Advanced Card Systems Ltd.
Card & Reader Technologies

ACR1251U
带 SAM 的 NFC 读卡器

应用程序编程接口 V1.08

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

版本历史

发布日期	修订说明	版本号
2013-05-31	• 初始发布	1.00
2014-01-08	• 删除 6.4 节以下命令 o 设置自动 PPS o 读取自动 PPS o 天线场控制 o 读取天线场状态 o 读取用户额外保护时间 o 设置“616C” 自动操作选项 o 读取 “616C”自动操作选项	1.01
2014-03-07	• 重新排列各节	1.02
2014-04-03	• 修改 5.7.6、5.7.7、5.6.8 和 5.6.9 节： 设置/读取 LED 和蜂鸣器状态指示器的信息（针对固件版本 205 及以下，和固件版本为 206 及以上情况） • 新增 MIFARE 商标及归属说明	1.03
2014-05-23	• 更新 5.8 节：NFC 点对点通信相关命令 • 更新格式 • 将缩写语附录移至 3.0 节：缩略语: • 将兼容 ACR122U 的命令移至 5.9 节 • 在 2.0 节特性中，新增 KC 认证	1.04
2014-10-16	• 更新 2.0 节：特性	1.05
2014-10-30	• 新增 5.3.1 节：获取数据	1.06
2015-10-06	• 将 ACR1251U-A1 更名为 ACR1251U • 在 2.0 节特性中新增 REACH 认证	1.07
2017-09-20	• 更新 2.0 节：特性	1.08

目录

图目录

表目录

1.0. 简介

ACR1251U 是一款在频率为 13.56 MHz 的非接触式技术上开发出来的连机 NFC 智能卡读写器。它继 ACS 最成功的，同时也是全球首款符合 CCID 标准的非接触式 NFC 读卡器—ACR122U 之后，提供了更多先进的功能。它既支持 ISO 14443 的 A 类和 B 类卡，也支持 MIFARE®, FeliCa 卡以及全部四种 NFC 标签和设备。

做为电脑与卡片之间的中间设备，ACR1251U 会执行来自于电脑的命令，专门与非接触式标签、SAM 卡及外围设备（LED 或蜂鸣器）进行通信。它会执行来自于电脑的命令，专门与非接触式标签、SAM 卡及外围设备（LED 或蜂鸣器）进行通信。它的两种界面（PICC 界面和 SAM 界面）均符合 PC/SC 标准。本 API 文件则会详细介绍如何执行 PC/SC APDU 命令来支持非接触式界面和以及控制 ACR1251U 的外围设备。

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/写速率高达 424 Kbps
    - 内置天线用于读写非接触式标签，读取智能卡的距离可达 50 mm（视标签的类型而定）
    - 支持 ISO 14443 第 4 部分 A 类和 B 类卡、MIFARE 卡、FeliCa 卡和全部四种 NFC（ISO/IEC 18092）标签
    - 内建防冲突特性（任何时候都只能读写 1 张标签）
    - NFC 支持：
      - 读写器模式
      - 点到点通信模式
  - SAM 接口
    - 1 个 SAM 卡槽
    - 支持 ISO 7816 A 类、B 类和 C 类 SAM 卡
- 内置外围设备：
  - 用户可控的双色 LED 指示灯
  - 用户可控的蜂鸣器
- 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）
- 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：
  - EN 60950/IEC 60950
  - ISO 7816 (SAM 卡槽)
  - ISO 14443
  - ISO 18092
  - FeliCa 性能认证
  - PC/SC
  - CCID
  - CE
  - FCC
  - RoHS 2
  - REACH
  - J-LIS (日本)
  - VCCI (日本)

[1] 使用 *ACS 定义的安卓库*

- MIC (日本)
- KC (韩国)
- Microsoft® WHQL

3.0. 缩略语

缩略语	说明
ATR	属性请求和属性响应
DEP	数据交换协议请求及数据交换协议响应
DSL	取消请求和取消响应
PSL	参数选择请求和参数选择响应
RLS	发布请求和发布响应
WUP	唤醒请求和唤醒响应
DID	设备 ID
BS	发送比特周期
BR	接收比特周期
PP	协议参数
Gi	发起人可选信息域
PFB	交易控制信息
FSL	帧长度的最大值
LLCP	逻辑链路控制协议

