



ZLG600S 系列用户指南

集成电路卡读写器

UM01010101 V1.01 Date: 2013/11/06



| 类别 | 内容 |
|-----|---|
| 关键词 | ZLG600SP/T、ISO14443、ISO7816 |
| 摘 要 | 本文档详细介绍了模块的硬件管脚配置、通讯协议及各个命令详解，可指导用户正确使用该模块。 |



修订历史

| 版本 | 日期 | 原因 |
|-------|------------|-----------|
| V0.00 | 2013/03/22 | 创建文档 |
| V1.00 | 2013/8/20 | 产品型号确定 |
| V1.01 | 2013/11/06 | 补全缺少的产品型号 |

销售与服务网络（一）

广州致远电子股份有限公司

地址：广州市天河区车陂路黄洲工业区 3 栋 2 楼

邮编：510660

传真：(020)38601859

网址：www.zlg.cn

新浪微博：ZLG-周立功（<http://weibo.com/ligongzhou>）



免费服务电话：400 888 4005

销售及技术支持：

x86 工控：

销售电话：(020) 28267893 22644261 28267985

技术支持：(020) 28872451

x86.support@zlg.cn

MiniARM 及 ARM 工控：

销售电话：(020) 22644261 28267985 28267893

技术支持：(020) 28872684 28872412

miniarm.support@zlg.cn

分析仪器：

销售电话：(020) 28872453 28872569

技术支持：(020) 22644375

support@zlg.cn

编程器：

销售电话：(020) 28872453 28872569

技术支持：(020) 22644371

support@zlg.cn

仿真器：

销售电话：(020) 28872453 28872569

技术支持：(020) 22644360

TKScope@zlg.cn

楼宇自动化：

销售电话：(020) 28872573

技术支持：(020) 22644389 28267806

mjs.support@zlg.cn

mifare.support@zlg.cn

CAN-bus：

销售电话：(020) 28872524 28872342 28872450

技术支持：(020) 22644381 22644382

can.support@zlg.cn

ZigBee 及以太网：

销售电话：(020) 28872524 28872342 28872450

技术支持：(020) 22644380 22644385

ethernet.support@zlg.cn

串行通讯：

销售电话：(020) 28872524 28872342 28872450

技术支持：(020) 28267800 22644385

serial@zlg.cn

无线通讯：

销售电话：(020) 28872524 28872342 28872450

技术支持：(020) 22644386

DTU@zlg.cn

数据采集及 iCAN：

销售电话：(020) 28872524 28872342 28872450

技术支持：(020) 28267821

ican@zlg.cn

电源产品：

销售电话：(020) 28267835 28267859

技术支持：(020) 22644373 28267925

power@zlg.cn

销售与服务网络（二）

广州周立功单片机科技有限公司

地址：广州市天河北路 689 号光大银行大厦 12 楼 F4
邮编：510630
传真：(020)38730925
网址：www.zlgmcu.com
电话：(020)38730916 38730917 38730972 38730976 38730977
新浪微博：ZLG-周立功（<http://weibo.com/ligongzhou>）



广州专卖店

地址：广州市天河区新赛格电子城 203-204 室
电话：(020)87578634 87569917
传真：(020)87578842

南京周立功

地址：南京市珠江路 280 号珠江大厦 1501 室
电话：(025) 68123901 68123902
传真：(025) 68123900

北京周立功

地址：北京市海淀区知春路 113 号银网中心 A 座
1207-1208 室（中发电子市场斜对面）
电话：(010)62536178 62536179 82628073
传真：(010)82614433

重庆周立功

地址：重庆市石桥铺科园一路二号大西洋国际大厦
（赛格电子市场）2705 室
电话：(023)68796438 68796439
传真：(023)68796439

杭州周立功

地址：杭州市天目山路 217 号江南电子大厦 502 室
电话：(0571)89719480 89719481 89719482
89719483 89719484 89719485
传真：(0571)89719494

成都周立功

地址：成都市一环路南二段 1 号数码科技大厦 403 室
电话：(028)85439836 85437446
传真：(028)85437896

深圳周立功

地址：深圳市福田区深南中路 2072 号电子大厦 12 楼
电话：(0755)83781788（5 线）
传真：(0755)83793285

武汉周立功

地址：武汉市洪山区广埠屯珞瑜路 158 号 12128 室（华中电脑数码市场）
电话：(027)87168497 87168297 87168397
传真：(027)87163755

上海周立功

地址：上海市北京东路 668 号科技京城东楼 12E 室
电话：(021)53083452 53083453 53083496
传真：(021)53083491

西安办事处

地址：西安市长安北路 54 号太平洋大厦 1201 室
电话：(029)87881296 83063000 87881295
传真：(029)87880865

厦门办事处

E-mail: sales.xiamen@zlgmcu.com

沈阳办事处

E-mail: sales.shenyang@zlgmcu.com

目 录

| | |
|--------------------------------------|----|
| 1. 功能简介..... | 1 |
| 1.1 功能特点..... | 1 |
| 1.2 技术参数..... | 1 |
| 1.3 极限参数..... | 1 |
| 1.4 直流参数..... | 2 |
| 1.5 订购信息..... | 2 |
| 1.6 封装信息..... | 2 |
| 1.7 产品图片..... | 3 |
| 2. 操作说明..... | 5 |
| 2.1 I/O 设置..... | 5 |
| 2.1.1 通信接口设置 (J1) | 5 |
| 2.1.2 天线接口设置 (J2) | 6 |
| 2.1.3 接触式 IC 卡及电源控制接口设置 (J3) | 7 |
| 2.1.4 RS-232C/RS-485 接口设置 (J6) | 7 |
| 2.2 典型应用..... | 7 |
| 2.2.1 UART 接口应用..... | 7 |
| 2.2.2 I ² C 接口应用..... | 8 |
| 2.2.3 RS-232C/RS-485 接口的应用..... | 9 |
| 2.2.4 多从机方案应用..... | 10 |
| 2.2.5 双天线应用..... | 11 |
| 3. 通讯协议..... | 13 |
| 3.1 各通信模式时序..... | 13 |
| 3.1.1 自动侦测模式时序..... | 13 |
| 3.1.2 UART 模式时序..... | 14 |
| 3.1.3 I ² C 模式时序..... | 14 |
| 3.1.4 模块上电时序参数..... | 15 |
| 3.2 UART 通信设置..... | 15 |
| 3.2.1 波特率检测..... | 15 |
| 3.2.2 UART 数据通信..... | 16 |
| 3.3 I ² C 设置..... | 16 |
| 3.4 旧帧格式..... | 16 |
| 3.4.1 旧命令帧格式..... | 16 |
| 3.4.2 旧回应帧格式..... | 17 |
| 3.5 新帧格式..... | 18 |
| 3.5.1 新帧格式物理链路层..... | 18 |
| 3.5.2 新帧格式协议层..... | 19 |
| 4. 旧帧格式应用命令详述..... | 23 |
| 4.1 设备控制类命令 (CmdClass = 0x01) | 24 |
| 4.1.1 读设备信息 (Cmd = A) | 24 |
| 4.1.2 配置 IC 卡接口 (Cmd = B) | 25 |
| 4.1.3 关闭 IC 卡接口 (Cmd = C) | 25 |

| | | |
|--------|--------------------------------------|----|
| 4.1.4 | 设置 IC 卡接口协议（工作模式）（Cmd = D） | 26 |
| 4.1.5 | 装载 IC 卡密钥（Cmd = E） | 27 |
| 4.1.6 | 设置 IC 卡接口的寄存器值（Cmd = F） | 28 |
| 4.1.7 | 获取 IC 卡接口的寄存器值（Cmd = G） | 28 |
| 4.1.8 | 设置波特率（Cmd = H） | 29 |
| 4.1.9 | 设置天线驱动方式（Cmd = I） | 30 |
| 4.1.10 | 设置新旧帧格式（Cmd = K） | 30 |
| 4.1.11 | 设置设备工作模式（Cmd = U） | 31 |
| 4.1.12 | 获取设备工作模式（Cmd = V） | 32 |
| 4.1.13 | 装载用户密钥（Cmd = a） | 32 |
| 4.1.14 | 读 E ² PROM（Cmd = b） | 33 |
| 4.1.15 | 写 E ² PROM（Cmd = c） | 34 |
| 4.2 | Mifare S50/S70 卡类命令（CmdClass = 0x02） | 35 |
| 4.2.1 | 请求（Cmd = A） | 35 |
| 4.2.2 | 防碰撞（Cmd = B） | 36 |
| 4.2.3 | 卡选择（Cmd = C） | 37 |
| 4.2.4 | 卡挂起（Cmd = D） | 38 |
| 4.2.5 | E ² 密钥验证（Cmd = E） | 39 |
| 4.2.6 | 直接密钥验证（Cmd = F） | 40 |
| 4.2.7 | Mifare 卡读（Cmd = G） | 41 |
| 4.2.8 | Mifare 卡写（Cmd = H） | 42 |
| 4.2.9 | UltraLight 卡写（Cmd = I） | 43 |
| 4.2.10 | Mifare 值操作（Cmd = J） | 43 |
| 4.2.11 | 卡复位（Cmd = L） | 44 |
| 4.2.12 | 卡激活（Cmd = M） | 45 |
| 4.2.13 | 自动检测（Cmd = N） | 46 |
| 4.2.14 | 读自动检测数据（Cmd = O） | 49 |
| 4.2.15 | 设置值块的值（Cmd = P） | 49 |
| 4.2.16 | 获取值块的值（Cmd = Q） | 50 |
| 4.2.17 | 数据交互命令（Cmd = X） | 51 |
| 4.3 | ISO7816-3 类命令（CmdClass = 0x05） | 52 |
| 4.3.1 | 接触式 IC 卡复位(自动处理 PPS) | 52 |
| 4.3.2 | 接触式 IC 卡传输协议（自动处理 T = 0 和 T = 1 协议） | 53 |
| 4.3.3 | 接触式 IC 卡冷复位 | 54 |
| 4.3.4 | 接触式 IC 卡热复位 | 55 |
| 4.3.5 | 接触式 IC 卡停活 | 56 |
| 4.3.6 | 接触式 IC 卡协议和参数选择（PPS） | 57 |
| 4.3.7 | 接触式 IC 卡传输协议（T = 0） | 58 |
| 4.3.8 | 接触式 IC 卡传输协议（T = 1） | 59 |
| 4.4 | ISO14443（PICC）卡类命令（CmdClass = 0x06） | 61 |
| 4.4.1 | A 型卡请求（Cmd = A） | 61 |
| 4.4.2 | A 型卡防碰撞（Cmd = B） | 61 |
| 4.4.3 | A 型卡选择（Cmd = C） | 61 |
| 4.4.4 | A 型卡挂起（Cmd = D） | 62 |

| | | |
|--------|--|----|
| 4.4.5 | A 型卡 RATS (Cmd = E) | 62 |
| 4.4.6 | A 型卡 PPS (Cmd = F) | 62 |
| 4.4.7 | A 型卡解除激活 (Cmd = G) | 63 |
| 4.4.8 | T=CL (Cmd = H) | 64 |
| 4.4.9 | 数据交换 (Cmd = J) | 64 |
| 4.4.10 | A 型卡复位 (Cmd = L) | 65 |
| 4.4.11 | A 型卡激活 (Cmd = M) | 66 |
| 4.4.12 | B 型卡激活 (Cmd = N) | 67 |
| 4.4.13 | B 型卡复位 (Cmd = O) | 67 |
| 4.4.14 | B 型卡请求 (Cmd = P) | 68 |
| 4.4.15 | B 型卡防碰撞 (Cmd = Q) | 69 |
| 4.4.16 | B 型卡修改传输属性 (Cmd = R) | 69 |
| 4.4.17 | B 型卡挂起 (Cmd = S) | 70 |
| 4.5 | PLUS CPU 卡类命令 (CmdClass = 0x07) | 72 |
| 4.5.1 | SL0 个人化更新数据 (Cmd = B) | 72 |
| 4.5.2 | SL0 提交个人化 (Cmd = C) | 73 |
| 4.5.3 | SL3 首次验证 (直接密钥验证) (Cmd = J) | 74 |
| 4.5.4 | SL3 首次验证 (E ² 密钥验证) (Cmd = K) | 75 |
| 4.5.5 | SL3 跟随验证 (直接密钥验证) (Cmd = L) | 75 |
| 4.5.6 | SL3 跟随验证 (E ² 密钥验证) (Cmd = M) | 76 |
| 4.5.7 | SL3 复位验证 (Cmd = N) | 77 |
| 4.5.8 | SL3 读数据块 (Cmd = O) | 77 |
| 4.5.9 | SL3 写数据块 (Cmd = P) | 79 |
| 4.5.10 | SL3 值块操作 (Cmd = S) | 80 |
| 5. | 新帧格式应用命令详述 | 81 |
| 5.1 | 设备控制类命令 (CmdClass = 0x01) | 82 |
| 5.1.1 | 读设备信息 (Cmd = A) | 82 |
| 5.1.2 | 配置 IC 卡接口 (Cmd = B) | 83 |
| 5.1.3 | 关闭 IC 卡接口 (Cmd = C) | 84 |
| 5.1.4 | 设置 IC 卡接口协议 (工作模式) (Cmd = D) | 84 |
| 5.1.5 | 装载 IC 卡密钥 (Cmd = E) | 85 |
| 5.1.6 | 设置 IC 卡接口的寄存器值 (Cmd = F) | 86 |
| 5.1.7 | 获取 IC 卡接口的寄存器值 (Cmd = G) | 87 |
| 5.1.8 | 设置波特率 (Cmd = H) | 87 |
| 5.1.9 | 设置 IC 卡接口的输出 (输出时钟或载波) (Cmd = I) | 88 |
| 5.1.10 | 设置新旧帧格式 (Cmd = K) | 89 |
| 5.1.11 | 设置设备工作模式 (Cmd = U) | 90 |
| 5.1.12 | 获取设备工作模式 (Cmd = V) | 91 |
| 5.1.13 | 装载用户密钥 (Cmd = a) | 92 |
| 5.1.14 | 读 E ² PROM (Cmd = b) | 92 |
| 5.1.15 | 写 E ² PROM (Cmd = c) | 93 |
| 5.2 | Mifare S50/S70 卡类命令 (CmdClass = 0x02) | 95 |
| 5.2.1 | 请求 (Cmd = A) | 95 |
| 5.2.2 | 防碰撞 (Cmd = B) | 96 |

| | | |
|--------|--|-----|
| 5.2.3 | 卡选择 (Cmd = C) | 98 |
| 5.2.4 | 卡挂起 (Cmd = D) | 99 |
| 5.2.5 | E ² 密钥验证 (Cmd = E) | 99 |
| 5.2.6 | 直接密钥验证 (Cmd = F) | 100 |
| 5.2.7 | Mifare 卡读 (Cmd = G) | 101 |
| 5.2.8 | Mifare 卡写 (Cmd = H) | 102 |
| 5.2.9 | UltraLight 卡写 (Cmd = I) | 103 |
| 5.2.10 | Mifare 值操作 (Cmd = J) | 104 |
| 5.2.11 | 卡复位 (Cmd = L) | 105 |
| 5.2.12 | 卡激活 (Cmd = M) | 106 |
| 5.2.13 | 自动检测 (Cmd = N) | 107 |
| 5.2.14 | 读自动检测数据 (Cmd = O) | 109 |
| 5.2.15 | 设置值块的值 (Cmd = P) | 110 |
| 5.2.16 | 获取值块的值 (Cmd = Q) | 111 |
| 5.2.17 | 数据交互命令 (Cmd = X) | 112 |
| 5.3 | ISO7816-3 类命令 (CmdClass = 0x05) | 114 |
| 5.3.1 | 接触式 IC 卡复位(自动处理 PPS) | 114 |
| 5.3.2 | 接触式 IC 卡传输协议 (自动处理 T = 0 和 T = 1 协议) | 115 |
| 5.3.3 | 接触式 IC 卡冷复位 | 116 |
| 5.3.4 | 接触式 IC 卡热复位 | 117 |
| 5.3.5 | 接触式 IC 卡停活 | 118 |
| 5.3.6 | 接触式 IC 卡协议和参数选择 (PPS) | 119 |
| 5.3.7 | 接触式 IC 卡传输协议 (T = 0) | 120 |
| 5.3.8 | 接触式 IC 卡传输协议 (T = 1) | 121 |
| 5.4 | ISO14443 (PICC) 卡类命令 (CmdClass = 0x06) | 123 |
| 5.4.1 | A 型卡请求 (Cmd = A) | 123 |
| 5.4.2 | A 型卡防碰撞 (Cmd = B) | 123 |
| 5.4.3 | A 型卡选择 (Cmd = C) | 123 |
| 5.4.4 | A 型卡挂起 (Cmd = D) | 124 |
| 5.4.5 | A 型卡 RATS (Cmd = E) | 124 |
| 5.4.6 | A 型卡 PPS (Cmd = F) | 124 |
| 5.4.7 | A 型卡解除激活 (Cmd = G) | 125 |
| 5.4.8 | T=CL (Cmd = H) | 126 |
| 5.4.9 | 数据交换 (Cmd = J) | 127 |
| 5.4.10 | A 型卡复位 (Cmd = L) | 128 |
| 5.4.11 | A 型卡激活 (Cmd = M) | 129 |
| 5.4.12 | B 型卡激活 (Cmd = N) | 129 |
| 5.4.13 | B 型卡复位 (Cmd = O) | 130 |
| 5.4.14 | B 型卡请求 (Cmd = P) | 131 |
| 5.4.15 | B 型卡防碰撞 (Cmd = Q) | 132 |
| 5.4.16 | B 型卡修改传输属性 (Cmd = R) | 132 |
| 5.4.17 | B 型卡挂起 (Cmd = S) | 133 |
| 5.5 | PLUS CPU 卡类命令 (CmdClass = 0x07) | 135 |
| 5.5.1 | SL0 个人化更新数据 (Cmd = B) | 135 |

| | | |
|--------|--|-----|
| 5.5.2 | SL0 提交个人化 (Cmd = C) | 136 |
| 5.5.3 | SL3 首次验证 (直接密钥验证) (Cmd = J) | 137 |
| 5.5.4 | SL3 首次验证 (E ² 密钥验证) (Cmd = K) | 138 |
| 5.5.5 | SL3 跟随验证 (直接密钥验证) (Cmd = L) | 139 |
| 5.5.6 | SL3 跟随验证 (E ² 密钥验证) (Cmd = M) | 139 |
| 5.5.7 | SL3 复位验证 (Cmd = N) | 140 |
| 5.5.8 | SL3 读数据块 (Cmd = O) | 141 |
| 5.5.9 | SL3 写数据块 (Cmd = P) | 142 |
| 5.5.10 | SL3 值块操作 (Cmd = S) | 143 |
| 6. | 免责声明 | 145 |

1. 功能简介

1.1 功能特点

- 符合 ISO14443A、ISO14443B、ISO7816-3 标准；
- 集成 TypeB、Mifare UltraLight、Mifare1 S50/S70、PLUS CPU、SAM 卡的操作命令；
- 提供 ISO14443-4 的半双工块传输协议接口，可方便支持符合 ISO14443-4A 的 CPU 卡及符合 ISO14443-4B 的 TypeB 卡片；
- 可接双天线，能识别是哪一个天线上有卡；
- 支持串口、I²C 两种通信接口，其中串口可支持 UART、RS-232C、RS-485 方式；
- 可主动检测卡进入，检测到卡时可产生中断并且通过串口、I²C 输出数据；
- 硬件接口和通信协议完全兼容早期的 ZLG522S 系列模块；
- 可以连接多从机，从机数最多可达 127 个。

1.2 技术参数

表 1.1 ZLG600S 系列技术参数表

| | |
|--------|---|
| 产品型号 | ZLG600S 系列(ZLG600SP-T2、ZLG600SP-T4、ZLG600SP-LT2、ZLG600SP-LT4、ZLG600SP-LT、ZLG600SP/T、ZLG600SP-L ZLG600SP) |
| 功率消耗 | 平均电流：5V 直流供电/77mA 峰值电流：小于 150mA |
| 工作频率 | 13.56MHz |
| 读卡距离 | TypeA 卡：7.5cm TypeB 卡：4cm PLUS CPU 卡：5cm |
| 对外接口 | I ² C、UART、RS-232C、RS-485 |
| 数据传输速率 | I ² C：300K UART/RS-232C/RS-485：9600~230400bit/s |
| 支持卡类型 | 接触式：SAM 卡 非接触式：Mifare 1 S50、Mifare 1 S70、Mifare UltraLight、Mifare Desfire、PLUS CPU 卡、符合 ISO14443A 的逻辑加密卡和 CPU 卡、符合 ISO14443B 的卡片 |
| 物理特性 | 尺寸：天线一体化 54mm×34.5mm×6mm 分体式 25mm×16mm×6mm |
| 环境 | 工作温度：摄氏-20~85 度 湿度：相对湿度 5%~95% |

1.3 极限参数

表 1.2 极限参数表

| 符号 | 参数 | 最小 | 最大 | 单位 |
|------|-------------------|------|------|----|
| Top | 工作温度 | -25 | +80 | ℃ |
| Tstg | 存储温度 | -40 | +85 | ℃ |
| Vn1 | J1 任意管脚对 GND 电压 | -0.3 | +5.5 | V |
| Iol1 | J1 的 I/O 口低电平输入电流 | — | 20 | mA |

1.4 直流参数

温度范围：-25℃~+80℃。

表 1.3 直流参数表

| 符号 | 参数 | 条件 | 最小 | 典型 | 最大 | 单位 |
|-------|------------|----------------------------|------|-----|------|----|
| Icc51 | 电源电流, 正常工作 | Vcc=5V, 上电后或运行 config()后 | — | 77 | 150 | mA |
| Icc52 | 电源电流, 休眠模式 | Vcc=5V, 运行 close()后 | — | 15 | 20 | mA |
| Icc31 | 电源电流, 正常工作 | Vcc=3.3V, 上电后或运行 config()后 | — | 72 | 150 | mA |
| Icc32 | 电源电流, 休眠模式 | Vcc=3.3V, 运行 close()后 | — | 15 | 20 | mA |
| Vil1 | 输入低电压 | 仅 SCL、SDA | — | — | 1.5 | V |
| Vil2 | 输入低电压 | 除 SCL、SDA | — | — | 0.99 | V |
| Vih | 输入高电压 | — | 2.31 | — | 5.5 | V |
| Vol | 低电平输出电压 | Iol=20mA | — | 0.6 | 1.0 | V |
| | | Iol=3.2mA | — | 0.2 | 0.3 | |
| Voh | 高电平输出电压 | 仅 J1.1、J1.2、J1.3 | — | 开漏 | — | V |
| | | 除 J1.1、J1.2、J1.3, Ioh=20μA | 3 | 3.3 | — | |
| Iil | 逻辑 0 输入电流 | Vin=0.4V | — | — | -80 | μA |

1.5 订购信息

表 1.4 订购信息表

| 型号 | 供电电源 | 接口 | 备注 | 可替换的 ZLG522 模块型号 |
|--------------|------|-------------------------------|-------|------------------|
| ZLG600SP/T | 5V | I ² C、UART | 天线一体化 | ZLG522S/T |
| ZLG600SP-L | 3.3V | I ² C、UART | 天线分体式 | ZLG522S/L |
| ZLG600SP | 5V | I ² C、UART | 天线分体式 | ZLG522S |
| ZLG600SP-T2 | 5V | I ² C、UART、RS-232C | 天线一体化 | ZLG522S/T+ |
| ZLG600SP-T4 | 5V | I ² C、UART、RS-485 | 天线一体化 | 无 |
| ZLG600SP-LT2 | 3.3V | I ² C、UART、RS-232C | 天线一体化 | ZLG522S/LT+ |
| ZLG600SP-LT4 | 3.3V | I ² C、UART、RS-485 | 天线一体化 | 无 |
| ZLG600SP-LT | 3.3V | I ² C、UART | 天线一体化 | ZLG522S/LT |

