用。



## 4.17. 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。

- 01h = 启用天线场
- 00h = 停用天线场



## 4.18. 设置串行通信模式 (Set Serial Communication Mode)

本命令用于设置通信速度和通信模式。

Set Serial Communication Mode 的命令结构 (2 个字节)

命令	Byte 0	Byte 1
Set Serial Communication Mode	44h	模式选择

Set Serial Communication Mode 的响应结构 (2 个字节)

响应	Byte 0	Byte 1
结果	90h	模式选择

模式选择 (1 个字节) – 通信速度和模式选择

偏移	参数	描述
Bit 0-3	串行通信速度	000b= 9600 bps (默认) 001b= 19200 bps 010b= 38400 bps 011b= 57600 bps 100b= 115200 bps 101b= 128000 bps 110b= 230400 bps 其他值保留为将来使用。
Bit 4 - 6	RFU	RFU
Bit 7	Interrupt-In 消息 (仿 CCID 架构)	1 = 报告 Interrupt-In 消息。 0 = 无报告 (默认)

**注:** 成功修改通信速度后, 程序必须对通信速度进行调整, 以便继续剩下的数据交换。