Advanced Card Systems Ltd.
Card & Reader Technologies

ACR1581U

USB 双界面读写器

参考手册 V1.06

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

目录

图目录

表目录

1.0. 简介

ACR1581 DualBoost III 延续了 ACR1281U 的成功经验，是 ACS DualBoost 系列读写器的第三代产品。做为一款双界面读写器，它能够读写各类接触式和非接触式智能卡，通过 USB CCID 类驱动程序以及 USB 接口与电脑连接，接受来自计算机应用的卡片命令。除了支持传统的 ISO 7816 MCU 卡、MIFARE®卡以及 ISO 14443 的 A 类和 B 类非接触卡，它还支持 FeliCa 卡和 ISO 15693 卡。

做为计算机与卡的中间设备，ACR1581U 会执行来自计算机的命令，与非接触式标签、MCU 卡、SAM 卡及外围设备（LED 或蜂鸣器）进行通信。它的三种接口（PICC 接口、ICC 接口和 SAM 接口）均符合 PC/SC 标准，其中接触式接口使用 ISO 7816 标准定义的 APDU 命令。关于接触式 MCU 卡的操作，请参阅相关卡片的文档以及 PC/SC 标准。

本 API 文件会详细介绍如何执行 PC/SC APDU 命令来支持非接触式接口和接触式存储卡，以及控制 ACR1581U 的外设。

2.0. 特性

- USB全速接口
- 符合CCID标准
- 智能卡读写器：
  - 非接触接口：
    - 读写速率达 26 kbps（ISO 15693 卡）以及 848 kbps（ISO 14443 卡）
    - 内置天线用于读写非接触式标签，智能卡读取距离可达 70 mm（视标签类型而定）
    - 支持 ISO 15693 卡
    - 支持 ISO 14443 第 4 部分的 A 类和 B 类卡，以及 MIFARE 系列卡
    - 内建防冲突特性
    - 支持扩展的 APDU（最大 64 KB）
  - 接触式接口：
    - 支持 ISO 7816 的 A 类、B 类和 C 类（5V、3V、1.8V）卡
    - 支持通用权限卡（CAC）
    - 支持个人身份验证卡（PIV）
    - 支持 T=0 或 T=1 协议的微处理器卡
    - 支持协议和参数选择（PPS）
    - 具有短路保护功能
    - 支持扩展的 APDU（T=1：最高 64 K 字节；T=0：最高 512+10 字节）
  - SAM接口：
    - 1个SAM卡槽
    - 支持ISO 7816 A类SAM卡
- 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：
  - 2 个用户可控的 LED 指示灯（蓝色和绿色）
  - 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 15693
  - ISO 7816
  - PC/SC
  - CCID
  - CE
  - UKCA
  - FCC
  - RoHS
  - REACH
  - Microsoft® WHQL

[1] *使用 ACS 定义的安卓库*

3.0. ACR1581U 的结构

3.1. 读写器功能框图

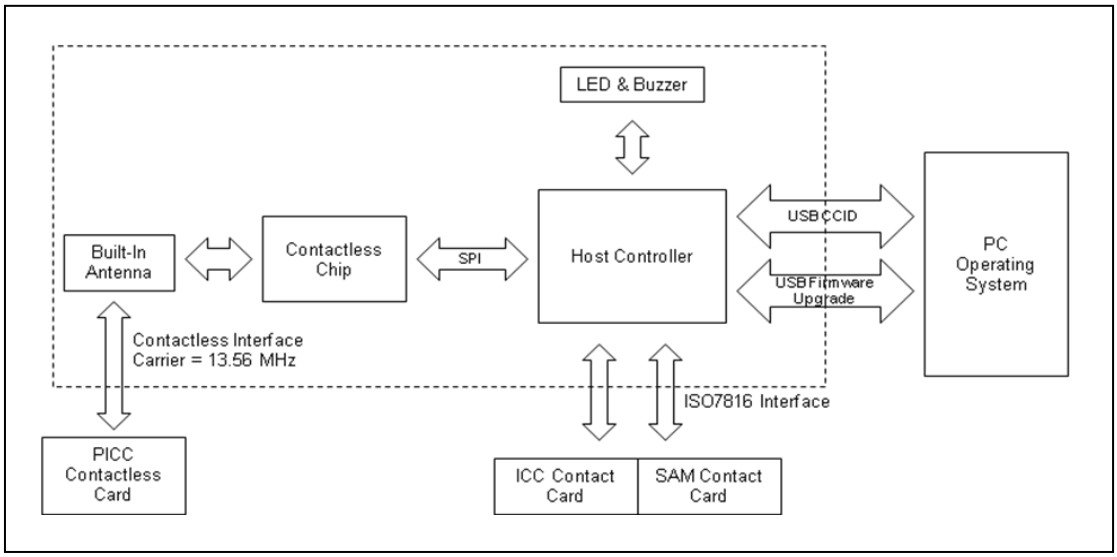

图 1：ACR1581U 读写器功能框图

3.2. PC/SC 驱动与 ICC、PICC 和 SAM 间的通信

ACR1581U 与计算机之间使用 CCID 协议进行通信。而 ICC、PICC 和 SAM 间的通信则完全符合 PC/SC 标准。

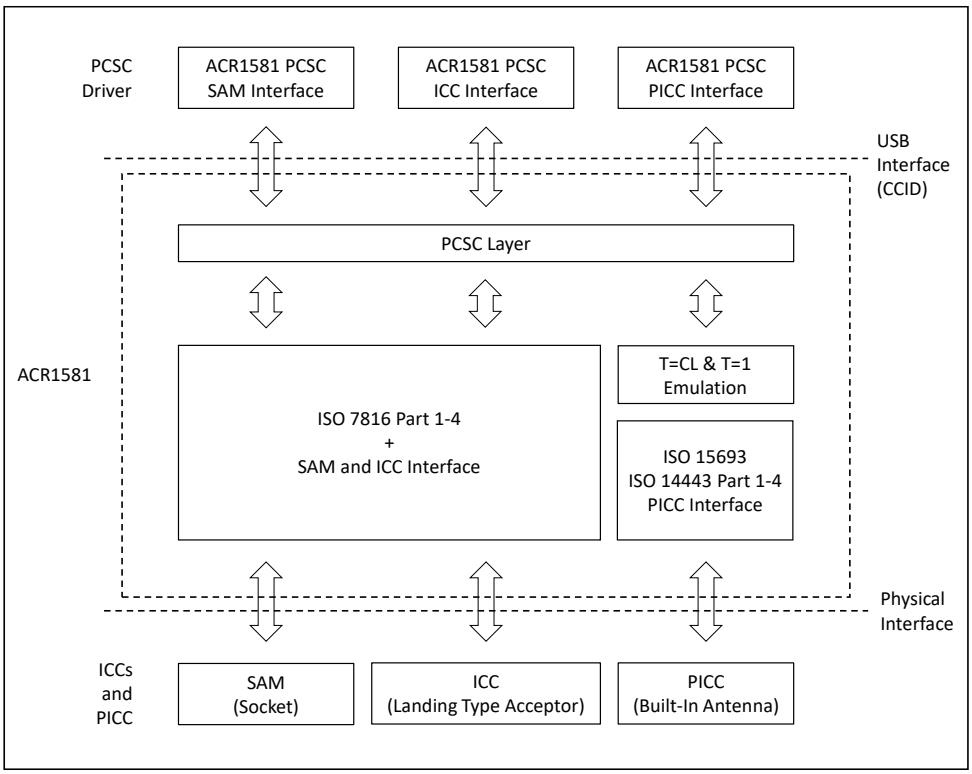

图 2：ACR1581U 的结构

4.0. 硬件设计

4.1. USB

ACR1581U 通过符合 USB 标准的 USB 接口与计算机连接。

4.1.1. 通信参数

ACR1581U 按照 USB2.0 规范的要求通过 USB 接口与计算机建立连接，支持 USB 全速模式，速率为 12 Mbps。

引脚	信号	功能
1	V_{BUS}	为读写器提供+5 V 的电源
2	D-	ACR1581U-C1 和 PC 间以差分信号传输数据
3	D+	ACR1581U-C1 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 ACR1581U 通过 USB 接口正常工作，应该先安装设备驱动程序。*

4.1.2. 端点

ACR1581U 通过下列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 ACR1581U 的命令（数据包大小为 64 字节）

Bulk-IN – 用于从 ACR1581U 发送至主计算机的响应（数据包大小为 64 字节）

Interrupt-IN – 用于从 ACR1581U 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. 接触式智能卡接口

ACR1581U 与插入的智能卡之间的接口符合 ISO 7816-3 标准，并进行了某些限制或提升来增强 ACR1581U 的实用功能。

4.2.1. 智能卡电源 VCC（C1）

插入的智能卡电流消耗不得大于 60 mA。

4.2.2. 卡片类型选择

激活插入的卡片之前，处于控制地位的计算机需要向 ACR1581U 发送正确的命令来选择卡片类型。这些卡片包括存储卡和基于 MCU 的卡。

对于基于 MCU 的卡片来说，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 卡仅支持一种协议类型（T=0 或 T=1），读写器会自动采用该协议类型，而不管应用程序选择了哪一种。

4.2.3. 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 5 MHz。

4.3. 非接触智能卡接口

ACR1581U 与非接触卡之间的接口遵循 ISO 14443 标准，并进行了某些限制或提升来增强 ACR1581U 的实用功能。

4.3.1. 载波频率

ACR1581U-C1 的载波频率为 13.56MHz。

4.3.2. 卡片轮询

ACR1581U-C1 会自动检测进入工作场的非接触卡。此功能支持 ISO 14443-4 的 A 类和 B 类卡、ISO 15693 卡以及 MIFARE 卡。

4.4. 用户接口

4.4.1. 蜂鸣器和 LED

ACR1581U 配有 LED 指示灯和单音蜂鸣器，用于指示接触式和非接触式接口的状态。蓝色 LED 是 PICC 状态指示灯，绿色 LED 是 ICC 状态指示灯。

读写器状态	蜂鸣器	绿色 LED (ICC)	蓝色 LED (PICC)
1. 插入读写器	响一次	● >> ● >> ●	
2. 待机（非接触卡轮询，不存在 ICC 卡或 PICC 卡）	关闭	关闭	●
3. 待机（无轮询，不存在 ICC 卡或 PICC 卡）	关闭	关闭	关闭
4. 刷非接触卡	响一次	基于 ICC 状态	●
5. 非接触卡存在	关闭	基于 ICC 状态	●
6. 非接触卡移出	关闭	基于 ICC 状态	待机 / 基于 ICC 状态
7. 非接触卡通信中	关闭	基于 ICC 状态	快速闪烁
8. 插入接触式卡	响一次	●	关闭 / 基于 PICC 状态
9. 接触式卡存在	关闭	●	关闭 / 基于 PICC 状态
10. 接触式卡移出	关闭	关闭	●
11. 接触式卡通信中	关闭	快速闪烁	基于 PICC 状态

表 2：蜂鸣器和 LED 指示灯

5.0. 软件设计

5.1. PCSC API

本节介绍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 工作组规格网站。

5.1.1. 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

在执行任何其他 PCSC 操作前，应当先执行此函数。

```
#define SCARD_SCOPE_USER 0

SCARDCONTEXT hContext;
int retCode;
void main ()
{
   // To establish the resource manager context and assign it to "hContext"
   retCode = SCardEstablishContext(SCARD_SCOPE_USER,
                  NULL,
                  NULL,
                  &hContext);
   if (retCode != SCARD_S_SUCCESS)
   {
      // Establishing resource manager context failed
   }
   else
   {
      // Establishing resource manager context successful
      // Further PCSC operation can be performed
   }

}
```

例如：

5.1.2. SCardListReaders

SCardListReaders 函数用来获取系统中在指定读写器组集合中的读写器名字列表（消除重复项）。

调用方提供一个读写器组列表，函数返回这些指定组里面的读写器名字列表。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供使用的组里面的读写器。

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

```
#define SCARD_SCOPE_USER 0

SCARDCONTEXT hContext; // Resource manager context
int retCode;
char readerName [256]; // List reader name

void main ()
{
   // To establish the resource manager context and assign to "hContext"
   retCode = SCardEstablishContext(SCARD_SCOPE_USER,
                  NULL,
                  NULL,
                  &hContext);
   if (retCode != SCARD_S_SUCCESS)
   {
      // Establishing resource manager context failed
   }
   else
   {
      // Establishing resource manager context successful
      // List the available reader which can be used in the system
      retCode = SCardListReaders (hContext,
                  NULL,
                  readerName,
                  &size);
      if (retCode != SCARD_S_SUCCESS)
      {
         // Listing reader fail
      }
      if (readerName == NULL)
      {
         // No reader available
      }
      else
      {
         // Reader listed
      }
   }
}
```

例如：

5.1.3. SCardConnect

SCardConnect 函数利用特定资源管理器上下文，在应用程序与特定读写器包含的智能卡之间建立连接。如果特定读写器中没有卡片，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

例如：

```
#define SCARD_SCOPE_USER 0

SCARDCONTEXT       hContext;          // Resource manager context
SCARDHANDLE        hCard;             // Card context handle
unsigned long      dwActProtocol;     // Establish active protocol
int                retCode;
char               readerName [256]; // List reader name
char               rName [256];       // Reader name for connection

void main ()
{
   ...
      if (readerName == NULL)
      {
         // No reader available
      }
      else
      {
         // Reader listed
      rName = "ACS ACR1581 1S Dual Reader PICC 0"; // Depends on what
                                                   reader be used
                                                   // Should connect to
                                                   PICC interface
         retCode = SCardConnect(hContext,
               rName,
               SCARD_SHARE_SHARED,
               SCARD_PROTOCOL_T0,
               &hCard,
               &dwActProtocol);
         if (retCode != SCARD_S_SUCCESS)
         {
            // Connection failed (May be because of incorrect reader
         name, or no card was detected)
         }
         else
         {
            // Connection successful
         }
      }
   }
```

5.1.4. SCardControl

SCardControl 函数提供对读写器的直接控制，可以在成功调用 SCardConnect 函数后，并且尚未成功调用 SCardDisconnect 函数前随时调用此函数。它对读写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：直接（Escape）命令要使用此 API 进行发送。

例如：

```
#define SCARD_SCOPE_USER    0

#define EscapeCommand 0x310000 + 3500*4
   SCARDCONTEXT          hContext;          // Resource manager context
   SCARDHANDLE           hCard;             // Card context handle
   unsigned long         dwActProtocol;     // Established active protocol
   int                   retCode;
   char                  readerName [256]; // Lists reader name
   char                  rName [256];       // Reader name for connection
   BYTE                  SendBuff[262],     // APDU command buffer
                         RecvBuff[262];     // APDU response buffer
   BYTE                  FWVersion [20],    // For storing firmware
                                               version message
   BYTE                  ResponseData[50]; // For storing card response
   DWORD                 SendLen,           // APDU command length
                         RecvLen;           // APDU response length
void main ()
   {
      ...
      rName = "ACS ACR1581 1S Dual Reader PICC 0";  // Depends on what
                                                    reader will be used
                                                    // Should connect to
                                                    PICC interface

         retCode = SCardConnect(hContext,
            rName,
            SCARD_SHARE_DIRECT,
            SCARD_PROTOCOL_T0| SCARD_PROTOCOL_T1,
            &hCard,
            &dwActProtocol);
         if (retCode != SCARD_S_SUCCESS)
         {
            // Connection failed (may be because of incorrect reader
         name, or no card was detected)
         }
         else
         {
            // Connection successful
            RecvLen = 262;
            // Get firmware version
            SendBuff[0] = 0xE0;
            SendBuff[1] = 0x00;
            SendBuff[2] = 0x00;
            SendBuff[3] = 0x18;
            SendBuff[4] = 0x00;
```

```
            SendLen = 5;
            retCode = SCardControl ( hCard,
                      EscapeCommand,
                      SendBuff,
                      SendLen,
                      RecvBuff,
                      RecvLen,
                      &RecvLen);
            if (retCode != SCARD_S_SUCCESS)
            {
               // APDU sending failed
               return;
            }
            else
            {
               // APDU sending successful
               // The RecvBuff stores the firmware version message.
               for (int i=0;i< RecvLen-5;i++)
               {
                  FWVersion[i] = RecvBuff [5+i];
               }
            }
            // Connection successful
            RecvLen = 262;

            // Turn Green LED on, turn Red LED off
            SendBuff[0] = 0xE0;
            SendBuff[1] = 0x00;
            SendBuff[2] = 0x00;
            SendBuff[3] = 0x29;
            SendBuff[4] = 0x01;
            SendBuff[5] = 0x02;  // Green LED On, Red LED off
            SendLen = 6;
            retCode = SCardControl ( hCard,
                      EscapeCommand,
                      SendBuff,
                      SendLen,
                      RecvBuff,
                      RecvLen,
                      &RecvLen);
            if (retCode != SCARD_S_SUCCESS)
            {
               // APDU sending failed
               return;
            }
            else
            {
               // APDU sending success
            }
```

5.1.5. 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：使用此API发送APDU命令（即：发送给已建立连接的卡片的命令、**PICC的PCSC私有（Pseudo）APDU（带专有扩展）**和**PCSC** 2.0 第 3 部分示例*

1. 开始透明会话

   命令：**FF C2 00 00 02 81 00**

   响应：**C0 03 00 90 00 90 00**

2. 关闭天线场

   命令：**FF C2 00 00 02 83 00**
   响应：**C0 03 00 90 00 90 00**

3. 打开天线场

   命令：**FF C2 00 00 02 84 00**

   响应：**C0 03 00 90 00 90 00**

4. 激活 ISO 14443-4A

   命令： **FF C2 00 02 04 8F 02 00 04**

   响应： **C0 03 01 64 01 90 00** （如果不存在卡片）

   C0 03 00 90 00 5F 51 [Len] [ATR] 90 00

5. 将 PCB 设为 0Ah，并在传输数据中启用 CRC、奇偶校验和协议头。

   命令：**FF C2 00 01 0A 90 02 00 00 FF 6E 03 07 01 0A**

   响应：**C0 03 00 90 00 90 00**

6. 发送 APDU “80B2000008”至卡片并取响应。

   命令：**FF C2 00 01 0E 5F 46 04 40 42 0F 00 95 05 80 B2 00 00 08**

   响应：**C0 03 00 90 00 92 01 00 96 02 00 00 97 0C [卡片响应] 90 00**

7. 结束透明会话。

   命令：**FF C2 00 00 02 82 00**

   响应：**C0 03 00 90 00 90 00**

PICC 的专属私有（Pseudo）APDU）。

例如：

```
#define SCARD_SCOPE_USER      0

SCARDCONTEXT      hContext;          // Resource manager context
SCARDHANDLE       hCard;             // Card context handle
unsigned long     dwActProtocol;     // Established active protocol
int               retCode;
char              readerName [256]; // List reader name
char              rName [256];       // Reader name for connect
BYTE              SendBuff[262],     // APDU command buffer
                  RecvBuff[262];     // APDU response buffer
BYTE              CardID [8],        // For storing the FeliCa IDM/
                                     MIFARE UID
BYTE              ResponseData[50]; // For storing card response
DWORD             SendLen,           // APDU command length
                  RecvLen;           // APDU response length
SCARD_IO_REQUEST  ioRequest;

void main ()
{
   ...
   rName = "ACS ACR1581 1S Dual Reader PICC 0"; // Depends on what
                                                  reader should be used
                                                  // Should connect to PICC
                                                  interface
      retCode = SCardConnect(hContext,
                     rName,
                     SCARD_SHARE_SHARED,
                     SCARD_PROTOCOL_T0,
                     &hCard,
                     &dwActProtocol);
      if (retCode != SCARD_S_SUCCESS)
      {
         // Connection failed (May be because of incorrect reader
         name, or no card was detected)
      }
      else
      {
         // Connection successful
         ioRequest.dwProtocol = dwActProtocol;
         ioRequest.cbPciLength = sizeof(SCARD_IO_REQUEST);
         RecvLen = 262;
```

```
// Get MIFARE UID/ FeliCa IDM
SendBuff[0] = 0xFF;
SendBuff[1] = 0xCA;
SendBuff[2] = 0x00;
SendBuff[3] = 0x00;
SendBuff[4] = 0x00;
SendLen = 5;
   retCode = SCardTransmit( hCard,
                  &ioRequest,
                  SendBuff,
                  SendLen,
                  NULL,
                  RecvBuff,
                  &RecvLen);

if (retCode != SCARD_S_SUCCESS)
{
   // APDU sending failed
   return;
}
else
{
   // APDU sending successful
   // The RecvBuff stores the IDM for FeliCa / the UID for
   MIFARE.
   // Copy the content for further FeliCa access
   for (int i=0;i< RecvLen-2;i++)
   {
      CardID [i] = RecvBuff[i];
   }
}
```

5.1.6. SCardDisconnect

SCardDisconnect 函数用来断开先前在应用程序和目标读写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

此函数用于结束 PCSC 操作。

例如：

```
#define SCARD_SCOPE_USER 0

SCARDCONTEXT       hContext;          // Resource manager context
SCARDHANDLE        hCard;             // Card context handle
unsigned long      dwActProtocol;     // Established active protocol
int                retCode;

void main ()
{
   ...
         // Connection successful
      ...
      retCode = SCardDisconnect(hCard,SCARD_RESET_CARD);
      if (retCode != SCARD_S_SUCCESS)
      {
         // Disconnection failed
      }
      else
      {
         // Disconnection successful
      }
   }
}
```