1.6 封装信息

单位：mm。

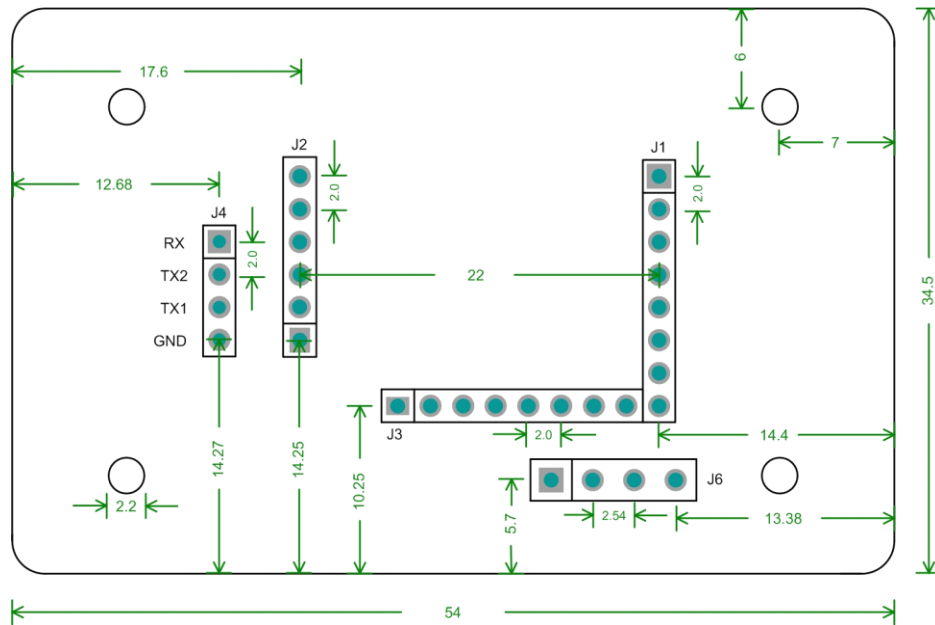


图 1.1 ZLG600SP/T 尺寸图

2.0mm 排针接口 J1、J2、J3、J4 焊盘孔径为：0.7mm；2.54mm 排针接口 J6 焊盘孔径为：0.9mm。

ZLG600SP 即是由 ZLG600SP/T 中间 J1、J2、J3 所围成的区域组成，外边的部分单独为天线部分，其尺寸可以参考图 1.1

1.7 产品图片

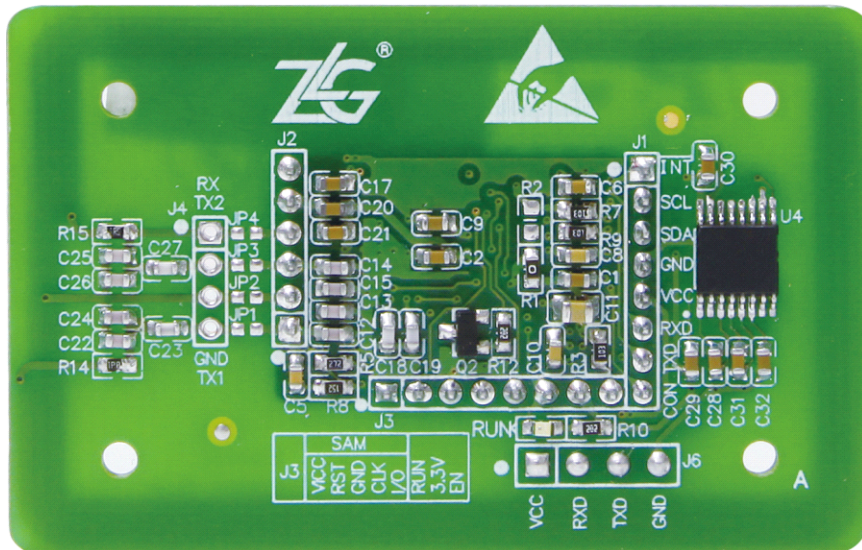


图 1.2 ZLG600SP/T 正面图片

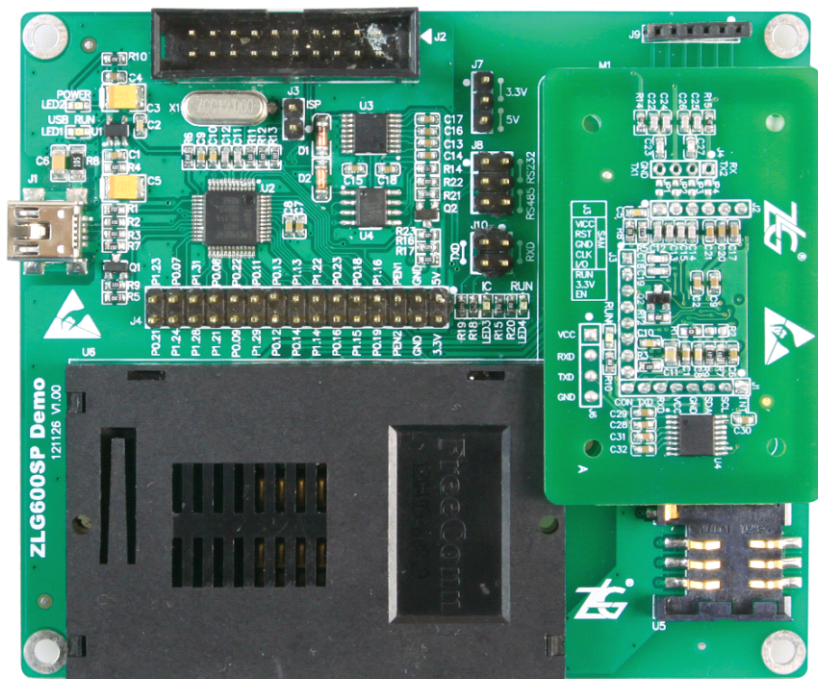


图 1.3 ZLG600SP Demo 正面图片

2. 操作说明

2.1 I/O 设置

2.1.1 通信接口设置 (J1)

通信接口为 J1，本模块有两种不可同时使用的通信接口：UART 和 I²C 接口。上电时模块根据 J1 通信管脚不同的电平状态进入不同的通信模式：自动侦测模式、UART 通信模式、I²C 通信模式。下面详细描述。

1. 自动侦测模式

上电时，若 J1 接口除电源外的其它管脚均为高电平，则模块进入自动侦测模式。进入此模式后，先判断模块内部有没有已经配置好的工作模式（比如模块地址、接口工作模式、波特率等），如果有相应的配置，则进入相应工作模式；如果内部没有已经配置好的工作模式，则 UART、I²C 两种接口都处于接收状态，只要其中一个接口先收到有效数据，模块将肯定以此方式与外界通信，并且关闭另外一种接口，如表 2.1 所示。模块出厂状态内部没有设置工作模式，进入自动侦测后，UART、I²C 将进行仲裁，并且模块默认地址为 0xB2。

表 2.1 自动侦测模式 J1 管脚定义

| 管脚 | 符号 | 类型 | 上电状态 | 描述 |
|------|------|-------|------|---|
| J1-1 | /INT | 输出 | 高电平 | 中断输出管脚，集电极开路；I ² C 通信模式时，命令执行完毕后，此管脚输出低电平；当响应自动检测命令，且使能中断输出时，当检测到卡片，此管脚输出低电平 |
| J1-2 | SCL | 输入 | 高电平 | I ² C 时钟输入管脚，开漏状态，内部无上拉 |
| J1-3 | SDA | 输入/输出 | 高电平 | I ² C 数据输入/输出管脚，开漏状态，内部无上拉 |
| J1-4 | GND | PWR | — | 电源负端 |
| J1-5 | VCC | PWR | — | 电源正端，5V |
| J1-6 | RXD | 输入 | 高电平 | UART 接收端 |
| J1-7 | TXD | 输出 | 高电平 | UART 发送端 |
| J1-8 | CON | 输入 | 高电平 | RS-485 通信时的控制管脚（0：输入；1：输出） |

注：方形焊盘为第 1 管脚。J1-2、J1-3 需要通过外部接上拉电阻。

2. UART 通信模式

上电时，若 J1-2 为低电平，则模块进入 UART 通信模式，波特率由 J1-3 管脚设置，如表 2.2 所示。

表 2.2 UART 模式 J1 管脚定义

| 管脚 | 符号 | 类型 | 上电状态 | 描述 |
|------|------|----|------|---|
| J1-1 | /INT | 输出 | — | 中断输出管脚，集电极开路；当响应自动检测命令，且使能中断输出时，当检测到卡片，此管脚输出低电平 |
| J1-2 | SBR | 输入 | 低电平 | 设置模块上电进入 UART 模式 |
| J1-3 | DBR | 输入 | — | UART 波特率设置输入管脚（当 SBR=0 时才有效） DBR=0：使用上次通信设置的波特率 DBR=1：默认波特率为 19200bit/s |

续上表

| 管脚 | 符号 | 类型 | 上电状态 | 描述 |
|------|-----|-----|------|----------------------------|
| J1-4 | GND | PWR | — | 电源负端 |
| J1-5 | VCC | PWR | — | 电源正端，5V |
| J1-6 | RXD | 输入 | 高电平 | UART 接收端 |
| J1-7 | TXD | 输出 | 高电平 | UART 发送端 |
| J1-8 | CON | 输入 | 高电平 | RS-485 通信时的控制管脚（0：输入；1：输出） |

注：方形焊盘为第 1 管脚。

3. I²C 通信模式

上电时，若 J1-6~J1-8 中任意管脚为低电平，则模块进入 I²C 通信模式，通信地址由这 3 个管脚共同决定，如表 2.3 所示。

表 2.3 I²C 模式 J1 管脚定义

| 管脚 | 符号 | 类型 | 上电状态 | 描述 |
|------|------|-------|------|--|
| J1-1 | /INT | 输出 | — | 中断输出管脚，集电极开路；I ² C 通信模式时，命令执行完毕后，此管脚输出低电平 |
| J1-2 | SCL | 输入 | 高电平 | I ² C 时钟输入，开漏状态，内部无上拉 |
| J1-3 | SDA | 输入/输出 | 高电平 | I ² C 数据输入/输出，开漏状态，内部无上拉 |
| J1-4 | GND | PWR | — | 电源负端 |
| J1-5 | VCC | PWR | — | 电源正端，5V |
| J1-6 | ADR1 | 输入 | — | I ² C 地址设置脚 1，上电时若为低电平，则模块使用 I ² C 通信，地址由 SADR 决定 |
| J1-7 | ADR0 | 输入 | — | I ² C 地址设置脚 0，上电时若为低电平，则模块使用 I ² C 通信，地址由 SADR 决定 |
| J1-8 | SADR | 输入 | — | I ² C 地址设置使能脚 SADR=0：使用 ADR0、ADR1 设置的地址进行通信，地址为 1 0 1 1 0 ADR1 ADR0 1B（LSB） SADR=1：使用默认的地址 0xB2 |

注：方形焊盘为第 1 管脚。J1-2、J1-3 需要通过外部接上拉电阻。

2.1.2 天线接口设置（J2）

表 2.4 天线接口 J2 管脚定义

| 管脚 | 符号 | 类型 | 描述 |
|------|-----|----|--|
| J2-1 | TX1 | 输出 | 天线输出驱动 1 |
| J2-2 | GND | 地 | 天线地 |
| J2-3 | TX2 | 输出 | 天线输出驱动 2 |
| J2-4 | RX | 输入 | 在双天线应用中，需要将该脚与 TX2 短接；TX1、TX2 同时驱动一个天线时，该脚不能与 TX2 短接 |
| J2-5 | GND | 地 | 天线地 |
| J2-6 | NC | — | 悬空脚 |

注：方形焊盘为第 1 管脚。

2.1.3 接触式 IC 卡及电源控制接口设置（J3）

产品用户手册

©2012 Guangzhou ZHIYUAN Electronics Stock Co., Ltd.

表 2.5 接触式 IC 卡及电源控制接口 J3 管脚定义

| 管脚 | 符号 | 类型 | 描述 |
|------|---------|-----------|---|
| J3-1 | SAM_VCC | PWR | 接触式 IC 卡的电源正端 |
| J3-2 | SAM_RST | 输出 | 接触式 IC 卡控制的 RST 管脚 |
| J3-3 | SAM_GND | PWR | 接触式 IC 卡的电源负端 |
| J3-4 | SAM_CLK | 输出 | 接触式 IC 卡控制的 CLK 管脚 |
| J3-5 | SAM_I/O | 输入/ 输出 | 接触式 IC 卡控制的数据输入/输出管脚 |
| J3-6 | RUN | 输入 | 通信/运行指示，低电平有效 |
| J3-7 | 3.3V | 输出 | 电源 3.3V 输出端，可提供 100mA 电流输出 |
| J3-8 | EN | 输入 | 电源控制管脚，内部自带 10K 电阻上拉到 VCC，置低时，将关闭整个模块电源 |

注：方形焊盘为第 1 管脚。可以通过控制 EN 管脚的电平来控制整个模块的电源通断。

2.1.4 RS-232C/RS-485 接口设置（J6）

该管脚是 RS-232C、RS-485 的共用接口，可以通过焊接模块的 U4、U5 来切换（U4：RS-232C、U5：RS-485），这两个芯片不能同时使用，否则将无法通信，甚至烧坏芯片。

表 2.6 RS-232C/RS-485 接口 J6 管脚定义

| 管脚 | 符号 | 类型 | 描述 |
|------|-----|-----|-------------------------|
| J6-1 | VCC | PWR | 电源正端，与 J1-5 连接，5V |
| J6-2 | RXD | 输入 | RS-232C 接收管脚、RS-485B 管脚 |
| J6-3 | TXD | 输出 | RS-232C 发送管脚、RS-485A 管脚 |
| J6-4 | GND | PWR | 电源负端 |

注：方形焊盘为第 1 管脚。当 J6 用作 RS-485 功能时，同时还应通过控制 J1-8 的 CON 管脚控制模块 RS-485 的发送、接收（0：接收；1：发送）。

2.2 典型应用

2.2.1 UART 接口应用

利用模块的 J1.6 和 J1.7 接口可以与主机进行 UART 通信，只要主机提供一个 UART 接口即可。

1. 自动侦测模式 UART 应用

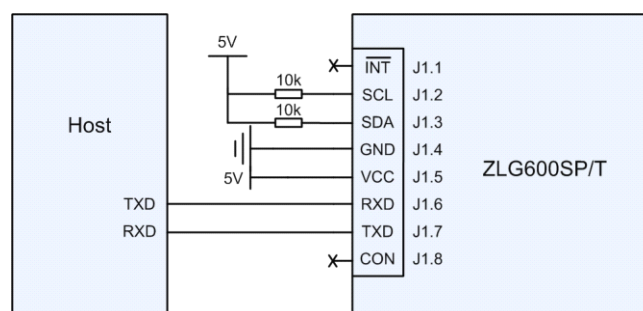


图 2.1 自动侦测模式 UART 典型应用

如图 2.1 所示，该应用中主机只需要提供 UART 接口与模块连接，模块其他通信接口接上拉即可。该应用中模块上电后需要执行波特率检测后才能执行主机命令，其中执行波特率检测是连续发送两次 0x20，模块将确定通信波特率，并回应 0x06，若不先进行这一步操作，模块不响应任何主机发送的命令。

2. UART 模式默认波特率应用

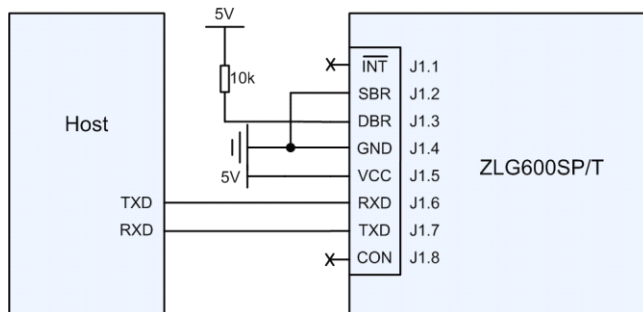


图 2.2 UART 模式使用默认波特率典型应用

如图 2.2 所示，该应用中主机只需要提供 UART 接口与模块连接，波特率为 19200，模块的 J1.2 (SBR) 应接到地，J1.3 (DBR) 接上拉。通信过程中可以用命令修改通信波特率。

3. UART 模式使用上次波特率应用

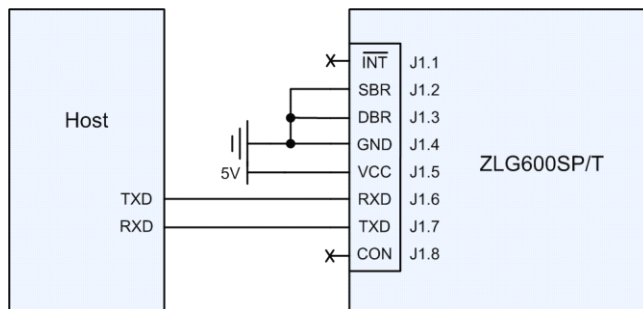


图 2.3 UART 模式使用上次波特率典型应用

如图 2.3 所示，该应用中主机只需要提供 UART 接口与模块连接，模块的 J1.2 (SBR)、J1.3 (DBR) 应接到地，此时模块使用掉电前一次的波特率进行通信，通信过程中可以用命令修改通信波特率，设置成功后的波特率掉电不丢失，直到再次使用波特率设置命令修改。出厂的默认波特率为 19200。

2.2.2 I²C 接口应用

利用模块的 J1.1~J1.3 接口可以与主机进行 I²C 通信，只要主机提供任意三个 I/O 口即可，注意本模块管脚内部并没有接上拉，实际应用中，用户应当外部接上拉。

1. 自动侦测模式 I²C 应用

如图 2.4 所示，该应用中主机只需要提供 I²C 接口和一个 I/O（用于检测模块的应答）与模块连接，模块其它通信接口连接上拉或者悬空（建议连接上拉）。该模式下模块 I²C 从机地址固定为 0xB2。

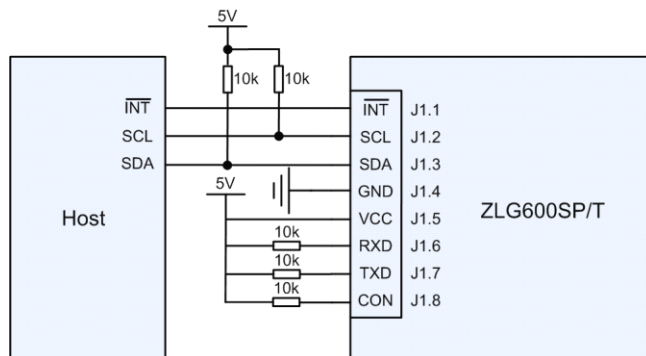


图 2.4 自动侦测 I²C 典型应用

2. I²C 模式硬件方式设置从机地址应用

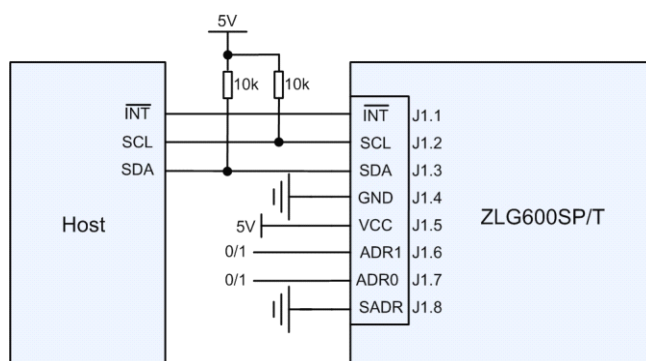


图 2.5 I²C 模式硬件方式设置从机地址典型应用

如图 2.5 所示，该应用中主机只需要提供 I²C 接口和一个检测中断的 I/O 管脚（用于检测模块的应答）与模块连接，模块的 J1.8 接地，通过 ADR0、ADR1 配置 I²C 从机地址：1 1 ADR1 ADR0 R/W (LSB)。在该模式下可同时在 I²C 总线上连接 4 个模块，如图 2.6 所示。

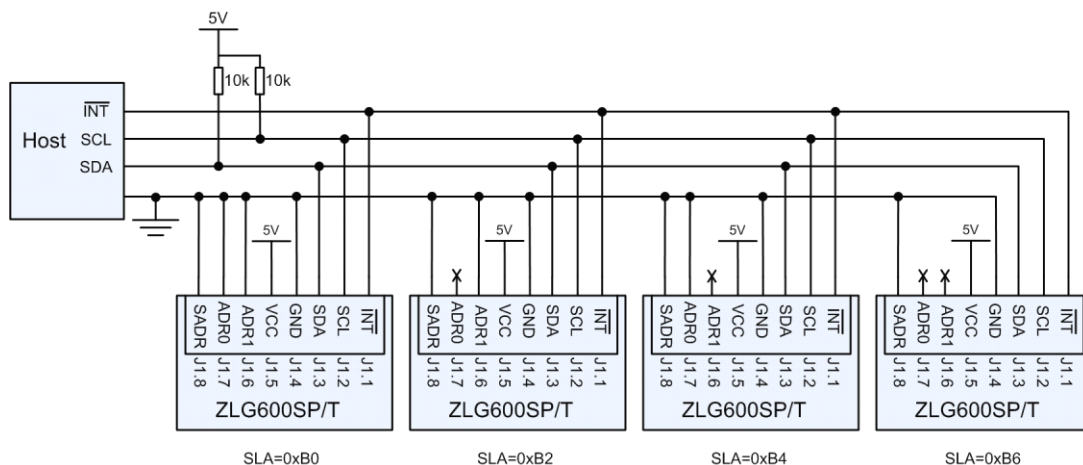


图 2.6 多模块 I²C 连接典型应用

2.2.3 RS-232C/RS-485 接口的应用

利用 ZLG600SP/T 模块的 J6 管脚，可以实现 RS-232C、RS-485 通信。如图 2.7 所示，J6 可以直接与 PC 机的 RS-232C 接口连接进行通信，必需先焊接 U4 芯片，此时也可以将 J1.2（SBR）接地来实现固定波特率通信；当使用 RS-485 功能时，必需取下 U4 芯片，焊接 U5

芯片，同时 J1.8 (CON) 作为 RS-485 发送接收的控制管脚。

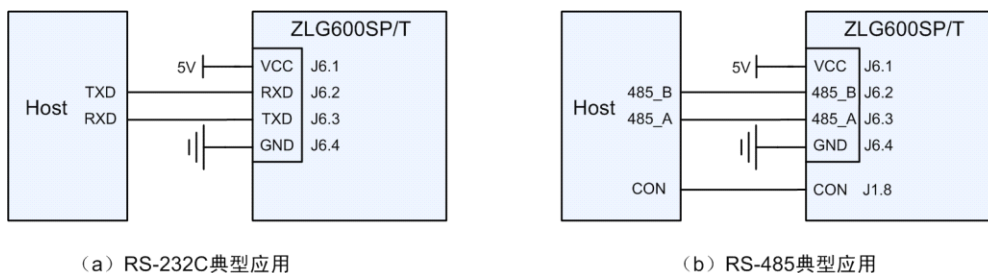


图 2.7 RS-232C/RS-485 典型应用图

2.2.4 多从机方案应用

前面章节提到在 I²C 模式下，通过硬件方式来设定模块的地址，此方式最多只能设置四个从机 (0xB0 0xB2 0xB4 0xB6)，局限性较大。为了适应 I²C 或 RS-485 多从机的应用，模块内部的地址是可以通过命令进行设置的，模块地址由一个字节组成，最低位是读写位，符合 I²C 地址格式，所以最多可以设置 127 个从机 (其中 0x00 不可用)，实际能连接从机数与使用的 RS-485 收发器等外界因素有关。

不管是 I²C 模式还是 RS-485 模式，首先，模块应先通过命令进行配置，主要是配置模块的工作模式、模块地址、波特率等信息，该信息掉电不丢失，且工作模式设置后必需重上电后才生效；其次，模块上电应设置成自动侦检模式，即是上电时 J1 管脚除电源外，其它管脚都为高电平，由于 J1 的通信管脚都是由 MCU 直接引出，其中 J1-1、J1-6、J1-7、J1-8 管脚在模块上电初始化时设置成了上拉状态，所以是高电平，外部可以不再接上拉电阻，但 J1-2、J1-3 管脚比较特殊，只有开漏模式，所以外部必需接上拉电阻，如图 2.8 所示；只有严格通过前面两步的操作，各模块才能连接在一起，形成多从机的方案应用。

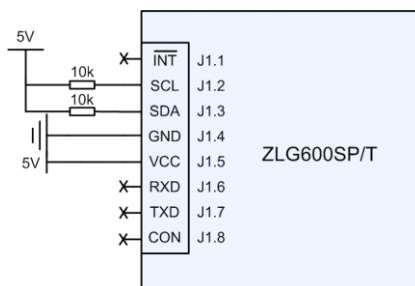


图 2.8 上电进入自动侦检的硬件设置

1. I²C 软件方式多从机应用

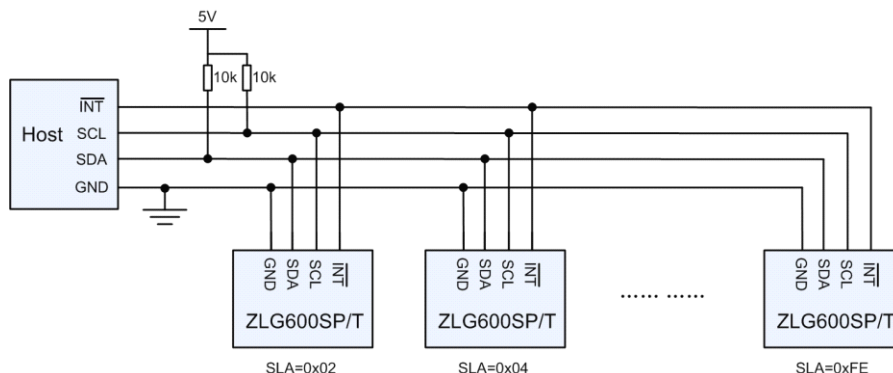


图 2.9 I²C 多从机连接示意图

图 2.9 是 I²C 多从机的连接示意图，其中从机的地址可以设置为 0x02~0xFE 共 127 种。在连接之前，每个从机模块应先通过命令配置成 I²C 工作模式，模块地址设置为自己想要的地址，命令的配置参考“设备控制类命令→设置工作模式”的说明；上电时，应保证 J1 各管脚都是高电平状态，应如图 2.8 把 J1-2、J1-3 接上拉。

注：在 I²C 通信方式下，新帧格式、旧帧格式都支持多从机连接，两种帧格式说明参照后文所述。

2. RS-485 软件方式多从机应用

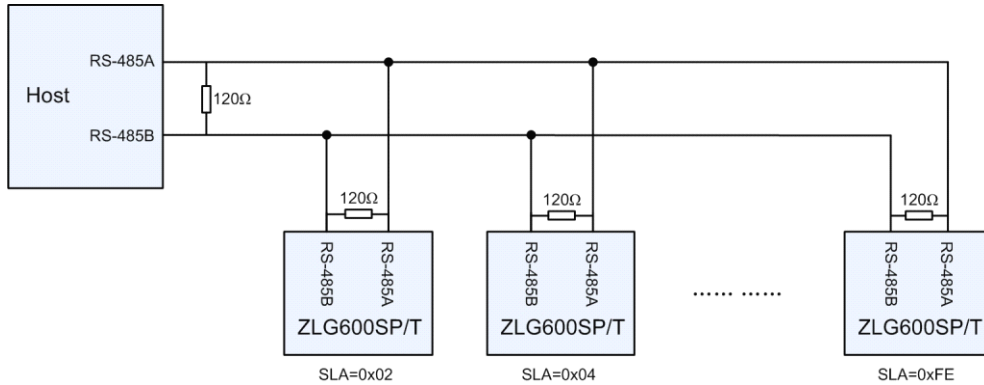


图 2.10 RS-485 多从机连接示意图

图 2.10 是 RS-485 多从机的连接示意图，与 I²C 通信方式相同，从机地址可以设置为 0x02~0xFE 共 127 种。在连接之前，每个从机模块应先通过命令配置成 UART 工作模式，模块地址设置为自己想要的地址，命令的配置参考“设备控制类命令→设置工作模式”的说明；上电时，应保证 J1 各管脚都是高电平状态，应如图 2.8 把 J1-2、J1-3 接上拉；另外，需要在模块上焊接 U5（RS-485 电平转换）芯片，通过 J6 与 RS-485 主机通信。