表1： 缩略语

4.0. 架构

ACR1251U 与计算机之间的数据通讯采用 CCID 协议。而 PICC 和 SAM 间的通信则完全符合 PC/SC 标准。

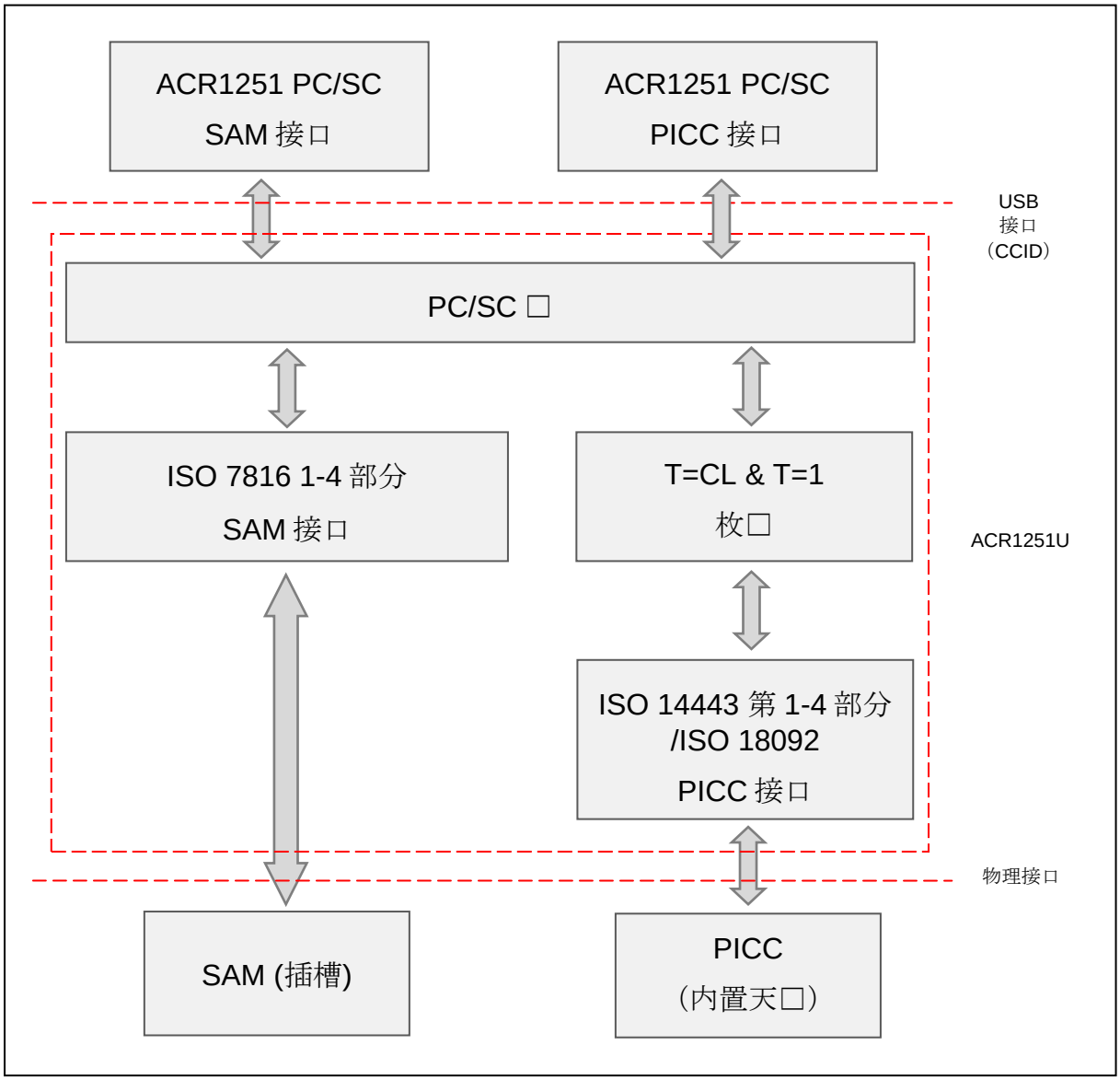

图1： ACR1251U-A1 架构

5.0. 主机编程（连机）API

5.1. PC/SC API

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

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

5.1.2. SCardListReaders

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

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

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

5.1.3. SCardConnect

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

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

5.1.4. SCardControl

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

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

注:5.7 *节的命令使用此 API 进行发送。*

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。

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

注: *APDU 命令（即：发送给已建立连接的卡片的命令，**5.3 节 - PICC 命令**）使用此 API 进行发送。*

5.1.6. SCardDisconnect

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

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

5.1.7. APDU 流程图

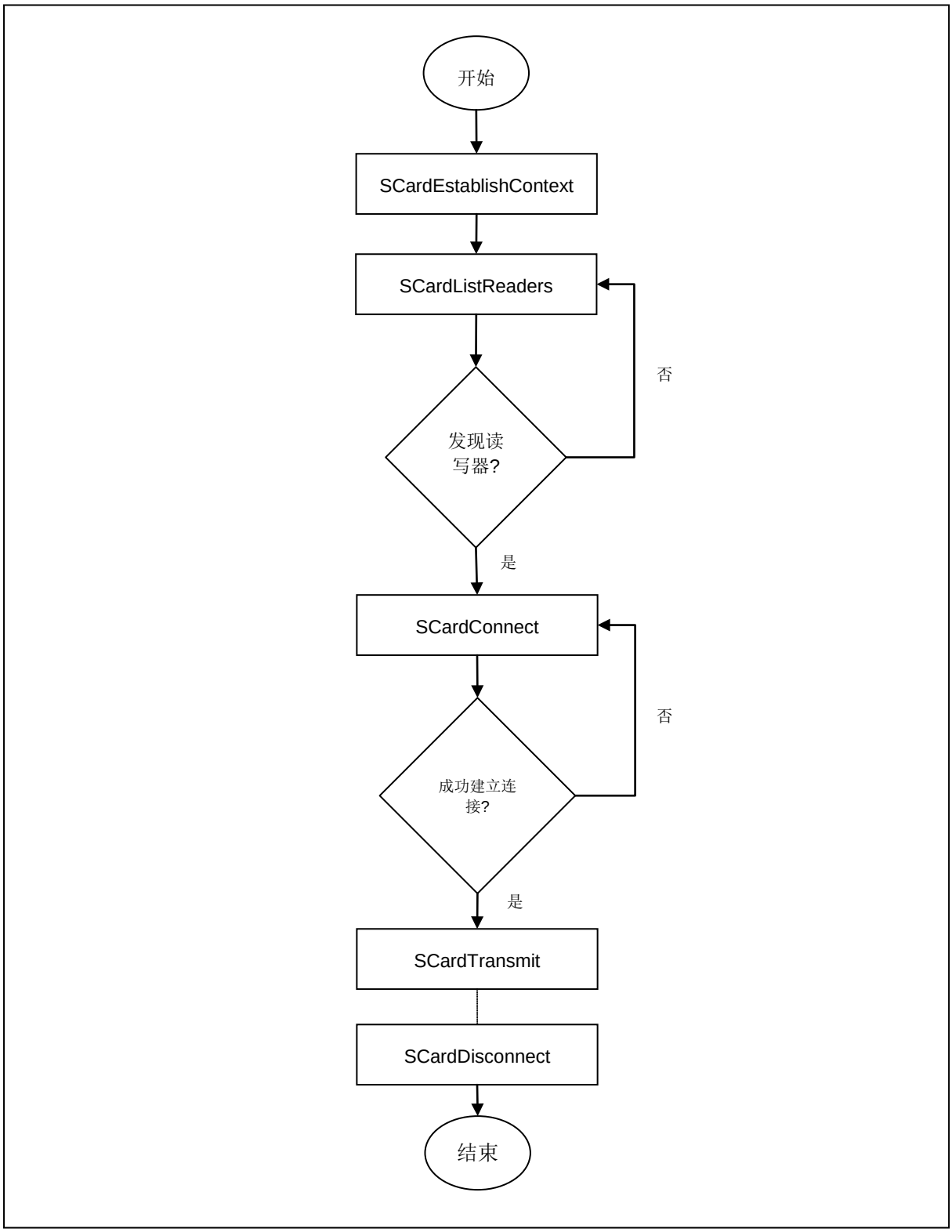

图2： ACR1251U-A1 APDU 流程图

5.1.8. 直接命令（Escape Command）流程图

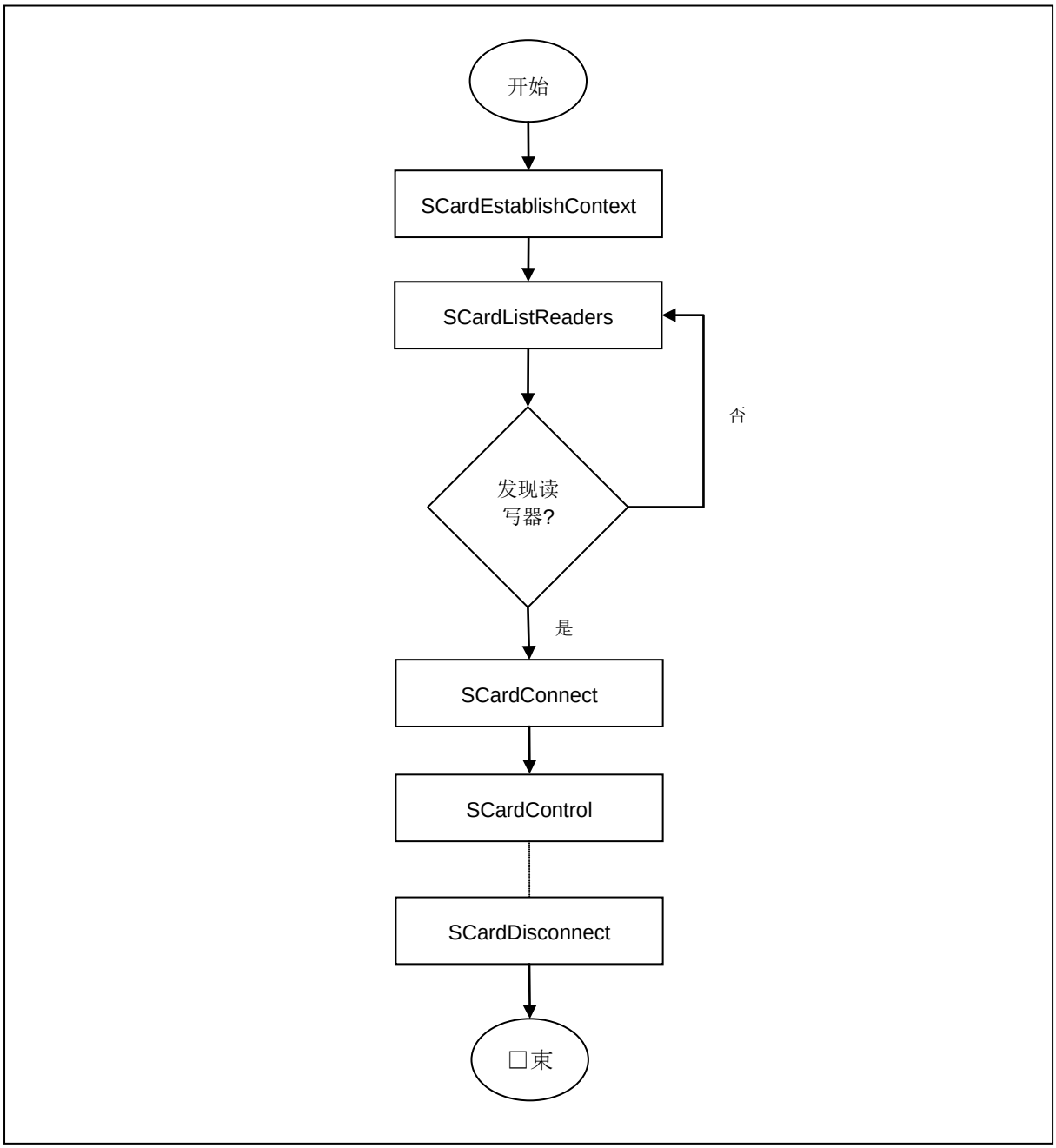

图3：ACR1251U-A1 直接命令（Escape Command）流程图

5.2. 非接触式智能卡协议

5.2.1. ATR 的生成

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

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

字节	Value	标记	说明
0	3Bh	初始头部	-

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 至 3+N	80h	T1	类别指示字节，80 表示在可选的 COMPACT-TLV 数据对象中可能存在状态指示。
	4Fh	Tk	应用标识符存在标识。
	0Ch		Length
	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：

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 30: Topaz 和 Jewel
00 02：MIFARE Classic 4K	00 3B: FeliCa
00 03：MIFARE Ultralight®	FF 28: JCOP 30
00 26：MIFARE Mini	FF [SAK]：未定义标签

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

字节	Value	标记	说明

<table>
<tr><th>字节</th><th>Value</th><th>标记</th><th>说明</th></tr>
<tr><td>0</td><td>3Bh</td><td>初始头部</td><td>-</td></tr>
<tr><td>1</td><td>8N</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
至
3 + N</td><td>XX</td><td>T1</td><td rowspan="2">历史字节：

ISO 14443-A：
来自 ATS 应答的历史字节。参考 ISO 14443-4 标准。

ISO 14443-B：
<table><tr><th>Byte1-4</th><th>Byte5-7</th><th>Byte8</th></tr><tr><td>ATQB 的应用数据</td><td>ATQB 的协议信息字符</td><td>高半字节 =ATTRIB 命令的 MBLI；低半字节 (RFU)=0</td></tr></table></td></tr>
<tr><td>XX
XX
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 bytes of 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. 非接触接口的私有 APDU 指令

5.3.1. 获取数据（Get Data）

此命令用于获取“已建立连接的 PICC”的序列号或 ATS。

Get UID 的 APDU 结构（5 字节）

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

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