5.1.7. APDU 流程图

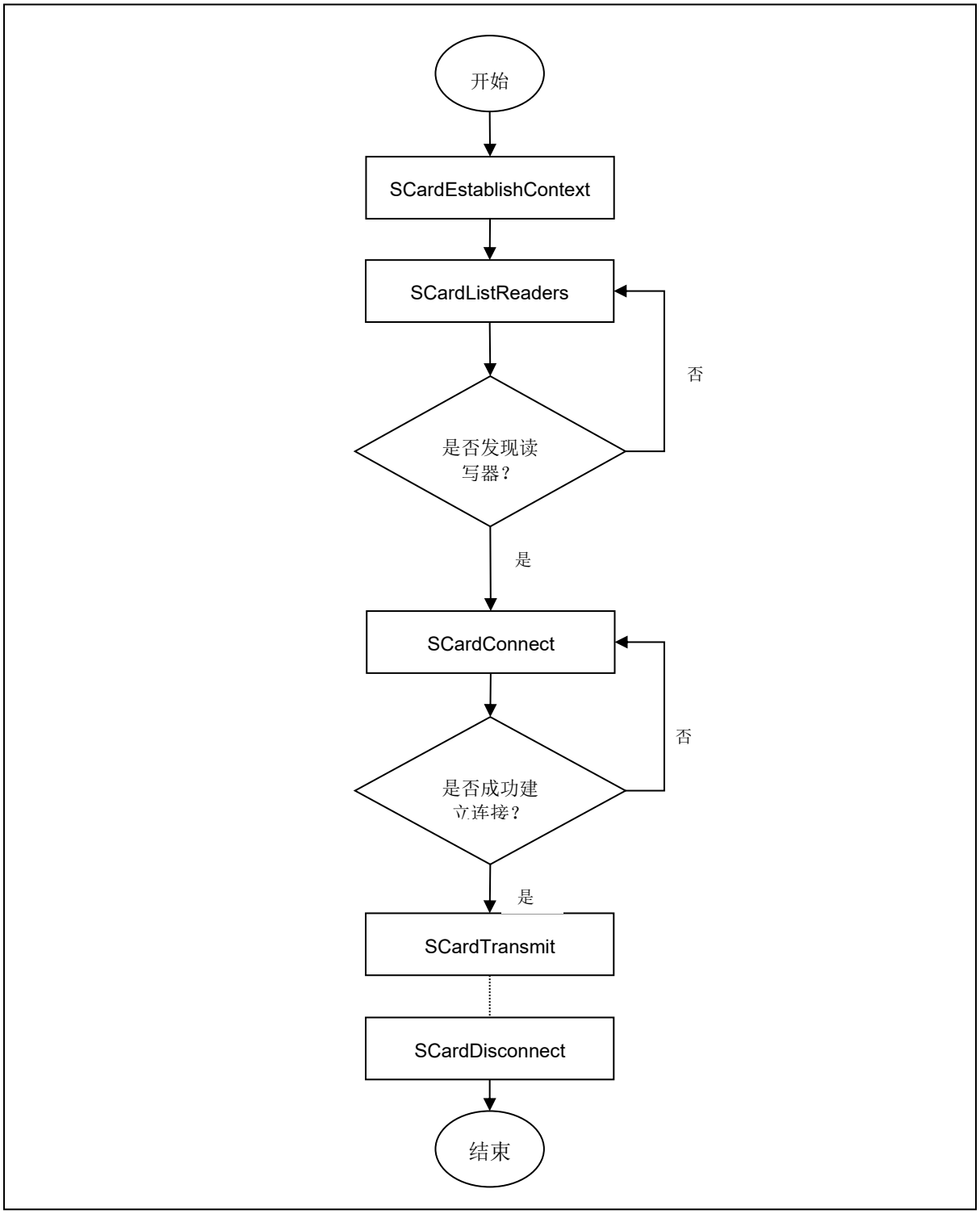

图 3：ACR1581U APDU 流程图

5.1.8. 直接命令流程图

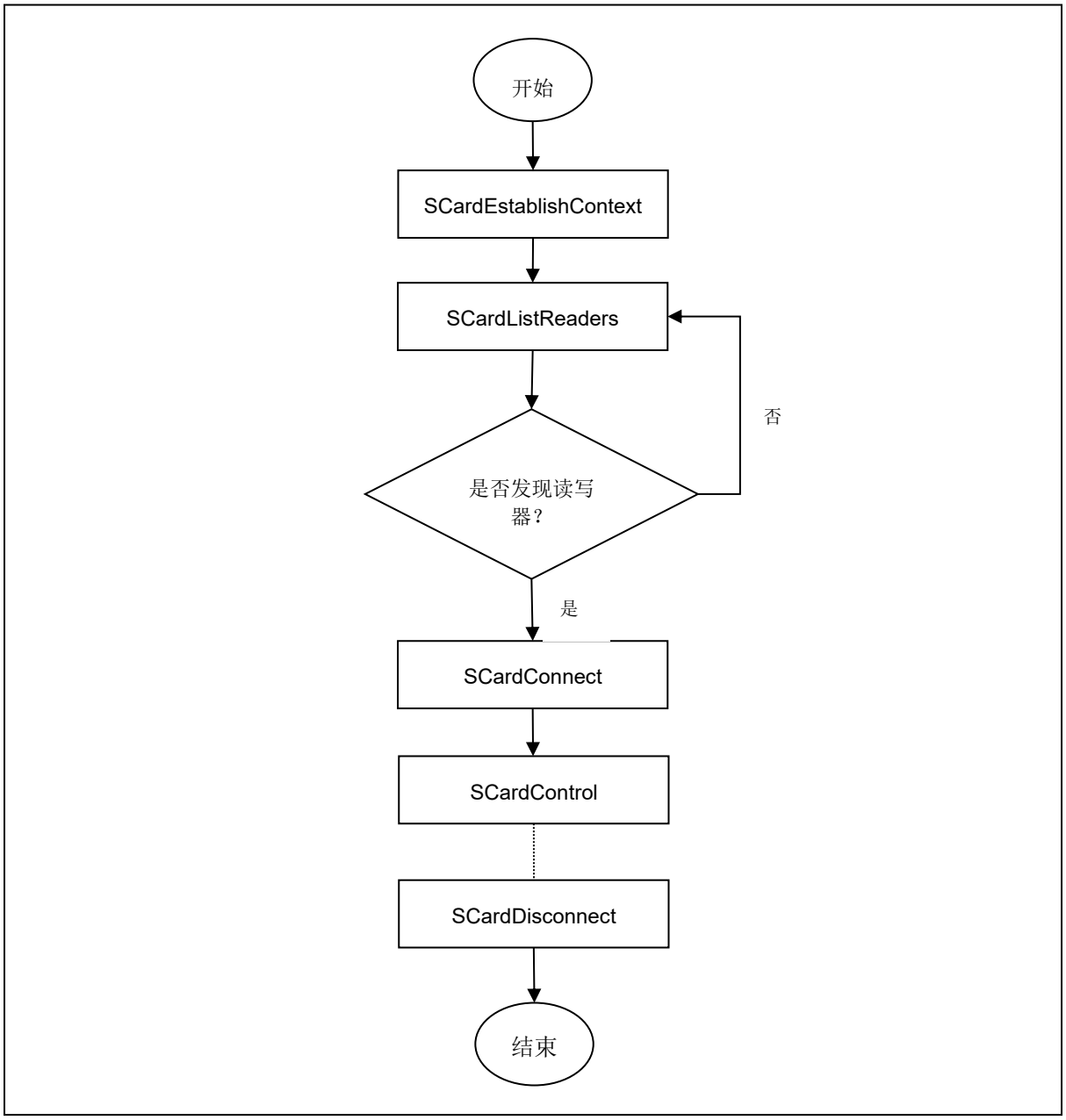

图 4： ACR1581U 直接命令流程图

5.2. 接触式智能卡协议

5.2.1. 存储卡 – 1/2/4/8/16 kb I2C 卡

5.2.1.1. 选择卡片类型（Select Card Type）

此命令用于对选定的卡片进行上电/下电，同时进行卡片复位操作。

命令

命令	CLA	INS	P1	P2	Lc	卡片类型
Select Card Type	FFh	A4h	00h	00h	01h	01h

响应状态码

结果	SW1	SW2	含义
成功	90h	00h	操作成功完成。

5.2.1.2. 选择页面大小（Select Page Size）

此命令会选择用于卡片读取的页面大小，默认值是 8 字节页写。在卡片移出，读写器下电时会重置为默认值。

命令

命令	CLA	INS	P1	P2	Lc	响应数据域
Select Page Size	FFh	01h	00h	00h	01h	页面大小

响应状态码

结果	SW1	SW2	含义
成功	90h	00h	操作成功完成。

页面大小：1 个字节

状态	说明
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

5.2.1.3. 读取存储卡（Read Memory Card）

此命令会从指定的地址位置开始读取存储卡的内容。

命令

命令	CLA	INS	P1	P2	Le
Read Memory Card	FFh	B0h	存储地址		长度

响应状态码

结果	SW1	SW2	含义
成功	90h	00h	操作成功完成。

5.2.1.4. 写入存储卡（Write Memory Card）

此命令会从指定的地址位置开始向存储卡写入内容。

命令

命令	CLA	INS	P1	P2	Le	命令数据域
Write Memory Card	FFh	D0h	存储地址		长度	数据

响应状态码

结果	SW1	SW2	含义
成功	90h	00h	操作成功完成。

5.2.2. 存储卡 – 32/64/128/256/512/1024 kb I2C 卡

5.2.2.1. Select Card Type

此命令用于对选定的卡片进行上电/下电，同时进行卡片复位操作。

命令

命令	CLA	INS	P1	P2	Lc	卡片类型
Select Card Type	FFh	A4h	00h	00h	01h	02h

响应状态码

结果	SW1	SW2	含义
成功	90h	00h	操作成功完成。

5.2.2.2. 选择页面大小（Select Page Size）

此命令会选择用于卡片读取的页面大小，默认值是 8 字节页写。在卡片移出，读写器下电时会重置为默认值。

命令

命令	CLA	INS	P1	P2	Lc	响应数据域
Select Page Size	FFh	01h	00h	00h	01h	页面大小

响应状态码

结果	SW1	SW2	含义
成功	90h	00h	操作成功完成。

页面大小：1 个字节

状态	说明
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

5.2.2.3. 读取存储卡（Read Memory Card）

此命令会从指定的地址位置开始读取存储卡的内容。

命令

命令	CLA	INS	P1	P2	Le
Read Memory Card	FFh		存储地址		长度

响应状态码

结果	SW1	SW2	含义
成功	90h	00h	操作成功完成。

注： INS *B0h = 用于 32, 64, 128, 256, 512 kb I2C 卡*

*1011 000*b；其中*是 17 位地址的 MSB = 用于 1024 kb I2C 卡*

5.2.2.4. 写入存储卡（Write Memory Card）

此命令会从指定的地址位置开始向存储卡写入内容。

命令

命令	CLA	INS	P1	P2	Le	命令数据域
Write Memory Card	FFh		存储地址		长度	数据

响应状态码

结果	SW1	SW2	含义
成功	90h	00h	操作成功完成。

注： INS *B0h = 用于 32, 64, 128, 256, 512 kb I2C 卡*

*1011 000*b；其中*是 17 位地址的 MSB = 用于 1024 kb I2C 卡*

5.2.3. 存储卡 – ATMEL AT88SC153

5.2.3.1. 选择卡片类型（Select card type）

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

命令

私有 APDU						
命令	CLA	INS	P1	P2	Lc	卡片类型
Select Card Type	FFh	A4h	00h	00h	01h	03h

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

5.2.3.2. 读取存储卡（Read memory card）

此命令会从指定的地址位置开始读取存储卡的内容。

命令

私有 APDU					
命令	CLA	INS	P1	字节地址	MEM_L
Read Memory Card	FFh		00h		

其中：

INS （1 个字节）

读取分区 00b, INS = B0h

读取分区 01b, INS = B1h

读取分区 10b, INS = B2h

读取分区 11b, INS = B3h

读取熔丝标志, INS = B4h

字节地址 （1 个字节）

存储卡的内存地址位置。

MEM_L （1 个字节）

要从存储卡读取的数据的长度。

响应

响应	字节 1	…	…	字节 N	SW1	SW2
结果						

其中：

字节(1…N) 从存储卡读取的数据。

SW1 SW2 = 90 00h（若操作成功完成）

5.2.3.3. 写入存储卡（Write memory card）

此命令会从指定地址位置开始向存储卡写入内容。

命令

私有 APDU									
命令	CLA	INS	P1	字节地址	MEM_L	字节 1	…	…	字节 N
Write Memory Card	FFh		00h						

其中：

INS （1 个字节）

读取分区 00b, INS = D0h

读取分区 01b, INS = D1h

读取分区 10b, INS = D2h

读取分区 11b, INS = D3h

读取熔丝标志, INS = D4h

字节地址 （1 个字节）

存储卡的内存地址位置。

MEM_L （1 个字节）

要写入存储卡的数据的长度。

字节(1…N) 要写入存储卡的数据

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

5.2.3.4. 校验密码（Verify password）

此命令用于校验用户输入的 PIN 是否与存储卡的密码相同。

命令

私有 APDU									
命令	CLA	INS	P1	P2	Lc	RP	PW (0)	PW (1)	PW (2)
Verify Password	FFh	20h	00h		03h				

其中：

PW (0), PW (1), PW (2) = 要发送给存储卡的密码

P2 （1 个字节）

= 0000 00r pb

其中“r p”两个位标识待比较的密码

r = 0：“写”密码

r = 1：“读”密码

p = 密码集编号

r p = 01b：安全密码

响应

响应	响应数据域	
结果	SW1	ErrorCnt

其中：

SW1 = 90h

ErrorCnt （1 个字节）

= 错误计数器

FFh 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。

5.2.3.5. 初始化认证（Initialize authentication）

此命令用于初始化存储卡认证。

命令

私有 APDU									
命令	CLA	INS	P1	P2	Lc	Q (0)	Q (1)	…	Q (7)
Initialize Authentication	FFh	84h	00h	00h	08h				

其中：

Q (0…7) （8 个字节）

= 主机随机数

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

5.2.3.6. 校验认证（Verify authentication）

此命令用于校验存储卡认证。

命令

私有 APDU									
命令	CLA	INS	P1	P2	Lc	Ch (0)	Ch (1)	…	Ch (7)
Verify Authentication	FFh	82h	00h	00h	08h				

其中：

Ch (0…7) （8 个字节）

= 主机挑战数

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

5.2.4. 存储卡 – ATMEL AT88SC1608

5.2.4.1. 选择卡片类型（Select card type）

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

命令

私有 APDU						
命令	CLA	INS	P1	P2	Lc	卡片类型
Select Card Type	FFh	A4h	00h	00h	01h	04h

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

5.2.4.2. 读取存储卡（Read memory card）

此命令会从指定的地址位置开始读取存储卡的内容。

命令

私有 APDU					
命令	CLA	INS	区域地址	字节地址	MEM_L
Read Memory Card	FFh				

其中：

INS （1 个字节）
读取用户区，INS = B0h
读取配置区或读取熔丝标志，INS = B1h

区域地址 （1 个字节）
= 00000 A10 A9 A8b，其中 A10 是分区地址的 MSB
** 读熔丝标志时无关

字节地址 （1 个字节）
= A7 A6 A5 A4 A3 A2 A1 A0b 是存储卡的内存地址位置
读熔丝标志时，字节地址 = 1000 0000b

MEM_L （1 个字节）
要从存储卡读取的数据的长度。

响应

响应	字节 1	…	…	字节 N	SW1	SW2
结果						

其中：

字节(1…N) 从存储卡读取的数据。

SW1 SW2 = 90 00h（若操作成功完成）

5.2.4.3. 写入存储卡（Write to memory card）

此命令会从指定地址位置开始向存储卡写入内容。

命令

私有 APDU									
命令	**CLA**	**INS**	**区域地址**	**字节地址**	**MEM_L**	**字节 1**	**…**	**…**	**字节 N**
Write Memory Card	FFh								

其中：

INS （1 个字节）

读取用户区，**INS = D0h**

读取配置区或读取熔丝标志，**INS = D1h**

区域地址 （1 个字节）

= 00000 A10 A9 A8b，其中 A10 是分区地址的 MSB

** 读熔丝标志时无关

字节地址 （1 个字节）

= A7 A6 A5 A4 A3 A2 A1 A0b 是存储卡的内存地址位置

读熔丝标志时，字节地址 = 1000 0000b

MEM_L （1 个字节）

要写入存储卡的数据的长度。

字节(1…N) 要写入存储卡的数据

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

5.2.4.4. 校验密码（Verify password）

此命令用于校验用户输入的 PIN 是否与存储卡的密码相同。

命令

私有 APDU									
命令	CLA	INS	P1	P2	Lc	RP	PW (0)	PW (1)	PW (2)
Verify Password	FFh	20h	00h	00h	04h				

其中：

PW (0), PW (1), PW (2) = 要发送给存储卡的密码

RP （1 个字节）

= 0000 r p2 p1 p0b

其中“r p2 p1 p0”两个位标识待比较的密码

r = 0 ： “写”密码

r = 1： “读”密码

p2 p1 p0 = 密码集编号

r p2 p1 p0 = 0111b：安全密码。

响应

响应	响应数据域	
结果	SW1	ErrorCnt

其中：

SW1 = 90h

ErrorCnt （1 个字节）

= 错误计数器

FFh 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。

5.2.4.5. 初始化认证（Initialize authentication）

此命令用于初始化存储卡认证。

命令

私有 APDU									
命令	CLA	INS	P1	P2	Lc	Q (0)	Q (1)	…	Q (7)
Initialize Authentication	FFh	84h	00h	00h	08h				

其中：

Q (0…7) （8 个字节）

= 主机随机数

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

5.2.4.6. 校验认证（Verify authentication）

此命令用于校验存储卡认证。

命令

私有 APDU									
命令	CLA	INS	P1	P2	Lc	Ch (0)	Ch (1)	…	Ch (7)
Verify Authentication	FFh	82h	00h	00h	08h				

其中：

Ch (0…7) （8 个字节）

= 主机挑战数

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

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

5.2.5.1. 选择卡片类型（Select card type）

此命令用于对选定的卡片进行上电/下电，之后进行卡片复位操作。

命令

命令	CLA	INS	P1	P2	Lc	卡片类型
Select Card Type	FFh	A4h	00h	00h	01h	05h

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

5.2.5.2. 读取存储卡（Read memory card）

此命令会从指定的地址位置开始读取存储卡的内容。

命令

命令	CLA	INS	字节地址		MEM_L
			MSB	LSB	
Read Memory Card	FFh	B0h			

其中：

MSB 字节地址 （1 个字节）

= 0000 00 A9 A8b 是存储卡的内存地址位置

LSB 字节地址 （1 个字节）

= A7 A6 A5 A4 A3 A2 A1 A0b 是存储卡的内存地址位置

MEM_L （1 个字节）

待从存储卡读取的数据的长度。

响应

响应	字节 1	…	…	字节 N	SW1	SW2
结果						

其中：

字节(1…N) 从存储卡读取的数据。

SW1 SW2 = 90 00h（若操作成功完成）

5.2.5.3. 读取输入错误计数器（Read presentation error counter memory card）（仅限 SLE4428 和 SLE5528）

此命令用于读取密码输入错误计数器。

命令

命令	CLA	INS	P1	P2	MEM_L
Read Presentation Error Counter	FFh	B1h	00h	00h	03h

响应

响应	ErrCnt	虚拟数 1	虚拟数 2	SW1	SW2
结果					

其中：

ErrCnt （1 个字节）

密码输入错误计数器的值

FFh = 表示验证正确

00h = 表示密码被锁定（超过最大重试次数）

其它值表示验证失败。

虚拟数 1, 虚拟数 2 （2 个字节）

从卡片读取的虚拟数据

SW1 SW2 = 90 00h（若操作成功完成）

5.2.5.4. 读取保护位（Read protection bit）

此命令用于读取保护位。

命令

命令	CLA	INS	字节地址		MEM_L
			MSB	LSB	
Read Protection Bit	FFh	B2h			

其中：

MSB 字节地址 （1 个字节）

存储卡的内存地址位置

= 0000 00 A9 A8b

LSB 字节地址 （1 个字节）

存储卡的内存地址位置

= A7 A6 A5 A4 A3 A2 A1 A0b

MEM_L （1 个字节）

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，应当发送下面的私有 APDU：

FF B1 00 10 01h

响应

响应	PROT 1	…	…	PROT L	SW1	SW2
结果						

其中：

PROT (1..L) 含有保护位的字节

SW1 SW2 = 90 00h（若操作成功完成）

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 是响应数据中字节 *x* 的保护位：

0 = 字节写保护

1 = 字节可以被写入

5.2.5.5. 写入存储卡（Write memory card）

此命令会向指定地址位置写入存储卡内容。

命令

命令	CLA	INS	字节地址		MEM_L	字节 1	…	…	字节 N
			MSB	LSB					
Write Memory Card	FFh	D0h							

其中：

MSB 字节地址 （1 个字节）

= 0000 00 A9 A8b 是存储卡的内存地址位置

LSB 字节地址 （1 个字节）

= A7 A6 A5 A4 A3 A2 A1 A0b 是存储卡的内存地址位置

MEM_L （1 个字节）

要写入存储卡的数据的长度。

字节(1…N) 要写入存储卡的数据

5.2.5.6. 写保护存储卡（Write protection memory card）

此命令将命令中指定的每一个字节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

命令	CLA	INS	字节地址		MEM_L	字节 1	…	…	字节 N
			MSB	LSB					
Write Protection Memory Card	FFh	D1h							