注：在 RS-485 通信方式下，只有新帧格式支持多从机连接，旧帧格式不支持多从机连接。

2.2.5 双天线应用

本模块可以同时连接两个天线，读写卡时可以识别是哪个天线上有卡。硬件连接如图 2.11 所示，J2 口连接两个天线，两个天线分别由 TX1、TX2 驱动。

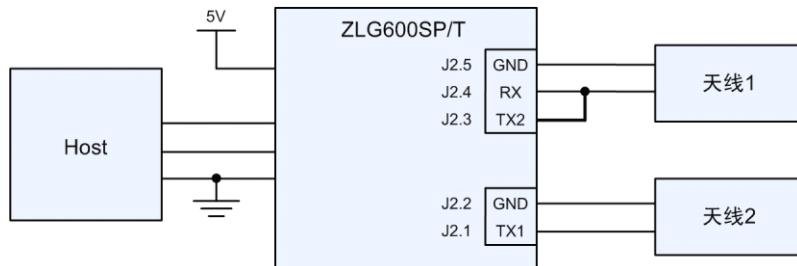


图 2.11 双天线典型应用图

在硬件连接方面，除了 J2.3（TX2）、J2.4（RX）要短接之外，JP1~JP4 应视情况连接或断开，TX1、TX2 两端的接收回路器件都应焊接。另外，天线的连接也可以考虑使用同轴电缆方式，这不在本指南的解释范围。软件上，应使用设备控制类命令中的“设置 IC 卡接口的寄存器值”、“设置 IC 卡接口的输出”等命令进行 TX1、TX2 管脚驱动的切换。

3. 通讯协议

本模块有两种不可同时使用的通信接口：串口和 I²C 接口。外部与模块通过这两种接口通信，必需按照规定的协议进行，为了兼容早期 ZLG522S 系列模块的同时提供更高质量的通信，ZLG600SP/T 制定了两种协议：旧帧格式、新帧格式，其中旧帧格式完全兼容 ZLG522S 系列模块、新帧格式拥有更高质量的通信。两种帧格式不能互相替代，在同一时间只能使用其中一种，过程中可以通过命令进行新旧帧通信的切换，出厂默认为旧帧格式。

本模块以命令——响应方式工作，在系统中模块属于从属地位，不会主动发送数据（响应自动检测卡命令除外），通常主机首先发出命令，然后等待模块响应。

3.1 各通信模式时序

前面章节已经详细描述了进入不同通信模式的方法，本小节描述各通信模式下的时序情况。

3.1.1 自动侦测模式时序

该模式完全兼容 ZLG522S 系列模块的通信接口模式，该模式下模块可以直接替换 ZLG522S 系列模块。上电后，若模块从 UART 通信线上检测到有效的波特率，则模块使用 UART 通信；若模块从 I²C 总线上收到 SLA（该模式下 SLA 为 0xB2），则模块使用 I²C 通信。

在自动侦测模式下，要进入 UART 模式，必需先发送两次 0x20，其时间间隔为 t_{BRC} ，如图 3.1 所示。

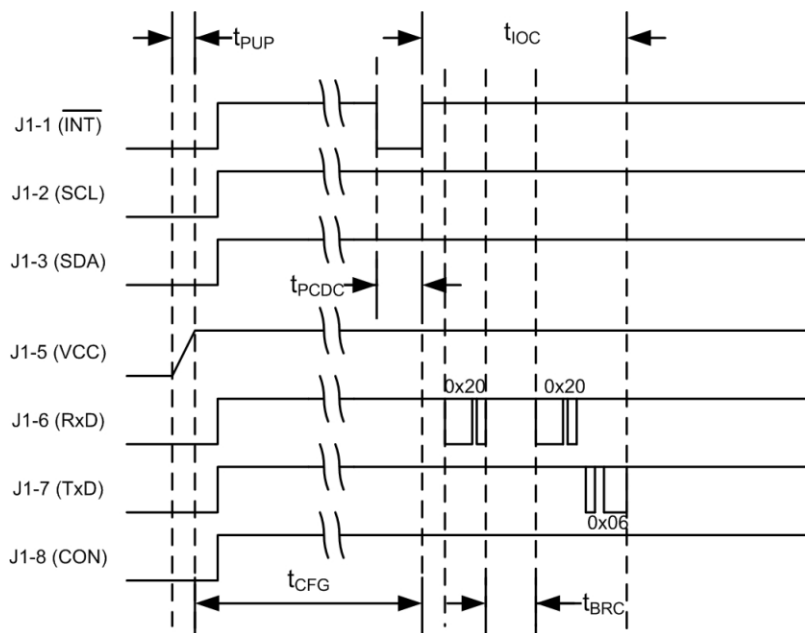


图 3.1 自动侦测模式下 UART 通信示意图

在自动侦测模式下，要进入 I²C 模式，首先会接收到 J1-2、J1-3 管脚上的 I²C 信号，并且该数据的传送地址与模块地址一至，将进入 I²C 模式，发送完命令帧后发送回应帧的间隔为 t_{CMD} ，如图 3.2 所示。

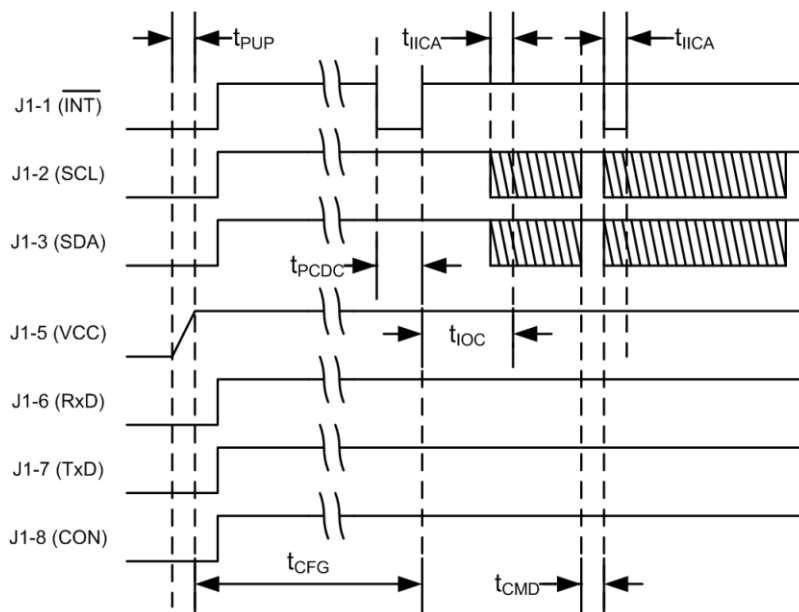


图 3.2 自动侦测模式下 I²C 通信示意图

3.1.2 UART 模式时序

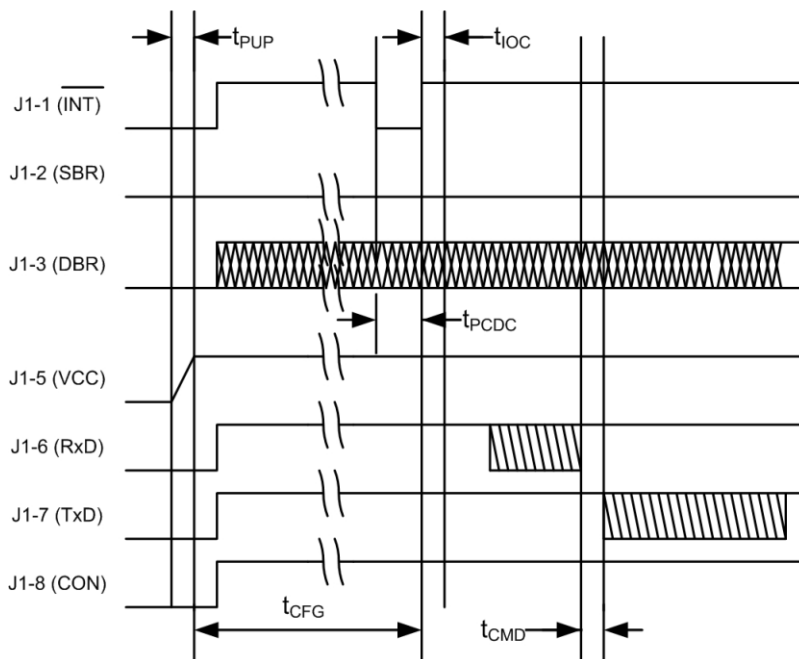


图 3.3 UART 模式通信示意图

当模块上电设定为 UART 模式时，模块不必像自动侦检模式时进入 UART 模式那样先发两次 0x20，上电设定为 UART 模式的情况下，可以直接响应命令帧，命令帧与回应帧的间隔为 t_{CMD} ，如图 3.3 所示。

3.1.3 I²C 模式时序

当模块上电设定为 I²C 模式时，与自动侦测模式下的 I²C 模式情况一样。

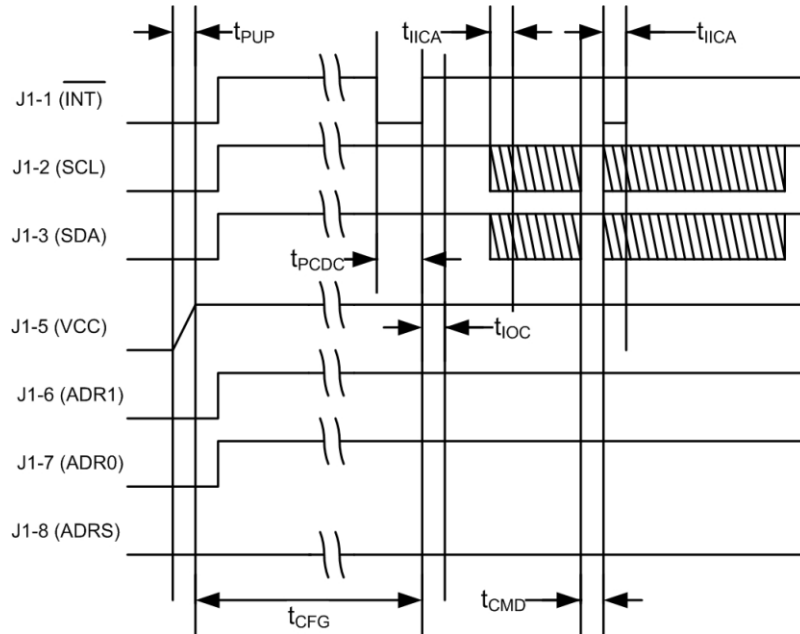


图 3.4 I²C 模式通信示意图

3.1.4 模块上电时序参数

模块上电初始化包括 t_{PUP} 、 t_{CFG} 和 t_{IOC} 三个阶段，在 UART/I²C 接口模式 t_{IOC} 的时间很短。建议在上电前就将模块的通信模式配置好，然后再上电/复位；模块上电后等待 100ms 才向模块发命令或者是执行波特率自适应流程。

表 3.1 模块上电各时序参数表

| 符号 | 定义 | 条件 | 最小 | 典型 | 最大 | 单位 |
|------------|----------------------------|----------------------------|-------|-------|----|----|
| t_{PUP} | 电源电压从 0V 上升到 0.7VCC 的时间 | — | — | — | 1 | ms |
| t_{CFG} | 模块上电初始化时间 | — | — | 3.643 | — | ms |
| t_{PCDC} | 模块读写芯片配置时间 | — | — | 2.478 | — | ms |
| t_{IOC} | 模块通信模式检测设置时间 | 自动侦测接口模式 | 20 | — | — | ms |
| | | UART/I ² C 接口模式 | — | 20 | — | |
| t_{BRC} | 波特率自适应模式下，2 次发送 0x20 的时间间隔 | — | 30 | 958 | — | us |
| t_{CMD} | 命令执行时间 | — | 200us | 5ms | — | — |

3.2 UART 通信设置

UART 接口的数据格式为：1 个起始位、8 个数据位、无奇偶校验位、1 个停止位。ZLG600SP/T 复位初始化之后，立即检测通信 I/O 口设置，若是自动侦测接口模式则进入波特率检测流程，直至检测到一个可用的波特率或收到从 I²C 接口发来的有效数据。因此，自动侦测接口模式，主机在与之进行 UART 通信之前必须主动执行波特率设置程序。

3.2.1 波特率检测

波特率检测一次有效，也即检测到一个主机发来的设置波特率，则固定以这个波特率与主机通信，此后若主机再企图尝试通过波特率检测的方式来设置另一个波特率，则

ZLG600SP/T 对这个设置不予理睬。除非 ZLG600SP/T 发生了复位，则主机可以设置另一个波特率。ZLG600SP/T 可以设置的波特率有 9600、19200、28800、38400、57600、115200、172800、230400。主机设置好自己波特率后，便可对 ZLG600SP/T 进行波特率设置，流程如图 3.5 所示。

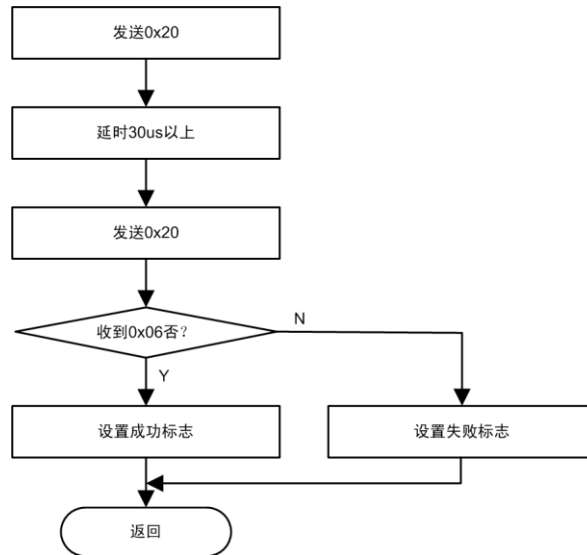


图 3.5 设置波特率流程

若波特率设置成功，主机便可发送命令，待模块执行命令完成，可通过 UART 主动发回响应。

3.2.2 UART 数据通信

UART 通信模式下，主机向模块发送命令，模块收到命令后解析并执行，执行完毕后主动将数据发送给主机。若命令错误，则模块直接丢弃接收到的数据，且不做任何回应。

3.3 I²C 设置

300KHz 的硬件 I²C 接口，从机地址可设为：0xB0、0xB2、0xB4 和 0xB6。若采用自动侦测接口模式，则从机地址只能设置为 0xB2。只要从机地址正确，速率合适，则模块可有效地收到主机命令，待模块执行命令完成，会输出中断信号（/INT 脚变低），通知主机读取响应。

3.4 旧帧格式

为了兼容早期的 ZLG522S 系列模块，ZLG600SP/T 的所有通信有两套帧格式，其中，旧帧格式完全兼容 ZLG522S 系列模块；新帧格式提供了更高质量的通信。两种帧格式不能替代，同一时间只能使用其中一种进行通信，过程中可以通过命令进行切换。

旧帧格式不分通信接口，都采用相同的数据格式。

3.4.1 旧命令帧格式

命令帧是外部主机为了使模块执行不同功能任务而向模块发送的一串数据。命令帧总是以一帧为单位进行通信，不足一帧的数据无效，连续多个命令帧时，模块只响应最先发送的命令帧，等到执行完该命令帧并向主机发送完回应帧后才继续等待新的命令帧。该旧命令帧数据结构如表 3.2 所示。

表 3.2 旧命令帧数据结构

| 帧长 FrameLen | 命令类型 CmdType | 命令 Cmd | 信息长度 Length | 信息 Info | 校验和 BCC | 帧结束符 ETX |
|----------------|-----------------|-----------|----------------|------------|------------|-------------|
| 1byte | 1byte | 1byte | 1byte | Nbyte | 1byte | 1byte |

表 3.3 旧命令帧各字段说明表

| 字段 | 长度 | 说明 | 备注 |
|-----------------|--------|---|-----------------|
| 帧长 FrameLen | 1 | 命令帧的长度，包括它自己 | — |
| 命令类型 CmdType | 1 | 命令类型 0x01: 设备控制类命令，如读写 I/O、读写寄存器等 0x02: Mifare 卡类命令（包括 US14443-3A） 0x05: ISO7816-3 类命令 0x06: ISO14443（PICC）类命令 0x07: PLUS CPU 卡类命令 其它值保留 从机返回相同的 CmdType | 该字段主机发送和接收的应该相同 |
| 命令 Cmd | 1 | 命令代码，代表执行一类命令里的不同功能，详见后面章节每类命令的描述 | — |
| 信息长度 Length | 1 | 该帧所带信息数据的长度 | — |
| 信息 Info | Length | 数据信息 | — |
| 校验和 BCC | 1 | 校验和，从 FrameLen 开始到 Info 的最后一字节异或取反，C 语言程序描述如下：（SerBfr 为一帧数据缓冲区首址） BCC = 0; for (i = 0; i < (SerBfr[0] - 2); i++) { BCC ^= SerBfr[i]; } SerBfr[SerBfr[0]-2] = ~BCC; | — |
| 帧结束符 ETX | 1 | 0x03: “End of Text” 标准的控制字符，是一帧的结束标志 | — |

3.4.2 旧回应帧格式

表 3.4 旧回应帧数据结构

| 帧长 FrameLen | 命令类型 CmdType | 状态 Status | 信息长度 Length | 信息 Info | 校验和 BCC | 帧结束符 ETX |
|----------------|-----------------|--------------|----------------|------------|------------|-------------|
| 1byte | 1byte | 1byte | 1byte | Nbyte | 1byte | 1byte |

表 3.5 旧回应帧各字段说明表

| 字段 | 长度 | 说明 | 备注 |
|-----------------|--------|--|-----------------|
| 帧长 FrameLen | 1 | 同命令帧 | — |
| 命令类型 CmdType | 1 | 同命令帧 | 该字段主机发送和接收的应该相同 |
| 状态 Status | 1 | 0x00——成功；其它——失败 | — |
| 信息长度 Length | 1 | 该帧所带信息数据的长度 如果 Status 不为 0 时，Length 为 0 | — |
| 信息 Info | Length | 同命令帧 | — |
| 校验和 BCC | 1 | 同命令帧 | — |
| 帧结束符 ETX | 1 | 同命令帧 | — |

使用旧帧格式，必需满足如下规则。

- 无论何时，如果接收方在 4.44 毫秒内未接收到一个字节，则表示下一个接收的字节为一帧的开始，即准备接收的是帧长（FrameLen）；
- 一帧的结束一定是 0x03，但接收到 0x03 则不一定是帧结束；
- 帧长必须不小于 6 字节，最大不能超过 70 字节，且帧长必须等于信息长度加 6；
- BCC 计算必需正确；
- 无论是主机还是从机所接收的数据必须符合以上规则，否则从机不会执行任何命令，也不会有任何错误响应，主机也必须丢弃这帧数据，以找出错误原因，从而纠正错误。

3.5 新帧格式

新帧格式根据 I²C、UART 两种不同的通信接口，在使用上稍有不同，新帧格式的通信协议分为 2 层，分别如下。

- 物理链路层
- 协议层

其中，I²C 通信接口必需符合这两层定义，而 UART 通信接口忽略物理链路层，只需按照协议层即可。

3.5.1 新帧格式物理链路层

物理链路层是基于 I²C 的有器件子地址模式，器件地址固定为 0xB2，器件子地址为 2 字节。存储/数据交互空间为 542 字节，其中前 256 字节为保留使用，后 286 字节为命令帧/回应帧使用。详细描述见表 3.6 所示。

表 3.6 模块存储空间分配

| 子地址 | 意义 | 备注 |
|---------------|-----------|---|
| 0x0000~0x00FF | 保留使用 | — |
| 0x0100 | 保留对齐使用 | 无特殊意义，任意读写 |
| 0x0101 | 主机控制/模块状态 | <p>写入 ‘STATUS_EXECUTING’ (0x8D) 将启动模块执行地址 0x0104~0x021D 中的命令（命令在写入结束后才开始执行），写入其他值模块无动作</p> <p>读出是模块当前的状态：</p> <p>STATUS_EXECUTING (0x8D) —— 命令还未执行</p> <p>STATUS_BUSY (0x8C) —— 命令正在执行</p> <p>STATUS_IDLE (0x8A) —— 模块空闲</p> <p>其他值 —— 执行结果</p> |
| 0x0102~0x0103 | 命令/回应帧长度 | 命令/回应帧的长度（小端模式） |
| 0x0104~0x021D | 命令/回应帧 | 命令/回应帧，见表 3.7 和表 3.8 |

注意：

1. 若一次性向包含 ‘0x0101’ 地址的连续存储空间写入数据，只有写入结束后，写入 ‘0x0101’ 地址的数据才起效。
2. ‘0x0101’ 地址的值为 ‘STATUS_EXECUTING’ (0x8D) 和 ‘STATUS_BUSY’ (0x8C) 时请勿向 ‘0x0101~0x021D’ 地址写入任何数据，因为此时 ‘0x0101~0x021D’ 地址空间是被模块内部使用。向其写数据会造成不可估计的错误或异常情况。
3. ‘STATUS_IDLE’ (0x8A) 只有上电时才会自动出现。执行命令后亦不会自动恢复成 ‘STATUS_IDLE’ 状态，只会保持命令执行后的状态。若有需要，请在执行命令结束后将 ‘0x0101’ 地址的值改为 ‘STATUS_IDLE’。
4. 为了减少从机处理通信中断的次数，在命令执行期间，请勿频繁访问模块的存储空间，即使是查询命令执行的状态。在命令执行期间，建议根据实际情况，2~10ms 查询一次，或者使用模块中断输出脚（模块命令执行完毕，中断脚会输出一个低电平，该电平持续到接收到本模块的 SLA+W 或 SLA+R 为止）。
5. 只有命令帧格式有效的情况下，模块才执行命令，命令帧格式无效的情况下只会产生状态，而不会产生中断。
6. 该层不适合于 UART 通信接口，UART 通信接口忽略该层。

模块 I²C 支持的最大速率为 400Kbps，命令执行完毕中断输出脚会产生一个低电平，该电平持续到模块收到本模块的 SLA+W 或 SLA+R 为止。可以通过检查该中断来判断命令是否执行完毕；也可以查询 ‘0x0101’ 地址的值来判断模块执行的情况。命令执行完毕后可以读取 ‘0x0102~0x0103’ 的值来获取回应帧的长度，以便于确定读取回应帧的字节数。

3.5.2 新帧格式协议层

表 3.7 新命令帧数据结构

| 地址 LocalAddr | 卡槽索引 SlotIndex | 安全报文/包号 SMCSeq | 命令类型 CmdClass | 命令代码 CmdCode | 信息长度 InfoLength |
|-----------------|-------------------|-------------------|------------------|-----------------|--------------------|
| 1 字节 | 1 字节 | 1 字节 | 1 字节 | 2 字节 | 2 字节 |
| 信息 Info | | | | | 校验和 Checksum |
| n 字节 | | | | | 2 字节 |

表 3.8 新回应帧数据结构

| 地址 LocalAddr | 卡槽索引 SlotIndex | 安全报文/包号 SMCSeq | 命令类型 CmdClass | 执行状态 Status | 信息长度 InfoLength |
|-----------------|-------------------|-------------------|------------------|----------------|--------------------|
| 1 字节 | 1 字节 | 1 字节 | 1 字节 | 2 字节 | 2 字节 |
| 信息 Info | | | | | 校验和 Checksum |
| n 字节 | | | | | 2 字节 |

表 3.9 新命令帧数据结构说明

| 字段 | 长度 (字节) | 说明 | 备注 |
|------------|------------|--|--------------------|
| LocalAddr | 1 | 同 I ² C 地址模式相同，高 7 位为本机地址，低位为方向。其中 ‘0x00’ 为通用地址 | — |
| SlotIndex | 1 | IC 卡卡槽的索引编号（本模块保留该字节，帧里面该字节填入 0x00 即可）。 | — |
| SMCSeq | 1 | 高 4 位为安全报文控制位（本模块不支持安全报文模式，即高 4 位无效）；低 4 位为该命令帧的序号。可以用来作为通信间的错误检查，从机接收到主机发来的信息，在应答信息中发出一个同样的“包号”信息，主机可以通过此信息检查是否发生的“包丢失”的错误。 | 可以为任意值 |
| CmdClass | 1 | 0x01: 设备控制类命令 0x02: Mifare S50/S70 卡类命令(包括 US114443-3A) 0x05: ISO7816-3 类命令 0x06: ISO14443 (PICC) 类命令 0x07: PLUS CPU 卡类命令 0x08: ISO15693 (VICC) 类命令 0x09: ISO18000-6C 0x0A: ISO18092(NFCIP-1) 0x0B: 二代身份证类命令 | 不同的模块，支持的命令类型是不一致的 |
| CmdCode | 2 | 命令代码 | — |
| InfoLength | 2 | 该帧所带信息的字节数 | — |
| Info | InfoLength | 数据信息 | — |
| Checksum | 2 | 校验和，从地址字节开始到信息的最后字节的累加和取反 | — |

注：“命令码”、“信息长度”和“校验和”均以小端模式存放，即低字节在前。信息长度可以为 0，即没有信息。

ZLG600SP/T 模块收到命令帧后，检测帧格式是否正确，若不正确，则丢弃当前数据，也不做任何回应；若正确则进一步进行处理。处理完毕后将处理的结果组成回应帧返回。

表 3.10 新回应帧数据结构说明

| 字段 | 长度 (字节) | 说明 | 备注 |
|------------|------------|---------------------------|----|
| LocalAddr | 1 | 高 7 位同命令帧，低位为方向 | — |
| SlotIndex | 1 | 同命令帧 | — |
| SMCSeq | 1 | 同命令帧 | — |
| CmdClass | 1 | 同命令帧 | — |
| Status | 2 | 执行状态 | — |
| InfoLength | 2 | 该帧所带信息的字节数 | — |
| Info | InfoLength | 数据信息 | — |
| Checksum | 2 | 校验和，从地址字节开始到信息的最后字节的累加和取反 | — |