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

响应	响应数据域		
结果	ATS	SW1	SW2

响应状态码

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

例如：

获取“已经建立连接的 PICC”的序列号:

```
UINT8 GET_UID[5] = {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};
```

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

5.4.1. 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 Classic 存储卡的特定扇区。读写器提供了两种认证密钥位置：易失密钥位置和非易失密钥位置。

Load Authentication Keys 的 APDU 结构（11 字节）

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

其中：

密钥结构 1 字节。

00h = 密钥被载入读写器的易失存储器。

其它 = 保留。

密钥号 1 字节。

00h – 01h =用于存储密钥的非易失性存储器。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保留在读写器存储里。读写器的非易失性存储器内可以存储最多 32 个密钥。

注：*默认值是 FF FF FF FF FF FFh。*

密钥 6 字节。

载入读写器的密钥值。例：FF FF FF FF FF FFh

Load Authentication Keys 的响应结构（2 字节）

响应	响应数据域	
结果	SW1	SW2

Load Authentication Keys 命令的响应状态码

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

例：

// 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}

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

此命令使用存储在读写器内的密钥来验证 MIFARE Classic 卡（PICC）。涉及两种认证密钥：TYPE_A 和 TYPE_B。

Load Authentication Keys 的 APDU 结构（6 字节）[弃用]

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

Load Authentication Keys 的 APDU 结构（10 字节）

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

认证数据字节（5 字节）

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

其中：

块号 1 字节。待验证的存储块。

MIFARE Classic 1K 卡的内存划分为 16 个扇区，每个扇区包含 4 个连续的块。（例如：扇区 00h 包含块{00h、01h、02h 和 03h}；扇区 01h 包含块{04h、05h、06h 和 07h}；最后一个扇区 0Fh 包含块{3Ch、3Dh、3Eh 和 3Fh}。）验证通过后，读取同一扇区内的其他块不需要再次进行验证。详情请参考 MIFARE Classic 1K/4K 卡标准。

注： *一旦该块被成功验证，即可访问属于同一扇区的所有块。*

密钥类型 1 字节。

60h = 该密钥被用作 TYPE A 密钥进行验证

61h = 该密钥被用作 TYPE B 密钥进行验证

密钥号 1 字节。

00 ~ 01h = 用于存储密钥的易失性存储器。一旦读写器与电脑断开连接，密钥就会消失。提供了两个易失性密钥。 可以用作不同会话的过程密钥。

Load Authentication Keys 的响应结构（2 字节）

响应	响应数据域	
结果	SW1	SW2

Load Authentication Keys 命令的响应状态码

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

例如：

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

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

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

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

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2： 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
扇区 32	80h – 8Eh	8Fh
扇区 33	90h – 9Eh	9Fh
..	..	..
..	..	..
扇区 38	E0h – EEh	EFh
扇区 39	F0h – FEh	FFh

4 KB

表3： MIFARE Classic 4K 卡的内存结构

例如：

// 要使用{TYPE A，密钥号 00h}验证块 04h。

// PC/SC V2.01，弃用

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

// 要使用{TYPE A，密钥号 00h}验证块 04h。

// PC/SC V2.07

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

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	内部	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□