其中：

MSB 字节地址 （1 个字节）

= 0000 00 A9 A8b 是存储卡的内存地址位置

LSB 字节地址 （1 个字节）

= A7 A6 A5 A4 A3 A2 A1 A0b 是存储卡的内存地址位置

MEM_L （1 个字节）

=要写入存储卡的数据的长度

字节(1…N)

=要与卡片内始于字节地址的数据做比较的字节值。字节 1 与在字节地址的数据比较；字节 N 与在字节地址 + N -1 的数据比较。

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

5.2.5.7. 提交存储卡密码（Present code memory card）（仅限 SLE44428 和 SLE5528）

此命令用于向存储卡提交密码，从而启用对 SLE4428 卡和 SLE5528 卡的写操作。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。
3. 擦除密码输入错误计数器。

命令

命令	CLA	INS	P1	P2	MEM_L	密码	
						字节 1	字节 2
Present Code Memory Card	FFh	20h	00h	00h	02h		

其中：

密码 （3 个字节）

密码（PIN）

响应

响应	响应数据域	
结果	90h	ErrorCnt

其中：

ErrorCnt （1 个字节）

错误计数器

FFh = 表示验证正确。

00h = 表示密码被锁定（超过最大重试次数）。

其它值表示验证失败。

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

5.2.6.1. 选择卡片类型（Select card type）

此命令用于对选定的卡片进行上电/下电，之后进行卡片复位操作。

命令

命令	CLA	INS	P1	P2	Lc	卡片类型
Select Card Type	FFh	A4h	00h	00h	01h	06h

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

5.2.6.2. 读取存储卡（Read memory card）

此命令会从指定的地址位置开始读取存储卡的内容。

命令

命令	CLA	INS	P1	字节地址	MEM_L
Read Memory Card	FFh	B0h	00h		

其中：

字节地址 （1 个字节）

=A7 A6 A5 A4 A3 A2 A1 A0b 是存储卡的内存地址位置

MEM_L （1 个字节）

待从存储卡读取的数据的长度。

响应

响应	字节 1	…	…	字节 N	PROT1	PROT2	PROT3	PROT4	SW1	SW2
结果										

其中：

字节(1…N) 从存储卡读取的数据

PROT (1…4) 含有保护位的字节

SW1 SW2 = 90 00h（若操作成功完成）

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 是响应数据中字节 *x* 的保护位：

0 = 字节写保护

1 = 字节可以被写入

5.2.6.3. 读取输入错误计数器（Read presentation error counter memory card）（仅限 SLE4442 和 SLE5542）

此命令用于读取密码输入错误计数器。

命令

命令	CLA	INS	P1	P2	MEM_L
Read Presentation Error Counter	FFh	B1h	00h	00h	04h

响应

响应	ErrCnt	虚拟数 1	虚拟数 2	虚拟数 3	SW1	SW2
结果						

其中：

ErrCnt （1 个字节）
密码输入错误计数器的值
07h = 表示验证正确
00h = 表示密码被锁定（超过最大重试次数）
其它值表示验证失败。

虚拟数 1, 虚拟数 2, 虚拟数 3 (3 个字节）
从卡片读取的虚拟数据

SW1 SW2 = 90 00h（若操作成功完成）

5.2.6.4. 读取保护位（Read Protection Bit）

此命令用于读取前 32 个字节的保护位。

命令

命令	CLA	INS	P1	P2	MEM_L
Read Protection Bit	FFh	B2h	00h	00h	04h

响应

响应	PROT 1	PROT 2	PROT 3	PROT 4	SW1	SW2
结果						

其中：

PROT (1..4) 含有保护位的字节

SW1 SW2 = 90 00h（若操作成功完成）

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 响应数据中字节的保护位：
0 = 字节写保护
1 = 字节可以被写入

5.2.6.5. 写入存储卡（Write memory card）

此命令会向指定地址位置写入存储卡内容。

命令

命令	CLA	INS	P1	字节地址	MEM_L	字节 1	…	…	字节 N
Write Memory Card	FFh	D0h	00h						

其中：

字节地址 （1 个字节）

= A7 A6 A5 A4 A3 A2 A1 A0b 是存储卡的内存地址位置

MEM_L （1 个字节）

要写入存储卡的数据的长度。

字节(1…N) 要写入存储卡的数据

响应

响应	响应数据域	
结果	SW1	SW2

其中：**SW1 SW2** = 90 00h（若操作成功完成）

5.2.6.6. 写保护存储卡（Write protection memory card）

此命令将命令中指定的每一个字节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

命令	CLA	INS	P1	字节地址	MEM_L	字节 1	…	…	字节 N
Write Protection Memory Card	FFh	D1h	00h						

其中：

字节地址 （1 个字节）

= 000A4 A3 A2 A1b (00h – 1Fh)是存储卡的保护内存地址位置

MEM_L （1 个字节）

要写入存储卡的数据的长度。

字节(1…N) 要与卡片内始于字节地址的数据做比较的字节值。字节 1 与在字节地址的数据比较；字节 N 与在字节地址 + N -1 的数据比较。

响应

响应	响应数据域	
结果	SW1	SW2

其中：**SW1 SW2** = 90 00h（若操作成功完成）

5.2.6.7. 提交存储卡密码（Present code memory card）（仅限 SLE4442 和 SLE5542）

此命令用于向存储卡提交密码，从而启用对 SLE4442 卡和 SLE5542 卡的写操作。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。
3. 擦除密码输入错误计数器。

命令

命令	CLA	INS	P1	P2	MEM_L	密码		
						字节 1	字节 2	字节 3
Present Code Memory Card	FFh	20h	00h	00h	03h			

其中：

密码 （3 个字节）

密码（PIN）

响应

响应	响应数据域	
结果	SW1	ErrorCnt

其中：

ErrorCnt （1 个字节）

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）

其它值表示验证失败。

5.2.6.8. 修改存储卡密码（Change code memory card）（仅限 SLE4442 和 SLE5542）

此命令用于将特定数据作为新密码写入卡片。执行此命令之前，需要先使用“Present Code”命令向卡片提交当前密码。

命令

命令	CLA	INS	P1	P2	MEM_L	密码		
						字节 1	字节 2	字节 3
Change Code Memory Card	FFh	D2h	00h	01h	03h			

其中：

密码 （3 个字节）

密码（PIN）

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

5.2.7. 存储卡 – SLE4406/SLE4436/SLE5536/SLE6636

5.2.7.1. 选择卡片类型（Select card type）

此命令用于对选定的卡片进行上电/下电，之后进行卡片复位操作。

命令

命令	CLA	INS	P1	P2	Lc	卡片类型
Select Card Type	FFh	A4h	00h	00h	01h	07h

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

5.2.7.2. 读取存储卡（Read Memory Card）

此命令会从指定的地址位置开始读取存储卡的内容。

命令

命令	CLA	INS	P1	字节地址	MEM_L
Read Memory Card	FFh	B0h	00h		

其中：

字节地址 （1 个字节）

存储卡的内存地址位置

MEM_L （1 个字节）

待从存储卡读取的数据的长度。

响应

响应	字节 1	…	…	字节 N	SW1	SW2
结果						

其中：

字节(1…N) 从存储卡读取的数据

SW1 SW2 = 90 00h（若操作成功完成）

5.2.7.3. 写入 1 个字节（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写入模式，通过命令数据域内的标志加以区分。

a. Write

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。因此，该模式仅适用于卡内计数器的值的更新。

c. Write with backup enabled (仅限 SLE4436、SLE5536 和 SLE6636)

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. Write with carry and backup enabled (仅限 SLE4436、SLE5536 和 SLE6636)

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。因此，该模式仅适用于卡内计数器的值的更新。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”。SLE4436 卡和 SLE5536 卡的备份模式可以在写操作中被启用或禁用。

命令

命令	CLA	INS	P1	字节地址	MEM_L	模式	字节
Read Memory Card	FFh	D0h	00h		02h		

其中：

字节地址 （1 个字节）
存储卡的内存地址位置

模式 （1 个字节）
指定写入模式和备份选项
00h = Write
01h = Write with carry
02h = Write with backup enabled (仅限 SLE4436、SLE5536 和 SLE6636)
03h = Write with carry and with backup enabled (仅限 SLE4436、SLE5536 和 SLE6636)

字节 （1 个字节）
待写入卡片的字节值

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

5.2.7.4. 提交存储卡密码（Present code memory card）

此命令用于向存储卡提交密码，从而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。

命令

命令	CLA	INS	P1	P2	MEM_L	密码			
						Addr	字节 1	字节 2	字节 3
Present Code Memory Card	FFh	20h	00h	00h	04h	09h			

其中：

Addr （1 个字节）

输入错误计数器的字节地址

密码 （3 个字节）

密码（PIN）

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

5.2.7.5. 认证存储卡（Authenticate memory card）（仅限 SLE4436、SLE5536 和 SLE6636）

此命令用于从卡片读取认证证书。执行的操作如下：

1. 根据命令在卡片中选择 Key 1 或 Key 2。
2. 将命令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 2 个步骤：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

命令	CLA	INS	P1	P2	MEM_L	密码				
						密钥	CLK_CNT	字节 1	…	字节 6
Send Authentication Certificate	FFh	84h	00h	00h	08h					

其中：

密钥 （1 个字节）

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（仅限 SLL5536 和 SLE6636）

81h = Key 2，带密码块链接（仅限 SLL5536 和 SLE6636）

CLK_CNT （1 个字节）

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）。

字节(1...6) 卡片随机数据。

响应

响应	SW1	SW2
结果	61h	02h

步骤 2：获取认证数据（Get Response）。

命令

命令	CLA	INS	P1	P2	MEM_L
Get Authentication Data	FFh	C0h	00h	00h	02h

响应

响应	Cert		SW1	SW2
结果				

其中：

Cert （2 个字节）

卡片计算出的 16 位的认证数据。字节 1 的 LSB 是从卡片中读取的第一个认证位。

SW1 SW2 = 90 00h（若操作成功完成）

5.2.8. 存储卡 – SLE4404

5.2.8.1. 选择卡片类型（Select card type）

此命令用于对选定的卡片进行上电/下电，之后进行卡片复位操作。

命令

命令	CLA	INS	P1	P2	Lc	卡片类型
Select Card Type	FFh	A4h	00h	00h	01h	08h

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

5.2.8.2. 读取存储卡（Read memory card）

此命令会从指定的地址位置开始读取存储卡的内容。

命令

命令	CLA	INS	P1	字节地址	MEM_L
Read Memory Card	FFh	B0h	00h		

其中：

字节地址 （1 个字节）
存储卡的内存地址位置

MEM_L （1 个字节）
待从存储卡读取的数据的长度。

响应

响应	字节 1	…	…	字节 N	SW1	SW2
结果						

其中：

字节(1…N) 从存储卡读取的数据

SW1 SW2 = 90 00h（若操作成功完成）

5.2.8.3. 写入存储卡（Write memory card）

此命令会向指定地址位置写入存储卡内容。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

命令	CLA	INS	P1	字节地址	MEM_L	字节 1	···	···	字节 N
Write Memory Card	FFh	D0h	00h						

其中：

字节地址 （1 个字节）

存储卡的内存地址位置

MEM_L （1 个字节）

要写入存储卡的数据的长度。

字节(1···N) 要写入存储卡的数据

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

5.2.8.4. 擦除暂存内存（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”。

命令

命令	CLA	INS	P1	字节地址	MEM_L
Erase Scratch Pad	FFh	D2h	00h		00h

其中：

字节地址 （1 个字节）

暂存存储器的内存字节地址位置。通常为 02h。

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

5.2.8.5. 校验用户密码（Verify user code）

此命令用于向插入的卡片提交用户密码（2 个字节）。该密码允许用户访问卡片的内存。

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

命令	CLA	INS	错误计数器 LEN	字节地址	MEM_L	密码	
						字节 1	字节 2
Verify User Code	FFh	20h	04h	08h	02h		

其中：

错误计数器 LEN （1 个字节）

密码输入错误计数器的长度，单位为比特

字节地址 （1 个字节）

卡片中密钥的字节地址

密码 （1 个字节）

用户密码

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

= 63 00h（若剩余重试次数为 0）

注：*收到响应 SW1 SW2 = 90 00h 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。*

5.2.8.6. 校验存储密码（Verify memory code）

此命令用于向插入的卡片提交存储密码（4 个字节）。该存储密码可授权用户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。

注：*存储错误计数器的内容不能被擦除。*

命令

命令	CLA	INS	错误计数器 LEN	字节地址	MEM_L	密码			
						字节 1	字节 2	字节 3	字节 4
Verify Memory Code	FFh	20h	40h	28h	04h				

其中：

错误计数器 LEN （1 个字节）

密码输入错误计数器的长度，单位为比特

字节地址 （1 个字节）

卡片中密钥的字节地址

密码 （4 个字节）

存储密码

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

= 63 00h（若剩余重试次数为 0）

注：*收到响应 SW1 SW2 = 90 00h 后，应当再次读取用户错误计数器，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。*

5.2.9. 存储卡 – AT88SC101/AT88SC102/AT88SC1003

5.2.9.1. 选择卡片类型（Select card type）

此命令用于对插入读写器的已选定卡片进行上电/下电，同时进行卡片复位操作。

命令

私有 APDU						
命令	CLA	INS	P1	P2	Lc	卡片类型
Select Card Type	FFh	A4h	00h	00h	01h	09h

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

5.2.9.2. 读取存储卡（Read Memory Card）

此命令会从指定的地址位置开始读取存储卡的内容。

命令

私有 APDU					
命令	CLA	INS	P1	字节地址	MEM_L
Read Memory Card	FFh	B0h	00h		

其中：

字节地址 （1 个字节）

存储卡的内存地址位置。

MEM_L （1 个字节）

要从存储卡读取的数据的长度。

响应

响应	字节 1	…	…	字节 N	SW1	SW2
结果						

其中：

字节(1…N) 从存储卡读取的数据

SW1 SW2 = 90 00h（若操作成功完成）

5.2.9.3. 写入存储卡（Write Memory Card）

此命令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

私有 APDU									
命令	CLA	INS	P1	字节地址	MEM_L	字节 1	…	…	字节 N
Write Memory Card	FFh	D0h	00h						

其中：

字节地址 （1 个字节）
存储卡的内存地址位置。

MEM_L （1 个字节）
要写入存储卡的数据的长度。

字节(1…N) 要写入卡片的字节值。

响应

响应	响应数据域	
结果	SW1	SW2

其中：**SW1 SW2** = 90 00h（若操作成功完成）

5.2.9.4. 擦除非应用区域（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：

- 指定的 Erase Application Zone With Erase 命令
- 指定的 Erase Application Zone With Write and Erase 命令
- 指定的 Verify Security Code 命令

命令

私有 APDU					
命令	CLA	INS	P1	字节地址	MEM_L
Erase Non-Application Zone	FFh	D2h	00h		00h

其中：

字节地址 （1 个字节）
要擦除的字的内存字节地址位置

响应

响应	响应数据域	
结果	SW1	SW2

其中：**SW1 SW2** = 90 00h（若操作成功完成）

5.2.9.5. 擦除应用区域（擦除）（Erase Application Zone with Erase）

此命令可用于下列情况：

- AT88SC101：擦除应用区域中的数据，EC 功能禁用
- AT88SC102：擦除应用区域 1 中的数据
- AT88SC102：擦除应用区域 2 中的数据，EC2 功能禁用
- AT88SC1003：擦除应用区域 1 中的数据
- AT88SC1003：擦除应用区域 2 中的数据，EC2 功能禁用
- 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。
2. 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

私有 APDU										
命令	CLA	INS	错误计数器 LEN	字节地址	MEM_L	密码				
						字节 1	字节 2	…	…	字节 N
Erase Application Zone with Erase	FFh	20h	00h							

其中：

错误计数器 LEN （1 个字节）
= 密码输入错误计数器的长度，单位为比特。值始终是 00h。

字节地址 （1 个字节）
= 卡片中应用区域密钥的字节地址。正确值请参阅下表：

MEM_L （1 个字节）
=“擦除”密钥的长度。正确值请参阅下表：

密码(1…N) =“擦除”秘钥

情形	字节地址	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
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

响应

响应	响应数据域	
结果	SW1	SW2

其中：**SW1 SW2** = 90 00h（若操作成功完成）

注：*收到状态字 SW1SW2 = 90 00h 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”，证明先前的验证成功。*

5.2.9.6. 擦除应用区域（写入和擦除）（Erase Application Zone with Write and Erase）

此命令可用于下列情况：

- AT88SC101：擦除应用区域中的数据，EC 功能启用
- AT88SC102：擦除应用区域 2 中的数据，EC2 功能启用
- 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 标识位没有被更改并处于“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

私有 APDU									
命令	CLA	INS	错误计数器 LEN	字节地址	MEM_L	密码			
						字节 1	字节 2	字节 3	字节 4
Erase Application Zone with Write and Erase	FFh	20h	80h		04h				

其中：

错误计数器 LEN （1 个字节）

= 密码输入错误计数器的长度，单位为比特。值始终是 80h。

字节地址 （1 个字节）

= 卡片中应用区域密钥的字节地址。正确值请参阅下表：

密码 （4 个字节）

= “擦除”密钥

情形	字节地址
AT88SC101	96h
AT88SC102	9Ch
AT88SC1003	5Ch

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

= 63 00（若剩余重试次数为 0）

注：*收到状态字 SW1SW2 = 90 00 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”，证明先前的验证成功。*

5.2.9.7. 校验安全密码（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

私有 APDU							
命令	CLA	INS	错误计数器 LEN	字节地址	MEM_L	密码	
						字节 1	字节 2
Verify Security Code	FFh	20h	08h	0Ah	02h		

其中：

错误计数器 LEN （1 个字节）

= 密码输入错误计数器的长度，单位为比特。

字节地址 （1 个字节）

= 卡片中密钥的字节地址。

密码 （2 个字节）

= 安全密码

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

= 63 00（若剩余重试次数为 0）

注：*收到响应 SW1SW2 = 90 00h 后，应当再次读取安全密码尝试计数器（SCAC），检查 Verify User Code 是否正确。如果 SCAC 已经被擦除并且等于“FFh”，证明先前的验证成功。*

5.2.9.8. 更改标识位（Blow 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

私有 APDU									
命令	CLA	INS	错误计数器 LEN	字节地址	MEM_L	密码			
						标识位地址(高)	标识位地址(低)	FUS Pin 状态	RST Pin 状态
Blown Fuse	FFh	05h	00h	00h	04h			01h	00h 01h

其中：

标识位地址 （2 个字节）

= 标识位的位地址。正确值请参阅下表：

FUS Pin 状态 （1 个字节）

= FUS Pin 的状态，始终应该是 01h。

RST Pin 状态 （1 个字节）

= RST pin 的状态，正确值请参阅下表。

		标识位地址(高)	标识位地址(低)	RST Pin 状态
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

表 3：更改标识位码值

响应

响应	响应数据域	
结果	SW1	SW2

其中：

SW1 SW2 = 90 00h（若操作成功完成）

= 63 00（若剩余重试次数为 0）

5.2.10. ACOS6-SAM 卡命令

本节介绍 SAM 专用命令。CCID 主机可以使用 PCSC API 中的 SCardTransmit()向读写器发送卡片专有命令（Card Native Command）或 APDU。

注：*请注意，ACOS6-SAM 卡仅随附于 ACR1581U 读写器的配套 SDK，标准的零售版读写器内不包含 ACOS6-SAM 卡。如需了解 ACOS6-SAM 命令的所有信息和应用场景，请联系 ACS 销售代表索取 ACOS6-SAM 参考手册。*

5.2.10.1. 密钥生成（Generate Key）

该命令利用客户卡序列号等偏差数据生成分散密钥，并导入 ACOS3/6 或其它卡片中，用于满足客户发卡的目的。

APDU	说明	
CLA	80h	
INS	88h	
P1	00h	生成 8 字节密钥
	01h	生成 16 字节密钥
	02h	生成 24 字节密钥
P2	用于生成分散密钥的主密钥的索引	
P3	08h	
数据	输入数据	

特定的响应报文状态字节：

SW1 SW2	说明
69 86h	未选择 DF
6A 86h	P1 或 P2 无效
67 00h	P3 不正确，必须是 08h
6A 83h	在 EF2 中找不到指定的密钥记录
69 81h	EF2 无效（记录大小、文件类型等）
6A 88h	找不到 EF2
62 83h	当前 DF 被锁定；EF2 被锁定
69 83h	使用计数器为 0
69 82h	不满足安全条件
6A 87h	指定的主密钥不支持 3DES 加密
61 08h	命令完成，发送 GET RESPONSE 取结果