注：“命令码”、“信息长度”和“校验和”均以小端模式存放，即低字节在前。信息长度可以为 0，即没有信息。

命令帧和回应帧的格式基本一致，只有“命令码”和“执行状态”之分。帧的最小长度为 10 字节(没有信息的情况)，最长理论上可以到 65545 字节。实际上没有必需。ZLG600SP/T 模块帧的最大长度为 282 字节，信息的最大长度为 272 字节，完全满足短 APDU 的处理。命令/回应帧的详细定义如程序清单 3.1 所示。

程序清单 3.1 命令帧/回应帧结构体定义

```
#define CMD_PROTOCOL_LENGTH      10          //!< 通讯帧协议字节数
#define CMD_INFO_MAX_LENGTH      272         //!< 通讯帧最大字节数。
                                           //!< 短 APDU 的字节数为 255 + 6 字节；
                                           //!< T=CL 协议长度为 5 字节；
                                           //!< ExchangeBlock()函数的头为 4 字节。
                                           //!< 升级数据包的最大长度为 16 + 256 字节

#define CMD_PACKET_MIN_SIZE      (CMD_PROTOCOL_LENGTH)  //!< 通讯帧信息最小字节数
#define CMD_PACKET_MAX_SIZE      (CMD_PROTOCOL_LENGTH + CMD_INFO_MAX_LENGTH)

//! @struct CommandFrame 命令帧结构体
typedef struct CommandFrame
{
    uint8_t      LocalAddr;                //!< 本机地址(最低位为方向位，同 I2C 地址相同)
    uint8_t      SlotIndex;                //!< 卡槽索引
    uint8_t      SMCSeq;                   //!< 安全报文/包号
    uint8_t      CmdClass;                 //!< 命令类
    union
    {
        uint16_t  CmdCode;                 //!< 命令码
        uint16_t  Status;                  //!< 命令执行状态
    };
    uint16_t      InfoLength;              //!< 信息长度
}
```

```
uint8_t      Info[CMD_INFO_MAX_LENGTH + 2];    //!< 信息和 2 字节的累加和取反校验
} CommandFrame;
```

“校验和”为从地址字节开始到信息的最后字节（若信息长度为 0，则不计算信息）的累加和取反（只取低 16 位，高 16 位丢弃），具体实现方式见程序清单 3.2。

程序清单 3.2 计算字节累加和

```
// =====
// @brief      计算字节累加和
// @param[in]   *p      -- 计算的数据
// @param[in]   nBytes  -- 字节数
// @return      字节累加和
// =====

uint32_t GetByteSum(const void *p, uint32_t nBytes)
{
    const uint8_t  *pBuf = (const uint8_t *)p;
    uint32_t       sum   = 0;

    while (nBytes--) {
        sum += *pBuf++;
    }

    return sum;
}
```

4. 旧帧格式应用命令详述

ZLG600SP 系列模块的应用命令共分为以下几类。

- 设备控制类命令；
- Mifare S50/S70 卡类命令；
- ISO7816-3 类命令；
- ISO14443 (PICC) 卡类命令；
- PLUS CPU 卡类命令；

下面的章节，将以旧帧格式进行命令的描述，如果要使用新帧，需要自行切换。

4.1 设备控制类命令（CmdClass = 0x01）

设备控制类命令汇总如表 4.1 所示。

表 4.1 设备控制类命令一览表

| 命令码 | 意义 |
|-----|-----------------------|
| ‘A’ | 读设备信息 |
| ‘B’ | 配置 IC 卡接口 |
| ‘C’ | 关闭 IC 卡接口 |
| ‘D’ | 设置 IC 卡接口协议（工作模式） |
| ‘E’ | 装载 IC 卡密钥 |
| ‘F’ | 设置 IC 卡接口的寄存器值 |
| ‘G’ | 获取 IC 卡接口的寄存器值 |
| ‘H’ | 设置波特率 |
| ‘I’ | 设置天线驱动方式 |
| ‘K’ | 设置新旧帧格式 |
| ‘U’ | 设置设备工作模式 |
| ‘V’ | 获取设备工作模式 |
| ‘a’ | 装载用户密钥 |
| ‘b’ | 读 E ² PROM |
| ‘c’ | 写 E ² PROM |

4.1.1 读设备信息（Cmd = A）

该命令能够获取模块的型号所用版本信息。

声明：*unsigned char GetDvcInfo(unsigned char *pDvcInfo)*

1. 主机命令

命令类型（CmdClass）： 0x01
 命令代码（CmdCode）： ‘A’
 信息长度（InfoLength）： 0
 信 息（Info）： none
 例 如： 获取模块的版本号

表 4.2 读设备信息命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 06 | 01 | 41 | 00 | none | B9 | 03 |

2. 从机应答

状 态（Status）： 0——成功，其它——失败
 信息长度（InfoLength）： 0x14
 信 息（Info）： ‘ZLG600SP/T V1.00’
 例 如： 获取模块信息成功后，返回模块的版本信息

表 4.3 读设备信息回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|--|-----|-----|
| 1A | 01 | 00 | 14 | 5A 4C 47 36 30 30 53 50 2F 54 20 56 31 2E 30 30 00 00 00 00 | 86 | 03 |

4.1.2 配置 IC 卡接口 (Cmd = B)

该命令配置 IC 卡的接口形式，这个命令执行后，默认为 ISO14443A 形式。

声明: `uint8_t CD_Config(CDProtType type)`

1. 主机命令

命令类型 (CmdClass): 0x01

命令代码 (CmdCode): 'B'

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 配置 IC 卡的接口形式

表 4.4 配置 IC 卡接口命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 06 | 01 | 42 | 00 | none | BA | 03 |

2. 从机应答

状 态 (Status): 0——成功，其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 配置 IC 卡接口成功后的回应

表 4.5 配置 IC 卡接口成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 01 | 00 | 00 | none | F8 | 03 |

4.1.3 关闭 IC 卡接口 (Cmd = C)

该命令关闭 IC 卡接口，执行该命令后，IC 卡相关命令将不能工作，如果还需要执行读/写卡相关操作，必需先执行“配置 IC 卡接口”命令。

声明: `void CD_Close()`

1. 主机命令

命令类型 (CmdClass): 0x01

命令代码 (CmdCode): 'C'

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 关闭 IC 卡的接口

表 4.6 关闭 IC 卡接口命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 06 | 01 | 43 | 00 | none | BB | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: 关闭 IC 卡接口成功后的回应

表 4.7 关闭 IC 卡接口成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 01 | 00 | 00 | none | F8 | 03 |

4.1.4 设置 IC 卡接口协议 (工作模式) (Cmd = D)

该命令设置 IC 卡接口协议, 与“配置 IC 卡接口”命令不同之处在于该命令即可以配置 IC 卡接口为 ISO14443-3A 形式, 也可以配置成 ISO14443-3B 形式。配置只对当前上电期间有效, 掉电后, 又恢复至默认的 ISO14443-3A 形式。

声明: `uint8_t CD_SetISOType(CDProtType type)`

1. 主机命令

命令类型 (CmdClass): 0x01
 命令代码 (CmdCode): 'D'
 信息长度 (InfoLength): 0x01
 信 息 (Info): IC 卡接口协议 (1 字节): 0x00——ISO14443-3A 形式
 0x04——ISO14443-3B 形式
 例 如: 配置 IC 卡的接口为 ISO14443-3B 形式

表 4.8 设置 IC 卡接口为 ISO14443-3B 命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 07 | 01 | 44 | 01 | 04 | B8 | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: 配置 IC 卡接口为 ISO14443-3B 成功后的回应

表 4.9 设置 IC 卡接口为 ISO14443-3B 成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 01 | 00 | 00 | none | F8 | 03 |

4.1.5 装载 IC 卡密钥 (Cmd = E)

该命令是将输入的密钥保存在模块内部，模块掉电后该密钥不丢失，ZLG600SP/T 模块共能保存 A 密钥 16 组、B 密钥 16 组。

声明： `uint8_t CD_LoadKey(uint8_t keyType, uint8_t nKeySector, const void *pKey)`

1. 主机命令

命令类型 (CmdClass): 0x01

命令代码 (CmdCode): 'E'

信息长度 (InfoLength): 若是 6 字节密钥，则为 8
若是 16 字节密钥，则为 18

信 息 (Info): 密钥类型 (1 字节): 0x60——密钥 A
0x61——密钥 B

密钥区号 (1 字节): 取值范围 0~15

密钥 (6 字节或 16 字节)

例 如: 向密钥 01 区装载密钥 A: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF

表 4.10 向密钥 01 区装载密钥命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|-------------------------|-----|-----|
| 0E | 01 | 45 | 08 | 60 01 FF FF FF FF FF FF | DC | 03 |

2. 从机应答

状 态 (Status): 0——成功，其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 装载密钥成功模块的回应

表 4.11 装载密钥成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 01 | 00 | 00 | none | F8 | 03 |

3. 说明

此命令是向模块内装载密码，并非改变 Mifare1 卡内扇区的密码。模块内有 6 个密码区 (区号 0~15) 可供装载，每个区分密钥 A (0x60) 和密钥 B (0x61) 两个，总共 32 个密码。装载成功后，可用该密钥对 Mifare1 卡或 PLUS CPU 卡进行验证。装载时若输入的密钥为 6 字节，则模块自动将 6 字节密钥采用复制拼接的方式扩展为 16 字节的密钥。例如密钥为: 0xA0 0xA1 0xA2 0xA3 0xA4 0xA5, 经扩展后为: 0xA0 0xA1 0xA2 0xA3 0xA4 0xA5 0xA0 0xA1 0xA2 0xA3 0xA4 0xA5 0xA0 0xA1 0xA2 0xA3, 扩展后的密钥用于 PLUS CPU 卡的 AES 验证，若需要提高安全性，则直接输入 16 字节的密钥。

若要改变 Mifare1 卡内的密钥，可在用原密码验证通过后，直接用写块数据指令，将密码块改写。

4.1.6 设置 IC 卡接口的寄存器值 (Cmd = F)

该命令用于设置模块上读写卡芯片内部的寄存器值，通过该命令，我们可以实现很多现有命令不能完成的工作。

声明: *void CD_SetReg(uint8_t nRegAdr, uint8_t nRegVal)*

1. 主机命令

命令类型 (CmdClass): 0x01

命令代码 (CmdCode): 'F'

信息长度 (InfoLength): 0x02

信 息 (Info): 寄存器地址 (1 字节): 取值范围 0x00~0x3F
寄存器值 (1 字节)

例 如: 设置 TX1、TX2 天线驱动管脚的阻抗 (0x12 寄存器)

表 4.12 设置寄存器值命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|-------|-----|-----|
| 08 | 01 | 46 | 02 | 12 3F | 9F | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 设置寄存器值成功的回应

表 4.13 设置寄存器值成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 01 | 00 | 00 | none | F8 | 03 |

4.1.7 获取 IC 卡接口的寄存器值 (Cmd = G)

该命令用于设置模块上读写卡芯片内部的寄存器值, 通过该命令, 我们可以实现很多现有命令不能完成的工作。

声明: *uint8_t CD_GetReg(uint8_t nRegAdr)*

1. 主机命令

命令类型 (CmdClass): 0x01

命令代码 (CmdCode): 'G'

信息长度 (InfoLength): 0x01

信 息 (Info): 寄存器地址 (1 字节): 取值范围 0x00~0x3F

例 如: 读取 TX1、TX2 天线驱动管脚的阻抗 (0x12 寄存器)

表 4.14 读取寄存器值命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 07 | 01 | 47 | 01 | 12 | AD | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0x01

信 息 (Info): 寄存器值

例如：读 0x12 寄存器返回的值

表 4.15 读取寄存器值成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 07 | 01 | 00 | 01 | 3F | C7 | 03 |

4.1.8 设置波特率 (Cmd = H)

该命令用于在 UART 通信过程中改变通信的波特率，该命令执行完毕，等到返回成功信息以后才会使新设置的通信波特率生效，掉电后该设置值保留。

声明：void PCDSetsBaud(unsigned char ucBaudNum)

1. 主机命令

命令类型 (CmdClass): 0x01

命令代码 (CmdCode): 'H'

信息长度 (InfoLength): 0x01

信息 (Info): 波特率编号 (1 字节): 取值范围 0~7 如表 4.16 所示

表 4.16 波特率编号对应表

| 编号 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----|------|-------|-------|-------|-------|--------|--------|--------|
| 波特率 | 9600 | 19200 | 28800 | 38400 | 57600 | 115200 | 172800 | 230400 |

例如：设置 UART 通信波特率为 115200

表 4.17 设置 UART 通信波特率为 115200 命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 07 | 01 | 48 | 01 | 05 | B5 | 03 |

2. 从机应答

状态 (Status): 0——成功，其它——失败

信息长度 (InfoLength): 0

信息 (Info): none

例如：设置波特率成功的返回

表 4.18 设置 UART 波特率成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 01 | 00 | 00 | none | F8 | 03 |

4.1.9 设置天线驱动方式 (Cmd = I)

该命令用于设置天线驱动方式，可以打开或关闭 TX1、TX2 中的任意一个管脚，特别适用于双天线应用的设置。

声明：void PCDSetsTX(unsigned char ucSelTX)

1. 主机命令

命令类型 (CmdClass): 0x01

命令代码 (CmdCode): 'I'

信息长度 (InfoLength): 0x01

信 息 (Info): 天线驱动模式 (1 字节): 0x01——仅 TX1 驱动天线
0x02——仅 TX2 驱动天线
0x03——TX1、TX2 同时驱动天线
0x00——同时关闭 TX1、TX2

例 如: 将模块的天线驱动模式改为仅 TX2 输出

表 4.19 设置仅 TX2 驱动命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 07 | 01 | 49 | 01 | 02 | B3 | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 更改天线驱动模式成功的返回

表 4.20 更改天线驱动模式成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 01 | 00 | 00 | none | F8 | 03 |

4.1.10 设置新旧帧格式 (Cmd = K)

该命令用于设置模块通信的帧格式, 前面章节已经描述过, ZLG600SP/T 模块支持新、旧两种帧格式, 模块出厂默认上电是旧帧格式, 可以通过该命令把模块设置成新帧格式, 设置成功后掉电不丢失。

声明: *void PCDSetsTX(unsigned char ucSelTX)*

1. 主机命令

命令类型 (CmdClass): 0x01

命令代码 (CmdCode): 'K'

信息长度 (InfoLength): 0x01

信 息 (Info): 新旧帧格式 (1 字节): 0x00——设置成旧帧格式
0x01——设置成新帧格式

例 如: 将模块的通信设置成新帧格式

表 4.21 设置成新帧格式命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 07 | 01 | 4B | 01 | 01 | B2 | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例如：将模块设置成新帧格式成功后返回如下内容。在新帧情况下要设置回旧帧，应当以新帧形式发送：B2 00 00 01 4B 00 01 00 00 00 FF 切换回旧帧。

表 4.22 设置新旧帧格式成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 01 | 00 | 00 | none | F8 | 03 |

注：在串口通信方式时设置了新旧帧格式后将以原来的帧格式返回成功信息，但在 I²C 通信方式时，收到“设置新旧帧格式”命令后，即默认进入了设置后帧格式，不返回成功信息，下一次与模块通信需要使用设置后的帧格式与模块通信。

4.1.11 设置设备工作模式 (Cmd = U)

该命令用于设置模块上电时的工作模式，同时可以设置模块的从机地址，在一主多从的应用中，应通过该命令先设置好模块的从机地址。

1. 主机命令

命令类型 (CmdClass): 0x01

命令代码 (CmdCode): 'U'

信息长度 (InfoLength): 2 字节

信息 (Info): 工作模式 (1 字节): 该字节包含了模块的各种工作模式，当模块上电没有硬件管脚设置模块工作模式时，通过判断该字节的定义进入相关模式，出厂默认是自动侦测的从机模式，该字节设置后掉电不丢失，字节的描述如下所示：

表 4.23 工作模式字节描述

| B7~B4 | B3~B0 |
|---|---|
| 模块主从模式： 0000: 从机模式 0001: 自动检测卡片模式 (主机模式) | 当模块上电时，没有硬件设置工作模式将通过该 4 位进入相应模式： 0000: 自动侦测模式 0001: I ² C 通信模式 0010: UART 通信(波特率固定为 19200) 0011: UART 通信(波特率固定为上次设置的波特率) 其它: 保留 |

从机地址 (1 字节): 该字节保存从机的地址，设置该字节只在没有硬件设置工作模式时才有效，其它通过硬件设定工作模式时该字节值由硬件决定，如果硬件没有设定从机地址则采用默认 0xB2 地址。另外，从机地址采用 I²C 地址的格式，最低位是读写位，所以从机地址最多只有 127 种。

例如：设置模块的从机地址为 0x02，工作模式为从机的自动侦测模式

表 4.24 设置设备工作模式的命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|-------|-----|-----|
| 08 | 01 | 55 | 02 | 00 02 | A3 | 03 |

2. 从机应答

状态 (Status): 0——成功，其它——失败

信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: 设置模块工作模式成功的回应

表 4.25 设置设备工作模式成功的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 01 | 00 | 00 | none | F8 | 03 |

4.1.12 获取设备工作模式 (Cmd = V)

该命令用于获取模块的工作模式，包括从机地址。

1. 主机命令

命令类型 (CmdClass): 0x01
 命令代码 (CmdCode): 'V'
 信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: 获取模块工作模式信息

表 4.26 获取设备工作模式的命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 06 | 01 | 56 | 00 | none | AE | 03 |

2. 从机应答

状 态 (Status): 0——成功，其它——失败
 信息长度 (InfoLength): 2
 信 息 (Info): 工作模式 (1 字节): 详细说明见表 4.23
 从机地址 (1 字节)
 例 如: 获取模块工作模式信息成功的回应

表 4.27 获取设备工作模式成功的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|-------|-----|-----|
| 08 | 01 | 00 | 02 | 00 B2 | 46 | 03 |

4.1.13 装载用户密钥 (Cmd = a)

该命令用于装载用户密钥，模块里面提供了 2 个 16 字节的存储空间用于保存用户密钥。

1. 主机命令

命令类型 (CmdClass): 0x01
 命令代码 (CmdCode): 'a'
 信息长度 (InfoLength): 0x11
 信 息 (Info): 扇区号 (1 字节): 范围 0 ~1
 密钥数据 (16 字节)
 例 如: 往用户密钥存储 1 区存放 16 字节的密钥

表 4.28 装载用户密钥的命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|---|-----|-----|
| 17 | 01 | 61 | 11 | 01 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF | 98 | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: 往用户密钥存储 1 区存放 16 字节的密钥成功后的回应

表 4.29 装载用户密钥成功的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 01 | 00 | 00 | none | F8 | 03 |

注: 该用户密钥存储区不等同于 IC 卡密钥存储区, 2 个 16 字节大小的存储区, 16 字节的长度刚好与密钥长度相同, 方便用户扩展使用。

4.1.14 读 E²PROM (Cmd = b)

模块内部拥有一个 256Byte 的 E²PROM, 该存储空间掉电不丢失, 通过“读 E²PROM”、“写 E²PROM”命令可以对该存储器的数据进行读写。

1. 主机命令

命令类型 (CmdClass): 0x01
 命令代码 (CmdCode): 'b'
 信息长度 (InfoLength): 0x02
 信 息 (Info): E²PROM 地址 (1 字节): 范围 0 ~ 255
 读取数据的长度 (1 字节): 最大为 249 字节
 例 如: 读取 E²PROM 从 0x08 开始 8 字节的数据

表 4.30 读 E²PROM 的命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|-------|-----|-----|
| 08 | 01 | 62 | 02 | 08 08 | 96 | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): 0x08
 信 息 (Info): none
 例 如: 读取 E²PROM 里面从 0x08 地址开始 8 字节的数据的返回

表 4.31 读 E²PROM 成功的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|-------------------------|-----|-----|
| 0E | 01 | 00 | 08 | 00 00 00 00 00 00 00 00 | F8 | 03 |

注: 由于旧帧格式上的限制, 一次性最多能读取 249 字节, 超过 249 会产生读溢出。

4.1.15 写 E²PROM (Cmd = c)

该命令是往 E²PROM 里面写数据。

1. 主机命令

命令类型 (CmdClass): 0x01

命令代码 (CmdCode): 'c'

信息长度 (InfoLength): 要写的数据长度+2

信 息 (Info): E²PROM 地址 (1 字节): 范围 0 ~255

写数据的长度 (1 字节): 最大为 247 字节

要写入的数据信息 (n 字节)

例 如: 往 E²PRPM 里面 0x02 地址开始写入 4 字节的数据

表 4.32 写 E²PROM 的命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|-------------------|-----|-----|
| 0C | 01 | 63 | 06 | 02 04 FF FF FF FF | 91 | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 往 E²PROM 里面 0x02 地址开始写入 4 字节数据成功的返回

表 4.33 写 E²PROM 成功的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 01 | 00 | 00 | none | F8 | 03 |

注: 由于旧帧格式上的限制, 一次性最多能写入 247 字节, 超过 247 会产生写溢出。

4.2 Mifare S50/S70 卡类命令（CmdClass = 0x02）

Mifare S50/S70 卡类命令总汇如表 4.34 所示。

表 4.34 Mifare S50/S70 卡类命令一览表

| 命令码 | 意义 |
|-----|---------------------|
| ‘A’ | 请求 |
| ‘B’ | 防碰撞 |
| ‘C’ | 卡选择 |
| ‘D’ | 卡挂起 |
| ‘E’ | E ² 密钥验证 |
| ‘F’ | 直接密钥验证 |
| ‘G’ | Mifare 卡读 |
| ‘H’ | Mifare 卡写 |
| ‘I’ | UltraLight 卡写 |
| ‘J’ | Mifare 值操作 |
| ‘L’ | 卡复位 |
| ‘M’ | 卡激活 |
| ‘N’ | 自动检测 |
| ‘O’ | 读自动检测数据 |
| ‘P’ | 设置值块的值 |
| ‘Q’ | 获取值块的值 |
| ‘X’ | 数据交互命令 |

前 4 条命令（命令 A~D）是 ISO14443A 标准定义的命令，只要符合该标准的卡都应能发出响应；中间 6 条命令（命令 E~J）为 Mifare1 卡的专用命令，只有先进行验证（命令 E、F）成功之后才能进行；后四条命令（L、M、N、O）为实用的扩展命令；X 命令为读写器与卡交换数据块，该命令用于 ISO14443-4 标准。

注意：

命令 C 和 M 命令都做了 Mifare 卡和 PLUS CPU 卡自动辨别功能，并根据卡的类型不同自动调用相应的命令，该功能使用户的卡片由 M1 卡升级到 PLUS CPU 卡不必修改，若要执行其它符合 CPU 卡操作，建议使用 PLUS CPU 卡类命令。

4.2.1 请求（Cmd = A）

该命令用于 Mifare 卡的请求操作。

声明：`uint8_t PiccA_Request(uint8_t reqCode, uint8_t *pATQ)`

1. 主机命令

命令类型（CmdClass）： 0x02

命令代码（CmdCode）： ‘A’

信息长度（InfoLength）： 0x01

信 息（Info）： 请求模式（1 字节）： 0x26——IDLE 模式
0x52——ALL 模式

例 如： 请求天线范围内所有的卡

表 4.35 请求卡命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 07 | 02 | 41 | 01 | 52 | E8 | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): 0x02
 信 息 (Info): 请求应答 ATQ (2 字节, 低位在前)

表 4.36 ATQ 字节描述

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|-----|-----|-----|-----|-----|-----|----|----|--|----|-----|-----------------------|----|----|----|----|
| RFU | | | | | | | | UID 大小 00:4bytes 01:7bytes 10:10bytes | | RFU | 如果有任何位为 1, 则为比特帧防冲突方式 | | | | |

表 4.37 列举了各种类型的卡返回的 ATQ。

表 4.37 返回 ATQ 一览表

| 卡类型 | Mifare1 S50 | Mifare1 S70 | Mifare1 Light | Mifare0 UltraLight | Mifare3 DESFire | SHC1101 | SHC1102 | 11RF32 |
|-----|----------------|----------------|------------------|-----------------------|--------------------|---------|---------|--------|
| ATQ | 0x0004 | 0x0002 | 0x0010 | 0x0044 | 0x0344 | 0x0004 | 0x3300 | 0x0004 |

例 如: S50 卡返回的 ATQ

表 4.38 请求成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|-------|-----|-----|
| 08 | 02 | 00 | 02 | 04 00 | F3 | 03 |

3. 说明

卡进入天线后, 从射频场中获取能量, 从而得电复位, 复位后卡处于 IDLE 模式, 用两种请求模式的任一种请求时, 此时的卡均能响应; 若对某一张卡成功进行了挂起操作 (Halt 命令或 DeSelect 命令), 则进入了 Halt 模式, 此时的卡只响应 ALL (0x52) 模式的请求, 除非将卡离开天线感应区后再进入。

注: DeSelect 为 ISO14443-4 命令。另外, 对 Mifare1 卡连续进行请求操作, 总是一次成功, 一次失败, 循环往复。

4.2.2 防碰撞 (Cmd = B)

该命令用于 Mifare 卡的防碰撞操作, 需要执行成功一次请求命令, 并返回请求成功, 才能进行防碰撞操作, 否则返回错误。

声明: *uint8_t PiccA_Anticoll(uint8_t mode, uint8_t selCode, uint8_t *pUID, uint8_t nBitCnt)*

1. 主机命令

命令类型 (CmdClass): 0x02
 命令代码 (CmdCode): 'B'
 信息长度 (InfoLength): 若位计数=0, 则长度=2

表 4.39 防碰撞命令帧

表 4.40 防碰撞回应帧

0x95——第二级防碰撞

0x97——第三级防碰撞

UID（4 字节）：前一个防碰撞命令返回的 UID

例 如： 第一级选择，UID 为 0xEB1C1814

表 4.41 卡选择命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|----------------|-----|-----|
| 0B | 02 | 43 | 05 | 93 14 18 1C EB | D8 | 03 |