表4：MIFARE Ultralight 卡的内存结构

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

5.4.3. 读取二进制块（Read Binary Blocks）

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

Read Binary 的 APDU 结构（5 字节）

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

其中：

块号 1 字节。起始块。

待读取的字节数 1 字节。

MIFARE Classic 1K/4K 卡的待读字节的长度应该是 16 字节的倍数；MIFARE Ultralight 卡应该是 4 字节的倍数。

MIFARE Ultralight 卡的待读字节数最大为 16。

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

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

例 1： 10h (16 字节)。仅起始块。（单块模式）

例 2： 40h（64 字节）。从起始块至起始+3 块。（多块模式）

注： *出于安全原因，多块模式仅用于读写数据块。尾部块不能在多块模式下读写，请使用单块模式对其进行读写。*

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

响应	响应数据域		
结果	数据（4/16 字节的倍数）	SW1	SW2

Read Binary Block 的响应状态码

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

例如：

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

5.4.4. 更新二进制块（Update Binary Blocks）

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

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

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

其中：

块号 1 字节。待更新的起始块

待更新的字节数 1 字节。

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

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

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

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

例 1：10h (16 字节)。仅起始块。（单块模式）

例 2：30h（48 字节）。从起始块至起始 + +2 块。（多块模式）

注：*出于安全原因，多块模式仅用于访问数据块。尾部块不能在多块模式下读写，请使用单块模式对其进行读写。*

Update Binary Block 的响应状态码（2 字节）

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

例如：

// 将 MIFARE 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}

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

此命令用于对基于数值的交易进行操作（例如：增加值块的值）。

Value Block Operation 的 APDU 结构（10 字节）

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

其中：

块号 1 字节。待操作的值块。

VB_OP 1 字节。

00h = 将 VB_Value 存入该块，然后该块将变为值块。

01h = 使值块的值增加 VB_Value。该命令仅用于操作值块。

02h = 使值块的值减少 VB_Value。该命令仅用于操作值块。

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

例 1： Decimal –4 = {FFh, FFh, FFh, FCh}

VB_Value			
MSB			LSB
FFh	FFh	FFh	FCh

例 2： Decimal 1 = {00h, 00h, 00h, 01h}

VB_Value			
MSB			LSB
00h	00h	00h	01h

Value Block Operation 的响应结构（2 字节）

响应	响应数据域	
结果	SW1	SW2

Value Block Operation 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。

5.4.6. 读取值块（Read Value Block）

此命令用于获取值块中的数值，该命令仅用于操作值块。

Read Value Block 的 APDU 结构（5 字节）

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

其中：

块号 1 字节。待读写的值块。

Read Value Block 的响应结构（4 + 2 字节）

响应	响应数据域		
结果	Value {MSB ..LSB}	SW1	SW2

其中：

值 4 字节。卡片返回的数值，是一个有符号长整数（4 字节）。

例 1：Decimal –4 = {FFh, FFh, FFh, FCh}

Value			
MSB			LSB
FFh	FFh	FFh	FCh

例 2：Decimal 1 = {00h, 00h, 00h, 01h}

Value			
MSB			LSB
00h	00h	00h	01h

Read Value Block 命令的响应状态码

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

5.4.7. 复制值块（Copy Value Block）

此命令用于将一个值块中的数值复制到另外一个值块。

Copy Value Block 命令的 APDU 结构（7 字节）

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

其中：

源块号 1 字节。源值块中的值会被复制到目标值块。

目标块号 1 字节。要恢复的值块。源值块和目标值块必须位于同一个扇区。

Copy Value Block 的响应结构（2 字节）

响应	响应数据域	
结果	SW1	SW2

Copy Value Block 命令的响应状态码

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

例如：

// 将数值“1”存入块 05h

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

// 读取值块 05h

APDU = {FF B1 00 05 04h}

将值块 05h 的值复制到值块 06h

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

// 使值块 05h 的值增加“5”

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

5.5. 访问符合 PC/SC 的标签（ISO 14443-4）

基本上，所有符合 ISO 14443-4 标准的卡片（PICC 卡）都可以理解 ISO 7816-4 规定的 APDUs。ACR1251U-K1 读写器与符合 ISO 14443-4 标准的卡片进行通信时，需要对 ISO 7816-4 规定的 APDU 和响应进行转换。ACR1251U 会在内部处理 ISO 14443 第 1-4 部分协议。

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 **5.3 节**。

ISO 7816-4 规定的 APDU 报文的结构

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

ISO 7816-4 规定的响应报文的结构（数据 + 2 字节）

响应	响应数据域		
结果	响应数据	SW1	SW2

通用的 ISO 7816-4 命令的响应状态码

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

典型的操作顺序为：

1. 出示标签，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"来获取 ATS。*

例：

// 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]

5.6. 访问 FeliCa 标签

访问 FeliCa 标签的命令不同于访问符合 PC/SC 标签的命令及访问 MIFARE 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

   RES = [IDM (8bytes)] 90 00h

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

3. FeliCa 命令访问。

   例：以“读取” 内存块为例：

   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

   RES = Memory Block Data

5.7. 外设控制

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

5.7.1. 获取固件版本号（Get Firmware Version）

此命令用于获取读写器的固件信息。

Get Firmware Version 的命令结构（5 字节）

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

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

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

例：

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

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

固件版本号 (ASCII) = “ACR1251U_V204.0”

5.7.2. LED 控制（LED Control）

此命令用于控制 LED 的输出。

LED Control 命令的结构（6 字节）

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

LED Control 命令的响应结构（6 字节）

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

LED 状态（1 字节）

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

5.7.3. LED 状态（LED Status）

此命令用于检查当前 LED 的状态。

LED Status 命令的结构（5 字节）

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

LED Status 的响应结构（6 字节）

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

LED 状态（1 字节）

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

5.7.4. 蜂鸣器控制（Buzzer Control）

此命令用于控制蜂鸣器的输出。

Buzzer Control 的命令结构（6 字节）

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

其中：

蜂鸣器持续时间 1 字节。

00h = 关闭

01 - FFh = 持续时间（单位：10 ms)

Buzzer Control 的响应结构（6 字节）

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

5.7.5. 蜂鸣器状态（Buzzer Status）

此命令用于检查当前蜂鸣器的状态。

Buzzer Status 的命令结构（5 字节）

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

Buzzer Status 的响应结构（6 字节）

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

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

此命令用于设置 LED 和蜂鸣器作为状态指示器的各种操作。 本节中此命令的结构只适用于 205 版本或更低版本的固件。

注：*该设置将保存在非易失存储器中。*

Set LED and Buzzer Status Indicator Behaviors 的 命令结构（6 字节）

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

操作（1 字节）

操作	模式	说明
Bit 0	SAM 激活状态 LED	显示 SAM 接口的激活状态。 1 = 启用；0 = 禁用
Bit 1	PICC 轮询状态 LED	显示 PICC 轮询状态 1 = 启用；0 = 禁用
Bit 2	PICC 激活状态 LED	显示 PICC 界面的激活状态 1 = 启用；0 = 禁用
Bit 3	卡片插入和卡片移除事件蜂鸣器	每次检测到卡片插入或者卡片移出就会发出哔的一声。（PICC） 1 = 启用；0 = 禁用
Bit 4 – 6	RFU	RFU
Bit 7	卡片操作闪烁 LED	LED 在卡片（PICC）被读写时会闪烁。

注：*操作的默认值 = 8Fh*

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

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

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

此命令用于读取 LED 和蜂鸣器的当前默认操作。 本节中此命令的结构只适用于 205 版本或更低版本的固件。

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

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

Read LED and Buzzer Status Indicator Behavior 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