5.2.10.2. 密钥数据分散（或载入）（Diversify (or Load) Key Data））

该命令通过密钥分散和密钥载入使 SAM 卡准备好执行加密操作。它将序列号和 CBC 初始向量作为命令数据输入。

APDU	说明								
CLA	80h								
INS	72h								
P1	b7	b6	b5	b4	b3	b2	b1	b0	说明
	-	0	0	0	0	0	0	1	密码(Sc)
	-	0	0	0	0	0	1	0	帐户密钥(K_{ACCT})
	-	0	0	0	0	0	1	1	终端密钥
	-	0	0	0	0	1	0	0	卡片密钥
	-	0	0	0	0	1	0	1	批量加密密钥(非分散)
	-	0	0	0	0	1	1	0	初始向量
	0	-	-	-	-	-	-	-	16 字节密钥
	1	-	-	-	-	-	-	-	24 字节密钥
P2	主密钥的索引： Bit7: 1 = 当前 EF2 中的局部密钥； 0 = 全局密钥 EF2 Bit6-Bit5: 00b - RFU Bit4-Bit0: 密钥索引								
P3	若 P1 = 1-4, 则 P3 = 8/16,(如果算法为 AES，则 P3 = 8/16) 若 P1 = 5, 则 P3 = 0 若 P1 = 6, P3 = 8 (主密钥的算法为 DES/ 3DES/ 3KDES) P3 = 16 (主密钥的算法为 AES)								
数据	如果 P1 = 1-4，客户卡的序列号，(若算法为 AES，数据是客户卡的序列号，或者客户卡的序列号后面再加上“0000000000000000”) 如果 P1 = 5, 无命令数据. 如果 P1 = 6, DES/3DES/3KDES/AES CBC 初始向量。								

特定的响应报文状态字节：

SW1 SW2	说明
69 86h	未选择 DF
6A 86h	P1 错误，P1 必须为 1-6
67 00h	P3 错误，P3 必须为 8（或 0）
62 83h	当前 DF 被锁定，或者 EF2 被锁定
69 82h	不满足安全条件
6A 88h	找不到 EF2

SW1 SW2	说明
6A 83h	EF2 中找不到指定的主密钥
69 81h	EF2 无效（FDB、MRL 等不一致）
6A 87h	指定的密钥不支持认证
69 83h	指定的密钥被锁定
90 00h	已生成目标密钥，存在 SAM 存储器中

5.2.10.3. 加密（Encrypt）

该命令使用 DES 或 3DES 算法来加密数据，它会使用：

1. 与 ACOS3/6、DESFire®、DESFire® EV1/EV2/Light 或 MIFARE Plus 卡片相互认证生成的过程密钥。
2. 分散密钥（密码）。
3. 批量加密密钥。
4. 使用过程密钥对分散密码进行加密。
5. 给定一个非安全报文命令，准备 ACOS3 安全报文命令。

APDU	说明								
CLA	80h								
INS	74h								
P1	b7	b6	b5	b4	b3	b2	b1	b0	说明
	-	0	0	0	0	0	0	-	ECB 模式
	-	0	0	0	0	0	1	-	CBC 模式
	-	0	0	0	0	1	0	-	零售 MAC 模式
	-	0	0	0	0	1	1	-	MAC 模式
	-	0	0	0	1	0	0	-	准备 ACOS3 SM 命令
	-	1	0	0	1	0	1	-	MIFARE DESFire 加密
	-	1	0	0	1	1	0	-	MIFARE DESFire EV1/EV2/Light 加密
	-	0	0	0	1	1	1	-	CMAC
	-	0	1	0	0	0	0		MIFARE Plus 命令
	-	0	1	0	0	0	1		MIFARE Plus 响应
	0	-	-	-	-	-	-	0	3DES
	0	-	-	-	-	-	-	1	DES
	1	-	-	-	-	-	-	0	3K DES
	1	-	-	-	-	-	-	1	AES
	-	-	-	-	-	-	-	-	所有其他值 - RFU

APDU	说明
P2	P2 代表使用 Load Key 功能在 SAM 集中分散出的密钥： 1 – 使用过程密钥 *Ks* 对数据进行加密 2 – 使用分散密钥 *Sc* 对数据进行加密 3 – 使用批量加密密钥对数据进行加密 0 – 返回 ENC (*Sc*, *Ks*) 如果 P1.b3 = 1 或 b5=1, P2 必须为 1 如果 P2 = 0h, P1 可以是 0 或 1
P3	P3 < 128 如果 P1 的 bit 3 不等于 1，并且 P1 的 bit 5 也不等于 1 - 如果 P2 = 1-3, 8(DES/3DES/3KDES)或 16(AES)的倍数，最高 128 字节 - 如果 P2 = 0, 0
数据	明文 如果 P2 b6 = 1，数据格式应该是： • 明文数据的长度 • DESFire 卡片的命令和卡片头的长度 • DESFire 卡片的命令和卡片头 • 明文 若 P1 = A1h，该加密用于 MIFARE Plus 命令 • 如果 MFP 命令是一个值操作命令，数据的格式应该是：Command Code(1 个字节)+BlockNum(2/4 个字节)+Value(4 个字节)。 • 如果 MFP 命令是接近度检测，数据的格式应该是：Command Code(1 个字节)+ PPS1(1 个字节)。 • 如果 MFP 命令是读，数据的格式应该是：Command Code(1 个字节)+ BlockNum(2 个字节)。 • 如果 MFP 命令是写，数据的格式应该是 Command Code(1 个字节)+ BlockNum(2 个字节)+plaintext P1=A3h, • 由 ICC 返回的数据（不包括 SC 码也不包括 RMAC（如果存在 RMAC)） 若先前已使用 Authenticate EV2 First/ Authenticate EV2NonFirst 命令进行认证，则 P1 应为 CDh 或 8Fh。 • 如果 P1=CDh，则加密操作用于 DESFire EV2/Light 命令。数据的格式应该是： 明文数据 • 如果 P1=8Fh，则 CMAC 是用于 DESFire EV2/Light 命令或响应。数据的格式应该是： ➢ EV2 命令：完整命令（不含 CMAC 数据） ➢ EV2 响应：来自 EV2 的完整响应 ➢ Light 命令：完整命令（不含 CMAC 数据和 CLA/P1/P2/Lc/Le 字节）。 ➢ Light 响应：来自 Light 的 SW2 和响应数据。 如果 EV2/Light 卡仅返回状态码和明文数据，则数据应当是 00h，并发送至 SAM 卡。

特定的响应报文状态字节：

SW1 SW2	说明
69 86h	未选择 DF
6A 86h	P1 或 P2 无效
67 00h	P3 不正确
6A 83h	ACOS 目标密钥未准备就绪（使用 Diversify 命令生成密钥）
61 XX	加密完成，使用 GET RESPONSE 获取结果

5.2.10.4. 解密（Decrypt）

该命令用于通过 DES、3DES 或 AES 算法来解密数据，它会使用：

1. 与ACOS3/6、MIFARE DESFire、MIFARE DESFire EV1/EV2/Light或MIFARE Plus卡片相互认证生成的过程密钥。
2. 分散密钥（密码）。
3. 批量加密密钥。
4. 使用过程密钥对分散密码进行解密。
5. 查验并解密ACOS3安全报文响应数据

查验并解密 ACOS3 安全报文响应数据

APDU	说明								
CLA	80h								
INS	76h								
	b7	b6	b5	b4	b3	b2	b1	b0	说明
	-	0	0	0	0	0	0	-	ECB 模式
	-	0	0	0	0	0	1	-	CBC 模式
	-	0	0	0	1	0	0	-	查验并解密 ACOS3 安全报文应答
	-	1	0	0	1	0	1	-	MIFARE DESFire 解密
P1	-	1	0	0	1	1	0	-	MIFARE DESFire EV1 解密
	-	0	1	0	0	1	0	-	MIFARE Plus 解密
	0	-	-	-	-	-	-	0	3DES
	0	-	-	-	-	-	-	1	DES
	1	-	-	-	-	-	-	0	3K DES
	1	-	-	-	-	-	-	1	AES
	0	0	0	0	-	-	-	-	所有其他值 - RFU
P2	P2 代表使用 Load Key 功能在 SAM 集中分散出的密钥： 1 – 使用过程密钥 *Ks* 对数据进行解密 2 – 使用分散密钥 *Sc* 对数据进行解密 3 – 使用批量加密密钥对数据进行解密 0 – 返回 DEC(*Sc*, *Ks*)								

APDU	说明
P3	P3 < 128 如果 P1 = A5h, P3=16/32/48 如果 P1 的 bit 3 不等于 1 - 如果 P2 = 1-3, 8(DES/3DES/3KDES)或 16(AES)的倍数，最高 128 字节 - 如果 P2 = 0, 0
数据	密文 如果 P1 = A5h，数据是加密的文本 如果 P2 b6 = 1，数据格式应该是： • 明文数据的长度，如果未知，使用 00 • DESFire 卡片的命令和卡片数据头的长度 • DESFire 卡片的命令和卡片数据头 • 加密的文本 若先前已使用 Authenticate EV2 First/ Authenticate EV2NonFirst 命令完成认证，且 EV2/Light 卡返回经加密及附加 CMAC 的数据，则 P1 应为 CDh。数据的格式应该是： • EV2 响应：来自 EV2 的完整响应 • Light 响应：来自 Light 的 SW2 和响应数据。

特定的响应报文状态字节：

SW1 SW2	说明
69 86h	未选择 DF
6A 86h	P1 或 P2 无效
67 00h	P3 不正确
6A 83h	ACOS 目标密钥未准备就绪（使用 Diversify 命令生成密钥）
61 XX	解密完成，使用 GET RESPONSE 获取结果

5.2.10.5. 认证准备（Prepare Authentication)

该命令用于验证 SAM 卡（作为终端）对于 ACOS 3/6 卡，或者 MIFARE Ultralight C/MIFARE DESFire 卡/MIFARE DESFire EV1/EV2/Light 卡/MIFARE Plus 卡的合法性。

APDU	说明
CLA	80h
INS	78h
P1	00h – 3DES 01h – DES 02h – 3KDES (MIFARE DESFire EV1/ACOS3) 03h – AES (MIFARE DESFire EV1/MIFARE Plus/ACOS3) 80h – 3DES (仅限 MIFARE DESFire 验证)

APDU	说明
	81h – DES (仅限 MIFARE DESFire 验证) 其它 – RFU
P2	0h – 查验 ACOS3/6 验证返回信息 01h – MIFARE Ultralight C/DESFire 验证，通过（分散的）终端密钥 05h – MIFARE Ultralight C/DESFire 验证，通过批量加密密钥 02h – MIFARE Plus 认证。从 SL1 到 SL3 的首次认证 03h – MIFARE Plus 认证。从 SL1 到 SL2 中的认证 04h – MIFARE Plus 认证。从 SL2 到 SL3 的跟随认证 06h – MIFARE DESfire EV2/Light AuthenticateEV2First/AuthenticateEV2NonFirst/Mifare Plus EV1 安全报文发送
P3	8 – (P1 = 00h, 01h, 02h, 80h, 81h) 16 – (P1 = 03h)
数据	卡片随机数据

特定的响应报文状态字节：

SW1 SW2	说明
69 86h	未选择 DF
6A 86h	P1 或 P2 无效
67 00h	P3 不正确，必须是 08h
6A 83h	ACOS 密钥（KT 或 KC）未准备就绪，（使用 Diversify 生成该密钥）
69 82h	不满足安全条件
61 10h	命令完成，发送 GET RESPONSE 取结果

5.2.10.6. 校验认证（Verify Authentication）

此命令用于校验 ACOS 3/6、MIFARE Ultralight C、MIFARE DESFire/MIFARE DESFire EV1 或 MIFARE Plus 卡对于终端的合法性，也会在内部生成过程密钥 Ks。

APDU	说明
CLA	80h
INS	7Ah
P1	00h – 3DES (P2 = 0) 01h – DES (P2 = 0) 02h – 3KDES (P2 = 0，ACOS3) 03h – AES (P2 = 0，ACOS3) 其它 – RFU
P2	00h – 查验 ACOS3/6 认证返回信息 01h – 查验 MIFARE Ultralight C®/ DESFire®/ DESFire® EV1 认证返回信息

APDU	说明
	02h – 查验 MIFARE Plus 认证返回信息，或者 MIFARE DESFire EV2/Light 的 AuthenticateEV2First/AuthenticateEV2NonFirst 返回信息
P3	08h – (P2 = 0, P2 = 1，且过程密钥采用 DES/3DES) 16h – (P2 = 1，且过程密钥采用 3KDES/AES) 16h – (P2=02，且 MIFARE Plus 返回数据 ek(RndA’)) 32h – (P2=02，且 MIFARE Plus 返回数据 ek(TI+PICCcap2+PCDcap2))
数据	ACOS 3/6：DES (K_S, RND_T) MIFARE DESFire/ DESFire EV1/EV2/Light 返回数据：ek(RndA’) MIFARE Plus 返回数据 ek(RndA’)或 ek(TI+PICCcap2+PCDcap2)

特定的响应报文状态字节：

SW1 SW2	说明
69 86h	未选择 DF
6A 86h	P1 或 P2 无效
67 00h	P3 不正确，必须是 08h
6A 83h	ACOS-SAM 过程密钥或 RND_T 没有准备就绪。使用 PREPARE AUTHENTICATION 命令来生成这些密钥。
69 82h	数据不正确
90 00h	数据正确，ACOS 相互认证成功

5.2.10.7. ACOS 查询帐户校验（Verify ACOS Inquire Account）

该命令用于检查 ACOS3/6 卡片的查询帐户钱包命令。它会使用 SAM 的分散密钥验证 ACOS3/6 返回的 MAC 校验和是否正确。

APDU	说明								
CLA	80h								
INS	7Ch								
P1	b7	b6	b5	b4	b3	b2	b1	b0	说明
	-	0	0	0	0	-	0	-	ACOS INQ_AUT 未启用
	-	0	0	0	0	-	1	-	ACOS INQ_AUT 启用
	-	0	0	0	0	0	-	-	ACOS INQ_ACC_MAC 未启用
	-	0	0	0	0	1	-	-	ACOS INQ_ACC_MAC 启用
	0	-	-	-	-	-	-	0	3DES
	0	-	-	-	-	-	-	1	DES
	1	-	-	-	-	-	-	0	3K DES（仅 ACOS3）
	1	-	-	-	-	-	-	1	AES（仅 ACOS3）
P2	0h								

APDU	说明
P3	1Dh
数据	客户 ACOS 卡片的 INQUIRE ACCOUNT 命令返回的数据块。

特定的响应报文状态字节：

SW1 SW2	说明
69 86h	未选择 DF
6A 86h	P1 或 P2 无效
67 00h	P3 不正确
6A 83h	ACOS 密钥 K_S 或 K_{ACCT} 未准备就绪；使用 DIVERSIFY 命令生成 K_{ACCT}；如适用，通过“Prepare Authentication”生成 K_S。
6F 00h	数据块的 MAC 不正确
90 00h	数据块的 MAC 正确

5.2.10.8. ACOS 账户交易准备（Prepare ACOS Account Transaction）

为了生成 ACOS3/6 充值(Credit)/扣款(Debit)命令，必须计算 MAC 供 ACOS3/6 进行校验。

APDU	说明								
CLA	80h								
INS	7Eh								
P1	b7	b6	b5	b4	b3	b2	b1	b0	说明
	-	0	0	0	0	0	0	-	ACOS TRNS_AUT 未启用
	-	0	0	0	0	0	1	-	ACOS TRNS_AUT 启用
	0	-	-	-	-	-	-	0	3DES
	0	-	-	-	-	-	-	1	DES
	1	-	-	-	-	-	-	0	3K DES（仅 ACOS3）
	1	-	-	-	-	-	-	1	AES（仅 ACOS3）
P2	E2h: 充值 E6h: 扣款								
P3	0Dh								
数据	数据块								

特定的响应报文状态字节：

SW1 SW2	说明
69 86h	未选择 DF
6A 86h	P1 或 P2 无效

SW1 SW2	说明
67 00h	P3 不正确，必须是 0Dh
6A 83h	ACOS 密钥 K_S 或 K_{ACCT} 未准备就绪；使用 DIVERSIFY 命令生成 K_{ACCT}；如适用，通过“Prepare Authentication”生成 K_S。
61 0Bh	命令完成，发送 GET RESPONSE 取结果

5.2.10.9. 扣款证书查验（Verify Debit Certificate）

对于 ACOS3/6，若 DEBIT 命令中的 P1=1，会返回一个扣款证书。可以通过比较此命令的结果和 ACOS3 的响应报文对该扣款证书进行检查。

APDU	说明								
CLA	80h								
INS	70h								
P1	b7	b6	b5	b4	b3	b2	b1	b0	说明
	-	0	0	0	0	0	0	-	ACOS TRNS_AUT 未启用
	-	0	0	0	0	0	1	-	ACOS TRNS_AUT 启用
	0	-	-	-	-	-	-	0	3DES
	0	-	-	-	-	-	-	1	DES
	1	-	-	-	-	-	-	0	3K DES（仅 ACOS3）
	1	-	-	-	-	-	-	1	AES（仅 ACOS3）
P2	0h								
P3	14h								
数据	数据块								

特定的响应报文状态字节：

SW1 SW2	说明
69 86h	未选择 DF
6A 86h	P1 或 P2 无效
67 00h	P3 不正确，必须是 14h
6A 83h	ACOS 密钥 K_S 或 K_{ACCT} 未准备就绪；使用 DIVERSIFY 命令生成 K_{ACCT}；如适用，运行 PREPARE AUTHENTICATION 生成 K_S。
69 82h	不满足安全条件
6F 00h	DEBIT CERTIFICATE 无效
90 00h	成功，DEBIT CERTIFICATE 有效

5.2.10.10. 取密钥（Get Key)

取密钥命令使密钥从当前 SAM 的密钥文件（SFI=02h）安全地注入另外一张 ACOS6/ACOS6-SAM 卡片，这一过程可以通过也可以不通过密钥分散来实现。这样做可以确保待导入的密钥受到加密和消息验证代码的保护。

此外，该命令还可以通过密钥分散，使密钥安全地从当前 SAM 的密钥文件（SFI=02h）注入 ACOS7/10、MIFARE DESFire、MIFARE DESFire EV1/EV2/Light 或 MIFARE Plus 卡。这样做可以确保待导入的密钥受到加密和消息验证代码的保护。

若**卡片头模块**（见 ACOS6-SAM 参考手册第 4.2 节）设置了特殊功能标志 bit7（仅密钥注入标志），且密钥文件已被激活，必须使用 Get Key 才可以载入或变更卡片内的密钥。Bit7 设置后，密钥文件一旦激活，在任何情况下都禁用对其使用 Read Record 命令。

在取密钥命令执行之前，已经通过**相互认证**（ACOS6-SAM 参考手册第 6.3 节）中讲述的相互认证过程，或者是 MIFARE Plus/MIFARE DESFire 的相互认证过程在目标卡片中建立了过程密钥。

注：*GET KEY 命令只能获取密钥数据。*

APDU	说明			
CLA	80h			
INS	CAh			
P1	取密钥，供 ACOS 卡写/重装密钥			
	00h	响应数据是 MSAM 中的密钥		
	01h	响应数据是 16 个字节的分散密钥		
	02h	响应数据是 24 个字节的分散密钥		
	03h	响应数据是 MIFARE Plus 卡的 Change Key 命令		
	取密钥，供 DESFire 卡更改密钥，响应数据供 DESFire/DESFire EV1 更改密钥。			
		卡片类型	验证密钥号和修改密钥号*	密钥长度
	80h	MIFARE DESFire	在 MIFARE DESFire 卡片中是不同的	16 字节
	81h	MIFARE DESFire EV1	在 MIFARE DESFire EV1 卡片中是不同的	16 字节
	82h	MIFARE DESFire EV1	在 MIFARE DESFire EV1 卡片中是不同的	24 字节
	88h	MIFARE DESFire	在 MIFARE DESFire 卡片中是相同的	16 字节
	89h	MIFARE DESFire EV1	在 MIFARE DESFire EV1 卡片中是相同的	16 字节
	8Ah	MIFARE DESFire EV1	在 MIFARE DESFire EV1 卡片中是相同的	24 字节
P2	SAM 中的 Key ID（用于变更的新密钥）			
P3	若 P1 = 00h，P3 = 08h 若 P1 = 01/02h，P3 = 10h 若 P1 = 03h，P3 = 0Bh 若 P1 = 80/81/82/88/89/8Ah：P3 = 0Bh 或 0Dh			

APDU	说明
	若 P1 = 00h，命令数据为 RND_{Target}
	若 P1 = 01/02h，命令数据为 RND_{Target} + 目标卡片的序列号（或批号）
	若 P1 = 03h - 目标卡片的序列号（8 字节） - 写命令（A0 或 A1）（1 个字节） - BNr (2 个字节)
数据	若 P1 = 80/81/82/88/89/8Ah： - 目标卡片的序列号（8 字节） - 初始 Key ID（SAM 卡中的 Key 存储了初始 key，00=DESFire 卡的默认 Key) - Key No.(DESFire 卡的 Key No.) - Key Version（DESFire 卡的 Key 版本，如未使用，值=00） - 命令：C6（可选） - KeySetNo（可选）

* 此列表指出所列卡片是否具有不同的 Change Key 和 Authenticate Key，或者两个密钥是否使用相同的值。

特定的响应报文状态字节：

SW1 SW2	说明
69 85h	SAM 过程密钥未准备就绪
62 83h	当前 DF 被锁定，或目标 EF 被锁定
69 86h	未选择 DF
69 81h	KEY 文件的类型错误，应该是内部线性变长文件
69 82h	目标文件头块的校验和错误，或者不满足安全条件
6A 86h	P1 或 P2 无效
67 00h	P3 不正确
6A 83h	目标密钥未准备好或密钥长度小于 16
61 1Ch	命令成功，使用 GET RESPONSE 获取结果

5.3. 非接触式智能卡协议

5.3.1. ATR 的生成

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

5.3.1.1. ATR 信息格式（适用于 ISO14443-3 PICC）

字节	值	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 表示历史字符的个数（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 ~ 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 06
	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	UU	TCK	T0至Tk的所有字节按位异或