2. 从机应答

状 态（Status）： 0——成功，其它——失败

信息长度（InfoLength）： 0x01

信 息（Info）： 选择应答 SAK，如表 4.42 所示，其中 Bit 2 位是 Cascade 位，表示 UID 是否完整。

若 Bit 2 = 0，表示 UID 完整

若 Bit 2 = 1，表示 UID 不完整，还有部分 UID 未读出

表 4.42 返回 SAK 一览表

| 卡类型 | Mifare1 S50 | Mifare1 S70 | Mifare1 Light | Mifare0 UltraLight | Mifare3 DESFire | SHC1101 | SHC1102 | 11RF32 |
|-----|----------------|----------------|------------------|-----------------------|--------------------|---------|---------|--------|
| SAK | 0x08 | 0x18 | 0x01 | 0x04 | 0x24 | 0x22 | — | 0x08 |

例 如： 返回 S50 卡应答

表 4.43 卡选择成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 07 | 02 | 00 | 01 | 08 | F3 | 03 |

3. 说明

卡的序列号长度有三种：4 字节、7 字节和 10 字节。4 字节的只要用一级选择即可得到完整的序列号，如 Mifare1 S50/S70 等；7 字节的要求用二级选择才能得到完整的序列号，前一级所得到的序列号的最低字节为级联标志 0x88，在序列号内只有后 3 字节可用，后一级选择能得到 4 字节序列号，两者按顺序连接即为 7 字节序列号，如 UltraLight 和 DesFire 等；10 字节的以此类推，但至今还未发现此类卡。

在程序中可用 SAK.2 位来判断是还有序列号未读出，如 if(SAK & 0x04){...}。

4.2.4 卡挂起（Cmd = D）

该命令用于 Mifare 卡的挂起操作，使所选择的卡进入 HALT 状态，在 HALT 状态下，卡将不响应读卡器发出的 IDLE 模式的请求，除非将卡复位或离开天线感应区后再进入。但它会响应读卡器发出的 ALL 请求。

声明：uint8_t PiccA_Halt(void)

1. 主机命令

命令类型（CmdClass）： 0x02

命令代码（CmdCode）： ‘D’

例如：将已激活的卡挂起，使之不响应请求空闲卡命令

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 06 | 02 | 44 | 00 | none | BF | 03 |

例如：挂起命令执行成功的回应

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 02 | 00 | 00 | none | FB | 03 |

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
|----------|-------|-----|--------|------|-----|-----|

| | | | | | | |
|----|----|----|----|----------------------|----|----|
| 0D | 02 | 45 | 07 | 60 14 18 1C EB 01 04 | 2C | 03 |
|----|----|----|----|----------------------|----|----|

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例如：验证成功返回的信息

表 4.47 E2 密钥验证成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 02 | 00 | 00 | none | FB | 03 |

4.2.6 直接密钥验证 (Cmd = F)

该命令将密码作为参数传递，因此在此之前不需“装载 IC 卡密钥”命令。若当前卡为 PLUS CPU 卡的等级 2 或等级 3，且输入的密码只有 6 字节，则模块自动将输入的密码复制 2 次，取前 16 字节作为当前验证密钥。

声明: *uint8_t MF_Authent(uint8_t mode, const void *pKey, const uint8_t *pUID, uint8_t nBlock)*

1. 主机命令

命令类型 (CmdClass): 0x02

命令代码 (CmdCode): 'F'

信息长度 (InfoLength): 密钥为 6 字节, 则为 12

密钥为 16 字节，则为 22

信息 (Info): 密钥类型 (1 字节): 0x60——密钥 A

0x61——密钥 B

卡序列号 (4 字节)

密钥（6 字节或 16 字节）

卡块号 (1 字节): S50 (0~63)

S70 (0~255)

PLUS CPU 2K (0~127)

PLUS CPU 4K (0~255)

例如：用密钥“0xFF 0xFF 0xFF 0xFF 0xFF 0xFF”验证序列号为 0xEB1C1814 的卡的块 4

注：PLUS CPU 系列的卡的卡号有 4 字节和 7 字节之分，对于 7 字节卡号的卡，只需要将卡号的高 4 字节（等级 2 防碰撞得到的卡号）作为验证的卡号即可。

表 4.48 直接密钥验证命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
|----------|-------|-----|--------|------|-----|-----|

| | | | | | | |
|----|----|----|----|--|----|----|
| 12 | 02 | 46 | 0C | 60 14 18 1C EB FF FF FF FF FF FF 04 | 3A | 03 |
|----|----|----|----|--|----|----|

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例如：验证成功返回的信息

表 4.49 直接密钥验证成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 02 | 00 | 00 | none | FB | 03 |

4.2.7 Mifare 卡读 (Cmd = G)

该命令对 Mifare 卡进行读操作，读之前必需成功进行密钥验证。

声明: *uint8 t MF Read(uint8 t nStartBlock, uint8 t nBlockNum, void *pBuf)*

1. 主机命令

命令类型 (CmdClass): 0x02

命令代码 (CmdCode): 'G'

信息长度 (InfoLength): 0x01

| | | |
|-------------|-------------|---------------------|
| 信 息 (Info): | 卡块号 (1 字节): | S50 (0~63) |
| | | S70 (0~255) |
| | | PLUS CPU 2K (0~127) |
| | | PLUS CPU 4K (0~255) |

例如：读块 4 的数据

表 4.50 Mifare 卡读命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 07 | 02 | 47 | 01 | 04 | B8 | 03 |

2. 从机应答

状态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0x10

信 息 (Info): 块数据 (16 字节)

例 如：从卡的块 4 读出数据为：“0x7F 0x4B 0xD8 0x37 0xAA 0x99 0xF3 0xE0 0xA5 0xD9 0x93 0x70 0x8F 0x89 0xE2 0x64”

表 4.51 Mifare 读成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
|----------|-------|--------|--------|------|-----|-----|

| | | | | | | |
|----|----|----|----|--|----|----|
| 16 | 02 | 00 | 10 | 7F 4B D8 37 AA 99 F3 E0 A5 D9 93 70 8F 89 E2 64 | 1F | 03 |
|----|----|----|----|--|----|----|

3. 说明

在验证成功之后，才能读相应的块数据，所验证的块号与读块号必须在同一个扇区内，Mifare1 卡从块号 0 开始按顺序每 4 个块 1 个扇区，若要对一张卡中的多个扇区进行操作，在对某一扇区操作完毕后，必须进行一条读命令才能对另一个扇区直接进行验证命令，否则必须从请求开始操作。

对于 PLUS CPU 卡，若下一个读扇区的密钥和当前扇区的密钥相同，则不需要再次验证密钥，直接读即可。

4.2.8 Mifare 卡写 (Cmd = H)

该命令对 Mifare 卡进行写操作，写之前必需成功进行密钥验证。

声明: `uint8_t MF_Write(uint8_t nStartBlock, uint8_t nBlockNum, const void *pBuf)`

1. 主机命令

命令类型 (CmdClass): 0x02

命令代码 (CmdCode): 'H'

信息长度 (InfoLength): 0x11

信 息 (Info): 卡块号 (1 字节): S50 (0~63)
S70 (0~255)
PLUS CPU 2K (0~127)
PLUS CPU 4K (0~255)

数据 (16 字节)

例 如: 向块 4 写入 16 字节数据 “0x00 0x01 0x02 0x03 0x04 0x05
0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F”

表 4.52 Mifare 卡写命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|---|-----|-----|
| 17 | 02 | 48 | 11 | 04 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F | B7 | 03 |

2. 从机应答

状 态 (Status): 0——成功，其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 数据成功写入卡片模块的回应

表 4.53 Mifare 写成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 02 | 00 | 00 | none | FB | 03 |

3. 说明

对卡内某一块进行验证成功后，即可对同一扇区的各个进行写操作（只要访问条件允

许), 其中包括位于扇区尾的密码块, 这是更改密码的唯一方法。对于 PLUS CPU 卡等级 2、3 的 AES 密钥则是在其他位置修改密钥。

4.2.9 UltraLight 卡写 (Cmd = I)

该命令对 UltraLight 卡进行写操作。

声明: *uint8_t MF0_ULWrite(uint8_t nStartBlock, uint8_t nBlockNum, const void *pBuf)*

1. 主机命令

命令类型 (CmdClass): 0x02
命令代码 (CmdCode): 'I'
信息长度 (InfoLength): 0x05
信 息 (Info): 卡块号 (1 字节): 1~15
数据 (4 字节)
例 如: 写块 4 数据

表 4.54 UltraLight 卡写命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|----------------|-----|-----|
| 0B | 02 | 49 | 05 | 04 05 05 05 05 | BE | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
信息长度 (InfoLength): 0
信 息 (Info): none
例 如: 数据成功写入卡片模块的回应

表 4.55 UltraLight 卡写成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 02 | 00 | 00 | none | FB | 03 |

3. 说明

此命令只对 UltraLight 卡有效, 对 UltraLight 卡进行读操作与 Mifare1 卡一样。

4.2.10 Mifare 值操作 (Cmd = J)

该命令对 Mifare 卡的值块进行加减操作。

声明: *uint8_t MF1_Value(uint8_t operMode, uint8_t nSourceBlock, int32_t nValue, uint8_t nDestinationBlock)*

1. 主机命令

命令类型 (CmdClass): 0x02
命令代码 (CmdCode): 'J'
信息长度 (InfoLength): 0x07
信 息 (Info): 模式 (1 字节): 0xC0~减
0xC1~加
卡块号 (1 字节): S50 (0~63)

S70 (0~255)

PLUS CPU 2K (0~127)

PLUS CPU 4K (0~255)

值 (4 字节有符号数, 低字节在先)

传输块号 (1 字节)

例 如: 将块 4 的值减 1, 其结果保存到块 5

表 4.56 Mifare 值操作命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|----------------------|-----|-----|
| 0D | 02 | 4A | 07 | C0 04 01 00 00 00 05 | 7D | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 值块操作成功后模块的回应

表 4.57 Mifare 值操作成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 02 | 00 | 00 | none | FB | 03 |

3. 说明

要进行此类操作, 块数据必须要有值块的格式, 可参考 NXP 的相关文档。若卡块号与传输块号相同, 则将操作后的结果写入原来的块内; 若卡块号与传输块号不相同, 则将操作后的结果写入传输块内, 结果传输块内的数据被覆盖, 原块内的值不变。处于等级 2 的 PLUS CPU 卡不支持值块操作, 等级 1、3 支持。

4.2.11 卡复位 (Cmd = L)

该命令是通过将载波信号关闭指定的时间, 再开启来实现卡片复位。

声明: `void CD_PauseCarrier(uint8_t pause_ms, uint8_t wait_ms)`

1. 主机命令

命令类型 (CmdClass): 0x02

命令代码 (CmdCode): 'L'

信息长度 (InfoLength): 0x01

信 息 (Info): 时间 (1 字节), 以毫秒为单位, 0 为一直关闭

例 如: 将载波信号关闭 1ms

表 4.58 卡复位命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 07 | 02 | 4C | 01 | 01 | B6 | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: 执行卡复位成功模块的回应

表 4.59 卡复位成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 02 | 00 | 00 | none | FB | 03 |

3. 说明

该命令将天线信号关闭数毫秒，若一直关闭，则等到执行一个请求命令时打开。

4.2.12 卡激活 (Cmd = M)

该命令用于激活卡片，是请求、防碰撞和选择三条命令的组合。

声明: *uint8_t MF_Activate(uint8_t mode, uint8_t reqCode, PiccAResetInfo *pResetInfo)*

1. 主机命令

命令类型 (CmdClass): 0x02
 命令代码 (CmdCode): 'M'
 信息长度 (InfoLength): 0x02
 信 息 (Info): 保留 (1 字节)，设置为 0
 请求代码 (1 字节): 0x26~IDLE
 0x52~ALL

例 如: 以 IDLE 方式激活卡

表 4.60 卡激活命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|-------|-----|-----|
| 08 | 02 | 4D | 02 | 00 26 | 9C | 03 |

2. 从机应答

状 态 (Status): 0——成功，其它——失败
 信息长度 (InfoLength): Mifare1 S50、S70、Light 卡: 8 字节
 Mifare0 UltraLight 卡: 11 字节
 Mifare3 Desfire 卡: 11 字节
 Plus CPU 卡: 8 字节或 11 字节
 信 息 (Info): 请求应答 ATQ (2 字节)
 最后一级选择应答 SAK (1 字节)
 序列号长度 (1 字节)
 序列号 (N 字节，由序列号长度决定)
 例 如: 一张序列号为 0xEB1C1814 的 Mifare1 S50 卡返回的数据

表 4.61 卡激活成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|-------------------------|-----|-----|
| 0E | 02 | 00 | 08 | 04 00 08 04 14 18 1C EB | 08 | 03 |

4.2.13 自动检测 (Cmd = N)

该命令用于卡片的自动检测，执行该命令成功后，在 UART 模式下，模块将主动发送读取到卡片的数据。

声明: *unsigned char PiccAutoDetect(unsigned char ucADMode, unsigned char ucTxMode, unsigned char ucReqCode, unsigned char ucAuthMode, unsigned char ucKeyType, unsigned char *pKey, unsigned char ucBlock)*

1. 主机命令

命令类型 (CmdClass): 0x02

命令代码 (CmdCode): 'N'

信息长度 (InfoLength): 若验证模式 = “E”，则为 7
若验证模式 = “F”，则为 12 或 22
若验证模式 = 0，则为 4

信 息 (Info): 自动检测模式 *ADMode* (1 字节): 该字节内容如表 4.62 所示

表 4.62 *ADMode* 字节位描述

| B7~B4 | B3 | B2 | B1 | B0 |
|-------------|--|------------------------------|---|--|
| RFU 0000 | 执行完一次自动检测后的动作 0: 无动作 1: 最后执行 Halt 命令 | 数据输出后 0: 不继续检测 1: 继续检测 | 当 UART 接口，检测到有卡时 0: 不产生中断 1: 产生中断，当串口发送数据完毕后，中断消失；当 I ² C 接口时此位无效，应设置为 0，因肯定产生中断 | 当 UART 接口，检测到有卡时 0: 串口不发送 1: 串口主动发送，发送的数据格式见“检测卡回应格式”。当 I ² C 接口时此位无效，应设置为 0，因为是从模式 |

天线驱动方式 *TxMode* (1 字节): 字节描述如表 4.63 所示

表 4.63 *TxMode* 字节位描述

| B7~B2 | B1 | B0 |
|---------------|--|----|
| RFU 000000 | 00: TX1、TX2 交替驱动 01: 仅 TX1 驱动 10: 仅 TX2 驱动 11: TX1、TX2 同时驱动 | |

请求代码 *ReqCode* (1 字节): 0x26~IDLE
0x52~ALL

验证模式 *AuthMode* (1 字节): ‘E’~用 E2 密钥验证
‘F’~用直接密钥验证
0 ~不验证

密钥 AB *KeyType* (1 字节): 0x60~密钥 A
0x61~密钥 B

密钥 *Key*: 若验证模式为‘E’, 则为密钥区号 (1 字节)
若验证模式为‘F’, 则为密钥 (6 或 16 字节)

卡块号 *Block* (1 字节): S50 (0~63)
S70 (0~255)
PLUS CPU 2K (0~127)
PLUS CPU 4K (0~255)

例如: 设置模块检测到有卡时产生中断, 串口输出, 以 IDLE 方式激活卡, 用直接密码验证密钥 A (密码为 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF), 读出第 1 块数据内容

表 4.64 自动检测命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|--|-----|-----|
| 12 | 02 | 4E | 0C | 03 03 26 46 60 FF FF FF FF FF FF 01 | AC | 03 |

2. 从机应答

状态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信息 (Info): none

例如: 模块设置自动检测成功的回应

表 4.65 设置自动检测成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 02 | 00 | 00 | none | FB | 03 |

3. 检测卡回应格式

“从机应答”只是说明模块设置成自动检测成功, 在串口通信方式下, 自动检测模式使能后, 若允许串口主动发送 (即 *ADMode.0*=1), 有卡靠近模块, 模块将自动把检测到卡的相应信息按如下的格式发送。

状态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 若验证命令不为 0, 则为: 21+序列号长度
若验证命令为 0, 则为: 5+序列号长度

信息 (Info): 天线驱动 *TxDrv* (1 字节): 如表 4.66 所示

表 4.66 TxDrv 字节位描述

| B7~B2 | B1 | B0 |
|---------------|--|----|
| RFU 000000 | 00: TX1、TX2 交替驱动 01: 仅 TX1 驱动 10: 仅 TX2 驱动 11: TX1、TX2 同时驱动 | |

请求应答 *ATQ* (2 字节)

选择应答 *SAK* (1 字节)

序列号长度 *UIDLen* (1 字节)

序列号 *UID* (长度为各种卡序列号的实际长度)

块数据: 若验证命令不为 0, 则块数据为 16 字节

若验证命令为 0, 则块数据为 0 字节

例如: 检测到序列号为 0xEB1C1814 的 S50 卡, 并读出块 1 数据

表 4.67 检测卡成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|--|-----|-----|
| 1F | 02 | 00 | 19 | 03 04 00 08 04 14 18 1C EB 14 18 1C EB FB 88 04 00 47 C1 24 37 E1 00 11 06 | E4 | 03 |

4. 说明

执行自动检测命令成功后, 并且读取卡片信息成功返回, 整个过程相当于以下命令的组合: 请求——防碰撞——选择——验证 (若 *AuthMode*!=0)——读取 (若 *AuthMode*!=0)——挂起 (若 *AuthMode*.3=1)。当输入的密钥为 6 字节时, 模块内部将按 *Key*[0:15]=*pKey*[0:5]*pKey*[0:5]*pKey*[0:3]模式扩展。

串口主动发送之后, 模块状态由 *ADMode*.2 位来决定, 若 *ADMode*.2=1, 则自动进入自动检测模式; 否则结束自动检测模式, 主机可以发送其它任何命令。若 *ADMode*.1=1, 则模块检测到卡后产生中断信号, 可以通过读取自动检测数据命令 (*Cmd*=0) 读取。

当为 I²C 接口通信时, 因模块为从模式, 所以不主动发送数据, 但肯定输出中断信号, 应使 *ADMode*.1=0, 产生中断后, 主机可以通过以下两种方式读回数据。

- 一是直接读取, 这样读取之后的模块状态由 *ADMode*.2 位来决定: 若 *ADMode*.2=1, 则继续进入自动检测模式; 否则结束自动检测模式, 主机可以发送其它任何命令。
- 二是通过读取自动检测数据命令 (*Cmd*=0) 读取数据之后的模块状态由该函数的参数来决定: 在自动检测模式期间, 主机可以随时发出读取自动检测数据命令, 读取自动检测数据、查询自动检测状态、取消或继续自动检测; 验证和读命令只对 Mifare1 卡和 PLUS CPU 卡有效。

注: 在自动检测期间, 若主机发送任何除读自动检测数据外的, 且数据长度小于 3 (帧长小于 9) 的命令, 将退出自动检测模式, 如请求 *Picca-Request* 0 命令, 在此期间, 模块将不接收数据长度大于 2 (帧长大于 8) 的命令。

4.2.14 读自动检测数据 (*Cmd* = 0)

该命令用于读取自动检测的数据，特别适合于 I²C 通信模式下使用。通过该读取自动检测数据命令，可以决定读取数据后是否继续检测。

声明： *INT8U ReadAutoDetect(INT8U ReadMode);*

1. 主机命令

- 命令类型 (CmdClass): 0x02
- 命令代码 (CmdCode): ‘O’
- 信息长度 (InfoLength): 0x01
- 信 息 (Info): 读模式 (1 字节)，该字节内容如表 4.68 所示

表 4.68 读模式字节描述

| B7~B1 | B0 |
|---------------|---------------------------------|
| RFU 000000 | 数据发回之后： 00: 取消检测 01: 继续检测 |

例 如： 读取自动检测数据之后取消自动检测

表 4.69 读自动检测数据命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 07 | 02 | 4F | 01 | 00 | B4 | 03 |

2. 从机应答

- 状 态 (Status): 0——成功，其它——失败
- 信息长度 (InfoLength): 0x19
- 信 息 (Info): 自动检测读取成功保存的信息
- 例 如： 读自动检测数据成功的回应

表 4.70 读自动检测数据成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|--|-----|-----|
| 1F | 02 | 00 | 19 | 03 04 00 08 04 14 18 1C EB 14 18 1C EB FB 88 04 00 47 C1 24 37 E1 00 11 06 | E4 | 03 |

4.2.15 设置值块的值 (Cmd = P)

该命令用于设置值块的值。

声明： *uint8_t MF1_SetValue(uint8_t nBlock, int32_t nValue);*

1. 主机命令

- 命令类型 (CmdClass): 0x02
- 命令代码 (CmdCode): ‘P’
- 信息长度 (InfoLength): 0x05
- 信 息 (Info): 块地址 (1 字节): 将要写入数值的块地址
块值 (4 字节): 有符号的 32 位数据，低字节在前

例如：将 0x05 值块地址的值设置为 0x03

表 4.71 设置值块的值命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|----------------|-----|-----|
| 0B | 02 | 50 | 05 | 05 03 00 00 00 | A5 | 03 |

2. 从机应答

状态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信息 (Info): none

例如：将 0x05 值块地址的值设置为 0x03 成功后的返回

表 4.72 设置值块的值成功的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 02 | 00 | 00 | none | FB | 03 |

4.2.16 获取值块的值 (Cmd = Q)

该命令用于获取值块的值, 值块里面的数据只有是按照值格式存储时, 才能通过该命令读取成功, 否则返回失败。

声明: `uint8_t MF1_GetValue(uint8_t nBlock, int32_t *pValue);`

1. 主机命令

命令类型 (CmdClass): 0x02

命令代码 (CmdCode): 'Q'

信息长度 (InfoLength): 0x01

信息 (Info): 块地址 (1 字节): 将要读取数值的块地址

例如：读 0x06 值块地址的值

表 4.73 获取值块的值命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 07 | 02 | 51 | 01 | 06 | AC | 03 |

2. 从机应答

状态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 4

信息 (Info): 块值 (4 字节): 有符号的 32 位数据, 低字节在前

例如：读 0x06 值块地址的值成功后的返回

表 4.74 获取值块的值成功的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|-------------|-----|-----|
| 0A | 02 | 00 | 04 | 01 00 00 00 | F2 | 03 |

4.2.17 数据交互命令 (Cmd = X)

该命令用读写器与卡片的数据交互, 通过该命令可以实现读写卡器的所有功能。

声明: `uint8_t CD_ExchangeBlock(const ExchangeInputPara *pIn, ExchangeOutputPara *pOut)`

1. 主机命令

命令类型 (CmdClass): 0x02

命令代码 (CmdCode): 'X'

信息长度 (InfoLength): 交互数据块长度+2

信 息 (Info): 交互数据块 (其内容与实际使用的 CPU 卡有关)

WTXM_CRC (1 字节), 该字节内容如表 4.75 所示

表 4.75 WTXM_CRC 字节描述

| B7~B2 | B1 | B0 |
|-------|-----|--------|
| WTXM | RFU | CRC 禁能 |
| | 0 | CRC 使能 |

FWI (1 字节): 超时等待时间编码

超时时间 = ((0x01 << FWI) * 302us)

例 如: 向一张已被激活的 Mifare DESFire 卡发送“请求应答以选择 (RATS)”命令, 交互的数据块为该命令的命令帧 (0xE0,0x50), 帧长 2 字节 (不包括 CRC 校验, 其中 0xE0 是 RATS 的命令编码, 0x50 的高半字节为 FSDI, 低半字节为 CID, FSDI=5 表示最大交互帧为 64 字节)

表 4.76 数据交互命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|-------------|-----|-----|
| 0A | 02 | 58 | 04 | E0 50 01 04 | 1E | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0x06

信 息 (Info): ATS

例 如: RATS 命令执行成功的回应

表 4.77 数据交互成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|-------------------|-----|-----|
| 0C | 02 | 00 | 06 | 06 77 81 02 80 00 | 85 | 03 |

4.3 ISO7816-3 类命令 (CmdClass = 0x05)

ISO7816-3 类命令汇总表如表 4.78 所示。

表 4.78 ISO7816-3 类命令一览表

| 命令码 | 意义 |
|-----|----------------------------------|
| ‘A’ | 接触式 IC 卡复位(自动处理 PPS) |
| ‘B’ | 接触式 IC 卡传输协议 (自动处理 T=0 和 T=1 协议) |
| ‘C’ | 接触式 IC 卡冷复位 |
| ‘D’ | 接触式 IC 卡热复位 |
| ‘E’ | 接触式 IC 卡停活 (关闭电源和时钟) |
| ‘F’ | 接触式 IC 卡 PPS(传输协议协商) |
| ‘G’ | 接触式 IC 卡 T=0 传输协议 |
| ‘H’ | 接触式 IC 卡 T=1 传输协议 |

其中 ‘A’ 和 ‘B’ 命令是组合命令, 根据卡片的实际情况自动调整通信协议; ‘C’、‘D’、‘F’ ~ ‘H’ 命令需要使用者自己根据卡片的情况来调用不同的命令; ‘E’ 命令是停活命令, 执行该命令后, IC 卡处于掉电状态。实际上对用户来说, 只需要执行 ‘A’、‘B’ 命令即可。

注意: ‘D’ 命令没有控制电源, 执行该命令前必须保证该 IC 卡没有处于停活状态。

4.3.1 接触式 IC 卡复位(自动处理 PPS)

该命令是冷复位, 执行成功后会自动根据 IC 卡的复位信息来自动执行 PPS 命令, 然后再选择 ‘B’ 命令使用的传输协议 (T = 0 或 T = 1)。

声明: `uint8_t Cicc_Reset(uint8_t nSlotIndex, uint8_t nResetFD,`
`void *pATRBuf, uint32_t nBufSize, uint32_t *pATRBytes)`

1. 主机命令

命令类型(CmdClass): 0x05
命令代码(CmdCode): ‘A’
信息长度(InfoLength): 0x01
信 息(Info): IC 卡复位时的波特率 (1 字节): 0x11 — 9600bps
0x13 — 38400bps