操作（1 字节）

操作	模式	说明
Bit 0	SAM 激活状态 LED	显示 SAM 接口的激活状态。 1 = 启用；0 = 禁用
Bit 1	PICC 轮询状态 LED	显示 PICC 轮询状态 1 = 启用；0 = 禁用
Bit 2	PICC 激活状态 LED	显示 PICC 界面的激活状态 1 = 启用；0 = 禁用
Bit 3	卡片插入和卡片移除事件蜂鸣器	每次检测到卡片插入或者卡片移出就会发出哔的一声。（PICC） 1 = 启用；0 = 禁用
Bit 4 – 6	RFU	RFU
Bit 7	卡片操作闪烁 LED	LED 在卡片（PICC）被访问时会闪烁。

注：*操作的默认值 = 8Fh*

5.7.8. 设置 PICC 接口的 LED 和蜂鸣器状态指示器（Set LED and Buzzer Status Indicator Behavior for PICC interface）

此命令用于设置 PICC 接口的 LED 和蜂鸣器作为状态指示器的各种操作。 本节中此命令的结构只适用于 206 版本或更高版本的固件。

注：*该设置将保存在非易失存储器中。*

Set LED and Buzzer Status Indicator Behavior for PICC interface Format (6 bytes)

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

操作（1 字节）

操作	模式	说明
Bit 0	卡片操作闪烁 LED	读写 PICC 卡片，LED 都会闪烁。 1 = 启用；0 = 禁用
Bit 1	PICC 轮询状态 LED	显示 PICC 轮询状态 1 = 启用；0 = 禁用
Bit 2	PICC 激活状态 LED	显示 PICC 界面的激活状态 1 = 启用；0 = 禁用
Bit 3	卡片插入和卡片移除事件蜂鸣器	每次检测到卡片插入或者卡片移出就会发出哔的一声。（PICC） 1 = 启用；0 = 禁用
Bit 4	RFU	RFU
Bit 5	PN512 复位指示蜂鸣器	PN512 复位时发出哔的一声。 1 = 启用；0 = 禁用
Bit 6	选择颜色（绿色）	表示状态更改的绿色 LED 1 = 启用；0 = 禁用
Bit 7	选择颜色（红色）	表示状态更改的红色 LED 1 = 启用；0 = 禁用

注：*操作的默认值 = 7Fh*

Set LED and Buzzer Status Indicator Behaviors for PICC interface 的命令响应结构（6 字节）

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

5.7.9. 读取 PICC 接口的 LED 和蜂鸣器状态指示器的信息（Read LED and Buzzer Status Indicator Behavior for PICC interface）

此命令用于读取 PICC 接口的 LED 和蜂鸣器的当前默认操作。 本节中此命令的结构只适用于 206 版本或更高版本的固件。

Read LED and Buzzer Status Indicator Behavior for PICC interface 的命令结构（5 字节）

命令	CLA	INS	P1	P2	Lc
Read LED and Buzzer Status Indicator Behavior for PICC interface	E0h	00h	00h	21h	00h

Read LED and Buzzer Status Indicator Behavior for PICC interface 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

操作（1 字节）

操作	模式	说明
Bit 0	卡片操作闪烁 LED	读写 PICC 卡片时，LED 都会闪烁。 1 = 启用；0 = 禁用
Bit 1	PICC 轮询状态 LED	显示 PICC 轮询状态 1 = 启用；0 = 禁用
Bit 2	PICC 激活状态 LED	显示 PICC 界面的激活状态 1 = 启用；0 = 禁用
Bit 3	卡片插入和卡片移除事件蜂鸣器	每次检测到卡片插入或者卡片移出就会发出哔的一声。（PICC） 1 = 启用；0 = 禁用
Bit 4	RFU	RFU
Bit 5	PN512 复位指示蜂鸣器	PN512 复位时发出哔的一声。 1 = 启用；0 = 禁用
Bit 6	选择颜色（绿色）	表示状态更改的绿色 LED 1 = 启用；0 = 禁用
Bit 7	选择颜色（红色）	表示状态更改的红色 LED 1 = 启用；0 = 禁用

***注：** 操作的默认值 = 7Fh.*

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

此命令用于设置读写器的轮询模式。

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

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

注：*该设置将保存在非易失存储器中。*

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

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

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

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

轮询设置（1 字节）

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

注：*轮询设置的默认值 = 8Fh*

提示：

1. *建议启用"如果 PICC 闲置，关闭天线场"选项，这样闲置的 PICC 就不会一直暴露在天线场中，可以防止 PICC"发热"。*
2. *PICC 轮询间隔时间越长，节能效果越好。然而，PICC 轮询的响应时间也会增加。在节能状态下，空闲时的电流消耗约为 60 mA；而在非节能状态下，空闲时的电流消耗约为 130 mA。**注：**空闲时的电流消耗=PICC 尚未激活。*
3. *读写器会自动激活"ISO 14443A-4 PICC"的 ISO 14443A-4 模式。B 类 PICC 不会受此选项影响。*
4. *JCOP30 卡片有两种模式：ISO 14443A-3（MIFARE Classic 1K）和 ISO 14443A-4 模式。一旦 PICC 被激活，应用就必须要选定一种模式。*

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

此命令用于检查当前的 PICC 轮询设置。

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

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

Read the Configure Mode 的响应结构（6 字节）

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

轮询设置（1 字节）

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

注：*轮询设置的默认值 = 8Fh*

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

此命令用于设置 PICC 操作参数。

注：*该设置将保存在非易失存储器中。*

Set the PICC Operating Parameter 的命令结构（6 字节）

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

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

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1	00h	00h	00h	01h	操作参数

操作参数（1 字节）

操作参数	参数	说明	选项
Bit 0	ISO 14443 A 类	PICC 轮询要检测的标签类别	1 = 检测 0 = 跳过
Bit 1	ISO 14443 B 类		1 = 检测 0 = 跳过
Bit 2	FeliCa 212 Kbps		1 = 检测 0 = 跳过
Bit 3	FeliCa 424 Kbps		1 = 检测 0 = 跳过
Bit 4	Topaz		1 = 检测 0 = 跳过
Bit 5 - 7	RFU	RFU	RFU

注：*操作参数的默认值 = 1Fh*

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

此 命令用于检查 当前的 PICC 操作参数。

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

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

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

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

操作参数（1 字节）

操作参数	参数	说明	选项
Bit 0	ISO 14443 A 类	PICC 轮询要检测的标签类别。	1 = 检测 0 = 跳过
Bit 1	ISO 14443 B 类		1 = 检测 0 = 跳过
Bit 2	FeliCa 212 Kbps		1 = 检测 0 = 跳过
Bit 3	FeliCa 424 Kbps		1 = 检测 0 = 跳过
Bit 4	Topaz		1 = 检测 0 = 跳过
Bit 5 - 7	RFU	RFU	RFU