例如：

MIFARE Classic 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 Classic 1K)

标准(SS) = 03h：ISO 14443A，第3部分
= 11h：FeliCa

卡片名称(C0 ..C1)：

00 01：MIFARE Classic 1K
00 02：MIFARE Classic 4K
00 03：MIFARE Ultralight®
00 26：MIFARE Mini®
00 3A：MIFARE Ultralight® C
00 36：MIFARE Plus® SL1 2K
00 37：MIFARE Plus® SL1 4K
00 38：MIFARE Plus® SL2 2K
00 39：MIFARE Plus® SL2 4K
00 30：Topaz和Jewel
00 3B：FeliCa
FF 28：JCOP 30
FF [SAK]：尚未定义的标签

5.3.1.2. ATR 信息格式（适用于 ISO14443-4 PICC）

<table>
<tr><th>字节</th><th>值</th><th>标记</th><th>说明</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 表示历史字符的个数（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 ~
3+N</td><td>XX</td><td>T1</td><td rowspan="2">历史字节：

ISO 14443-A：
来自ATS响应的历史字节。参考ISO 14443-4标准。

ISO 14443-B：
<table>
<tr><th>字节1~4</th><th>字节5~7</th><th>字节8</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</td><td>Tk</td></tr>
<tr><td>4+N</td><td>UU</td><td>TCK</td><td>T0至Tk的所有字节按位异或</td></tr>
</table>

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

5.3.2. APDU、私有（Psedo）APDU 和卡片专有（Native）命令

CCID 主机可以使用 CCID 报文 PC_to_RDR_XfrBlock（对应于 PCSC API 中的 SCardTransmit()）向读写器发送卡片专有命令或 APDU。对于 PICC，如果卡片支持 ISO14443 第 4 部分协议或 Innovation 协议，读写器会将命令/APDU 打包到协议帧中直接发送给卡片，不会对命令/APDU 进行解析。如果卡片不支持这两种协议，则会向 CCID 主机返回消息“6A 81”。

注：由于 Microsoft Window 支持智能卡即插即用，Microsoft Window 可能在卡片出示时向卡片发送 APDU 指令。该操作会使 DESFire 卡进入 ISO APDU 模式，使得卡片无法接收专有命令，除非重置卡片。通常情况下，Microsoft Window 会在卡片处于无反应状态 10 秒后重置卡片（通过 PC_to_RDR_IccPowerOff）。

5.3.3. PICC 的 PCSC 私有（Pseudo）APDU（带专有扩展）

下列私有 APDU 用于间接访问非接触卡。CCID 主机可以使用 CCID 报文 PC_to_RDR_XfrBlock（对应于 PCSC API 中的 SCardTransmit()）向读写器发送这些 APDU。收到私有 APDU 后，读写器会解读生成低级别的卡片命令，然后发送给卡片。卡片处理完这些低级别命令后，读写器收集卡片响应并创建响应发回 CCID 主机。

5.3.3.1. 获取数据（Get Data）[FF CA …]

此命令用来读取激活过程中获得的数据，例如序列号、协议参数等。

命令

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

命令参数

P1	P2	含义
00h	00h	获取卡片的 UID/PUPI/SN
01h	00h	获取 A 类第 4 部分的 ATS
02h	00h	获取以下卡片类型相关数据，传输顺序： A 类：2 字节 ATQA/ATVA + 4/7/10 字节 UID + 1 字节最后一个 SAK。 B 类：12 字节 ATQB
80h	00h	获取以下卡片类型相关数据，传输顺序： A 类：2 字节 ATQA/ATVA + 4/7/10 字节 UID + 1/2/3 字节 SAK。 B 类：12 字节 ATQB FeliCa：17 字节 ATQ (+ 6 字节 ATTR，如已激活) SRI：8 字节 UID + 1 字节芯片 ID。

P1	P2	含义
		ISO15693：1 字节 DSFID + 8 字节 UID CTS：4 字节 SN + 2 字节 ATQT Innovatron：4 字节 SN + 1 字节标签地址。

响应

响应	响应数据域		
结果	数据	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	6X XXh	失败。

例如：

获取“已经建立连接的 PICC”的序列号：

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

获取“已经建立连接的 ISO 14443-A PICC”的 ATS：

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

5.3.3.2. 加载密钥（Load Key）[FF 82 …]

此命令用于向密钥缓冲区号指定的内部密钥缓冲区加载密钥数据。密钥缓冲区属于易失存储区，里面的内容将用于认证。此命令不会产生卡片数据传输。

命令

命令	CLA	INS	P1	P2	Lc	命令数据域
Load Authentication Keys	FFh	82h	00h	密钥缓冲区号（0-1）	密钥长度	密钥数据

密钥长度/数据

卡片类型	密钥长度 (Lc)	密钥数据 (按传输/存储顺序)
MIFARE Standard MIFARE Plus SL1	06h	6 字节 Crypto1 Key A/B。
MIFARE Plus SL1 MIFARE Plus SL2	16h	6 字节 Crypto1 Key A/B + 16 字节 AES Key。

卡片类型	密钥长度 (Lc)	密钥数据 (按传输/存储顺序)
MIFARE Plus SL2	06h	6 字节加密 Crypto1 Key A/B。
MIFARE Ultralight C MIFARE DESFire	10h	16 字节 2K3DES Key。

响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	6X XXh	失败。

例如：

// 向易失性存储器位置 00h 加载密钥 {FF FF FF FF FF FFh}。

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

5.3.3.3. 认证（Authenticate）[FF 86 00 00 05 …]

此命令用于向卡片执行认证过程。认证成功后，用户可以访问受保护的块/页。命令发送前，用户需通过 Load Key 命令将正确的密钥数据加载到**密钥缓冲区号**指定的缓冲区。

命令

命令	CLA	INS	P1	P2	Lc	命令数据域
Authenticate	FFh	86h	00h	00h	05h	见下表

命令数据

字节 0	字节 1	字节 2	字节 3	字节 4
01h	00h (RFU)	地址	密钥类型	密钥缓冲区号

地址和密钥类型

卡类型	地址	密钥类型
MIFARE Standard MIFARE Plus SL1 MIFARE Plus SL2	00h~FFh：块 0~255	60h：Crypto1 Key A 61h：Crypto1 Key B
MIFARE UltraLightC	00h (RFU)	80h：2K3DES
MIFARE DESFire	00h~0Eh：DESFire 密钥号 0~14	0Ah：2K3DES

响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	6X XXh	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 字节)	尾部块 (1 个块，16 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

1 KB

表 4： MIFARE Classic 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 字节)	尾部块 (1 个块，16 字节)
扇区 0	00h ~ 02h	03h
扇区 1	04h ~ 06h	07h
..		
..		
扇区 30	78h ~ 7Ah	7Bh
扇区 31	7Ch ~ 7Eh	7Fh

2 KB

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 字节)	尾部块 (1 个块，16 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

表 5： MIFARE Classic 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 6：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 V2.07

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。

5.3.3.4. 读取二进制块（Read Binary Blocks）[FF B0 …]

此命令用于从指定块/页地址的位置开始从 PICC 读取指定字节的数据。根据卡片类型的不同，调用此命令前，用户可能需要先进行认证并获得这些块/页的访问权限。

命令：

命令	CLA	INS	P1	P2	Le
Read Binary Blocks	FFh	B0h	模式和地址		待读取的字节数

P1/P2（模式和地址）

卡类型	P1[7:4] 模式	P1[3:0] + P2[7:0] 起始地址（MSB 在前）
MIFARE Standard MIFARE Plus SL1 MIFARE Plus SL2	0h：跳过尾部块 8h：包含尾部块	000h~0FFh：块 0~255
MIFARE Ultralight MIFARE Ultralight C	0h（保留）	000h~02Fh：页 0~47
SRIX4K/SRT512	0h（保留）	000h~07Fh：块 0~127 0FFh：系统区域
PicoPass	0h（保留）	000h~0FFh：块 0~255
ISO15693	0h（保留）	000h~7FFh：块 0~2047
Topaz/NFC Type-1 标签	0h（保留）	000h~7FFh：字节地址
CTS	0h（保留）	000h~01Fh：块 0~31

Le（待读取的字节数）

类型	字节 0	字节 1	字节 2
短	00h：读取 256 字节 01h~FFh：读取 1~255 字节	--	
长	00h	0000h：读取 65536 字节 0001h~FFFFh：读取 1~65535 字节	

响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	6X XXh	失败。

例如：

// 从二进制块 04h 中读取 16 字节（MIFARE Classic 1K 或 4K）

APDU = FF B0 00 04 10h

// 从二进制块 80h 开始读取 240 字节（MIFARE Classic 4K）

// 块 80h 至块 8Eh（15 个块）

APDU = FF B0 00 80 F0h

5.3.3.5. 更新二进制块（Update Binary Blocks）[FF D6 …]

此命令用于从指定块/页地址的位置开始向 PICC 写入指定字节（必须是块/页大小的倍数）的数据。根据卡片类型的不同，调用此命令前，用户可能需要先进行认证并获得这些块/页的访问权限。

向卡片内的块/页写入数据可能改变卡片的安全设置（例如 MIFARE 卡的尾部块），所以应当格外小心。如果写入错误的数据或者操作失败，可能会将卡锁死。为了减少卡片锁定的风险，不建议在涉及安全块/页时，通过一个 APDU 命令向多个块/页写入数据。

命令

命令	CLA	INS	P1	P2	Lc	命令数据域
Update Binary Blocks	FFh	D6h	模式和地址		待写入的字节数量	数据字节

P1/P2（模式和地址）和 Write Size 一致（块/页大小）

卡类型	P1[7:4] 模式	P1[3:0] + P2[7:0] 起始地址（MSB 在前）	块/页大小（字节）
MIFARE Standard MIFARE Plus SL1 MIFARE Plus SL2	0h：跳过尾部块 8h：包含尾部块	000h~0FFh：块 0~255	16
MIFARE Ultralight MIFARE UltraLightC	0h（保留）	000h~02Fh：页 0~47	4
SRIX4K/SRT512	0h（保留）	SRIX4K/SRT512	4
PicoPass	0h（保留）	PicoPass	8
ISO15693	0h（保留）	000h~7FFh：块 0~2047	ISO15693
Topaz/NFC Type-1 标签	0h：包括擦除 8h：不包括擦除	000h~7FFh：字节地址	1(地址 78h)或 8(其它)
CTS	0h（保留）	000h~01Fh：块 0~31	CTS

Lc（待写入的字节数量）

类型	字节 0	字节 1	字节 2
短	01h~FFh：写入 1~255 个字节	--	
长	00h	0001h~FFFFh：写入 1~65535 个字节	

响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	6X XXh	失败。

例如：

// 将 MIFARE Classic 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}

5.3.4. 透传（Pass Though）命令

5.3.4.1. 访问 ISO14443-3 标签

本节介绍如何使用透传命令结构发送符合 ISO14443-3 标准的命令，对固件的要求如下：

- ACR1581U-C FW 1.09.00 或更高版本

命令

命令	CLA	INS	P1	P2	Lc	命令数据域
ISO14443-3 命令	FFh	00h	00h	00h	数据长度	数据

响应

响应	响应数据域		
结果	数据	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	6X XXh	失败。

5.3.4.1.1. ISO14443-3 专有命令转换为透传命令

本节说明如何将 ISO14443 专有命令转换为透传格式。

例如：

Mifare Ultralight AES 获取版本号

60

透传格式

FF 00 00 00 01 60

5.3.4.2. 访问 ISO15693 标签

本节介绍如何使用透传命令结构发送符合 ISO15693 标准的命令，对固件的要求如下：

- ACR1581U-C FW 1.08.00 或更高版本

命令

命令	CLA	INS	P1	P2	Lc	命令数据域		Le
ISO15693 命令	FFh	FBh	00h	标志	数据长度+1	命令代码	数据	--/00h

其中：

标志： 00h（默认）：将标志位设置为 22h。在此情况下，数据字段无需包含 UID。

注：此选项不能与自定义（Custom）或专用（Proprietary）命令一起使用。

其他：将该值用作标志位的值

响应

响应	响应数据域		
结果	数据	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	6X XXh	失败。

5.3.4.2.1. ISO15693 命令转换为透传命令

本节介绍如何将 ISO15693 命令转换为透传格式。需要注意的是，透传命令模式仅支持 26 kbps 速率的命令。如需使用 53 kbps 的指令，请参考 *Error! Reference source not found.*。

例 1：

保持静默（强制命令）

22 02 XX XX XX XX XX XX XX XX

其中：

22 标志位

02 保持静默命令代码

XX XX XX XX XX XX XX XX 8 字节 UID

透传格式

FF FB 00 22 09 02 XX XX XX XX XX XX XX XX

或者

FF FB 00 00 01 02

例 2：

非寻址方式读取多个数据块（可选命令）

02 23 00 01

其中：

02　　标志位

23　　读多块命令代码

00　　第一个块的编号

01　　读取 2 个块

透传格式

FF FB 00 02 03 23 00 01

例 3：

在选定状态下获取随机数（NXP SLIX2 自定义命令）

12 B2 04

其中：

12　　标志位

B2　　取随机数命令代码

04　　NXP 生产商编码

透传格式

FF FB 00 12 02 B2 04

5.3.4.2.2.　支持的命令列表

下表列出了 ISO15693-3 标准规定的所有命令。标记为“N/A”的命令表示尚未进行测试。如需咨询这些命令的相关问题，请发送邮件至 info@acs.com.hk。请注意：ISO15693 卡可能不支持所有可选命令。具体支持情况请参阅对应卡片的技术规格书。

命令代码	类型	功能	支持
01h	强制	清点（Inventory）	是
02h	强制	保持静默（Stay Quiet）	是
20h	可选	读单块（Read Single Block）	是
21h	可选	写单块（Write Single Block）	是
22h	可选	锁数据块（Lock Block）	是
23h	可选	读多块（Read Multiple Blocks）	是
24h	可选	写多块（Write Multiple Blocks）	是
25h	可选	选择（Select）	是
26h	可选	重置为就绪（Reset to Ready）	是
27h	可选	写入 AFI（Write AFI）	是

命令代码	类型	功能	支持
28h	可选	锁定 AFI（Lock AFI）	是
29h	可选	写入 DSFID（Write DSFID）	是
2Ah	可选	锁定 DSFID（Lock DSFID）	是
2Bh	可选	获取系统信息（Get System Information）	是
2Ch	可选	获取多块安全状态（Get Multiple Block Security Status）	是
2Dh	可选	快速读多块（Fast Read Multiple Blocks）	否
30h	可选	扩展读单块（Extended Read Single Block）	是
31h	可选	扩展写单块（Extended Write Single Block）	是
32h	可选	扩展锁数据块（Extended Lock Block）	是
33h	可选	扩展读多块（Extended Read Multiple Blocks）	是
34h	可选	扩展写多块（Extended Write Multiple Blocks）	是
35h	可选	认证（Authenticate）	是
36h	可选	密钥更新（KeyUpdate）	N/A
37h	可选	认证通信加密格式指示符（AuthComm Crypto Format Indicator）	N/A
38h	可选	安全通信加密格式指示符（SecureComm Crypto Format Indicator）	N/A
39h	可选	挑战（Challenge）	否
3Ah	可选	读取缓冲区（ReadBuffer）	是
3Bh	可选	扩展获取系统信息（Extended Get System Information）	是
3Ch	可选	扩展获取多块安全状态（Extended Get Multiple Block Security Status）	是
3Dh	可选	快速扩展读多块（Fast Extended Read Multiple Blocks）	否
A0-DFh	自定义	IC 制造商相关数据（IC Mfg Dependent）	N/A
E0-FFh	专用	IC 制造商相关数据（IC Mfg Dependent）	N/A

5.3.4.3. 访问 FeliCa 标签

访问 FeliCa 标签的命令不同于访问符合 PCSC 和 MIFARE 标签的命令。此命令符合 FeliCa 规范，加了一个命令头。

FeliCa 命令结构

命令	CLA	INS	P1	P2	Lc	命令数据域
FeliCa 命令	FFh	00h	00h	00h	命令数据域的长度	FeliCa 命令（开始于长度字节）

FeliCa 的响应结构（数据 + 2 字节）

响应	响应数据域
结果	响应数据

以读取内存块为例：

1. 与 FeliCa 建立连接。

   ATR = 3B 8F 80 01 80 4F 0C A0 00 00 03 06 11 00 3B 00 00 00 00 42h

   其中，11 00 3Bh = FeliCa

2. 读取 FeliCa IDM。

   命令 = FF CA 00 00 00h

   响应 = [IDM (8 字节)] 90 00h

   例如：FeliCa IDM = 01 01 06 01 CB 09 57 03h

3. FeliCa 命令访问。

   例如：“读取”内存块。

   命令 = FF 00 00 00 10 10 06 01 01 06 01 CB 09 57 03 01 09 01 01 80 00h

   其中：

   Felica 命令 = 10 06 01 01 06 01 CB 09 57 03 01 09 01 01 80 00h

   IDM = 01 01 06 01 CB 09 57 03h

   响应 = 内存块数据

5.3.5. PCSC 2.0 第 3 部分支持的 APDU 指令（V2.02 及以上版本）

PCSC 2.0 第三部分规定的命令用于将数据从应用程序透明传递给非接触式标签，将接收到的数据透明返回给应用程序和协议，同时切换协议。

5.3.5.1. PCSC 2.0 第 3 部分流程

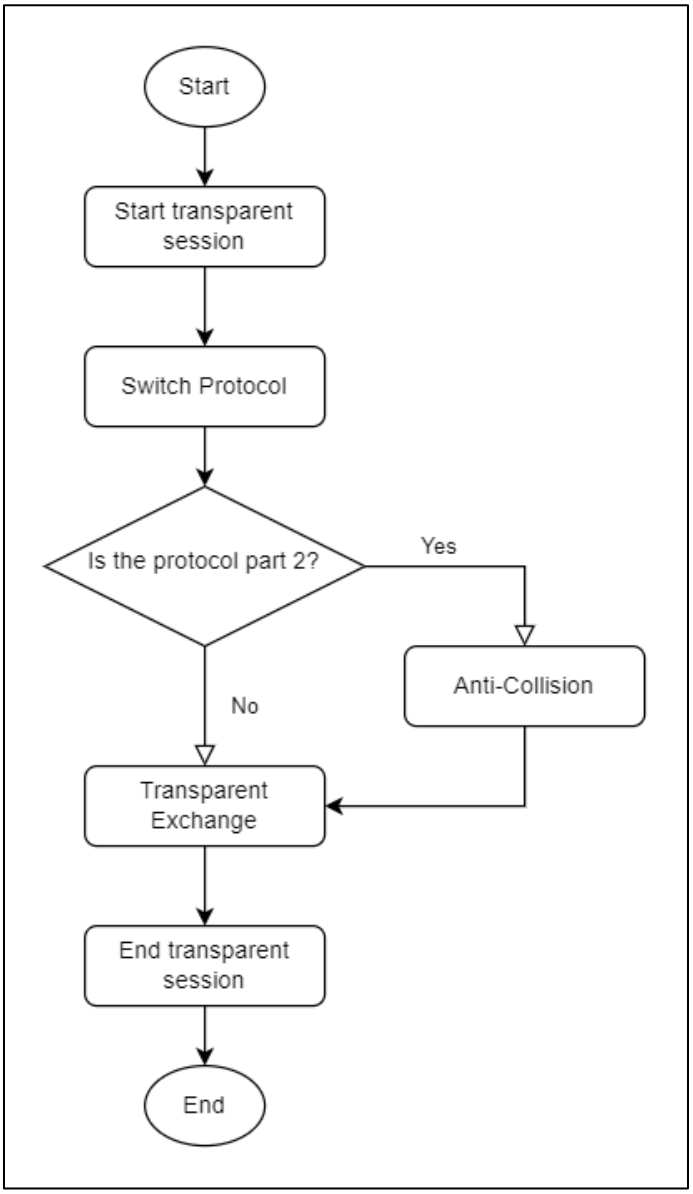

图 5：ACR1581U 透明会话流程图

5.3.5.2. 命令和响应的 APDU 格式

命令格式

CLA	INS	P1	P2	Lc	命令数据域
FFh	C2h	00h	功能	数据长度	数据[数据长度]

其中功能 (1 个字节）：

00h = 会话管理
01h = 透明交互
02h = 切换协议
其它 = RFU

响应格式

响应数据域	SW1	SW2
BER-TLV 编码的数据域		

每个命令都会返回 SW1 和 SW2 加上响应数据域（如有）。SW1 和 SW2 符合 ISO 7816 的规定。也应使用以下 C0 数据对象的 SW1 SW2。

C0 数据元格式

标签	长度（1 字节）	SW2
C0h	03h	错误状态

错误状态说明

错误状态	说明
XX SW1 SW2	XX = APDU 中错误数据对象的编号 00 = APDU 常见错误 01 = 第 1 个数据对象有错误 02 = 第 2 个数据对象有错误
00 90 00h	未发生错误
XX 62 82h	数据对象 XX 告警，请求信息不存在
XX 63 00h	未有信息
XX 63 01h	由于其它数据对象失败，停止执行
XX 6A 81h	不支持数据对象 XX
XX 67 00h	意外长度的数据对象 XX
XX 6A 80h	意外值的数据对象 XX
XX 64 00h	数据对象 XX 执行错误（IFD 无响应）
XX 64 01h	数据对象 XX 执行错误（ICC 无响应)
XX 6F 00h	数据对象 XX 失败，没有准确诊断