例 如: 初始波特率为 38400bps 的接触式 IC 卡复位命令帧如表 4.79 所示

表 4.79 初始波特率为 38400bps 的接触式 IC 卡复位命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 07 | 05 | 41 | 01 | 13 | AE | 03 |

2. 从机回应

执行状态 (Status): 0 — 执行成功; 其他 — 警告或失败
信息长度(InfoLength): 16 + (不同的卡回应的字节数不同)
信 息(Info): 保留信息 (16 字节, 该信息为任意值)
接触式 IC 卡复位信息 (不同的卡复位信息长度不同)

例如：接触式 IC 卡复位操作执行成功的回应帧如表 4.80 所示

表 4.80 接触式 IC 卡复位操作执行成功的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|--|-----|-----|
| 23 | 05 | 00 | 1D | 13 00 00 00 00 00 00 00 10 03 00 00 03 03 00 00 3B 69 00 00 57 44 37 51 BA CB 18 18 35 | A7 | 03 |

注意：表 4.80 中信息字段中的前 16 字节是无效字节，没有任何意义，保留为将来使用，用户不用理会；后 13 字节才是接触式 IC 卡的复位信息。

4.3.2 接触式 IC 卡传输协议（自动处理 T = 0 和 T = 1 协议）

该命令根据接触式 IC 卡的复位信息，自动选择 T = 0 或 T = 1 传输协议，整个过程不需要使用者干预。该命令用于传输 APDU 数据流。

声明：uint8_t Cicc_TPDU(const void *pSendBuf, uint32_t nSendBytes,
void *pRcvBuf, uint32_t nRcvBufSize, uint32_t *pRcvBytes)

1. 主机命令

命令类型(CmdClass): 0x05
命令代码(CmdCode): ‘B’
信息长度(InfoLength): 1~272
信息(Info): 发送到 IC 卡的数据
例如：

通过 FID（文件标识符）选择 MF（FID 为：3F00）。选择文件的 APDU 如表 4.81 所示，将其转换为数据流为：00 A4 00 00 02 3F 00 00（不需要区分 APDU 的 4 种情况，‘3F00’在数据流中是以大端模式存放，即高字节在前），该命令能自动处理，其命令帧如表 4.82 所示

表 4.81 某 CPU 卡选择文件的 APDU

| 代码 | 长度 (字节) | 值 (Hex) | 说明 |
|---|------------|------------|--|
| CLA | 1 | 00 | — |
| INS | 1 | A4 | — |
| P1 | 1 | 00/04 | P1=00，表示按文件标识符选择（P2 必须等于 0），可选择： • 当前目录（DF）下基本文件或子目录文件 • 同级目录文件（DF） P1=04，表示用 DF 名称选择，分如下两种情况： • P2=00，表示第一个或仅有一个 • P2=02，表示下一个 |
| P2 | 1 | 00/02 | — |
| Lc | 1 | xx | — |
| Data | xx | xx…xx | 文件标识符或 DF 名称 |
| Le | 1 | 00 | 对于 DF 而言为卡片自动返回的 FCI 的最大长度 |
| 注：在任何情况下均可通过标识符‘3F00’或目录名称 1PAY.SYS.DDF01 选择 MF | | | |

表 4.82 通过 FID 选择 MF (FID 为 ‘3F00’) 的命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|-------------------------|-----|-----|
| 0E | 05 | 42 | 08 | 00 A4 00 00 02 3F 00 00 | 27 | 03 |

2. 从机回应

执行状态 (Status): 0 — 执行成功; 其他 — 警告或失败
 信息长度(InfoLength): 不同的卡回应的字节数不同
 信 息(Info): IC 卡回复的数据
 例 如: 选择 MF 操作执行成功的回应帧如表 4.83 所示

表 4.83 选择 MF 操作执行成功的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|--|-----|-----|
| 1F | 05 | 00 | 19 | 6F 15 84 0E 31 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 03 88 01 01 90 00 | D5 | 03 |

表 4.83 中的前 23 字节为 MF 的 FCI, 最后 2 字节 ‘90 00’ 表示卡片处理成功。需要注意的是 Info 域的最后 2 字节表示卡片执行结果与回应帧中的 ‘Sataus’ 字段表示的不是同一状态, ‘Sataus’ 字段表示是通信链路层的状态; 而 Info 域的最后 2 字节表示卡片执行结果。

4.3.3 接触式 IC 卡冷复位

该命令是冷复位, 执行了接触式 IC 卡上电时序, 执行成功后会自动根据 IC 卡的复位信息来选择 ‘B’ 命令使用的传输协议 (T=0 或 T=1)。与 4.3.1 相比只是没有自动执行 PPS 命令, 需要用户根据复位信息来判断是否使用 `Cicc_PPS()` 来修改协议和参数。

声明: `uint8_t Cicc_ColdReset(uint8_t nSlotIndex, uint8_t nResetFD,`
`void *pATRBuf, uint32_t nBufSize, uint32_t *pATRBytes)`

1. 主机命令

命令类型(CmdClass): 0x05
 命令代码(CmdCode): ‘C’
 信息长度(InfoLength): 0x01
 信 息(Info): IC 卡复位时的波特率 (1 字节): 0x11 — 9600bps
 0x13 — 38400bps
 例 如: 初始波特率为 38400bps 的接触式 IC 卡冷复位命令帧和复位的命令帧除了命令码不同, 其他相同, 如表 4.84 所示

表 4.84 初始波特率为 38400bps 的接触式 IC 卡冷复位命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 07 | 05 | 43 | 01 | 13 | AC | 03 |

2. 从机回应

执行状态 (Status): 0 — 执行成功; 其他 — 警告或失败
 信息长度(InfoLength): 16 + (不同的卡回应的字节数不同)
 信 息(Info): 保留信息 (16 字节, 该信息为任意值)

接触式 IC 卡复位信息（不同的卡复位信息长度不同）

例 如： 接触式 IC 卡复位操作执行成功的回应帧如表 4.85 所示。和复位操作成功的回应帧相同，见表 4.80

表 4.85 接触式 IC 卡冷复位操作执行成功的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|--|-----|-----|
| 23 | 05 | 00 | 1D | 13 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 03 3B 69 00 00 57 44 37 51 BA CB 18 18 35 | 44 | 03 |

注意：表 4.80 中信息字段中的前 16 字节是无效字节，没有任何意义，保留为将来使用，用户不用理会；后 17 字节才是接触式 IC 卡的复位信息。

4.3.4 接触式 IC 卡热复位

该命令是热复位，没有执行了接触式 IC 卡上电时序，执行成功后会自动根据 IC 卡的复位信息来选择‘B’命令使用的传输协议（T=0 或 T=1）。该命令和 4.3.3 比较只是没有执行 IC 卡上电操作。需要用户根据复位信息来判断是否使用 Cicc_PPS()来修改协议和参数。该命令必须在 IC 卡时钟和电源均有效的情况下才能执行。

声明： *uint8_t Cicc_WarmReset(uint8_t nSlotIndex, uint8_t nResetFD,*
*void *pATRBuf, uint32_t nBufSize, uint32_t *pATRBytes)*

1. 主机命令

命令类型(CmdClass): 0x05

命令代码(CmdCode): ‘D’

信息长度(InfoLength): 0x01

信 息(Info): IC 卡复位时的波特率（1 字节): 0x11 — 9600bps
0x13 — 38400bps

例 如： 初始波特率为 38400bps 的接触式 IC 卡冷复位命令帧和复位的命令帧除了命令码不同，其他相同，如表 4.86 所示

表 4.86 初始波特率为 38400bps 的接触式 IC 卡热复位命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 07 | 05 | 44 | 01 | 13 | AB | 03 |

2. 从机回应

执行状态 (Status): 0 — 执行成功； 其他 — 警告或失败

信息长度(InfoLength): 16 + （不同的卡回应的字节数不同）

信 息(Info): 保留信息（16 字节，该信息为任意值）
接触式 IC 卡复位信息（不同的卡复位信息长度不同）

例 如： 接触式 IC 卡复位操作执行成功的回应帧如表 4.87 所示。和复位操作成功的回应帧相同，见表 4.80

表 4.87 接触式 IC 卡热复位操作执行成功的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|--|-----|-----|
| 23 | 05 | 00 | 1D | 13 53 59 53 2E 44 44 46 30 31 A5 03 3B 69 00 00 3B 69 00 00 57 44 37 51 BA CB 18 18 35 | 70 | 03 |

注意：表 4.80 中信息字段中的前 16 字节是无效字节，没有任何意义，保留为将来使用，用户不用理会；后 17 字节才是接触式 IC 卡的复位信息。

4.3.5 接触式 IC 卡停活

该命令是关闭接触式 IC 卡的电源和时钟。

声明： `uint8_t Cicc_Deactivation(void)`

1. 主机命令

命令类型(CmdClass): 0x05

命令代码(CmdCode): 'E'

信息长度(InfoLength): 0

信 息(Info): none

例 如： 关闭接触式 IC 卡电源和时钟的命令帧如表 4.88 所示

表 4.88 接触式 IC 卡停活命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 06 | 05 | 45 | 00 | none | B9 | 03 |

2. 从机回应

执行状态 (Status): 0 — 执行成功； 其他 — 警告或失败

信息长度(InfoLength): 0

信 息(Info): none

例 如： 接触式 IC 卡停活操作执行成功的回应帧如表 4.89 所示

表 4.89 接触式 IC 卡停活操作执行成功的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 05 | 00 | 00 | none | FC | 03 |

4.3.6 接触式 IC 卡协议和参数选择 (PPS)

该命令是冷复位或热复位之后且必须首先执行（协商模式下需要执行，专用模式不需要执行）。若对接触式 IC 卡不了解，建议使用 `Cicc_Reset()`（该命令自动处理了 PPS 命令），而不要使用 `Cicc_ColdReset()` + `Cicc_PPS()` 或 `Cicc_WarmReset()` + `Cicc_PPS()`。

注意：PPS 命令是修改通信协议和参数，必须在 IC 卡复位之后首先执行。修改的参数必须是 IC 卡支持的才可以。

声明： `uint8_t Cicc_PPS(const uint8_t *pPPS)`

1. 主机命令

命令类型(CmdClass): 0x05

命令代码(CmdCode): 'F'

信息长度(InfoLength): 0x04
 信 息(Info): PPS 参数 (4 字节)
 PPS[0] — 指定是否存在 PPS1、PPS2、PPS3
 PPS[0].3:0 — 保留
 PPS[0].4 = 1 — PPS1 存在; 0 — PPS1 不存在
 PPS[0].5 = 1 — PPS2 存在; 0 — PPS2 不存在
 PPS[0].6 = 1 — PPS3 存在; 0 — PPS3 不存在
 PPS[0].7 — 保留
 PPS[1] — F/D
 PPS[2] — N
 PPS[3] — 待定
 例 如: 将接触式 IC 卡通信波特率改为 115200bps (Fi 为 1; Di 为 8), 其他的不修改, 命令帧如表 4.90 所示

表 4.90 接触式 IC 卡协议和参数选择命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|-------------|-----|-----|
| 0A | 05 | 46 | 04 | 10 18 00 00 | BA | 03 |

2. 从机回应

执行状态 (Status): 0 — 执行成功; 其他 — 警告或失败
 信息长度(InfoLength): 0
 信 息(Info): none
 例 如: 接触式 IC 卡协议和参数选择操作执行成功的回应帧如表 4.91 所示

表 4.91 接触式 IC 卡协议和参数选择操作执行成功的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 05 | 00 | 00 | none | FC | 03 |

4.3.7 接触式 IC 卡传输协议 (T = 0)

该命令用于 T = 0 传输协议。若接触式 IC 卡的传输协议为 T = 0, 该命令等同于 Cicc_TPDU()。

声明: *uint8_t Cicc_TP0(const void *pSendBuf, uint32_t nSendBytes, void *pRcvBuf, uint32_t nRcvBufSize, uint32_t *pRcvBytes)*

1. 主机命令

命令类型(CmdClass): 0x05
 命令代码(CmdCode): 'G'
 信息长度(InfoLength): 1~272
 信 息(Info): 发送到 IC 卡的数据
 例 如: 通过 FID (文件标识符) 选择 MF (FID 为: 3F00)。选择文件的 APDU 如表 4.81 所示, 将其转换为数据流为: 00 A4 00

00 02 3F 00 00（不需要区分 APDU 的 4 种情况，‘3F00’在数据流中是以大端模式存放，即高字节在前），该命令能自动处理，其命令帧如表 4.92 所示

表 4.92 通过 FID 选择 MF（FID 为 ‘3F00’，T=0）的命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|-------------------------|-----|-----|
| 0E | 05 | 47 | 08 | 00 A4 00 00 02 3F 00 00 | 22 | 03 |

2. 从机回应

执行状态 (Status): 0 — 执行成功； 其他 — 警告或失败
 信息长度(InfoLength): 不同的卡回应的字节数不同
 信 息(Info): IC 卡回复的数据
 例 如: 选择 MF 操作执行成功的回应帧如表 4.93 所示

表 4.93 选择 MF（T=0）操作执行成功的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|--|-----|-----|
| 1F | 05 | 00 | 19 | 6F 15 84 0E 31 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 03 88 01 01 90 00 | D5 | 03 |

表 4.93 中的前 23 字节为 MF 的 FCI，最后 2 字节 ‘90 00’ 表示卡片处理成功。需要注意的是 Info 域的最后 2 字节表示卡片执行结果与回应帧中的 ‘Sataus’ 字段表示的不是同一状态，‘Sataus’ 字段表示是通信链路层的状态；而 Info 域的最后 2 字节表示卡片执行结果。

4.3.8 接触式 IC 卡传输协议（T = 1）

该命令用于 T=1 传输协议。若接触式 IC 卡的传输协议为 T=1，其等同于 Cicc_TPDU()。

声明: *uint8_t Cicc_TPDU(const void *pSendBuf, uint32_t nSendBytes,*
*void *pRcvBuf, uint32_t nRcvBufSize, uint32_t *pRcvBytes)*

1. 主机命令

命令类型(CmdClass): 0x05
 命令代码(CmdCode): ‘H’
 信息长度(InfoLength): 1~272
 信 息(Info): 发送到 IC 卡的数据
 例 如: 通过 FID（文件标识符）选择 MF（FID 为：3F00）。选择文件的 APDU 如表 4.81 所示，将其转换为数据流为：00 A4 00 00 02 3F 00 00（不需要区分 APDU 的 4 种情况，‘3F00’在数据流中是以大端模式存放，即高字节在前），该命令能自动处理，其命令帧如表 4.94 所示

表 4.94 通过 FID 选择 MF（FID 为 ‘3F00’，T=1）的命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|-------------------------|-----|-----|
| 0E | 05 | 48 | 08 | 00 A4 00 00 02 3F 00 00 | 2D | 03 |

2. 从机回应

执行状态 (Status): 0 — 执行成功; 其他 — 警告或失败
信息长度(InfoLength): 不同的卡回应的字节数不同
信 息(Info): IC 卡回复的数据
例 如: 选择 MF 操作执行成功的回应帧如表 4.95 所示

表 4.95 选择 MF (T=1) 操作执行成功的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|--|-----|-----|
| 1F | 05 | 00 | 19 | 65 15 84 0E 31 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 03 88 01 01 90 00 | DF | 03 |

表 4.95 中的前 23 字节为 MF 的 FCI，最后 2 字节 ‘90 00’ 表示卡片处理成功。需要注意的是 Info 域的最后 2 字节表示卡片执行结果与回应帧中的 ‘Sataus’ 字段表示的不是同一状态，‘Sataus’ 字段表示是通信链路层的状态；而 Info 域的最后 2 字节表示卡片执行结果。

4.4 ISO14443 (PICC) 卡类命令 (CmdClass = 0x06)

ISO14443 (PICC) 卡类命令总汇如表 4.96 所示。

表 4.96 ISO14443 (PICC) 卡类命令一览表

| 命令码 | 意义 |
|-----|----------------------------|
| ‘A’ | A 型卡请求 |
| ‘B’ | A 型卡防碰撞 |
| ‘C’ | A 型卡选择 |
| ‘D’ | A 型卡挂起 |
| ‘E’ | A 型卡 RATS |
| ‘F’ | A 型卡 PPS |
| ‘G’ | A 型卡解除激活 |
| ‘H’ | T=CL |
| ‘J’ | 数据交换 |
| ‘L’ | A 型卡复位 |
| ‘M’ | A 型卡激活 |
| ‘N’ | B 型卡激活 |
| ‘O’ | B 型卡复位 |
| ‘P’ | B 型卡请求 |
| ‘Q’ | B 型卡防碰撞 |
| ‘R’ | B 型卡修改传输属性 |
| ‘S’ | B 型卡挂起 |

前 4 条命令 (命令 A~D) 是 ISO14443-3A 标准定义的命令, 只要符合该标准的卡都应能发出响应; 中间 4 条命令 (命令 E~H) 为是 ISO14443-4 标准定义的命令。其中 A~D 命令和 Mifare S50/S70 卡类命令的 A~D 命令完全相同

4.4.1 A 型卡请求 (Cmd = A)

该命令用于 A 型卡的请求操作, 该命令的操作与 Mifare S50/S70 卡类的请求 (Cmd = A) 命令一样。

例 如: 请求天线范围内所有的 A 型卡。

主机命令: 07 06 41 01 52 EC 03。

4.4.2 A 型卡防碰撞 (Cmd = B)

该命令用于 A 型卡的防碰撞, 该命令的操作与 Mifare S50/S70 卡类的防碰撞 (Cmd = B) 命令一样。

例 如: 第一级防碰撞。

主机命令: 08 06 42 02 93 00 22 03。

4.4.3 A 型卡选择 (Cmd = C)

该命令用于 A 型卡的选择, 该命令的操作与 Mifare S50/S70 卡类的卡选择 (Cmd = C) 命令一样。

例 如: 第一级选择, UID 为 0xEB1C1814。

主机命令：0B 06 43 05 93 14 18 1C EB DC 03。

4.4.4 A 型卡挂起 (Cmd = D)

该命令用于 A 型卡的挂起，该命令的操作与 Mifare S50/S70 卡类的卡挂起 (Cmd = D) 命令一样。

例 如：将已激活的卡挂起，使之不响应请求空闲卡命令。

主机命令：06 06 44 00 BB 03。

4.4.5 A 型卡 RATS (Cmd = E)

RATS (request for answer to select) 是 ISO14443-4 协议的命令，模块发送 RATS，卡片发出 ATS (answer to select) 作为 RATS 的应答，在执行该命令前，必需先进行一次卡选择操作，且执行过一次 RATS 命令后，想再次执行 RATS 命令，必需先解除激活。

声明：uint8_t PiccA_RATS(uint8_t CID, void *pRATS,

uint32_t nRATSBufSize, uint32_t *pRATSBytes)

1. 主机命令

命令类型 (CmdClass): 0x06

命令代码 (CmdCode): 'E'

信息长度 (InfoLength): 0x01

信 息 (Info): CID (1 字节): 卡标识符 (card Identifier, 取值范围 0x00~0x0E)

例 如：向 PLUS CPU 卡发送 RATS 命令，CID 设备为 0x0A

表 4.97 A 型卡 RATS 命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 07 | 06 | 45 | 01 | 0A | B0 | 03 |

2. 从机应答

状 态 (Status): 0——成功，其它——失败

信息长度 (InfoLength): 0x0C (不同的卡，ATS 的字节数不同)

信 息 (Info): ATS

例 如：一张 SL3 的 PLUS CPU 卡会回应的 ATS

表 4.98 A 型卡响应 RATS 的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|--|-----|-----|
| 12 | 06 | 00 | 0C | 0C 75 77 80 02 C1 05 2F 2F 01 BC D6 | C4 | 03 |

4.4.6 A 型卡 PPS (Cmd = F)

PPS (protocol and parameter selection) 是 ISO14443-4 协议的命令，用于改变有关的专用协议参数，该命令不是必需的，命令只支持默认参数，即该命令的参数设置为 0 即可。在执行该命令前，必需先成功执行一次 RATS 命令。

声明：uint8_t PiccA_PPS(uint8_t DSI_DRI)

1. 主机命令

命令类型 (CmdClass): 0x06
 命令代码 (CmdCode): 'F'
 信息长度 (InfoLength): 0x01
 信 息 (Info): DSI_DRI (1 字节): 模块与卡通信波特率, 设置为 0 (106Kb/s)
 例 如: 将 PLUS CPU 卡与模块间的通信波特率设置为 106Kb/s

表 4.99 A 型卡 PPS 命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 07 | 06 | 46 | 01 | 00 | B9 | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: PLUS CPU 卡执行 PPS 成功后的回应

表 4.100 A 型卡响应 PPS 的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 06 | 00 | 00 | none | FF | 03 |

4.4.7 A 型卡解除激活 (Cmd = G)

该命令是 ISO14443-4 协议的命令, 用于将卡片置为挂起 (HALT) 状态, 处于挂起 (HALT) 状态的卡可以用“请求”命令 (请求代码为 ALL) 来重新激活卡, 只有执行“RATS”命令的卡才用该命令。

声明: *uint8_t PiccA_DeSelect(void)*

1. 主机命令

命令类型 (CmdClass): 0x06
 命令代码 (CmdCode): 'G'
 信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: 将激活的卡置为挂起状态

表 4.101 A 型卡解除激活命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 06 | 06 | 47 | 00 | none | B8 | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): 0
 信 息 (Info): none

例如：PLUS CPU 卡执行 PPS 成功后的回应

表 4.102 A 型卡响应解除激活的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 06 | 00 | 00 | none | FF | 03 |

4.4.8 T=CL (Cmd = H)

T=CL 是半双工分组传输协议，ISO14443-4 协议命令，用于读写器与卡片之间的数据交互，一般符合 ISO14443 协议的 CPU 卡均用该协议与读写器通信。调用该命令时只需要将 CPU 卡 COS 命令的数据作为输入即可，其他的如分组类型、卡标识符 CID、帧等待时间 FWT、等待时间扩展倍增因子 WTXM (waiting time extensionmultiplier)，等等由该命令自动完成。

声明： `uint8_t Picc_TPCL(const void *pSBuf, uint32_t nSBytes,
void *pRBuf, uint32_t nRBufSize, uint32_t *pRealBytes)`

1. 主机命令

命令类型 (CmdClass): 0x06

命令代码 (CmdCode): 'H'

信息长度 (InfoLength): COS 命令的长度

信息 (Info): COS 命令

例如：选择 FM1208 的 MF 标识符为 3F00，选择 COS 命令如下

表 4.103 FM1208 选择 MF 的命令编码

| 代码 | CLA | INS | P1 | P2 | Lc | Data | Le |
|----|-----|-----|----|----|----|-------|----|
| 值 | 00 | A4 | 00 | 00 | 02 | 3F 00 | — |

表 4.104 A 型卡 T=CL 命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|----------------------|-----|-----|
| 0D | 06 | 48 | 07 | 00 A4 00 00 02 3F 00 | 22 | 03 |

2. 从机应答

状态 (Status): 0——成功，其它——失败

信息长度 (InfoLength): COS 命令回应数据长度

信息 (Info): COS 命令回应数据

例如：FM1208 选择 MF 时响应的数据为嵌套的 TLV 格式的变长记录，其意义请参考《FMCOS 用户手册》

表 4.105 A 型卡响应 T=CL 的回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|---|-----|-----|
| 1C | 06 | 00 | 16 | 6F 15 84 0E 31 50 41 59 53 2E 44 44 46 30 31 A5 03 88 01 01 90 00 | FE | 03 |

4.4.9 数据交换 (Cmd = J)

该命令用读写器与卡片的数据交互，通过该命令可以实现读写卡器的所有功能。

声明: `uint8_t CD_ExchangeBlock(const ExchangeInputPara *pIn, ExchangeOutputPara *pOut)`

1. 主机命令

命令类型 (CmdClass): 0x06
命令代码 (CmdCode): 'J'
信息长度 (InfoLength): 交互数据块长度+2
信 息 (Info): 交互数据块 (其内容与实际使用的 CPU 卡有关)
WTXM_CRC (1 字节), 该字节内容如表 4.75 所示

表 4.106 WTXM_CRC 字节描述

| B7~B2 | B1 | B0 |
|-------|-----|--------|
| WTXM | RFU | CRC 禁能 |
| | 0 | CRC 使能 |

FWI (1 字节): 超时等待时间编码

超时时间 = ((0x01 << FWI) * 302us)

例 如: 向一张已被激活的 Mifare DESFire 卡发送“请求应答以选择 (RATS)”命令, 交互的数据块为该命令的命令帧 (0xE0, 0x50), 帧长 2 字节 (不包括 CRC 校验, 其中 0xE0 是 RATS 的命令编码, 0x50 的高半字节为 FSDI, 低半字节为 CID, FSDI=5 表示最大交互帧为 64 字节)

表 4.107 数据交互命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|-------------|-----|-----|
| 0A | 06 | 4A | 04 | E0 50 01 04 | 08 | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
信息长度 (InfoLength): 0x06
信 息 (Info): ATS
例 如: RATS 命令执行成功的回应

表 4.108 数据交互成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|-------------------|-----|-----|
| 0C | 06 | 00 | 06 | 06 77 81 02 80 00 | 81 | 03 |

4.4.10 A 型卡复位 (Cmd = L)

该命令是通过将载波信号关闭指定的时间, 再开启来实现卡片复位。

声明: `void CD_PauseCarrier(uint8_t pause_ms, uint8_t wait_ms)`

1. 主机命令

命令类型 (CmdClass): 0x06
命令代码 (CmdCode): 'L'
信息长度 (InfoLength): 0x01
信 息 (Info): 时间 (1 字节), 以毫秒为单位, 0 为一直关闭

例如：将载波信号关闭 1ms

表 4.109 卡复位命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 07 | 06 | 4C | 01 | 01 | B2 | 03 |

2. 从机应答

状态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信息 (Info): none

例如：执行卡复位成功模块的回应

表 4.110 卡复位成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 06 | 00 | 00 | none | FF | 03 |