注：*操作参数的默认值 = 1Fh。*

5.8. NFC 点到点相关命令

5.8.1. 发起方模式相关命令

本小节介绍了发起方模式下可用的命令。下图显示了该模式中命令的点到点流程。

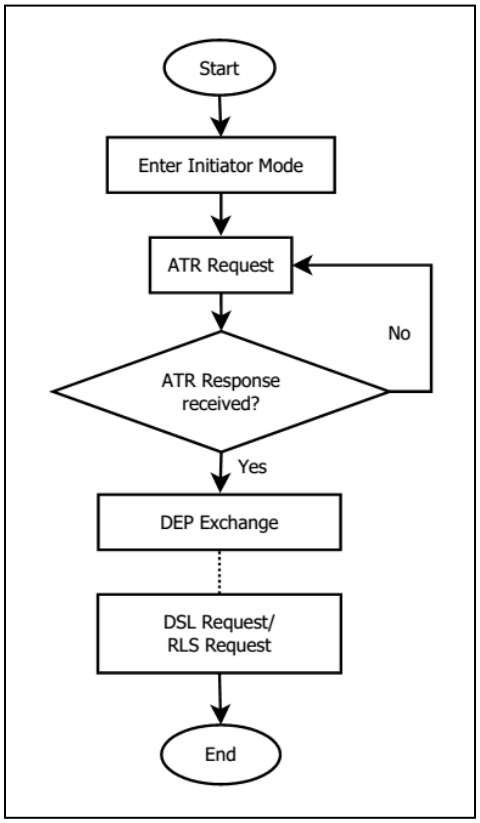

图4： 发起方模式的点到点流程图

5.8.1.1. 设置发起方模式的超时时间（Set Initiator Mode Timeout）

此命令用于设置发起方模式的超时时间。 一旦读写器进入发起方模式，它有 5 次重试机会（相邻两次重试机会的时间间隔为 250 ms），以便成功交换 SNEP 消息。

Set Initiator Mode Timeout 的命令结构 (7 字节)

命令	CLA	INS	P1	P2	Lc	命令数据域	
Set Initiator Mode	E0h	00h	00h	41h	02h	超时时间 (MSB)	超时时间(LSB)

注：*单位 = 10 ms；发起方模式超时时间的默认值 = 00 64h (100 * 10 ms = 1000 ms）。*

Set Initiator Mode Timeout 的响应结构 (7 字节)

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	E1h	00h	00h	00h	02h	超时时间 (MSB)	超时时间 (LSB)

其中：

Timeout 2 字节。发起方模式超时时间（10 ms）。

5.8.1.2. 进入发起方模式（Enter Initiator Mode）

此命令用于设置读写器进入发起方模式，以发送 SNEP 消息。

Enter Initiator Mode 的命令结构（8 字节）

命令	CLA	INS	P1	P2	Lc	命令数据域		
Enter Initiator Mode	E0h	00h	00h	40h	03h	NfcMode	OpMode	Speed

Enter Initiator Mode 的响应结构（8 字节）

响应	CLA	INS	P1	P2	Le	响应数据域		
结果	E1h	00h	00h	00h	03h	NfcMode	OpMode	Speed

其中：

NfcMode 1 字节。NFC 设备模式。
08h = 点到点发起方模式
00h = 卡片读卡器/写模式

OpMode 1 字节。主动模式/被动模式
01h = 主动模式
02h = 被动模式

速度 1 字节。通信速度。
01h = 106 Kbps
02h = 212 Kbps
03h = 424 Kbps

5.8.1.3. 发送 ATR 请求（Send ATR Request）

此命令用于轮询进入工作场内的点到点目标模式的设备。

ATR Request 的命令结构

命令	CLA	INS	P1	P2	Lc	命令数据域				
ATR Request	E0h	00h	00h	42h	Len	11h	Mode（1 字节）	Speed（1 字节）	NFCID（10 字节）	DID（1 字节）

命令数据域				
BS（1 字节）	BR（1 字节）	PP（1 字节）	LLCP 参数	
			GiLen （1 字节）	Gi（GiLen 字节）

ATR Request 的响应结构

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	Len	ATR 响应（Len 字节）

其中：

Mode 1 字节。工作模式。
01h = 主动
02h = 被动

速度 1 字节。通信速度。
01h = 106 Kbps
02h = 212 Kbps
03h = 424 Kbps

NFCID 10 字节。发起方设备的 NFCID。

DID 1 字节。发起方设备的设备标识。

BS 1 字节。发起方设备支持的发送比特速率。

BR 1 字节。发起方设备的支持比特速率。

PP 1 字节。发起方设备的可选参数。

Gi N 字节。LLCP 参数。

5.8.1.4. 交换 DEP（Exchange DEP）

此命令可以与目标设备交换 DEP。

DEP Exchange 的命令结构

命令	CLA	INS	P1	P2	Lc	命令数据域			
DEP Exchange	E0h	00h	00h	43h	Len	11h	PFB（1 字节）	DepLen（1 字节）	Dep (N 字节)

DEP Exchange 的响应结构

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	Len	Dep 响应（Len 字节）

其中：

PFB 1 字节。控制数据传输和纠错。

DepLen 1 字节。DEP 消息长度。

Dep N 字节。DEP 消息用于点到点通信。

5.8.1.5. 发送 DSL 请求（Send DSL Request）

此命令用于发送 DSL 请求给目标设备。

DSL Request 的命令结构

命令	CLA	INS	P1	P2	Lc	命令数据域	
DSL request	E0h	00h	00h	44h	02h	11h	DID（1 字节）

其中：

DID 1 字节。设备标识。

DSL Request 的响应结构

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	02h	返回码（2 字节）

返回码

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

5.8.1.6. 发送 RLS 请求（Send RLS Request）

此命令用于发送 RLS 请求给目标设备。

RLS request 的命令结构

命令	CLA	INS	P1	P2	Lc	命令数据域	
RLS request	E0h	00h	00h	45h	02h	11h	DID（1 字节）

其中：

DID 1 字节。设备标识。

RLS request 的响应结构

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	02h	返回码（2 字节）

返回码

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

5.8.1.7. 发送 PSL 请求（Send PSL Request）

此命令用于发送 PSL 请求给目标设备。

PSL request 的命令结构

命令	CLA	INS	P1	P2	Lc	命令数据域		
PSL request	E0h	00h	00h	46h	04h	11h	DID （1 字节）	BRS （1 字节）	FSL （1 字节）

其中：

DID 1 字节。设备标识。

BRS 1 字节。发起方和目标设备的比特率。

FSL 1 字节。帧长度。

PSL request 的响应结构

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	01h	DID (1 字节)