第一个字节的值表示错误数据对象 XX 的编号，最后两个字节是对错误的解释。允许使用 ISO 7816 规定的 SW1 SW2 值。

如果 C-APDU 数据域中存在多个数据对象，而且其中一个数据对象失败，那么在其它数据对象不依赖于失败的数据对象的情况下，IFD 可以处理接下来的数据对象。

5.3.5.3. 管理会话（Manage Session）[FF C2 00 00 …]

此命令允许用户开启会话并禁用轮询功能，以进行后续通信。通信完成后，用户应立即结束会话。

需要注意的是，如果不正确使用，此命令可能会使读写器无法检测到卡片是否存在，并且无法自动恢复，除非逻辑/物理断开读写器连接。

命令

命令	CLA	INS	P1	P2	Lc	命令数据域	Le
Manage Session	FFh	C2h	00h	00h	Cmd 数据长度	Cmd TLV	--/00h

响应状态码

响应数据	SW1 SW2	含义
--	90 00h	操作成功完成。
Rsp TLV	90 00h	Le = 0x00：Cmd TLV 之一失败。如需了解错误详情，请参考 Rsp TLV。
--	6X XXh	Le = --：Cmd TLV 之一失败。

Cmd TLV

Cmd	含义
Start Session: 81 00h	开始会话并禁用轮询。
RF Off: 83 00h	关闭 RF。
Timer: 5F 46 04h [TIME]	设置下一个 RF On/Off TLV 前的休眠时间 [TIME]: 4 字节值（MSB 在前），范围为 1000 到 100000 us。实际休眠时间将四舍五入到最接近的 1000us。
RF On: 84 00h	打开 RF。
End Session: 82 00h	结束会话，重新启用轮询。

Rsp TLV

Rsp	含义
TLV Error: C0 03 NN 6X XXh	第 NN 个命令 TLV 错误。

5.3.5.3.1. 开始会话数据对象（Start Session Data Object）

此命令用于开启透明会话。会话开始后，自动轮询功能被禁用，直到会话结束。

开始会话数据对象

标签	长度（1 字节）	值
81h	00h	-

5.3.5.3.2. 结束会话数据对象（End Session Data Object）

此命令用于终止透明会话。在新的会话开始之前，重置为自动轮询状态。

结束会话数据对象

标签	长度（1 字节）	值
82h	00h	-

5.3.5.3.3. 关闭 RF 数据对象（Turn Off the RF Data Object）

此命令用于关闭天线场。

关闭 RF 场数据对象

标签	长度（1 字节）	值
83h	00h	-

5.3.5.3.4. 开启 RF 数据对象（Turn On the RF Data Object）

此命令用于开启天线场。

开启 RF 场数据对象

标签	长度（1 字节）	值
84h	00h	-

5.3.5.3.5. 计时器数据对象（Timer Data Object）

此命令用于创建一个 32 位计时器数据对象，以 1 μs 为单位。

例如： 如果在关闭 RF 数据对象和开启 RF 数据对象之间有 5000 μs 的计时器数据对象，读写器会关闭 RF 场大约 5000μs，然后再重新开启 RF 场。

计时器数据对象

标签	长度（1 字节）	值
5F 46h	04h	计时器（4 个字节）

5.3.5.4. 透明交互（Transparent Exchange）[FF C2 00 01 …]

此命令允许用户向卡片发送/从卡片接收任意位或字节，并可以选择配置各种链路和传输层（例如 ISO14443 第 4 部分）以及一些链路层冗余（CRC 和奇偶校验）。用户可以将任何卡片特定的原始数据嵌入到这个私有 APDU 中，然后发送给卡片。

需要注意的是，此命令可能会干扰卡支持的内部处理过程，可能会在不通知驱动程序/固件的情况下更改卡片状态，并且可能需要重置和/或移除卡片才能使驱动程序/固件恢复正常。

命令

命令	CLA	INS	P1	P2	Lc	命令数据域	Le
Transparent Exchange	FFh	C2h	00h	01h	Cmd 数据长度	Cmd TLV	00h

响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	6X XXh	失败。

Cmd TLV

Cmd	含义
Transceive Flag: 90 02 [Flag] 00h	设置下列 Transceive TLV 的 Flag Flag[7:5]: RFU; 设为 0 Flag[4]: 设为禁用 ISO14443-4 Flag[3]: 设为禁止接收奇偶校验处理 Flag[2]: 设为禁止传输奇偶校验处理 Flag[1]: 设为禁止接收 CRC 处理 Flag[0]: 设为禁止传输 CRC 处理 如果此 TLV 缺失，则使用上一个命令中设置的 Flag 值。如果从未设置过 Flag 值，则使用当前协议值。
Transmit Bit Frame: 91 01h [NumBit]	设置下列 Transceive TLV 的 Bit Frame。如果此 TLV 缺失，则默认值为 0。 NumBit[7:3]: RFU; 设为 0 NumBit[2:0]: 最后一个字节中的有效位的数量（0 表示所有的位都有效）
Timer: 5F 46 04h [TIME]	设置下列 Transceive TLV 的超时时间。 [TIME]: 4 字节值（MSB 在前），范围为 1 us 到 1000000 us。实际超时时间将四舍五入到最接近的 302.07 x 20~15 us。 如果此 TLV 缺失，则使用先前设置的 FWTI 值作为超时时间。
Set FWTI:	设置 Transceive 的 FWT/超时。若先前未通过“FF C2h ...”命令设置 FWTI，则默认值为 0.

Cmd	含义
FF 6E 03 03 01h [FWTI]	FWTI: 0 ~ 15, FWT/超时 = 302.07 x 2FWTI us
Transceive: 95h [Size] [Data]	Size: BER-TLV 长度字段中编码数据的大小 Data: 待传输的数据

Rsp TLV

Rsp	含义
Receive Bit framing: 92 01h [NumBit]	NumBit[7:3]: RFU; 设为 0. NumBit[2:0]: 最后一个字节中的有效位的数量（0 表示所有的位都有效）
Response: 97h [Size] [Data]	Size: BER-TLV 长度字段中编码数据的大小 Data: 接收的数据。
Response Status: 96 02h [Status] 00h	Status [7:4]: RFU. Status[3]: 成帧错误。 Status[2]: 奇偶校验错误。 Status[1]: RFU. Status[0]: CRC 错误。

5.3.5.4.1. 发送和接收标志数据对象（Transmission and Reception Flag Data Object）

此命令用于为下列传输定义成帧参数和 RF 参数。

发送和接收标志数据对象

标签	长度（1 字节）	值		
		字节 0		字节 1
		位	说明	
90h	02h	0	0 – 在传输的数据后添加 CRC 1 – 不在传输的数据后添加 CRC	00h
		1	0 – 对接收数据进行 CRC 检查 1 – 不对接收数据进行 CRC 检查	
		2	0 – 在传输的数据中插入奇偶校验位 1 – 不插入奇偶校验位	
		3	0 – 期望接收的数据中含有奇偶校验位 1 – 不期望接收的数据中含有奇偶校验位（即不进行奇偶校验）	
		4	0 – 在传输数据中添加协议头，或者从响应中丢弃 1 – 不添加或者丢弃协议头（如有）（例如 PCB、CID、NAD）	
		5-7	RFU	

5.3.5.4.2. 发送位成帧数据对象（Transmission Bit Framing Data Object）

此命令用于定义待发送或待收发数据中最后一个字节的有效位数量。

发送位成帧数据对象

标签	长度（1 字节）	值	
		位	说明
91h	01h	0-2	最后一个字节中的有效位数量（0 表示所有的位都有效）
		3-7	RFU

发送位成帧数据对象只能和“发送”或“收发”数据对象一起使用。如果不存在此数据对象，则表明所有的位都有效。

5.3.5.4.3. 收发数据对象（Transceive Data Object）

此命令用于发送和接收来自 ICC 的数据。数据发送完成后，读写器会保持等待状态，直到计时器数据对象规定的时间结束。

如果没有在数据域中定义计时器数据对象，读写器会保持等待状态直到设置参数 FWTI 数据对象规定的时间段结束。如果没有设置 FWTI，读写器会等待大约 302 μs。

收发数据对象

标签	长度		值
95h	01-7Fh		数据（1~127 字节）
95h	81h	80h 或更多	数据（128~N 个字节）

5.3.5.4.4. 计时器数据对象（Timer Data Object）

此命令用于创建一个 32 位计时器数据对象，以 1 μs 为单位。

例如：如果有一个 5000 μs 的计时器数据对象，读写器会等待下列 Transceive TLV 大约 5000 μs 再超时。

计时器数据对象

标签	长度（1 字节）	值
5F 46h	04h	计时器（4 个字节）

5.3.5.4.5. 响应位成帧数据对象（Response Bit Framing Data Object）

此命令用于在响应中提示接收到的发送位成帧数据对象

标签	长度（1 字节）	值	
		位	说明
92h	01h	0-2	最后一个字节中的有效位数量（0 表示所有的位都有效）
		3-7	RFU

发送位成帧数据对象只能和“发送”或“收发”数据对象一起使用。如果不存在此数据对象，则表明所有的位都有效。

5.3.5.4.6. 响应状态数据对象（Response Status Data Object）

此命令用于在响应中提示接收到的数据状态

响应状态数据对象

标签	长度（1 字节）	值		
		字节 0		字节 1
		位	说明	
96h	02h	0	0 – CRC 正确，或未进行校验 1 – CRC 校验失败	RFU
		1	0 – 无冲突 1 – 检测到冲突	
		2	0 – 无奇偶校验位错误 1 – 检测到奇偶校验位错误	
		3	0 – 无成帧错误 1 – 检测到成帧错误	
		4 - 7	RFU	

5.3.5.4.7. 响应数据对象（Response Data Object）

此命令用于在响应中提示接收到的数据状态

响应数据对象

标签	长度			值
97h	01-7Fh			响应数据（1-127 字节）
	81h	80 - FFh		响应数据（128-255 字节）
	82h	-	-	响应数据（256-N 字节）

5.3.5.5. 切换协议（Switch Protocol）[FF C2 00 02 …]

此命令允许用户切换并指定协议，以及选择协议层和参数。

需要注意的是，此命令可能会干扰卡支持的内部处理过程，可能会在不通知驱动程序/固件的情况下更改卡片状态，并且可能需要重置和/或移除卡片才能使驱动程序/固件恢复正常。

命令

命令	CLA	INS	P1	P2	Lc	命令数据域	Le
Switch Protocol	FFh	C2h	00h	02h	Cmd 数据长度	Cmd TLV	00h

响应状态码

响应数据	SW1 SW2	含义
Rsp TLV	90 00h	数据成功。
--	90 00h	成功
--	6X XXh	失败。

Cmd TLV

Cmd	含义
Set Baud: FF 6E 03 05 01h [Baud]	设置要在协议切换过程中使用的第 4 部分/层的波特率。如果尚未通过“FF C2h ...”命令设置[Baud]，则默认值为 98h (106 kbps)。 ISO14443：98h (106 kbps)、99h (212 kbps)、9A (424 kbps)、9B (848 kbps)。 ISO15693：80h (26 kbps)、08h (53 kbps)
Switch Protocol: 8F 02h [RF] [Layer]	将协议切换到指定的 RF 和/或层。 [RF]: 00h: ISO14443A, 01h: ISO14443B 02h: ISO15693, 03h: FeliCa, FFh: 当前 RF 其它: RFU [Layer]: 02h: 第 2 层/部分 03h: 第 3 层/部分, 04h: 第 4 层/部分 (仅用于 A/B) 其它：RFU 注：如果切换到第 2 层/部分，则必须处于透明会话（禁用轮询）状态。

Rsp TLV

Rsp	含义
Response: 8Fh [Size] [Data]	Size: BER-TLV 长度字段中编码数据的大小 Data: ATR (如果是第 4 部分)、最终 SAK (如果是 A 类第 3 部分)、或者 ATQB 中的 PI (如果是 B 类第 3 部分)。

5.3.5.5.1. 切换协议数据对象（Switch Protocol Data Object）

此命令用于指定协议和不同标准层。

切换协议数据对象

标签	长度（1 字节）	值	
		字节 0	字节 1
8Fh	02h	00h – ISO/IEC14443 A 类 01h – ISO/IEC14443 B 类 02h – ISO15693 03h – FeliCa 其它 – RFU	02h – 切换到第二层 03h – 切换或激活到第三层 04h – 激活到第四层 其它 - RFU

5.3.5.5.2. 响应数据对象（Response Data Object）

此命令用于在响应中提示接收到的数据状态

响应数据对象

标签	长度（1 字节）	值
5F 51h	数据长度	ATR
8Fh	数据长度	最终 SAK (如果是 A 类第 3 部分)、或者 ATQB 中的 PI (如果是 B 类第 3 部分)。

5.3.5.6. PCSC 2.0 第 3 部分示例

8. 开始透明会话

   命令：**FF C2 00 00 02 81 00**

   响应：**C0 03 00 90 00 90 00**

9. 关闭天线场

   命令：**FF C2 00 00 02 83 00**
   响应：**C0 03 00 90 00 90 00**

10. 打开天线场

    命令：**FF C2 00 00 02 84 00**

    响应：**C0 03 00 90 00 90 00**

11. 激活 ISO 14443-4A

    命令： **FF C2 00 02 04 8F 02 00 04**

    响应： **C0 03 01 64 01 90 00** （如果不存在卡片）

    C0 03 00 90 00 5F 51 [Len] [ATR] 90 00

12. 将 PCB 设为 0Ah，并在传输数据中启用 CRC、奇偶校验和协议头。

    命令：**FF C2 00 01 0A 90 02 00 00 FF 6E 03 07 01 0A**

    响应：**C0 03 00 90 00 90 00**

13. 发送 APDU “80B2000008”至卡片并取响应。

    命令：**FF C2 00 01 0E 5F 46 04 40 42 0F 00 95 05 80 B2 00 00 08**

    响应：**C0 03 00 90 00 92 01 00 96 02 00 00 97 0C [卡片响应] 90 00**

14. 结束透明会话。

    命令：**FF C2 00 00 02 82 00**

    响应：**C0 03 00 90 00 90 00**

5.3.6. PICC 的专属私有（Pseudo）APDU

下列私有 APDU 用于间接访问非接触卡，是对 PCSC Pseudo APDU 的补充。这些 APDU 的内部处理过程与 PCSC Pseudo APDU 类似。

5.3.6.1. 写入值块（Write Value Block）[FF D7 …]

此命令用于将 4 个字节的值写入兼容 MIFARE 标准的卡的块。调用此命令前，用户应当先成功进行认证并获得对该块的访问权限。

命令

命令	CLA	INS	P1	P2	Lc	命令数据域
Write Value Block	FFh	D7h	00h	块号	05h	见下表

命令数据

字节 0	字节 1	字节 2	字节 3	字节 4
00h	4 个字节的值（MSB 在前）			

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

响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	6X XXh	失败。

5.3.6.2. 读取值块（Read Value Block） [FF B1 …]

此命令用于从兼容 MIFARE 标准的卡的有效值块中读取 4 个字节的值。调用此命令前，用户应当先成功进行认证并获得对该块的访问权限。

命令

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

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

响应

响应数据	SW1 SW2	含义
4 个字节的值（MSB 在前）	90 00h	数据成功。
--	6X XXh	失败。

5.3.6.3. 减少/增加值（Decrement/Increment Value）[FF D7 …]

此命令用于从源块减少/增加一个 4 字节的值，并将结果存入目标块（卡片需兼容 MIFARE 标准）。 如果要将结果存入同一个源块，则可以将目标块的编号设为 0 或者源块号。调用此命令前，用户应当先成功进行认证并获得对源块和目标块的访问权限。

命令

命令	CLA	INS	P1	P2	Lc	命令数据域
Decrement/Increment Value	FFh	D7h	目标块 #	源块#	05h	见下表

命令数据

字节 0	字节 1	字节 2	字节 3	字节 4
01h	4 个字节的增加值（MSB 在前）			
02h	4 个字节的减少值（MSB 在前）			

响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	6X XXh	失败。

5.3.6.4. 复制值块（Copy Value Block）[FF D7 …]

此命令用于将值从源块复制到目标块（卡片需兼容 MIFARE 标准）。调用此命令前，用户应当先成功进行认证并获得对源块和目标块的访问权限。

命令

命令	CLA	INS	P1	P2	Lc	命令数据域
Copy Value Block	FFh	D7h	00	源块#	02h	见下表

命令数据

字节 0	字节 1
03h	目标块#

响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	6X XXh	失败。

5.3.7. 访问符合 PCSC 的标签（ISO14443-4）

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。ACR1581U 读写器与符合 ISO 14443-4 标准的卡片进行通信时，只需要交互 ISO 7816-4 规定的 APDU 和响应。ACR1581U 会在内部处理 ISO 14443 第 1-4 部分协议。

另外 MIFARE Classic (1K/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。更多信息请参阅 **PICC 的 PCSC 私有（Pseudo）APDU（带专有扩展）**。

ISO 7816-4 规定的 APDU 报文结构

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 字节）

响应	响应数据域		
结果	响应数据	SW1	SW2

常见的 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。

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" 获取 ATS。*

例如：

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

5.3.8. 支持的 PICC ATR

默认支持下列 PICC 类型/技术。在读写器上刷卡后，PCSC API 中的 SCardStatus()会将下述 ATR 返回给 CCID 主机。

卡片类型/技术	ATR
MIFARE Std 1k[2]	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 01 00 00 00 00 6A
MIFARE Std 4k[2]	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 02 00 00 00 00 69
MIFARE UltraLight[2]	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 03 00 00 00 00 68
MIFARE Ultralight EV1[2]	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 3D 00 00 00 00 56
MIFARE Plus SL1 2k[2]	默认：与 MIFARE Std 1k 相同 备用：3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 36 00 00 00 00 5D
MIFARE Plus SL1 4k[2]	默认：与 MIFARE Std 4k 相同 备用：3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 37 00 00 00 00 5C
MIFARE Plus SL2 2k	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 38 00 00 00 00 53
MIFARE Plus SL2 4k	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 39 00 00 00 00 52
MIFARE Ultralight C[2]	默认：3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 3A 00 00 00 00 51 备用：与 MIFARE Ultralight 相同
SmartMX，模拟 MIFARE Std 1k[2]	默认：与 MIFARE Std 1k 相同 备用：与 ISO14443-4, Type A 相同
SmartMX，模拟 MIFARE Std 4k[2]	默认：与 MIFARE Std 4k 相同 备用：与 ISO14443-4, Type A 相同
ISO14443-4, Type A	3B 8n 80 01 T1 ..Tn Tck n = ATS 中历史字节的数量 T1 ..Tn = ATS 中的历史字节 Tck = 异或 8n 80 01 T1 ..Tn
ISO14443-4, Type B	3B 88 80 01 T1 ..T8 Tck T1 ..T4 = ATQB 中的应用数据 T5 ..T7 = ATQB 中的协议信息 T8 = ATA 中的 MBLI Tck = 异或 88 80 01 T1 ..T8
FeliCa	3B 8F 80 01 80 4F 0C A0 00 00 03 06 11 00 3B 00 00 00 00 42
ISO15693-3 Generic	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 00 00 00 00 00 63
Infineon My-D Vicinity (SRF55Vxxx)	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 0E 00 00 00 00 6D
ST LRI	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 13 00 00 00 00 70

[2] 关于备用 ATR 定义的配置和取消，请参阅**设置轮询/ATR 选项（Set Polling/ATR Option）[E0 00 00 23 01 …]**直接命令中的 Bit 3 和 Bit 7。

卡片类型/技术	ATR
NXP I-Code SLI	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 14 00 00 00 00 77
NXP I-Code SLIX/SLIX2	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 35 00 00 00 00 56
PicoPass 2K	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 17 00 00 00 00 79
PicoPass 2KS	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 18 00 00 00 00 76
PicoPass 16K	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 19 00 00 00 00 77
PicoPass 16KS	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1A 00 00 00 00 74
PicoPass 16K (8 x 2)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1B 00 00 00 00 75
PicoPass 16KS (8 x 2)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1C 00 00 00 00 72
PicoPass 32KS (16 + 16)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1D 00 00 00 00 73
PicoPass 32KS (16 + 8x2)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1E 00 00 00 00 70
PicoPass 32KS (8x2 + 16)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1F 00 00 00 00 71
PicoPass 32KS (8x2 + 8x2)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 20 00 00 00 00 4E

为了缩短常见应用的响应时间，默认禁用对下列 PICC 类型/技术的支持。用户可以通过直接命令“Set operation Mode”来启用对各种类型/技术的支持。如果相应类型/技术已启用并且在读写器上刷卡，PC_to_RDR_IccPowerOn 命令会将下列 ATR 返回给 CCID 主机。

卡片类型/技术	ATR
SRI (SRIX4K/SRT512)	3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 07 00 00 00 00 69
Topaz	3B 8F 80 01 80 4F 0C A0 00 00 03 06 02 00 30 00 00 00 00 5A
PicoPass 2K	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 17 00 00 00 00 75
PicoPass 2KS	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 18 00 00 00 00 7A
PicoPass 16K	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 19 00 00 00 00 7B
PicoPass 16KS	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 1A 00 00 00 00 78
PicoPass 16K (8 x 2)	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 1B 00 00 00 00 79
PicoPass 16KS (8 x 2)	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 1C 00 00 00 00 7E
PicoPass 32KS (16 + 16)	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 1D 00 00 00 00 7F
PicoPass 32KS (16 + 8x2)	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 1E 00 00 00 00 7C
PicoPass 32KS (8x2 + 16)	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 1F 00 00 00 00 7D
PicoPass 32KS (8x2 + 8x2)	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 20 00 00 00 00 42
Innovatron	3B 88 80 01 80 4F 05 F0 49 4E 4E 4F 35
CTS	3B 87 80 01 80 4F 04 F0 43 54 53 79