3. 说明

该命令将天线信号关闭数毫秒, 若一直关闭, 则等到执行一个请求命令时打开。

4.4.11A 型卡激活 (Cmd = M)

该命令用于激活卡片, 是请求、防碰撞和选择三条命令的组合。

声明: `uint8_t MF_Activate(uint8_t mode, uint8_t reqCode, PiccAResetInfo *pResetInfo)`

1. 主机命令

命令类型 (CmdClass): 0x06

命令代码 (CmdCode): 'M'

信息长度 (InfoLength): 0x02

信息 (Info): 保留 (1 字节), 设置为 0

请求代码 (1 字节): 0x26~IDLE

0x52~ALL

例如：以 IDLE 方式激活卡

表 4.111 卡激活命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|-------|-----|-----|
| 08 | 06 | 4D | 02 | 00 26 | 98 | 03 |

2. 从机应答

状态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): Mifare1 S50、S70、Light 卡: 8 字节

Mifare0 UltraLight 卡: 11 字节

Mifare3 Desfire 卡: 11 字节

Plus CPU 卡: 8 字节或 11 字节

信息 (Info): 请求应答 ATQ (2 字节)

最后一级选择应答 SAK (1 字节)

序列号长度 (1 字节)

序列号 (N 字节, 由序列号长度决定)

例 如: 一张序列号为 0xEB1C1814 的 Mifare1 S50 卡返回的数据

表 4.112 卡激活成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|-------------------------|-----|-----|
| 0E | 06 | 00 | 08 | 04 00 08 04 14 18 1C EB | 0C | 03 |

4.4.12 B 型卡激活 (Cmd = N)

该命令用于激活 B 型卡片, 在调用该命令前, 需要先执行设备控制类的“设置 IC 卡接口协议 (工作模式) (Cmd = D)”, 把模块先配置成 TypeB 模式。

声明: `uint8_t PiccB_Activate(uint8_t reqCode, uint8_t AFI, PiccBResetInfo *pResetInfo)`

1. 主机命令

命令类型 (CmdClass): 0x06

命令代码 (CmdCode): 'N'

信息长度 (InfoLength): 0x02

信 息 (Info): 请求代码 (1 字节): 0x00~IDLE
0x08~ALL

应用标识 (1 字节): 默认为 0x00

例 如: 以 IDLE 方式激活卡

表 4.113 卡激活命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|-------|-----|-----|
| 08 | 06 | 4E | 02 | 00 00 | BD | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0x0C

信 息 (Info): UID 相关信息

例 如: 一张 TypeB 卡激活后返回的数据

表 4.114 卡激活成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|--|-----|-----|
| 12 | 06 | 00 | 0C | 70 05 34 07 00 00 00 00 00 81 C1 00 | E1 | 03 |

4.4.13 B 型卡复位 (Cmd = O)

该命令是通过将载波信号关闭指定的时间, 再开启来实现卡片复位, 其实现方式与 A 型卡复位一样。

声明: `void CD_PauseCarrier(uint8_t pause_ms, uint8_t wait_ms)`

1. 主机命令

| | |
|--------------------|----------------------------|
| 命令类型 (CmdClass): | 0x06 |
| 命令代码 (CmdCode): | 'O' |
| 信息长度 (InfoLength): | 0x01 |
| 信 息 (Info): | 时间 (1 字节), 以毫秒为单位, 0 为一直关闭 |
| 例 如: | 将载波信号关闭 1ms |

表 4.115 卡复位命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 07 | 06 | 4F | 01 | 01 | B1 | 03 |

2. 从机应答

| | |
|--------------------|---------------|
| 状 态 (Status): | 0——成功, 其它——失败 |
| 信息长度 (InfoLength): | 0 |
| 信 息 (Info): | none |
| 例 如: | 执行卡复位成功模块的响应 |

表 4.116 卡复位成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 06 | 00 | 00 | none | FF | 03 |

4.4.14 B 型卡请求 (Cmd = P)

该命令用于 B 型卡请求。

声明: `uint8_t PiccB_Request(uint8_t reqCode, uint8_t AFI, uint8_t N, PiccBResetInfo *pResetInfo)`

1. 主机命令

| | |
|--------------------|--|
| 命令类型 (CmdClass): | 0x06 |
| 命令代码 (CmdCode): | 'P' |
| 信息长度 (InfoLength): | 0x03 |
| 信 息 (Info): | 请求代码 (1 字节): 0x00~IDLE 0x08~ALL 应用标识 (1 字节): 默认为 0x00 时隙总数 (1 字节): 范围 0~4 |
| 例 如: | 以 IDLE 方式请求卡片 |

表 4.117 卡请求命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|----------|-----|-----|
| 09 | 06 | 50 | 03 | 00 00 00 | A3 | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
信息长度 (InfoLength): 0x0C

信 息 (Info): UID 相关信息
例 如: 一张 TypeB 卡请求成功后返回的数据

表 4.118 卡请求成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|--|-----|-----|
| 12 | 06 | 00 | 0C | 70 05 34 07 00 00 00 00 00 81 C1 00 | E1 | 03 |

4.4.15 B 型卡防碰撞 (Cmd = Q)

该命令用于 B 型卡的防碰撞，一般在请求成功结束后执行。

注：该命令是保留命令，本模块暂时不处理该命令。

声明： *uint8_t PiccB_SlotMarker(uint8_t N, PiccBResetInfo *pResetInfo);*

1. 主机命令

命令类型 (CmdClass): 0x06
命令代码 (CmdCode): 'Q'
信息长度 (InfoLength): 0x01
信 息 (Info): 时隙标记 (1 字节): 范围 2~16, 该参数值与请求命令的时隙总数有关系, 假如请求命令的时隙总数为 n, 侧该时隙标记 $N < 2^n$
例 如: 时隙标记为 4 防碰撞

表 4.119 卡防碰撞命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 07 | 06 | 51 | 01 | 04 | AA | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
信息长度 (InfoLength): 0x0C
信 息 (Info): UID 相关信息
例 如: 一张 TypeB 卡防碰撞成功后返回的数据

表 4.120 卡防碰撞成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|--|-----|-----|
| 12 | 06 | 00 | 0C | 70 05 34 07 00 00 00 00 00 81 C1 00 | E1 | 03 |

4.4.16 B 型卡修改传输属性 (Cmd = R)

该命令用于 B 型卡修改传输属性 (卡选择)。

声明： *uint8_t PiccB_Attrib(const void *pPUP, uint8_t CID, uint8_t proType, void *pRBuf, uint32_t nRBufSize, uint32_t *pRealBytes);*

1. 主机命令

命令类型 (CmdClass): 0x06
命令代码 (CmdCode): 'R'

信息长度 (InfoLength): 0x06

信 息 (Info): PUPI (4 字节): 卡片标识符

CID (1 字节): 取值范围为 0 - 14, 若不支持 CID, 则设置为 0

proType (1 字节): 支持的协议, 由请求回应中的 ProtocolType 指定

proType.3: PCD 与 PICC 是否继续通信

1~PCD 中止与 PICC 继续通信

0~PCD 与 PICC 继续通信

proType.2:1: PICC EOF 和 PCD SOF 间的最小延迟

11~10 etu + 512 / fs

10~10 etu + 256 / fs

01~10 etu + 128 / fs

00~10 etu + 32 / fs

proType.0: 是否遵循 ISO14443-4

1~遵循 ISO14443-4;

0~不遵循 ISO14443-4. (二代证必须为 1)

例 如: 选择 PUPI 为 0x07340570

表 4.121 卡选择命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|-------------------|-----|-----|
| 0C | 06 | 52 | 06 | 70 05 34 07 00 01 | E6 | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 卡选择成功的回应

表 4.122 卡选择成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 06 | 00 | 00 | none | FF | 03 |

4.4.17 B 型卡挂起 (Cmd = S)

该命令用于 B 型卡挂起, 在执行挂起命令前, 必需先执行成功过一次请求命令。执行挂起命令成功后, 卡片处于挂起状态, 模块必需通过 ALL 方式请求卡片, 而不能用 IDLE 方式请求。

声明: `uint8_t PiccB_Halt(uint8_t *pPUPI);`

1. 主机命令

命令类型 (CmdClass): 0x06

命令代码 (CmdCode): 'S'

信息长度 (InfoLength): 0x04

信 息 (Info): PUPI (4 字节): 4 字节标识符
例 如: 挂起 PUPI 为: 0x38492295 的卡片

表 4.123 卡挂起命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|-------------|-----|-----|
| 0A | 06 | 53 | 04 | 95 22 49 38 | 62 | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
信息长度 (InfoLength): 0
信 息 (Info): none
例 如: 卡挂起成功的回应

表 4.124 卡挂起成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 06 | 00 | 00 | none | FF | 03 |

4.5 PLUS CPU 卡类命令（CmdClass = 0x07）

PLUS CPU 卡类命令总汇如表 4.125 所示，该命令集包括了 PLUS CPU 卡 SL0（Security Level 0）、SL3 的命令，其中等级 1 的命令与 Mifare S50/S70 卡（M1）相同，所以不在本命令集中。

表 4.125 PLUS CPU 卡类命令一览表

| 命令码 | 意义 |
|-----|--|
| ‘B’ | SL0 个人化更新数据 |
| ‘C’ | SL0 提交个人化 |
| ‘J’ | SL3 首次验证（直接密钥验证） |
| ‘K’ | SL3 首次验证（E² 密钥验证） |
| ‘L’ | SL3 跟随验证（直接密钥验证） |
| ‘M’ | SL3 跟随验证（E² 密钥验证） |
| ‘N’ | SL3 复位验证 |
| ‘O’ | SL3 读数据块 |
| ‘P’ | SL3 写数据块 |
| ‘S’ | SL3 值块操作 |

卡片激活后，只有通过“首次验证”之后才能使用“跟随验证”，卡片激活后，则只需要第一次验证命令使用“首次验证”命令，之后的验证命令都可以使用“跟随验证”，当然也可以都是用“首次验证”；若执行“复位验证”，则“复位验证”之后的第一次验证，也必须使用“首次验证”命令。两种验证的区别在于使用的时机不同，“首次验证”所需要的时间比“跟随验证”的时间要长些。

PLUS CPU 卡的密钥 A/B 是通过地址的奇偶数来区分，AES 的密钥地址与数据块的扇区关系对应如下。

- 密钥 A 地址=0x4000 + 扇区 × 2
- 密钥 B 地址=0x4000 + 扇区 × 2 + 1

除扇区密钥外，其它密钥不分密钥 A/B，详细的 PLUS CPU 卡地址分配请参阅 PLUS CPU 卡的数据手册。

4.5.1 SL0 个人化更新数据（Cmd = B）

该命令用于 SL0（Security Level 0，安全等级 0）的 PLUS CPU 卡个人化，PLUS CPU 卡出厂时的安全等级为 SL0，该等级下，不需要任何验证就可以向卡里写数据，写入的数据是作为其它安全等级的初始值，例如：

向 SL0 的 0x0003 块写入 0xA0 0xA1 0xA2 0xA3 0xA4 0xA5 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF，当卡片升级到 SL1 后，扇区 0 的 A 密钥为 0xA0 0xA1 0xA2 0xA3 0xA4 0xA5，而不是默认的 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF，即可以在 SL0 修改卡片的默认数据和密钥。

注意：PLUS CPU 卡在 SL0 的存储器地址均为 2 字节，其中地址 0x0000 ~ 0x00FF 为用户数据块，与 Mifare S50/S70 卡的数据/密钥块一一对应，该命令是 ISO14443-4 的命令。

声明：*uint8_t PLUS_WritePersoTCL(uint32_t usBNr, uint8_t *pBuf)*

1. 主机命令

命令类型 (CmdClass): 0x07
 命令代码 (CmdCode): 'B'
 信息长度 (InfoLength): 0x12
 信 息 (Info): BNr (2 字节): PLUS CPU 卡存储器地址
 Data (16 字节): 数据/AES 密钥/配置字
 例 如: 更改 PLUS CPU 卡的主控密钥 (地址为 0x9000)

表 4.126 SL0 个人化更新数据命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|--|-----|-----|
| 18 | 07 | 42 | 12 | 00 90 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF | 20 | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: 更改主控密钥成功的回应

表 4.127 SL0 个人化更新数据成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 07 | 00 | 00 | none | FE | 03 |

4.5.2 SL0 提交个人化 (Cmd = C)

该命令用于 SL0 (Security Level 0, 安全等级 0) 的 PLUS CPU 卡提交个人化数据, 命令“SL0 个人化更新数据”只是更新卡中的数据, 但该数据还未生效, 用户还不能直接使用。“SL0 提交个人化”使更新的个人化数据生效。执行该命令后, PLUS CPU 卡的安全等级提高到 SL1 或者 SL3 (若是支持 SL1 的卡, 则执行该命令后卡片安全等级提高到 SL1; 若是只支持 SL0 和 SL3 的卡, 则执行该命令后卡片安全等级提高到 SL3)。

注意: 在 SL0 的 PLUS CPU 卡, 只有修改了以下地址才能执行“SL0 提交个人化”命令:

- 0x9000 (主控密钥)
- 0x9001 (配置块密钥)
- 0x9002 (SL2 提升密钥, 只有支持 SL2 的卡才有该密钥)
- 0x9003 (SL3 主控密钥, 只有支持 SL3 的卡才有该密钥)

该命令是 ISO14443-4 的命令

声明: *uint8_t PLUS_CommitPersoTCL(void)*

1. 主机命令

命令类型 (CmdClass): 0x07
 命令代码 (CmdCode): 'C'
 信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: 将已修改主控密钥、配置块密钥、SL2 提升密钥和 SL3 主控密钥卡的安全等级提高到 SL1

表 4.128 SL0 提交个人化命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 06 | 07 | 43 | 00 | none | BD | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 更改主控密钥成功的回应

表 4.129 SL0 提交个人化成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 07 | 00 | 00 | none | FE | 03 |

4.5.3 SL3 首次验证 (直接密钥验证) (Cmd = J)

该命令用于 SL3 PLUS CPU 卡的密钥验证, 验证的密钥通过该命令的参数输入。

声明: `uint8_t PLUS_SL3FirstAuth(uint32_t uiKNr, const uint8_t *pKey)`

1. 主机命令

命令类型 (CmdClass): 0x07

命令代码 (CmdCode): 'J'

信息长度 (InfoLength): 0x12

信 息 (Info): AES 密钥地址 (2 字节)

AES 密钥 (16 字节)

例 如: 用密钥“FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF”

验证 1 扇区的 AES 密钥 A (1 扇区的 AES 密钥 A 对应的密钥地址为 0x4002)

表 4.130 SL3 首次验证 (直接密钥验证) 命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|--|-----|-----|
| 18 | 07 | 4A | 12 | 02 40 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF | FA | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 验证密钥成功的回应

表 4.131 SL3 首次验证 (直接密钥验证) 成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 07 | 00 | 00 | none | FE | 03 |

4.5.4 SL3 首次验证 (E² 密钥验证) (Cmd = K)

该命令也是用于 SL3 PLUS CPU 卡的密钥验证，验证的密钥来自模块内部，掉电不丢失的数据。

声明： *uint8_t PLUS_SL3FirstAuthE2(uint32_t uiKNr, uint8_t KeySector)*

1. 主机命令

命令类型 (CmdClass): 0x07

命令代码 (CmdCode): 'K'

信息长度 (InfoLength): 0x03

信 息 (Info): AES 密钥地址 (2 字节)
密钥区号 (1 字节)

例 如: 用密钥 1 区的密钥验证 1 扇区的 AES 密钥 A (密钥地址为 0x4002)

表 4.132 SL3 首次验证 (E² 密钥验证) 命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|----------|-----|-----|
| 09 | 07 | 4B | 03 | 02 40 01 | FA | 03 |

2. 从机应答

状 态 (Status): 0——成功，其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 验证密钥成功的回应

表 4.133 SL3 首次验证 (E² 密钥验证) 成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 07 | 00 | 00 | none | FE | 03 |

4.5.5 SL3 跟随验证 (直接密钥验证) (Cmd = L)

该命令用于 SL3 PLUS CPU 卡的跟随密钥验证，验证的密钥来自命令参数，只有执行过“首次验证”命令成功后才能使用该命令。

声明： *uint8_t PLUS_SL3FollowingAuth(uint32_t uiKNr, const uint8_t *pKey)*

1. 主机命令

命令类型 (CmdClass): 0x07

命令代码 (CmdCode): 'L'

信息长度 (InfoLength): 0x12

信 息 (Info): AES 密钥地址 (2 字节)
AES 密钥 (16 字节)

例 如: 用密钥“FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF”
验证 1 扇区的 AES 密钥 A (1 扇区的 AES 密钥 A 对应的密钥地址为 0x4002)

表 4.134 SL3 跟随验证（直接密钥验证）命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|--|-----|-----|
| 18 | 07 | 4C | 12 | 02 40 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF | FC | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 验证密钥成功的回应

表 4.135 SL3 跟随验证（直接密钥验证）成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 07 | 00 | 00 | none | FE | 03 |

4.5.6 SL3 跟随验证（E² 密钥验证）（Cmd = M）

该命令用于 SL3 PLUS CPU 卡的跟随密钥验证, 验证的密钥来自模块内部掉电不丢失的数据, 只有执行过“首次验证”命令成功后才能使用该命令。

声明: *uint8_t PLUS_SL3FollowingAuth(uint32_t uiKNr, const uint8_t *pKey)*

1. 主机命令

命令类型 (CmdClass): 0x07

命令代码 (CmdCode): 'M'

信息长度 (InfoLength): 0x03

信 息 (Info): AES 密钥地址 (2 字节)

密钥区号 (1 字节)

例 如: 用密钥 1 区的密钥验证 1 扇区的 AES 密钥 A (密钥地址为 0x4002)

表 4.136 SL3 跟随验证（E² 密钥验证）命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|----------|-----|-----|
| 09 | 07 | 4D | 03 | 02 40 01 | FC | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 验证密钥成功的回应

表 4.137 SL3 跟随验证（E² 密钥验证）成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 07 | 00 | 00 | none | FE | 03 |

4.5.7 SL3 复位验证（Cmd = N）

该命令用于 PLUS CPU 卡通过首次验证后的使用过程中，复位读写计数器和验证等信息。

声明：*uint8_t PLUS_SL3ResetAuth(void)*

1. 主机命令

命令类型 (CmdClass): 0x07
命令代码 (CmdCode): 'N'
信息长度 (InfoLength): 0
信 息 (Info): none
例 如: 复位验证卡片的验证信息

表 4.138 SL3 复位验证命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|------|-----|-----|
| 06 | 07 | 4E | 00 | none | B0 | 03 |

2. 从机应答

状 态 (Status): 0——成功，其它——失败
信息长度 (InfoLength): 0
信 息 (Info): none
例 如: 验证密钥成功的回应

表 4.139 SL3 复位验证成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 07 | 00 | 00 | none | FE | 03 |

3. 说明

若执行“复位验证”命令，读写计数器和所有的认证信息都将清空，若还需要对卡片进行操作，则必需使用“首次验证”命令或者将卡片重新激活。

4.5.8 SL3 读数据块 (Cmd = O)

该命令用于读取 SL3 的数据块，在读数据块之前必需成功执行一次密钥验证。

声明：*uint8_t PLUS_SL3Read(uint8_t ucMode, uint32_t usBNr, uint8_t ucExt, uint8_t *pBuf)*

1. 主机命令

命令类型 (CmdClass): 0x07
命令代码 (CmdCode): 'O'
信息长度 (InfoLength): 0x04
信 息 (Info): 读模式 (1 字节): 0x30~命令有 MAC; 数据密文; 回应无 MAC
0x31~命令有 MAC; 数据密文; 回应有 MAC
0x32~命令有 MAC; 数据明文; 回应无 MAC
0x33~命令有 MAC; 数据明文; 回应有 MAC
0x34~命令无 MAC; 数据密文; 回应无 MAC
0x35~命令无 MAC; 数据密文; 回应有 MAC

0x36~命令无 MAC；数据明文；回应无 MAC

0x37~命令无 MAC；数据明文；回应无 MAC

起始块号（2 字节）

读的块数（1 字节）： 范围 1~3

例 如： 从块 4 开始以“命令有 MAC，数据明文，回应无 MAC”的方式读 1 块数据

表 4.140 SL3 读数据块命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|-------------|-----|-----|
| 0A | 07 | 4F | 04 | 33 04 00 01 | 8F | 03 |

2. 从机应答

状 态（Status）： 0——成功，其它——失败

信息长度（InfoLength）： 0x10

信 息（Info）： 数据（16 字节）

例 如： 从卡中读出的数据为“05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05”

表 4.141 SL3 读数据块成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|---|-----|-----|
| 16 | 07 | 00 | 10 | 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 | FE | 03 |

3. 说明

在验证成功之后，才能读相应的块数据，若不同扇区的密钥相同，则所验证的块号与读块号不必在同一个扇区内。当读的块数不为 1 时，且读的块包含了区尾块（密钥/配置块），则该读操作会自动跳过区尾块读到下一个扇区的数据，若需要对区尾块进行访问时，则需要将读的起始地址设为区尾块的地址，读的块数设置为 1 即可。

PLUS CPU 卡的数据块、区尾块和 Mifare S50/70 卡分配相同，只是将块地址扩展为 2 字节。读命令可以根据需要设置如下不同的安全模式。

- 0x30~命令有 MAC；数据密文；回应无 MAC
- 0x31~命令有 MAC；数据密文；回应无 MAC
- 0x32~命令有 MAC；数据明文；回应无 MAC
- 0x33~命令有 MAC；数据明文；回应无 MAC
- 0x34~命令无 MAC；数据密文；回应无 MAC
- 0x35~命令无 MAC；数据密文；回应无 MAC
- 0x36~命令无 MAC；数据明文；回应无 MAC
- 0x37~命令无 MAC；数据明文；回应无 MAC

注意：PLUS S 系列的卡只支持“命令有 MAC，数据明文，回应无 MAC”这一种模式，数据是否加密是指——读写模块与卡之间的数据通信是否加密，而不是模块与主控制器间的数据是否加密。

4.5.9 SL3 写数据块（Cmd = P）

该命令用于写 SL3 的数据块，在写数据块之前必需成功执行一次密钥验证。

声明：`uint8_t PLUS_SL3Write(uint8_t ucMode, uint32_t usBNr, uint8_t ucExt, const uint8_t *pBuf)`

1. 主机命令

命令类型 (CmdClass): 0x07

命令代码 (CmdCode): 'P'

信息长度 (InfoLength): 写的块数×16+4

信 息 (Info): 写模式 (1 字节): 0xA0~命令有 MAC; 数据密文; 回应无 MAC
0xA1~命令有 MAC; 数据密文; 回应无 MAC
0xA2~命令有 MAC; 数据明文; 回应无 MAC
0xA3~命令有 MAC; 数据明文; 回应无 MAC

起始块号 (2 字节)

写的块数 (1 字节): 范围 1~3

写入的数据 (写的块数×16 字节)

例 如: 将“05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05”用“命令有 MAC, 数据明文, 回应无 MAC”的方式写到第 0x0004 块。

表 4.142 SL3 写数据块命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|--|-----|-----|
| 1A | 07 | 50 | 14 | A3 04 00 01 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 | 00 | 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 16

信 息 (Info): 数据 (16 字节)

例 如: 从卡中读出的数据为“05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05”

表 4.143 SL3 写数据块成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 07 | 00 | 00 | none | FE | 03 |

3. 说明

在验证成功之后, 才能写相应的块数据, 若不同扇区的密钥相同, 则所验证的块号与写块号不必在同一个扇区内。当写的块数不为 1 时, 且写的块包含了区尾块 (密钥/配置块), 则该写操作会自动跳过区尾块写到下一个扇区的数据, 若需要对区尾块进行访问时, 则需要将写的起始地址设为区尾块的地址, 写的块数设置为 1 即可。

PLUS CPU 卡的数据块、区尾块和 Mifare S50/70 卡分配相同, 只是将块地址扩展为 2 字节。写命令可以根据需要设置如下不同的安全模式。

- 0xA0~命令有 MAC; 数据密文; 回应无 MAC
- 0xA1~命令有 MAC; 数据密文; 回应无 MAC
- 0xA2~命令有 MAC; 数据明文; 回应无 MAC

- 0xA3～命令有 MAC；数据明文；回应 有 MAC

注意：PLUS S 系列的卡只支持“命令有 MAC，数据明文，回应 有 MAC”这一种模式，数据是否加密是指——读写模块与卡之间的数据通信是否加密，而不是模块与主控制器间的数据是否加密。

4.5.10 SL3 值块操作（Cmd = S）

该命令用于写 SL3 的数据块，在写数据块之前必需成功执行一次密钥验证。

声明：*uint8_t PLUS_SL3ValueOperTran(uint8_t ucMode, uint32_t usSBNr, uint32_t usDBNr, long lValue)*

1. 主机命令

命令类型（CmdClass）： 0x07
命令代码（CmdCode）： ‘S’
信息长度（InfoLength）： 0x09
信 息（Info）： 值操作模式（1 字节）： 0xB7～加值
0xB9～减值的符号位被忽略
源块号（2 字节）
目的块号（2 字节）
值数据（4 字节）： 4 字节有符号数，低字节在前，高字节的符号位被忽略
例 如： 将 0x0004 块的值用“加值传输模式，回应 有 MAC”方式加上 0x01234567 其结果存放到 0x0005。

表 4.144 SL3 值块操作命令帧

| FrameLen | CType | Cmd | Length | Info | BCC | ETX |
|----------|-------|-----|--------|----------------------------|-----|-----|
| 0F | 07 | 53 | 09 | B7 04 00 05 00 67 45 23 01 | 1B | 03 |

2. 从机应答

状 态（Status）： 0——成功，其它——失败
信息长度（InfoLength）： 0
信 息（Info）： none
例 如： 加值成功模块的回应

表 4.145 SL3 值块操作成功回应帧

| FrameLen | CType | Status | Length | Info | BCC | ETX |
|----------|-------|--------|--------|------|-----|-----|
| 06 | 07 | 00 | 00 | none | FE | 03 |