其中：

DID 1 字节。设备标识。

5.8.2. 目标模式相关命令

本小节介绍了目标模式下可用的命令。下图显示了该模式中命令的点到点流程。

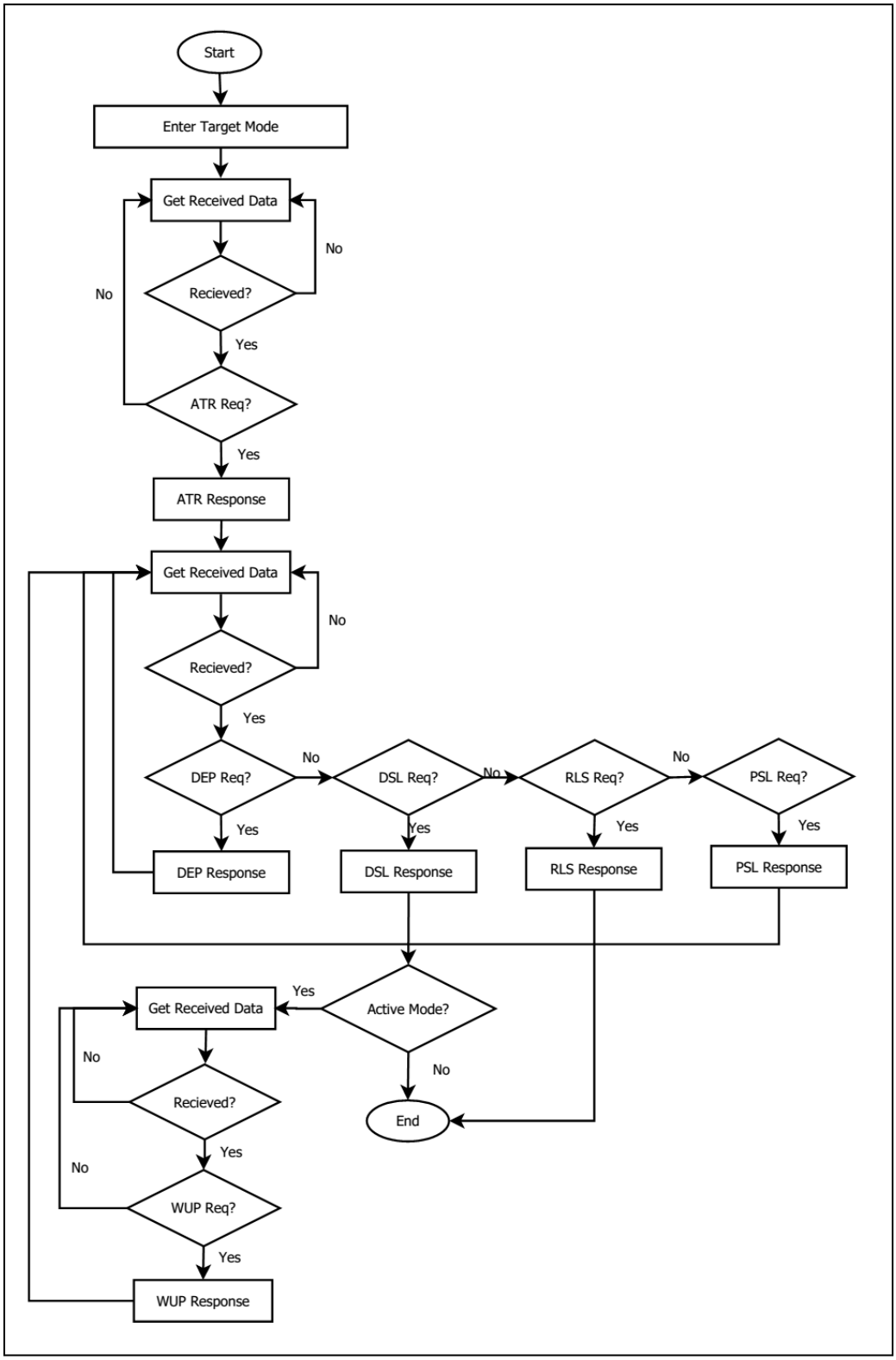

图5： 目标模式的点到点流程图

5.8.2.1. 设置目标模式的超时时间（Set Target Mode Timeout）

此命令用于设置目标模式的超时时间。

Set Target Timeout 的命令结构

命令	CLA	INS	P1	P2	Lc	命令数据域	
Set Target Timeout	E0h	00h	00h	59h	02h	Timeout (MSB)	Timeout (LSB)

注：*单位 100 µs；目标超时时间的默认值 = 00 C8h (200 * 100 µs = 20 ms)。*

Set Target Timeout 的响应结构

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	E1h	00h	00h	00h	02h	Timeout (MSB)	Timeout (LSB)

其中：

Timeout 2 字节。发起方模式超时时间（单位 = 100 ms）。

5.8.2.2. 进入目标模式（Enter Target Mode)

此命令用于设置读写器进入目标模式，以接收 SNEP 消息。

Enter Target Mode 的命令结构

命令	CLA	INS	P1	P2	Lc
Enter Target Mode	E0h	00h	00h	51h	00h

设置进入目标模式，波特率 106 Kbps 以及被动模式

或

命令	CLA	INS	P1	P2	Lc	命令数据域	
Enter Target Mode	E0h	00h	00h	51h	02h	Speed	OpMode

Enter Target Mode 的响应结构

响应	CLA	INS	P1	P2	Le	响应数据域	
结果	E1h	00h	00h	00h	02h	Speed	OpMode

其中：

速度 1 字节。通信速度。

01h = 106 Kbps

02h = 212 Kbps

03h = 424 Kbps

OpMode 1 字节。主动模式/被动模式

01h = 主动模式

02h = 被动模式

5.8.2.3. 发送 ATR 响应（Send ATR Response）

此命令用于发送对发起方 ATR 请求的 ATR 响应。

ATR Response 的命令结构

命令	CLA	INS	P1	P2	Lc	命令数据域
ATR Response	E0h	00h	00h	52h	Len	LLCP 参数 (N 字节)

其中：

LLCP 参数　　N 字节。ATR 响应的通用字节。

ATR 响应结构

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	02h	返回码（2 字节）

返回码

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

5.8.2.4. 发送 DEP 响应（Send DEP Response）

此命令用于发送对发起方 DEP 请求的 DEP 响应。

DEP Response 的命令结构

命令	CLA	INS	P1	P2	Lc	命令数据域	
DEP Response	E0h	00h	00h	53h	Len	PFB（1 字节）	DEP Message (N 字节)

其中：

PFB 1 字节。控制数据传输和纠错。

ATR **消息** N 字节。DEP 响应。

DEP 响应结构

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	02h	返回码（2 字节）

返回码

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

5.8.2.5. 发送 DSL 响应（Send DSL Response）

此命令用于发送对发起方 DSL 请求的 DSL 响应。

DSL Response 的命令结构

命令	CLA	INS	P1	P2	Lc
DSL Response	E0h	00h	00h	54h	00h

DSL 响应结构

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	02h	返回码（2 字节）

返回码

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

5.8.2.6. 发送 RLS 响应（Send RLS Response）

此命令用于发送对发起方 RLS 请求的 RLS 响应。

RLS Response 的命令结构

命令	CLA	INS	P1	P2	Lc
RLS Response	E0h	00h	00h	55h	00h

RLS 响应结构

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	02h	返回码（2 字节）

返回码

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

5.8.2.7. 发送 PSL 响应（Send PSL Response）

此命令用于发送对发起方 PSL 请求的 PSL 响应。

PSL Response 的命令结构

命令	CLA	INS	P1	P2	Lc	命令数据域	
PSL Response	E0h	00h	00h	56h	02h	BRS（1 字节）	FSL（1 字节）

其中：

BRS 1 字节。BRS 参数。

FSL 1 字节。FSL 参数。

PSL 响应结构

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	02h	返回码（2 字节）

返回码

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

5.8.2.8. 发送 WUP 响应（Send WUP Response）

此命令用于发送对发起方 WUP 请求的 WUP 响应。

WUP Response 的命令结构

命令	CLA	INS	P1	P2	Lc
WUP Response	E0h	00h	00h	57h	00h

WUP 响应结构

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	02h	返回码（2 字节）

返回码

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

5.8.2.9. 获取接收到的数据（Get Received Data）

此命令用于获取处于发起方模式下的设备消息。

Get Received Data 的命令结构

命令	CLA	INS	P1	P2	Lc
Get Received Data	E0h	00h	00h	58h	00h

Get Received Data 的响应结构

响应	CLA	INS	P1	P2	Le	响应数据域
结果	E1h	00h	00h	00h	Len	接收的消息 (N 字节)