6.0. 直接（Escape）命令

下列命令用于配置 PCD/NFC，以及访问读写器的特殊功能。CCID 主机可以使用 PCSC API 中 SCARD_CTL_CODE(3500)的 SCardControl()向读写器发送这些命令。收到 Escape 命令后，读写器会解读命令并执行各项操作，然后生成响应并发送回 CCID 主机。

注：
这些命令需通过正确的接口发送。例如, E0 00 00 25 01 00 (6.1.1 节)应当通过PICC接口发送（6.0 节）。

6.1. PICC 的 Escape 命令

6.1.1. RF 控制（RF Control）[E0 00 00 25 01 …]

此命令用于设置 RF 控制。

命令

命令	CLA	INS	P1	P2	Lc	响应数据域
RF Control	E0h	00h	00h	25h	01h	RF 状态

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	RF 状态

RF 状态：1 个字节

RF 状态	说明
00h	RF 关闭
01h	RF 开启，轮询
02h	RF 开启，不轮询

默认设置 – 01h (RF 开启，轮询)

6.1.2. 获取 PCD/PICC 状态（Get PCD/PICC Status）[E0 00 00 25 00]

此命令用于获取 PCD/PICC 的状态。

命令

命令	CLA	INS	P1	P2	Le
Get PCD/PICC Status	E0h	00h	00h	25h	00h

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	Get PCD/PICC Status

PCD/PICC 状态：1 个字节

RF 状态	说明
00h	RF 关闭
01h	无 PICC

RF 状态	说明
02h	PICC 已就绪
03h	PICC 已选定/已激活
FFh	错误

6.1.3. 获取轮询/ATR 选项（Get Polling/ATR Option）[E0 00 00 23 00]

此命令用于设置/获取轮询选项，无需其它命令即可保存设置。此命令仅用于最初的读写器配置。

命令

命令	CLA	INS	P1	P2	Le
Get Polling/ATR Option	E0h	00h	00h	23h	00h

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	PICC 轮询/ATR 选项

6.1.4. 设置轮询/ATR 选项（Set Polling/ATR Option）[E0 00 00 23 01 …]

此命令用于设置轮询选项。

命令

命令	CLA	INS	P1	P2	Lc	响应数据域
Set Polling/ATR Option	E0h	00h	00h	23h	01h	PICC 轮询/ATR 选项

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	PICC 轮询/ATR 选项

PICC 轮询/ATR 选项 - 1 个字节

操作参数	参数	说明	选项
Bit 0	启用轮询	PICC轮询要检测的标签类型。	1 = 检测 0 = 跳过
Bit 1	启用 RF 关闭间隔		
Bit 2	RFU		
Bit 3	启用第 3 部分卡片 ATR 对额外 MIFARE 类型的识别	PICC 轮询要检测的标签类型。	1 = 检测 0 = 跳过
Bit 4 ~ 5	RF 关闭间隔		见下表
Bit 6	RFU		
Bit 7	启用第 4 部分 ATR 适用于 SmartMX/JCOS 卡模拟 MIFARE	PICC 轮询要检测的标签类型。	1 = 检测 0 = 跳过

RF 关闭间隔 – 2 Bit　　**情形 1:** 禁用 RF 关闭（Bit 1=0）

操作参数		USB 运行(D0)
Bit 5	Bit 4	
0	0	无 RF 关闭
0	1	
1	0	
1	1	

情形 2: 启用 RF 关闭 (Bit 1 = 1)

操作参数		USB 运行(D0)
Bit 5	Bit 4	
0	0	250 ms
0	1	500 ms

操作参数		USB 运行(D0)
Bit 5	Bit 4	
1	0	1000 ms
1	1	2500 ms

默认设置 – 8Bh (启用轮询，启用 RF 关闭间隔, 启用第 3 部分卡片在 ATR 中对额外 MIFARE 类型的识别, RF 关闭间隔[00], 启用第 4 部分 ATR 适用于 SmartMX/JCOS 卡模拟 MIFARE)

6.1.5. 获取 PICC 轮询类型（Get PICC Polling Type）[E0 00 01 20 00]

此命令用于获取允许的技术/轮询类型，无需其它命令即可保存设置。仅用于最初的读写器配置。

命令

命令	CLA	INS	P1	P2	Le
Get PICC Polling Type	E0h	00h	01h	20h	00h

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域	
						字节 1	字节 0
结果	E1h	00h	00h	00h	02h	PICC 轮询类型	

6.1.6. 设置 PICC 轮询类型（Set PICC Polling Type）[E0 00 01 20 02 …]

此命令用于设置 PICC 轮询类型。

命令	CLA	INS	P1	P2	Lc	响应数据域	
						字节 1	字节 0
Set PICC Polling Type	E0h	00h	01h	20h	02h	PICC 轮询类型	

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域	
						字节 1	字节 0
结果	E1h	00h	00h	00h	02h	PICC 轮询类型	

PICC 轮询类型 - 2 个字节，位掩码如下

字节	操作参数	参数	说明	选项
字节 1	Bit 0	ISO 14443A	PICC轮询要检测的标签类型。 RFU位应设置为0。	1 = 检测 0 = 跳过
	Bit 1	ISO 14443B		
	Bit 2	FeliCa		

字节	操作参数	参数	说明	选项
	Bit 3	RFU		
	Bit 4	Topaz		
	Bit 5	Innovatron		
	Bit 6	SRI/SRIX		
	Bit 7	RFU		
字节 0	Bit 0	Picopass (ISO14443B)		
	Bit 1	Picopass (ISO15693)		
	Bit 2	ISO15693		
	Bit 3	CTS		
	Bit 4-7	RFU		

默认设置 – 字节 1：07h (ISO14443A、ISO14443B、FeliCa)

字节 0：05h (Picopass (ISO14443B)、ISO15693)

例如：

命令：E0 00 01 20 02 07 05

响应：E1 00 00 00 02 07 05

轮询类型： 字节 1 = 07h = 0000 0111b = ISO14443A、ISO14443B、FeliCa

字节 0 = 05h = 0000 0101b = Picopass (ISO14443B)、ISO15693

6.1.7. 获取自动 PPS（Get Auto PPS）[E0 00 00 24 00]

每次识别出 PICC，读写器都会尝试按照最大连接速度的定义更改 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。

命令

命令	CLA	INS	P1	P2	Le
Get Auto PPS	E0h	00h	00h	24h	00h

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域	
结果	E1h	00h	00h	00h	02h	最高速率	当前速率

6.1.8. 设置自动 PPS（Set Auto PPS）[E0 00 00 24 01 …]

此命令用于设置自动 PPS。

命令

命令	CLA	INS	P1	P2	Lc	响应数据域
Set Auto PPS	E0h	00h	00h	24h	01h	最高速率

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域	
结果	E1h	00h	00h	00h	02h	最高速率	当前速率

PPS的速率

速率	说明
00h	106 kbps; 等同于没有设置自动PPS
01h	212 kbps
02h	424 kbps
03h	848 kbps

默认设置 – 02h (424 kbps)

注：
1.通常来说，应用程序应了解正在使用的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。读写器只是以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.如果较高的速率设置影响到读写器运行，请切换回较低的速率设置。

6.1.9. 读取 PICC 类型（Read PICC Type）[E0 00 00 35 00]

此命令用于读取 PICC 类型。

命令

命令	CLA	INS	P1	P2	Le
Get PICC Type	E0h	00h	00h	35h	00h

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域	
结果	E1h	00h	00h	00h	02h	类型	状态

类型: 1 个字节

类型	说明
CCh	无 PICC
04h	Topaz
10h	MIFARE
11h	FeliCa

类型	说明
20h	Type A, Part 4
23h	Type B, Part 4
25h	Innovatron
28h	SRIX
30h	PicoPass
FFh	其它

状态：1 个字节

状态	说明
00h	RF 关闭
01h	无 PICC
02h	PICC 已就绪
03h	PICC 已选定/已激活
FFh	错误

6.1.10. 获取 RF 功率设置（Get RF Power Setting）[E0 00 00 50 00]

此命令用于读取 RF 的功率设置，固件要求如下所述：

- ACR1581U-C FW 1.09.00 或更高版本

命令

命令	CLA	INS	P1	P2	Le
Get RF Power Setting	E0h	00h	00h	50h	00h

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	RF 功率

6.1.11. 设置 RF 功率（Set RF Power Setting）[E0 00 00 50 01 …]

此命令用于设置 RF 功率，固件要求与 Get RF Power Setting 相同。

注：执行 Set RF Power Setting 后，读写器将复位一次。

命令

命令	CLA	INS	P1	P2	Lc	响应数据域
Set RF Power Setting	E0h	00h	01h	50h	01h	RF 功率

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	RF 功率

百分比模式

RF 功率 - 1 个字节

参数	说明
00h	禁用手动RF功率设置
01h	20%
02h	40%
03h	60%
04h	80%
05h	100%

默认设置 – 00h

* 由于硬件限制，百分比模式的 RF 功率值可能无法生效。

6.1.12. 获取选择性挂起设置（Get Selective Suspend Setting）[E0 00 00 E5 00]

此命令用于读取选择性挂起设置，固件要求如下所述：

- ACR1581U-C FW 1.09.00 或更高版本

命令

命令	CLA	INS	P1	P2	Lc
Get Selective Suspend Setting	E0h	00h	00h	E5h	00h

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	选择性挂起

6.1.13. 设置选择性挂起设置（Set Selective Suspend Setting）[E0 00 00 E5 01 …]

此命令用于设置选择性挂起设置，对固件的要求与“获取选择性挂起设置”命令相同。

注：当读写器处于键盘模拟模式时，无法启用选择性挂起功能，反之亦然。

命令

命令	CLA	INS	P1	P2	Lc	响应数据域
Set Selective Suspend Setting	E0h	00h	00h	E5h	01h	选择性挂起

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	选择性挂起

选择性挂起 - 1 个字节

参数	说明
00h	停用
01h	启用

默认设置 – 00h

6.1.14. PICC – HID 键盘的 Escape 命令

6.1.14.1. 获取输出格式（Get Output Format）[E0 00 00 90 00]

此命令用于获取输出格式。

命令

命令	CLA	INS	P1	P2	Le
Get Output Format	E0h	00h	00h	90h	00h

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域	
结果	E1h	00h	00h	00h	02h	输出格式	输出顺序

6.1.14.2. 设置输出格式（Set Output Format）[E0 00 00 90 02 …]

此命令用于设置输出格式。

下述固件版本可支持 02h 和 04h 输出顺序：

- ACR1581U-C FW 1.09.00 或更高版本

命令

命令	CLA	INS	P1	P2	Lc	响应数据域	
Set Output Format	E0h	00h	00h	90h	02h	输出格式	输出顺序

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域	
结果	E1h	00h	00h	00h	02h	输出格式	输出顺序

输出格式：1 个字节

操作参数	参数	说明	选项
Bit 7 ~ 4	字母大小写	PICC轮询要检测的标签类型。	1 = 检测 0 = 跳过
Bit 3 ~ 0	显示模式		

输出顺序：1 个字节

状态	说明
00h	默认设置(UID字节0, UID字节1 … UID字节N) 例如: aa cc bb dd (原始/实际UID顺序)
01h	所有卡片类型逆序(UID字节N, UID字节N-1 … UID字节0) 例如: dd bb cc aa (原始/实际UID顺序反转)
02h	仅ISO 14443和Felica逆序 例如: dd bb cc aa (所选卡片类型的原始/实际UID顺序反转)
04h	仅ISO 15693逆序 例如: dd bb cc aa (所选卡片类型的原始/实际UID顺序反转)

字母大小写：高 4 位(Bit 7 到 Bit 4)

状态(从 bit 7 到 bit 4)	说明(无需关注 x bit)
1xxx	保留
00x0	小写字母
00x1	大写字母
000x	仅支持4字节UID
001x	支持4、7、8、10字节UID

显示模式：低 4 位(Bit 3 到 Bit 0)

状态(从 bit 7 到 bit 4)	说明(无需关注 x bit)
0h	Hex
1h	Dec (逐字节)
2h	Dec
3h	6H-6H
4h	8H-8H
5h	10H-10H
6h	14H-14H
7h	20H-20H
8h	6H-8D
9h	6H-10D
Ah	8H-10D
Bh	10H-14D
Ch	2H4H-8D
Dh	14H-17D

默认设置 – 输出格式：30h

输出顺序：00h

6.1.14.3. 获取 UID 起始、中间和结束位字符（Get Character at Start, Between, at End UID）[E0 00 00 91 00]

此命令用于获取 UID 起始、中间和结束位置的字符。

命令

命令	CLA	INS	P1	P2	Le
Get Character of UID	E0h	00h	00h	91h	00h

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域		
结果	E1h	00h	00h	00h	03h	中间	结束	开始

6.1.14.4. 设置 UID 起始、中间和结束位字符（Set Character at Start, Between, at End UID）[E0 00 00 91 03 …]

此命令用于设置 UID 起始、中间和结束位置的字符。

命令

命令	CLA	INS	P1	P2	Lc	响应数据域		
Set Character of UID	E0h	00h	00h	91h	03h	中间	结束	开始

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域		
结果	E1h	00h	00h	00h	03h	中间	结束	开始

中间: 1 个字节（每个 UID 之间的字符）

状态	说明
FFh	中间没有字符
其它	请参阅通用串行总线（USB）HID使用表

结束: 1 个字节（输出末尾的字符）

状态	说明
FFh	中间没有字符
其它	请参阅通用串行总线（USB）HID使用表

起始: 1 个字节（输出开始的字符）

状态	说明
FFh	中间没有字符
其它	请参阅通用串行总线（USB）HID使用表

注：

1. *AZERTY键盘布局仅支持“;” “,” “,” “,” “-“ 作为中间的字符，不支持零(0)和退格键。*

6.1.14.5. 获取键盘布局语言（Get Keyboard Layout Language）[E0 00 00 92 00]

此命令用于获取键盘布局语言。

命令

命令	CLA	INS	P1	P2	Le
Get Keyboard Layout Language	E0h	00h	00h	92h	00h

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	键盘布局语言

6.1.14.6. 设置键盘布局语言（Set Keyboard Layout Language）[E0 00 00 92 01 …]

此命令用于设置键盘布局语言。

命令

命令	CLA	INS	P1	P2	Lc	响应数据域
Set Keyboard Layout Language	E0h	00h	00h	92h	01h	键盘布局语言

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	键盘布局语言

键盘布局语言：1 个字节

状态	说明
00h	英语
01h	法语
02h	保留
03h	立陶宛语

默认设置 – 00h（英语）

6.1.14.7. 获取主机接口（Get Host Interface） [E0 00 00 93 00]

此命令用于获取主机接口。

命令

命令	CLA	INS	P1	P2	Le
Get Host Interface	E0h	00h	00h	93h	00h

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	主机接口

6.1.14.8. 设置主机接口（Set Host Interface）[E0 00 00 93 01 …]

此命令用于设置主机接口。

命令

命令	CLA	INS	P1	P2	Lc	响应数据域
Set Host Interface	E0h	00h	00h	93h	01h	主机接口

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	主机接口

主机接口：1 个字节

状态	说明
00h	仅HID键盘
01h	仅CCID读写器
02h	HID键盘 + CCID读写器

默认设置 – 01h（仅限 CCID 读写器）

6.1.14.9. 获取屏幕键盘设置（Get On-Screen Keyboard Setting）[E0 00 00 94 00]

此命令用于获取屏幕键盘的启用状态，固件要求如下：

- ACR1581U-C　FW 1.09.00 或更高版本

命令

命令	CLA	INS	P1	P2	Lc
Get On-Screen Keyboard Setting	E0h	00h	00h	94h	00h

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	屏幕键盘设置

6.1.14.10. 设置屏幕键盘设置（Set On-Screen Keyboard Setting）[E0 00 00 94 01 …]

此命令用于设置屏幕键盘的启用状态，固件要求与 Get On-Screen Keyboard Setting 命令相同。

命令

命令	CLA	INS	P1	P2	Lc	响应数据域
Set On-Screen Keyboard Setting	E0h	00h	00h	94h	01h	屏幕键盘设置

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	屏幕键盘设置

屏幕键盘设置：1 个字节

状态	说明
00h	禁用键盘弹出
01h	启用键盘弹出

默认设置 – 00h (禁用键盘弹出)

6.1.15. PICC – 卡模拟的 Escape 命令

6.1.15.1. 进入卡模拟模式（Enter Card Emulation Mode）[E0 00 00 40 03 …]

此命令用于设置读写器进入卡模拟模式，以便模拟 NFC 论坛类型 2 标签或者 FeliCa 卡。

***注：**模拟 NFC 论坛类型 2 标签时不支持 Lock 字节。UID 可由用户编写。*

命令

命令	CLA	INS	P1	P2	Lc	响应数据域		
Enter Card Emulation Mode	E0h	00h	00h	40h	03h	NFC 模式	00h	00h

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	03h	NFC 模式

NFC 设备模式：3 个字节

状态	说明
02h	NFC论坛类型2标签模式
03h	FeliCa
其它	读写器模式

***注意：**在切换到不同的卡模拟模式之前，请先进入读写器模式。卡模拟模式初始完成后将显示响应。*

字节号	0	1	2	3	USB 访问字节地址
序列号	SN0	SN1	SN2	BCC0	Nil
保留	保留	保留	保留	保留	Nil
内部/锁	保留	内部	Lock0	Lock1	Nil
数据读/写	Data0	Data1	Data2	Data3	0-3
数据读/写	Data4	Data5	Data6	Data7	4-7
数据读/写	Data8	Data9	Data10	Data11	8-11
数据读/写	Data12	Data13	Data14	Data15	12-15
数据读/写	Data16	Data17	Data18	Data19	16-19
数据读/写	Data20	Data21	Data22	Data23	20-23
数据读/写	Data24	Data25	Data26	Data27	24-27
数据读/写	Data28	Data29	Data30	Data31	28-31
数据读/写	Data32	Data33	Data34	Data35	32-35
数据读/写	Data36	Data37	Data38	Data39	36-39
数据读/写	Data40	Data41	Data42	Data43	40-43
数据读/写	Data44	Data45	Data46	Data47	44-47
数据读/写	Data48	Data49	Data50	Data51	48-51
数据读/写	Data52	Data53	Data54	Data55	52-55
数据读/写		…			…
数据读/写	Data1984	Data1985	Data1986	Data1987	1984~1987

可访问区（1988 字节）

表 7：NFC 论坛类型 2 标签的内存结构（2000 字节）

内存	1 数据块（16 字节）	USB 访问字节地址
数据读/写	Block 0	0-15
数据读/写	Block 1	16-31
数据读/写	Block 2	32-47
数据读/写	Block 3	48-63
数据读/写	Block 4	64-79
数据读/写	Block 5	80-95
数据读/写	Block 6	96-111
数据读/写	Block 7	112-127
数据读/写	Block 8	128-143
数据读/写	Block 9	144-159

表 8：FeliCa 卡的内存结构（160 字节）

其中：

默认： 块 0 数据：{10h, 01h, 01h, 00h, 09h, 00h, 00h, 00h, 00h, 00h, 01h, 00h, 00h, 00h, 00h, 1Ch}

默认块 0 数据 NFC 类型 3 标签属性信息块

注：

1. *FeliCa 卡模拟支持不带加密读/写。*
2. *FeliCa 卡片识别号码（IDm）可由用户定义，而生厂商编码固定为(03 88)。*

6.1.15.2. 读取卡模拟数据（Read Card Emulation Data）（NFC 论坛类型 2 标签）[E0 00 00 60 04 …]

此命令用于读取所模拟卡片的内容。

命令

命令	CLA	INS	P1	P2	Lc	命令数据域			
Read Card Emulation Data	E0h	00h	00h	60h	04h	00h	NFC 模式	起始偏移量	长度

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	长度	数据

起始偏移量： 1 字节 – 表 7 中从 Data0 开始的地址

长度： 1 字节 – 字节数量

6.1.15.3. 写入卡模拟数据（Write Card Emulation Data）（NFC 论坛类型 2 标签）[E0 00 00 60 …]

此命令用于写入模拟的卡片内容。

命令

命令	CLA	INS	P1	P2	Lc	命令数据域				
Write Card Emulation Data	E0h	00h	00h	60h	长度 + 04h	01h	NFC 模式	起始偏移量	长度	数据

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域		
结果	E1h	00h	00h	00h	03h	长度	90h	00h

NFC 设备模式：1 个字节

状态	说明
02h	NFC论坛类型2标签模式
03h	FeliCa
其它	读写器模式

起始偏移量： 1 字节 – 表 7 中从 Data0 开始的地址

长度： 1 字节 – 字节数量

6.1.15.4. 读取卡模拟数据（Read Card Emulation Data）（NFC 论坛类型 2 标签）（扩展）

此命令用于读取所模拟卡片的内容。

命令

命令	CLA	INS	P1	P2	Lc		命令数据域			
Read Card Emulation Data	E0h	00h	01h	60h	05h	00h	NFC 模式	起始偏移量 Bit[15:8]	起始偏移量 Bit[7:0]	长度