5.9. ACR122U 兼容命令

5.9.1. 双色 LED 和蜂鸣器控制（Bi-color LED and Buzzer Control）

此命令用于控制双色 LED 指示灯和蜂鸣器的状态。

Bi-color LED and Buzzer Control 的命令结构（9 字节）

命令	CLA	INS	P1	P2	Lc	命令数据域 （4 字节）
Bi-color LED and Buzzer Control	FFh	00h	40h	LED 状态控制	04h	闪烁周期控制

P2 LED 状态控制

双色 LED 和蜂鸣器控制的结构（1 字节）

命令	项	说明
Bit 0	红色 LED 最终状态	1 = 开；0 = 关
Bit 1	绿色 LED 最终状态	1 = 开；0 = 关
Bit 2	红色 LED 状态掩码	1 = 更新其状态 0 = 不更改
Bit 3	绿色 LED 状态掩码	1 = 更新其状态 0 = 不更改
Bit 4	红色 LED 初始闪烁状态	1 = 开；0 = 关
Bit 5	绿色 LED 初始闪烁状态	1 = 开；0 = 关
Bit 6	红色 LED 闪烁掩码	1 = 闪烁 0 = 不闪烁
Bit 7	绿色 LED 闪烁掩码	1 = 闪烁 0 = 不闪烁

命令数据域 闪烁周期控制。

Bi-color LED Blinking Duration Control 的命令结构（4 字节）

字节 0	字节 1	字节 2	字节 3
T1 周期 初始闪烁状态 （单位 = 100 ms）	T2 周期 切换闪烁状态 （单位 = 100 ms）	重复次数	蜂鸣器响应

其中：

字节 3 蜂鸣器响应。在 LED 闪烁期间控制蜂鸣器的状态。

00h = 蜂鸣器不开启。

01h = 蜂鸣器在 T1 周期内开启。

02h = 蜂鸣器在 T2 周期内开启。

03h = 蜂鸣器在 T1 和 T2 周期内开启。

响应数据域　　SW1 SW2。读卡器返回的状态码。

状态码

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

LED 当前状态（1 字节）

状态	项	说明
Bit 0	当前的红色 LED	1 = 开；0 = 关
Bit 1	当前的绿色 LED	1 = 开；0 = 关
Bits 2 – 7	保留	

提示：

1. ***LED 状态****操作是在****LED 闪烁****操作之后进行的。*
2. *如果相应的****LED 状态掩码****未启用，则 LED 状态不会发生改变。*
3. *如果相应的****LED 闪烁掩码****未启用，则 LED 不会闪烁。同时，重复次数的值必须大于 0。*
4. *T1 和 T2 周期参数主要用于控制 LED 闪烁的工作周期和蜂鸣器的鸣响时间。比如说，如果 T1=1，T2=1，则工作周期 = 50%。*

   注：*工作周期 = T1/(T1 + T2)。*

1. *如果只想控制蜂鸣器，则将 P2"LED 状态控制"置为 0 即可。*
2. *要想使蜂鸣器工作，"重复次数"必须大于 0。*
3. *如果只想控制 LED，则将参数"蜂鸣器响应"置为 0 即可。*

5.9.2. 获取固件版本号（Get Firmware Version）

此命令用于获取读写器的固件版本号。

Get Firmware Version 的命令结构（5 字节）

命令	CLA	INS	P1	P2	Le
Get Firmware	FFh	00h	48h	00h	00h

Get Firmware Version 的响应结构（X 字节）

响应	响应数据域
结果	固件版本号

例：

响应 = 41 43 52 31 32 35 31 55 5F 56 32 30 34 2E 30h = ACR1251U_V204.0 (ASCII)

5.9.3. 获取 PICC 操作参数（Get PICC Operating Parameter）

此命令用于获取读写器的 PICC 操作参数。

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

命令	CLA	INS	P1	P2	Le
Get PICC Operation Parameter	FFh	00h	50h	00h	00h

Get the PICC Operating Parameter 的响应结构（2 字节）

响应	响应数据域	
结果	90h	PICC 操作参数

PICC 操作参数

位	参数	说明	选项
7	自动 PICC 轮询	启用 PICC 轮询	1 = 启用 0 = 停用
6	自动 ATS 生成	每次激活 ISO 14443-4 A 类标签都发送 ATS 请求。	1 = 启用 0 = 停用
5	轮询时间间隔	设置连续 PICC 轮询之间的时间间隔。	1 = 250 ms 0 = 500 ms
4	FeliCa 424 Kbps	PICC 轮询要检测的标签类别。	1 = 检测 0 = 跳过
3	FeliCa 212 Kbps		1 = 检测 0 = 跳过
2	Topaz		1 = 检测 0 = 跳过
1	ISO 14443 B 类		1 = 检测 0 = 跳过
0	ISO 14443 A 类 ***注：****要检测 MIFARE 标签，必须首先禁用自动 ATS 生成。*		1 = 检测 0 = 跳过

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

此命令用于设置读写器的 PICC 操作参数。

Set PICC operation Parameter 的命令结构（5 字节）

命令	CLA	INS	P1	P2	Le
Set PICC Operation Parameter	FFh	00h	51h	PICC 操作参数	00h

Set PICC operation Parameter 的响应结构（2 字节）

响应	响应数据域	
结果	90h	PICC 操作参数

PICC 操作参数

位	参数	说明	选项
7	自动 PICC 轮询	启用 PICC 轮询	1 = 启用 0 = 停用
6	自动 ATS 生成	每次激活 ISO 14443-4 A 类标签都发送 ATS 请求。	1 = 启用 0 = 停用
5	轮询时间间隔	设置连续 PICC 轮询之间的时间间隔。	1 = 250 ms 0 = 500 ms
4	FeliCa 424 Kbps	PICC 轮询要检测的标签类别。	1 = 检测 0 = 跳过
3	FeliCa 212 Kbps		1 = 检测 0 = 跳过
2	Topaz		1 = 检测 0 = 跳过
1	ISO 14443 B 类		1 = 检测 0 = 跳过
0	ISO 14443 A 类 ***注：*** *要检测 MIFARE 标签，必须首先禁用自动 ATS 生成。*		1 = 检测 0 = 跳过

附录A. SNEP 消息

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

例：

SNEP 消息 = {D1 02 0F 53 70 D1 01 0B 55 01 61 63 73 2E 63 6F 6D 2E 68 6Bh}

偏移	内容	Length	说明
0	D1	1	NDEF 头部。TNF = 01h, SR=1, MB=1, ME=1
1	02	1	记录名长度 （2 字节）
2	0F	1	智能海报数据的长度（15 字节）
3	53 70 (“Sp”)	2	记录名
5	D1	1	NDEF 头部。TNF = 01h, SR=1, MB=1, ME=1
6	01	1	记录名长度 （1 字节）
7	0B	1	URI 数据包的长度（11 字节）
8	55 (“U”)	1	记录类型“U”
9	01	1	缩写“http://www.”
10	61 63 73 2E 63 6F 6D 2E 68 6B	10	URL 本身。“acs.com.hk”