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	长度	数据

起始偏移量： 2 字节 – **表 7** 中从 SN0 起的开始读取地址

长度： 1 字节 – 待读取的字节数

6.1.15.5. 写入卡模拟数据（Write Card Emulation Data）（NFC 论坛类型 2 标签）（扩展）

此命令用于写入模拟的卡片内容。

命令

命令	CLA	INS	P1	P2	Lc		命令数据域				
Write Card Emulation Data	E0h	00h	01h	60h	长度 + 05h	01h	NFC 模式	起始偏移量 Bit[15:8]	起始偏移量 Bit[7:0]	长度	数据

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域		
结果	E1h	00h	00h	00h	03h	长度	90h	00h

NFC 设备模式：1 个字节

状态	说明
02h	NFC论坛类型2标签模式
其它	读写器模式

起始偏移量： 2 字节 – **表 7** 中从 SN0 起的开始写入地址

长度： 1 个字节 – 要写入的字节数

6.1.15.6. 设置 NFC 论坛类型 2 标签卡模拟 ID（Set Card Emulation of NFC Forum Type 2 Tag ID）[E0 00 00 61 03 …]

此命令用于设置所模拟的 NFC 论坛类型 2 标签的 UID。

命令

命令	CLA	INS	P1	P2	Lc	命令数据域
Set Card Emulation Lock Data	E0h	00h	00h	61h	03h	3 字节 UID

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域	
结果	E1h	00h	00h	00h	02h	90h	00h

6.1.15.7. 设置 NFC 卡模拟锁定数据（Set Card Emulation Lock Data in NFC）[E0 00 00 65 01 …]

此命令用于设置 NFC 通信过程中卡片模拟数据的锁定。数据锁定后，不能再通过 NFC 进行重写。

命令

命令	CLA	INS	P1	P2	Lc	命令数据域
Set Card Emulation Lock Data	E0h	00h	00h	65h	01h	锁

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	锁

锁：1 个字节 – 保护数据不能通过 NFC 通信进行重写

操作参数	参数	说明	选项
Bit 7 ~ 2	保留	保留	
Bit 1	启用 FeliCa 锁	数据不能通过 NFC 通信进行修改。数据仍然可以使用 USB 直接命令进行修改。	0：禁用锁 1：启用锁
Bit 0	启用 NFC 论坛类型 2 标签		

6.1.15.8. 设置卡模拟时 FeliCa 的 IDm（Set Card Emulation FeliCa IDm）[E0 00 00 64 06 …]

此命令用于在所模拟的 FeliCa 卡片上设置 6 字节 FeliCa 卡标识号。

命令

命令	CLA	INS	P1	P2	Lc	命令数据域
Set Card Emulation FeliCa IDm	E0h	00h	00h	64h	06h	IDm

响应状态码

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	06h	IDm

其中：

IDm 6 字节

6.1.15.9. 获取卡模拟状态（Get Card Emulation Status）[E0 00 00 69 00]

此命令用于获取 NFC 通信中卡片模拟数据的状态。

命令

命令	CLA	INS	P1	P2	Lc
Get Card Emulation Status	E0h	00h	00h	69h	00h

响应状态码

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

状态：1 个字节

操作参数	模式	说明
Bit 7 ~ 6	保留	保留
Bit 5	模拟卡已激活	1 = 已激活
Bit 4	模拟卡已移出	1 = 卡片已移出
Bit 3	模拟卡已全部读取	1 = 所有数据均已读取
Bit 2	模拟卡已读取	1 = 数据已读取
Bit 1	模拟卡已写入	1 = 数据已写入
Bit 0	模拟卡已被检测到	1 = 卡片检测中

6.1.15.10. 模拟 NFC 论坛类型 2 标签模式的示例命令集

此命令集以 ACR1581U 模拟 NFC 论坛类型 2 标签模式，触发 ACS 网站 https://www.acs.com.hk。步骤如下：

1. 通过下面的命令进入卡模拟模式：
   - 发送进入卡模拟模式（Enter Card Emulation Mode）

     E0 00 00 40 03 02 00 00
2. 通过下面的命令写 NDEF 数据：
   - 发送写入卡模拟数据（Write Card Emulation Data）（NFC 论坛类型 2 标签）

     E0 00 00 60 1A 01 02 00 18 E1 10 F4 00 03 0F D1 01 0B 55 02 61 63 73 2E 63 6F 6D 2E 68 6B FE

该命令集将触发一个示例长 URL 网站

https://www.example.com/this/is/a/very/long/url/that/keeps/going/on/and/on/with/even/more/segments/added/to/make/sure/it/exceeds/the/typical/length/limit/of/260/bytes/which/is/surprisingly/easy/to/do/if/you/keep/adding/more/and/more/segments/like/this/one/and/even/more

使用 ACR1581U 模拟 NFC 论坛类型 2 标签模式。步骤如下：

1. 通过下面的命令进入卡模拟模式：
   - 发送进入卡模拟模式（Enter Card Emulation Mode）命令

     E0 00 00 40 03 02 00 00

2. 通过下面的命令写 NDEF 数据：

   - 发送写入卡模拟数据（Write Card Emulation Data）（NFC 论坛类型 2 标签）命令。由于 NDEF 消息的长度超过 256 字节，需要拆分成两部分发送给 NFC 论坛类型 2 标签。

     E0 00 00 60 AC 01 02 00 A8 E1 10 F4 00 03 FF 01 09 C1 01 00 00 01 02 55 02 65 78 61 6D 70 6C 65 2E 63 6F 6D 2F 74 68 69 73 2F 69 73 2F 61 2F 76 65 72 79 2F 6C 6F 6E 67 2F 75 72 6C 2F 74 68 61 74 2F 6B 65 65 70 73 2F 67 6F 69 6E 67 2F 6F 6E 2F 61 6E 64 2F 6F 6E 2F 77 69 74 68 2F 65 76 65 6E 2F 6D 6F 72 65 2F 73 65 67 6D 65 6E 74 73 2F 61 64 64 65 64 2F 74 6F 2F 6D 61 6B 65 2F 73 75 72 65 2F 69 74 2F 65 78 63 65 65 64 73 2F 74 68 65 2F 74 79 70 69 63 61 6C 2F 6C 65 6E 67 74 68 2F 6C 69 6D 69 74 2F 6F 66 2F 32 36 30 2F 62 79 74

     E0 00 00 60 6E 01 02 A8 6A 65 73 2F 77 68 69 63 68 2F 69 73 2F 73 75 72 70 72 69 73 69 6E 67 6C 79 2F 65 61 73 79 2F 74 6F 2F 64 6F 2F 69 66 2F 79 6F 75 2F 6B 65 65 70 2F 61 64 64 69 6E 67 2F 6D 6F 72 65 2F 61 6E 64 2F 6D 6F 72 65 2F 73 65 67 6D 65 6E 74 73 2F 6C 69 6B 65 2F 74 68 69 73 2F 6F 6E 65 2F 61 6E 64 2F 65 76 65 6E 2F 6D 6F 72 65 FE

注：
如需了解更多关于NDEF（NFC 数据交互格式）的信息和规定，建议参考NDEF 规范；该规范就NDEF 记录的结构和使用提供了全面指引和详细信息，且这些 NDEF 记录常用于 NFC 数据交互。NDEF 规范有助于深入了解如何在 ACR1581U 设备环境中解读和利用 NDEF 命令和数据。

6.1.16. PICC – 发现模式的 Escape 命令

6.1.16.1. 进入发现模式（Enter Discovery Mode）[E0 00 00 6A 01 …]

此命令用于进入发现模式。

命令

命令	CLA	INS	P1	P2	Lc	响应数据域
Enter Discovery Mode	E0h	00h	00h	6Ah	01h	发现模式

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	发现模式

发现模式：1 个字节

状态	说明
00h	卡片读写器模式
02h	NFC论坛类型2标签模式
03h	FeliCa

6.2. ICC 的 Escape 命令

6.2.1. 获取专用模式（Get Exclusive Mode）[E0 00 00 2B 00]

此命令用于获取读写器的专用模式设置。

命令

命令	CLA	INS	P1	P2	Le
Get Exclusive Mode	E0h	00h	00h	2Bh	00h

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	专用模式

6.2.2. 设置专用模式（Set Exclusive Mode） [E0 00 00 2B 01 …]

此命令用于配置读写器进入/退出专用模式。

命令

命令	CLA	INS	P1	P2	Lc	响应数据域
Set Exclusive Mode	E0h	00h	00h	2Bh	01h	专用模式

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	专用模式

专用模式 (1 个字节)

专用模式	说明
00h	共享模式：ICC 和 PICC 界面可以同时工作。
01h	专用模式：插入 ICC 卡时自动轮询功能和天线关闭，PICC 停用（默认）
其它	RFU

默认设置 – 01h (专用模式)

6.2.3. 获取卡片电源配置（Get Card Power Config）[E0 00 00 0B 00]

此命令用于获取 ICC 卡的电源配置，仅用于最初的读写器配置。

命令

命令	CLA	INS	P1	P2	Le
Get Card Power Config	E0h	00h	00h	0Bh	00h

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	卡片电源配置

6.2.4. 设置卡片电源配置（Set Card Power Config）[E0 00 00 0B 01 …]

此命令用于设置和保存 ICC 卡的电源配置。仅用于最初的读写器配置。

命令

命令	CLA	INS	P1	P2	Lc	响应数据域
Set Card Power Config	E0h	00h	00h	0Bh	01h	配置

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	卡片电源配置

卡片电源配置（1 个字节）

卡片电源配置	说明
00h	自动检测，1.8V -> 3V -> 5V
01h	仅 5V
02h	仅 3V
03h	仅 1.8V
04h	自动检测，5V -> 3V -> 1.8V
其它	RFU

默认设置 – 04h (自动检测，5V -> 3V -> 1.8V)

6.3. 外设控制及其他的 Escape 命令

6.3.1. 获取固件版本（Get Firmware Version）[E0 00 00 18 …]

此命令用于获取读写器的固件信息。

命令

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

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	固件版本的长度	固件版本

例如：
命令： E0 00 00 18 00
响应状态码： E1 00 00 00 12 41 43 52 31 35 38 31 20 46 57 20 31 2E 30 30
十六进制固件版本： 41 43 52 31 35 38 31 20 46 57 20 31 2E 30 30
ASCII 固件版本：ACR1581 FW 1.00

6.3.2. 获取序列号（Get Serial Number）[E0 00 00 33 00]

此命令用于获取序列号。

命令

命令	CLA	INS	P1	P2	Le
Get Serial Number	E0h	00h	00h	33h	00h

响应状态码

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	序列号长度	序列号

6.3.3. 设置 USB 描述符中的 S/N（Set S/N in USB Descriptor）[E0 00 00 F0]

此命令用于设置 USB 描述符中的 S/N。

命令

命令	CLA	INS	P1	P2	Le	命令数据域	
Set S/N in USB Descriptor	E0h	00h	00h	F0h	02h	00h	启用 USB 描述符中的 SN

响应状态码

响应	CLA	INS	P1	P2	Le	响应数据域		
结果	E1h	00h	00h	00h	03h	启用 USB 描述符中的 SN	90h	00h

启用 USB 描述符中的 SN（1 字节）

启用 USB 描述符中的 SN	说明
00h	禁用 USB 描述符中的 SN
01h	启用 USB 描述符中的 SN

FW1.09.00 及以下版本的默认设置 – 00h

FW1.09.01 及以上版本的默认设置 – 01h

6.3.4. 设置蜂鸣器控制-单次（Set Buzzer Control - Single Time）[E0 00 00 28 01 …]

此命令用于设置单次蜂鸣器。

命令

命令	CLA	INS	P1	P2	Lc	响应数据域
Buzzer Control	E0h	00h	00h	28h	01h	蜂鸣器状态

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	蜂鸣器状态

蜂鸣器状态（1 个字节）

蜂鸣器状态	说明
00h	关闭
01 ~ FFh	开启，持续时间以 10ms 为单位

6.3.5. 设置蜂鸣器控制-重复（Set Buzzer Control - Repeatable）[E0 00 00 28 03 …]

此命令用于设置蜂鸣器的周期

命令

命令	CLA	INS	P1	P2	Lc	响应数据域
Buzzer Control	E0h	00h	00h	28h	03h	蜂鸣器状态

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	03h	蜂鸣器状态

蜂鸣器状态（3 个字节）

操作参数	蜂鸣器状态	说明
参数 1 – 字节 0	开启时间段	01 ~ FF: 开启的持续时间，以 10ms 为单位
参数 2 – 字节 1	关闭时间段	01 ~ FF: 关闭的持续时间，以 10ms 为单位
参数 3 – 字节 2	重复时间	01 ~ FF: 重复的次数

6.3.6. 获取 LED 状态（Get LED Status）[E0 00 00 29 00]

此命令用于获取当前 LED 的状态。

命令

命令	CLA	INS	P1	P2	Le
Get LED Status	E0h	00h	00h	29h	00h

响应状态码

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

6.3.7. 设置 LED 控制（Set LED Control）[E0 00 00 29 01 …]

此命令用于设置 LED 控制

命令

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

响应状态码

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

LED 状态（1 字节）

LED 状态	说明
Bit 0：蓝色 LED	1 = 开；0 = 关
Bit 1：绿色 LED	1 = 开；0 = 关
Bit 2-7：RFU	其它

6.3.8. 获取 UI 操作（Get UI Behaviour）[E0 00 00 21 00]

此命令用于获取 PCD UI 的操作，无需其它命令即可保存设置。仅用于最初的读写器配置。

命令

命令	CLA	INS	P1	P2	Le
Get PICC UI Behaviour	E0h	00h	00h	21h	00h

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	PICC UI 操作

6.3.9. 设置 UI 操作（Set UI Behaviour）[E0 00 00 21 01 …]

此命令用于设置 PICC UI 的操作。

UI 操作 bit 1、bit 2 和 bit 5 在下列固件版本中可用：

- ACR1581U-C　FW 1.09.00 或更高版本

命令

命令	CLA	INS	P1	P2	Lc	响应数据域
Set PICC UI Behaviour	E0h	00h	00h	21h	01h	PICC UI 操作

响应状态码

响应	CLA	INS	P1	P2	Le	命令数据域
结果	E1h	00h	00h	00h	01h	PICC UI 操作

UI 操作 - 1 个字节，位掩码如下

操作参数	参数	说明	选项
Bit 0	读写中（LED 快速闪烁）	读写器的UI操作	1 = 启用 0 = 停用
Bit 1	PICC轮询状态LED		
Bit 2	PICC激活状态LED		
Bit 3	卡片进入天线区域事件（蜂鸣器短暂鸣响）		
Bit 4	卡片移出天线区域事件（蜂鸣器短暂鸣响）		
Bit 5	上电事件（蜂鸣器短暂鸣响）		

PICC 默认设置 – 2Fh

注：

1. *获取/设置UI操作命令不适用于SAM接口。*

附录 A. NDEF 消息

本节介绍如何使用 NDEF 消息将 URL 编码到 Ntag 上。

如需了解数据结构，请参考“NFC Forum NFC Data Exchange Format (NDEF) Specifications 1.0”规范。

例如：

NDEF 消息 = { D1 01 0B 55 02 61 63 73 2E 63 6F 6D 2E 68 6Bh}

偏移	内容	长度	说明
0	D1	1	NDEF 数据头。TNF = 01h, SR=1, MB=1, ME=1
1	01	1	记录名长度（1 字节）
2	0B	1	URI 数据包的长度（11 字节）
3	55 (“U”)	1	记录类型：“U”
4	02	1	缩写：“https://www.”
5	61 63 73 2E 63 6F 6D 2E 68 6B	10	URL 本身。“acs.com.hk”

编码到 Ntag = {03 0F D1 01 0B 55 01 61 63 73 2E 63 6F 6D 2E 68 6B FEh}

偏移	内容	长度	说明
0	03	1	TLV 数据头。03h = NDEF 消息
1	0F	1	NDEF 消息的长度（15 字节）
2	D1 01 0B 55 01 61 63 73 2E 63 6F 6D 2E 68 6B	15	NDEF 消息
17	FE	1	TLV 数据头。 FEh = 记录结束

附录 B. ACR1281U 兼容性

本节介绍兼容 ACR1281U 的各种命令，固件要求 1.08.01 或者以上版本。

附录 B.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。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）

附录 B.2. MIFARE Classic (1K/4K)卡认证（Authentication for MIFARE Classic (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

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。

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需要再次进行验证。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。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）

附录 B.3. 手动 PICC 轮询（Manual PICC Polling）

此命令用于检测读写器感应范围内是否存在 PICC。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

附录 B.4. 读取 PICC 操作参数（Read PICC Operating Parameter）

此命令用于查看当前的 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 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

附录 B.5. 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 PICC Operating Parameter	E0h	00h	00h	20h	01h	操作参数

操作参数 (1 个字节)

操作参数	参数	说明	选项
Bit 0	ISO 14443 A 类	PICC 轮询需检测的标签类别	1 = 检测 0 = 跳过
Bit 1	ISO 14443 B 类		1 = 检测 0 = 跳过
Bit 2 – 7	RFU	RFU	RFU

注意：*操作的默认值 = 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

附录 B.6. 初始化卡片插入计数器（Initialize Cards Insertion Counter）

此命令用于初始化卡片插入/检测计数器。

Initialize Card Insertion Counter 的命令结构（9 字节）

命令	CLA	INS	P1	P2	Lc	命令数据域			
Initialize Card Insertion Counter	E0h	00h	00h	09h	04h	ICC Cnt (LSB)	ICC Cnt (MSB)	PICC Cnt (LSB)	PICC Cnt (MSB)

其中：

ICC Cnt (LSB) (1 字节)
ICC 插入计数器（LSB)

ICC Cnt (MSB) (1 字节)

ICC 插入计数器 (MSB)

PICC Cnt (LSB) (1 字节)

PICC 插入计数器 (LSB)

PICC Cnt (MSB) (1 字节)

PICC 插入计数器 (MSB)

Initialize Card Insertion Counter 的响应结构（5 字节）

响应	CLA	INS	P1	P2	Le
结果	E1h	00h	00h	00h	00h

附录 B.7. 读取卡片插入计数器（Read Cards Insertion Counter）

此命令用于查看卡片插入/检测计数器的值。

Read Card Insertion Counter 的命令结构（5 字节）

命令	CLA	INS	P1	P2	Lc
Read Card Insertion Counter	E0h	00h	00h	09h	00h

Read Card Insertion Counter 的响应结构（9 字节）

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

其中：

ICC Cnt (LSB) (1 字节)

ICC 插入计数器（LSB)

ICC Cnt (MSB) (1 字节)

ICC 插入计数器 (MSB)

PICC Cnt (LSB) (1 字节)

PICC 插入计数器 (LSB)

PICC Cnt (MSB) (1 字节)

PICC 插入计数器 (MSB)

附录 B.8. 更新卡片插入计数器（Update Card Insertion Counter）

此命令用于更新卡片插入/检测计数器的值。

Update Card Insertion Counter 的命令格式（5 个字节）

命令	CLA	INS	P1	P2	Lc
Update Card Insertion Counter	E0h	00h	00h	0Ah	00h

Update Card Insertion Counter 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	04h	ICC Cnt (LSB)	ICC Cnt (MSB)	PICC Cnt (LSB)	PICC Cnt (MSB)

其中：

ICC Cnt (LSB) (1 字节)
ICC 插入计数器（LSB)

ICC Cnt (MSB) (1 字节)
ICC 插入计数器 (MSB)

PICC Cnt (LSB) (1 字节)
PICC 插入计数器 (LSB)

PICC Cnt (MSB) (1 字节)
PICC 插入计数器 (MSB)

附录 B.9. 兼容性与迁移说明

在产品优化升级过程中，ACR1281 支持的一些命令在 ACR1581 中做了更新或者替换。

下表给出了明确的对应关系，以便协助客户在两款机型之间平稳过渡。

诸如 LED 控制、PICC 轮询这类最常用的功能依然能正常使用，但一些命令使用了新的名称或配置结构。

如果您在迁移或集成过程中遇到任何问题，欢迎发送邮件至 info@acs.com.hk，我们的技术团队将很高兴为您提供帮助。

注：

ACR1281U 所具备的卡片插入计数器功能，在 ACR1581U 中未予以支持。

依赖该项功能的应用程序需要进行相应的修改。

ACR1281U API	ACR1281U 指令	ACR1581U 是否支持	ACR1581U 对应指令	说明
LED 控制	E0 00 00 29 01	是	**设置 LED 控制（Set LED Control）[E0 00 00 29 01 …]**	概念相同，但 LED 颜色不同（蓝色/绿色，而非红色/绿色）
LED 状态	E0 00 00 29 00	是	**获取 LED 状态（Get LED Status）[E0 00 00 29 00]**	同上

ACR1281U API	ACR1281U 指令	ACR1581U 是否支持	ACR1581U 对应指令	说明
设置默认 LED 和蜂鸣器操作	E0 00 00 21 01	是	**设置 UI 操作（Set UI Behaviour）[E0 00 00 21 01 …]**	位定义发生改变，删除了独占模式蜂鸣器，并新增上电蜂鸣器控制
读取默认 LED 和蜂鸣器操作	E0 00 00 21 00	是	**获取 UI 操作（Get UI Behaviour）[E0 00 00 21 00]**	同上
设置自动 PICC 轮询	E0 00 00 23 01	是	**设置轮询/ATR 选项（Set Polling/ATR Option）[E0 00 00 23 01** …]	位定义发生变更，删除了 PICC 不工作时关闭天线的设置，并在 ATR 中为第 3 部分卡片增添额外的 MIFARE 类型识别
读取自动 PICC 轮询	E0 00 00 23 00	是	**获取轮询/ATR 选项（Get Polling/ATR Option）[E0 00 00 23 00]**	同上