注意：PLUS S 系列卡不支持该命令。

5. 新帧格式应用命令详述

ZLG600SP 系列模块的应用命令共分为以下几类。

- 设备控制类命令：
- Mifare S50/S70 卡类命令：
- ISO7816-3 类命令：
- ISO14443（PICC）卡类命令：
- PLUS CPU 卡类命令：

下面的章节，将以新帧格式进行命令的描述，如果要使用旧帧，需要自行切换。

5.1 设备控制类命令（CmdClass = 0x01）

设备控制类命令汇总如表 5.1 所示。

表 5.1 设备控制类命令一览表

| 命令码 | 意义 |
|-----|-----------------------|
| ‘A’ | 读设备信息 |
| ‘B’ | 配置 IC 卡接口 |
| ‘C’ | 关闭 IC 卡接口 |
| ‘D’ | 设置 IC 卡接口协议（工作模式） |
| ‘E’ | 装载 IC 卡密钥 |
| ‘F’ | 设置 IC 卡接口的寄存器值 |
| ‘G’ | 获取 IC 卡接口的寄存器值 |
| ‘H’ | 设置波特率 |
| ‘I’ | 设置 IC 卡接口的输出（输出时钟或载波） |
| ‘K’ | 设置新旧帧格式 |
| ‘U’ | 设置设备工作模式 |
| ‘V’ | 获取设备工作模式 |
| ‘a’ | 装载用户密钥 |
| ‘b’ | 读 E ² PROM |
| ‘c’ | 写 E ² PROM |

5.1.1 读设备信息（Cmd = A）

该命令能够获取模块的型号所用版本信息。

声明：*unsigned char GetDvcInfo(unsigned char *pDvcInfo)*

1. 主机命令

命令类型（CmdClass）： 0x01
 命令代码（CmdCode）： ‘A’
 信息长度（InfoLength）： 0
 信 息（Info）： none
 例 如： 获取模块的版本号

表 5.2 读设备信息命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 01 | 0041 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF0B |

2. 从机应答

状 态（Status）： 0——成功，其它——失败
 信息长度（InfoLength）： 0x14
 信 息（Info）： ‘ZLG600SP/T V1.00’
 例 如： 获取模块信息成功后，返回模块的版本信息

表 5.3 读设备信息回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|---|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 01 | 0000 | 0014 |
| Info | | | | | Checksum |
| 5A 4C 47 36 30 30 53 50 2F 54 20 56 31 2E 30 30 00 00 00 00 | | | | | FB59 |

5.1.2 配置 IC 卡接口 (Cmd = B)

该命令配置 IC 卡的接口形式，这个命令执行后，默认为 ISO14443A 形式。

声明: `uint8_t CD_Config(CDProtType type)`

1. 主机命令

命令类型 (CmdClass): 0x01

命令代码 (CmdCode): 'B'

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 配置 IC 卡的接口形式

表 5.4 配置 IC 卡接口命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 01 | 0042 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF0A |

2. 从机应答

状 态 (Status): 0——成功，其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 配置 IC 卡接口成功后的回应

表 5.5 配置 IC 卡接口成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 01 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF4B |

5.1.3 关闭 IC 卡接口 (Cmd = C)

该命令关闭 IC 卡接口，执行该命令后，IC 卡相关命令将不能工作，如果还需要执行读/写卡相关操作，必需先执行“配置 IC 卡接口”命令。

声明: `void CD_Close()`

1. 主机命令

命令类型 (CmdClass): 0x01

命令代码 (CmdCode): 'C'

表 5.6 关闭 IC 卡接口命令帧

2. 从机应答

表 5.7 关闭 IC 卡接口成功回应帧

5.1.4 设置 IC 卡接口协议（工作模式）（Cmd = D）

声明: `uint8 tCD SetISOType(CDProtType type)`

1. 主机命令

表 5.8 设置 IC 卡接口为 ISO14443-3B 命令帧

2. 从机应答

状态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: 配置 IC 卡接口为 ISO14443-3B 成功后的回应

表 5.9 设置 IC 卡接口为 ISO14443-3B 成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 01 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF4B |

5.1.5 装载 IC 卡密钥 (Cmd = E)

该命令是将输入的密钥保存在模块内部，模块掉电后该密钥不丢失，ZLG600SP/T 模块共能保存 A 密钥 16 组、B 密钥 16 组。

声明: *uint8_t CD_LoadKey(uint8_t keyType, uint8_t nKeySector, const void *pKey)*

1. 主机命令

命令类型 (CmdClass): 0x01
 命令代码 (CmdCode): 'E'
 信息长度 (InfoLength): 若是 6 字节密钥，则为 8
 若是 16 字节密钥，则为 18
 信 息 (Info): 密钥类型 (1 字节): 0x60——密钥 A
 0x61——密钥 B
 密钥区号 (1 字节): 取值范围 0~15
 密钥 (6 字节或 16 字节)
 例 如: 向密钥 01 区装载密钥 A: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF

表 5.10 向密钥 01 区装载密钥命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-------------------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 01 | 0045 | 0008 |
| Info | | | | | Checksum |
| 60 01 FF FF FF FF FF FF | | | | | F8A4 |

2. 从机应答

状 态 (Status): 0——成功，其它——失败
 信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: 装载密钥成功模块的回应

表 5.11 装载密钥成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 01 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF4B |

3. 说明

此命令是向模块内装载密码，并非改变 Mifare1 卡内扇区的密码。模块内有 6 个密码区（区号 0~15）可供装载，每个区分密钥 A（0x60）和密钥 B（0x61）两个，总共 32 个密码。装载成功后，可用该密钥对 Mifare1 卡或 PLUS CPU 卡进行验证。装载时若输入的密钥为 6 字节，则模块自动将 6 字节密钥采用复制拼接的方式扩展为 16 字节的密钥。例如密钥为：0xA0 0xA1 0xA2 0xA3 0xA4 0xA5，经扩展后为：0xA0 0xA1 0xA2 0xA3 0xA4 0xA5 0xA0 0xA1 0xA2 0xA3 0xA4 0xA5 0xA0 0xA1 0xA2 0xA3，扩展后的密钥用于 PLUS CPU 卡的 AES 验证，若需要提高安全性，则直接输入 16 字节的密钥。

若要改变 Mifare1 卡内的密钥，可在用原密码验证通过后，直接用写块数据指令，将密码块改写。

5.1.6 设置 IC 卡接口的寄存器值（Cmd = F）

该命令用于设置模块上读写卡芯片内部的寄存器值，通过该命令，我们可以实现很多现有命令不能完成的工作。

声明：*void CD_SetReg(uint8_t nRegAdr, uint8_t nRegVal)*

1. 主机命令

命令类型（CmdClass）： 0x01

命令代码（CmdCode）： ‘F’

信息长度（InfoLength）： 0x02

信 息（Info）： 寄存器地址（1 字节）： 取值范围 0x00~0x3F
寄存器值（1 字节）

例 如： 设置 TX1、TX2 天线驱动管脚的阻抗（0x12 寄存器）

表 5.12 设置寄存器值命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 01 | 0046 | 0002 |
| Info | | | | | Checksum |
| 12 3F | | | | | FEB3 |

2. 从机应答

状 态（Status）： 0——成功，其它——失败

信息长度（InfoLength）： 0

信 息（Info）： none

例 如： 设置寄存器值成功的回应

表 5.13 设置寄存器值成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 01 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF4B |

5.1.7 获取 IC 卡接口的寄存器值（Cmd = G）

该命令用于设置模块上读写卡芯片内部的寄存器值，通过该命令，我们可以实现很多现有命令不能完成的工作。

声明： *uint8_t CD_GetReg(uint8_t nRegAdr)*

1. 主机命令

命令类型 (CmdClass): 0x01

命令代码 (CmdCode): 'G'

信息长度 (InfoLength): 0x01

信 息 (Info): 寄存器地址 (1 字节): 取值范围 0x00~0x3F

例 如: 读取 TX1、TX2 天线驱动管脚的阻抗 (0x12 寄存器)

表 5.14 读取寄存器值命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 01 | 0047 | 0001 |
| Info | | | | | Checksum |
| 12 | | | | | FEF2 |

2. 从机应答

状 态 (Status): 0——成功，其它——失败

信息长度 (InfoLength): 0x01

信 息 (Info): 寄存器值

例 如: 读 0x12 寄存器返回的值

表 5.15 读取寄存器值成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 01 | 0000 | 0001 |
| Info | | | | | Checksum |
| 3F | | | | | FF0B |

5.1.8 设置波特率 (Cmd = H)

该命令用于在 UART 通信过程中改变通信的波特率，该命令执行完毕，等到返回成功信息以后才会使新设置的通信波特率生效，掉电后该设置值保留。

声明： *void PCDSetsBaud(unsigned char ucBaudNum)*

1. 主机命令

命令类型 (CmdClass): 0x01

命令代码 (CmdCode): 'H'

信息长度 (InfoLength): 0x01

信 息 (Info): 波特率编号 (1 字节): 取值范围 0~7 如表 5.16 所示

表 5.16 波特率编号对应表

| 编号 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----|------|-------|-------|-------|-------|--------|--------|--------|
| 波特率 | 9600 | 19200 | 28800 | 38400 | 57600 | 115200 | 172800 | 230400 |

例 如： 设置 UART 通信波特率为 115200

表 5.17 设置 UART 通信波特率为 115200 命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 01 | 0048 | 0001 |
| Info | | | | | Checksum |
| 05 | | | | | FEFE |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如： 设置波特率成功的返回

表 5.18 设置 UART 波特率成功响应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 01 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF4B |

5.1.9 设置 IC 卡接口的输出 (输出时钟或载波) (Cmd = I)

该命令用于设置天线驱动方式, 可以打开或关闭 TX1、TX2 中的任意一个管脚, 特别适用于双天线应用的设置。

声明: *void PCDSetsTX(unsigned char ucSelTX)*

1. 主机命令

命令类型 (CmdClass): 0x01

命令代码 (CmdCode): 'I'

信息长度 (InfoLength): 0x01

信 息 (Info): 天线驱动模式 (1 字节): 0x01——仅 TX1 驱动天线
0x02——仅 TX2 驱动天线
0x03——TX1、TX2 同时驱动天线

例 如： 将模块的天线驱动模式改为仅 TX2 输出

表 5.19 设置仅 TX2 驱动命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 01 | 0049 | 0001 |
| Info | | | | | Checksum |
| 02 | | | | | FF00 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none
例 如: 更改天线驱动模式成功的返回

表 5.20 更改天线驱动模式成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 01 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF4B |

5.1.10 设置新旧帧格式 (Cmd = K)

该命令用于设置模块通信的帧格式，前面章节已经描述过，ZLG600SP/T 模块支持新、旧两种帧格式，模块出厂默认上电是旧帧格式，可以通过该命令把模块设置成新帧格式，设置成功后掉电不丢失。

声明: *void PCDSerTX(unsigned char ucSelTX)*

1. 主机命令

命令类型 (CmdClass): 0x01
命令代码 (CmdCode): 'K'
信息长度 (InfoLength): 0x01
信 息 (Info): 新旧帧格式 (1 字节): 0x00——设置成旧帧格式
0x01——设置成新帧格式

例 如: 将模块的通信设置成旧帧格式

表 5.21 设置成旧帧格式命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 01 | 004B | 0001 |
| Info | | | | | Checksum |
| 00 | | | | | FF00 |

2. 从机应答

状 态 (Status): 0——成功，其它——失败
信息长度 (InfoLength): 0
信 息 (Info): none
例 如: 将模块设置成旧帧格式成功后返回如下内容。

表 5.22 设置新旧帧格式成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 01 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF4B |

注: 在串口通信方式时设置了新旧帧格式后将以原来的帧格式返回成功信息，但在 I²C 通信方式时，收到“设置新旧帧格式”命令后，即默认进入了设置后帧格式，不返回成功信息，下一次与模块通信需要使用设置后的帧格式与模块通信。

5.1.11 设置设备工作模式 (Cmd = U)

该命令用于设置模块上电时的工作模式，同时可以设置模块的从机地址，在一主多从的应用中，应通过该命令先设置好模块的从机地址。

1. 主机命令

- 命令类型 (CmdClass): 0x01
- 命令代码 (CmdCode): 'U'
- 信息长度 (InfoLength): 2 字节
- 信 息 (Info): 工作模式 (1 字节): 该字节包含了模块的各种工作模式，当模块上电没有硬件管脚设置模块工作模式时，通过判断该字节的定义进入相关模式，出厂默认是自动侦测的从机模式，该字节设置后掉电不丢失，字节的描述如下所示：

表 5.23 工作模式字节描述

| B7~B4 | B3~B0 |
|---|---|
| 模块主从模式： 0000: 从机模式 0001: 自动检测卡片模式 (主机模式) | 当模块上电时，没有硬件设置工作模式将通过该 4 位进入相应模式： 0000: 自动侦测模式 0001: I ² C 通信模式 0010: UART 通信(波特率固定为 19200) 0011: UART 通信(波特率固定为上次设置的波特率) 其它: 保留 |

从机地址 (1 字节): 该字节保存从机的地址，设置该字节只在没有硬件设置工作模式时才有效，其它通过硬件设定工作模式时该字节值由硬件决定，如果硬件没有设定从机地址则采用默认 0xB2 地址。另外，从机地址采用 I²C 地址的格式，最低位是读写位，所以从机地址最多只有 127 种。

- 例 如: 设置模块的从机地址为 0x02，工作模式为从机的自动侦测模式

表 5.24 设置设备工作模式的命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 01 | 0055 | 0002 |
| Info | | | | | Checksum |
| 00 02 | | | | | FEF3 |

2. 从机应答

- 状 态 (Status): 0——成功，其它——失败
- 信息长度 (InfoLength): 0
- 信 息 (Info): none
- 例 如: 设置模块工作模式成功的回应

表 5.25 设置设备工作模式成功的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 01 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF4B |

3. 说明

经过从机地址设置为 0x02 并返回设置成功后，下一命令帧的地址应为 0x02，如果地址保持原来的 0xB2，模块将不能响应。

5.1.12 获取设备工作模式（Cmd = V）

该命令用于获取模块的工作模式，包括从机地址。

1. 主机命令

命令类型（CmdClass）： 0x01

命令代码（CmdCode）： ‘V’

信息长度（InfoLength）： 0

信 息（Info）： none

例 如： 获取模块工作模式信息

表 5.26 获取设备工作模式的命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 01 | 0056 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FEF6 |

2. 从机应答

状 态（Status）： 0——成功，其它——失败

信息长度（InfoLength）： 2

信 息（Info）： 工作模式（1 字节）：详细说明见表 4.23
从机地址（1 字节）

例 如： 获取模块工作模式信息成功的回应

表 5.27 获取设备工作模式成功的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 01 | 0000 | 0002 |
| Info | | | | | Checksum |
| 00 B2 | | | | | FE97 |

5.1.13 装载用户密钥（Cmd = a）

该命令用于装载用户密钥，模块里面提供了 2 个 16 字节的存储空间用于保存用户密钥。

1. 主机命令

命令类型（CmdClass）： 0x01

命令代码（CmdCode）： ‘a’

信息长度 (InfoLength): 0x11
 信 息 (Info): 扇区号 (1 字节): 范围 0 ~1
 密钥数据 (16 字节)
 例 如: 往用户密钥存储 1 区存放 16 字节的密钥

表 5.28 装载用户密钥的命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|---|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 01 | 0061 | 0011 |
| Info | | | | | Checksum |
| 01 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF | | | | | EEE9 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: 往用户密钥存储 1 区存放 16 字节的密钥成功后的回应

表 5.29 装载用户密钥成功的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 01 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF4B |

注: 该用户密钥存储区不等同于 IC 卡密钥存储区, 2 个 16 字节大小的存储区, 16 字节的长度刚好与密钥长度相同, 方便用户扩展使用。

5.1.14 读 E²PROM (Cmd = b)

模块内部拥有一个 256Byte 的 E²PROM, 该存储空间掉电不丢失, 通过“读 E²PROM”、“写 E²PROM”命令可以对该存储器的数据进行读写。

1. 主机命令

命令类型 (CmdClass): 0x01
 命令代码 (CmdCode): ‘b’
 信息长度 (InfoLength): 0x02
 信 息 (Info): E²PROM 地址 (1 字节): 范围 0 ~255
 读取数据的长度 (1 字节)
 例 如: 读取 E²PROM 从 0x08 开始 8 字节的数据

表 5.30 读 E²PROM 的命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 01 | 0062 | 0002 |
| Info | | | | | Checksum |
| 08 08 | | | | | FED8 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): 0x08
 信 息 (Info): none
 例 如: 读取 E²PROM 里面从 0x08 地址开始 8 字节的数据的返回

表 5.31 读 E²PROM 成功的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-------------------------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 01 | 0000 | 0008 |
| Info | | | | | Checksum |
| 00 00 00 00 00 00 00 00 | | | | | FF43 |

5.1.15 写 E²PROM (Cmd = c)

该命令是往 E²PROM 里面写数据。

1. 主机命令

命令类型 (CmdClass): 0x01
 命令代码 (CmdCode): 'c'
 信息长度 (InfoLength): 要写的数据长度+2
 信 息 (Info): E²PROM 地址 (1 字节): 范围 0 ~255
 写数据的长度 (1 字节)
 要写入的数据信息 (n 字节)

例 如: 往 E²PRPM 里面 0x02 地址开始写入 4 字节的数据

表 5.32 写 E²PROM 的命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-------------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 01 | 0063 | 0006 |
| Info | | | | | Checksum |
| 02 04 FF FF FF FF | | | | | FAE1 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: 往 E²PROM 里面 0x02 地址开始写入 4 字节数据成功的返回

表 5.33 写 E²PROM 成功的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 01 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF4B |

5.2 Mifare S50/S70 卡类命令（CmdClass = 0x02）

Mifare S50/S70 卡类命令总汇如表 5.34 所示。

表 5.34 Mifare S50/S70 卡类命令一览表

| 命令码 | 意义 |
|-----|------------------------------------|
| ‘A’ | 请求 |
| ‘B’ | 防碰撞 |
| ‘C’ | 卡选择 |
| ‘D’ | 卡挂起 |
| ‘E’ | E² 密钥验证 |
| ‘F’ | 直接密钥验证 |
| ‘G’ | Mifare 卡读 |
| ‘H’ | Mifare 卡写 |
| ‘I’ | UltraLight 卡写 |
| ‘J’ | Mifare 值操作 |
| ‘L’ | 卡复位 |
| ‘M’ | 卡激活 |
| ‘N’ | 自动检测 |
| ‘O’ | 读自动检测数据 |
| ‘P’ | 设置值块的值 |
| ‘Q’ | 获取值块的值 |
| ‘X’ | 数据交互命令 |

前 4 条命令（命令 A~D）是 ISO14443A 标准定义的命令，只要符合该标准的卡都应能发出响应；中间 6 条命令（命令 E~J）为 Mifare1 卡的专用命令，只有先进行验证（命令 E、F）成功之后才能进行；后四条命令（L、M、N、O）为实用的扩展命令；X 命令为读写器与卡交换数据块，该命令用于 ISO14443-4 标准。

注意：

命令 C 和 M 命令都做了 Mifare 卡和 PLUS CPU 卡自动辨别功能，并根据卡的类型不同自动调用相应的命令，该功能使用户的卡片由 M1 卡升级到 PLUS CPU 卡不必修改，若要执行其它符合 CPU 卡操作，建议使用 PLUS CPU 卡类命令。

5.2.1 请求（Cmd = A）

该命令用于 Mifare 卡的请求操作。

声明：*uint8_t PiccA_Request(uint8_t reqCode, uint8_t *pATQ)*

1. 主机命令

命令类型（CmdClass）： 0x02

命令代码（CmdCode）： ‘A’

信息长度（InfoLength）： 0x01

信 息（Info）： 请求模式（1 字节）： 0x26——IDLE 模式
0x52——ALL 模式

例 如： 请求天线范围内所有的卡

表 5.35 请求卡命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 02 | 0041 | 0001 |
| Info | | | | | Checksum |
| 52 | | | | | FEB7 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): 0x02
 信 息 (Info): 请求应答 ATQ (2 字节, 低位在前)

表 5.36 ATQ 字节描述

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|-----|-----|-----|-----|-----|-----|----|----|--|-----|-----------------------|----|----|----|----|----|
| RFU | | | | | | | | UID 大小 00:4bytes 01:7bytes 10:10bytes | RFU | 如果有任何位为 1, 则为比特帧防冲突方式 | | | | | |

表 5.37 列举了各种类型的卡返回的 ATQ。

表 5.37 返回 ATQ 一览表

| 卡类型 | Mifare1 S50 | Mifare1 S70 | Mifare1 Light | Mifare0 UltraLight | Mifare3 DESFire | SHC1101 | SHC1102 | 11RF32 |
|-----|----------------|----------------|------------------|-----------------------|--------------------|---------|---------|--------|
| ATQ | 0x0004 | 0x0002 | 0x0010 | 0x0044 | 0x0344 | 0x0004 | 0x3300 | 0x0004 |

例 如: S50 卡返回的 ATQ

表 5.38 请求成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 02 | 0000 | 0002 |
| Info | | | | | Checksum |
| 04 00 | | | | | FF44 |

3. 说明

卡进入天线后, 从射频场中获取能量, 从而得电复位, 复位后卡处于 IDLE 模式, 用两种请求模式的任一种请求时, 此时的卡均能响应; 若对某一张卡成功进行了挂起操作 (Halt 命令或 DeSelect 命令), 则进入了 Halt 模式, 此时的卡只响应 ALL (0x52) 模式的请求, 除非将卡离开天线感应区后再进入。

注: DeSelect 为 ISO14443-4 命令。另外, 对 Mifare1 卡连续进行请求操作, 总是一次成功, 一次失败, 循环往复。

5.2.2 防碰撞 (Cmd = B)

该命令用于 Mifare 卡的防碰撞操作, 需要执行成功一次请求命令, 并返回请求成功, 才能进行防碰撞操作, 否则返回错误。

声明: `uint8_t PiccA_Anticoll(uint8_t mode, uint8_t selCode, uint8_t *pUID, uint8_t nBitCnt)`

该命令用于 Mifare 卡的选择操作。

声明: `uint8_t PiccA_Select(uint8_t selCode, const uint8_t *pUID, uint8_t *pSAK)`

1. 主机命令

命令类型 (CmdClass): 0x02

命令代码 (CmdCode): 'C'

信息长度 (InfoLength): 0x05

信 息 (Info): 选择代码 (1 字节): 0x93——第一级防碰撞

0x95——第二级防碰撞

0x97——第三级防碰撞

UID (4 字节): 前一个防碰撞命令返回的 UID

例 如: 第一级选择, UID 为 0xEB1C1814

表 5.41 卡选择命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|----------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 02 | 0043 | 0005 |
| Info | | | | | Checksum |
| 93 14 18 1C EB | | | | | FD3D |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0x01

信 息 (Info): 选择应答 SAK, 如表 5.42 所示, 其中 Bit 2 位是 Cascade 位,
表示 UID 是否完整。 若 Bit 2 = 0, 表示 UID 完整

若 Bit 2 = 1, 表示 UID 不完整, 还有部分 UID 未读出

表 5.42 返回 SAK 一览表

| 卡类型 | Mifare1 S50 | Mifare1 S70 | Mifare1 Light | Mifare0 UltraLight | Mifare3 DESFire | SHC1101 | SHC1102 | 11RF32 |
|-----|----------------|----------------|------------------|-----------------------|--------------------|---------|---------|--------|
| SAK | 0x08 | 0x18 | 0x01 | 0x04 | 0x24 | 0x22 | — | 0x08 |

例 如: 返回 S50 卡应答

表 5.43 卡选择成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 02 | 0000 | 0001 |
| Info | | | | | Checksum |
| 08 | | | | | FF41 |

3. 说明

卡的序列号长度有三种: 4 字节、7 字节和 10 字节。4 字节的只要用一级选择即可得到完整的序列号, 如 Mifare1 S50/S70 等; 7 字节的要求用二级选择才能得到完整的序列号, 前一级所得到的序列号的最低字节为级联标志 0x88, 在序列号内只有后 3 字节可用, 后一级选择能得到 4 字节序列号, 两者按顺序连接即为 7 字节序列号, 如 UltraLight 和 DesFire 等;

10 字节的以此类推，但至今还未发现此类卡。

在程序中可用 SAK.2 位来判断是还有序列号未读出，如 if(SAK & 0x04){...}。

5.2.4 卡挂起 (Cmd = D)

该命令用于 Mifare 卡的挂起操作，使所选择的卡进入 HALT 状态，在 HALT 状态下，卡将不响应读卡器发出的 IDLE 模式的请求，除非将卡复位或离开天线感应区后再进入。但它会响应读卡器发出的 ALL 请求。

声明：uint8_t PiccA_Halt(void)

1. 主机命令

命令类型 (CmdClass): 0x02

命令代码 (CmdCode): 'D'

信息长度 (InfoLength): 0

信 息 (Info): none

例 如：将已激活的卡挂起，使之不响应请求空闲卡命令

表 5.44 卡挂起命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 02 | 0044 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF07 |

2. 从机应答

状 态 (Status): 0——成功，其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如：挂起命令执行成功的回应

表 5.45 卡挂起成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 02 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF4A |

5.2.5 E² 密钥验证 (Cmd = E)

该命令用模块内部已存入的密钥与卡的密钥进行验证，所以使用该命令前，应先用“装载 IC 卡密钥”命令把密钥成功载入模块内，另外，需要验证的卡的扇区号不必与模块内密钥区号相等。

声明：uint8_t MF_Authent(uint8_t mode, const void *pKey, const uint8_t *pUID, uint8_t nBlock)

1. 主机命令

命令类型 (CmdClass): 0x02

命令代码 (CmdCode): 'E'

信息长度 (InfoLength): 0x07

0x61——密钥 B

卡序列号（4 字节）

密钥（6 字节或 16 字节）

卡块号（1 字节）： S50（0～63）

S70（0～255）

PLUS CPU 2K（0～127）

PLUS CPU 4K（0～255）

例 如： 用密钥“0xFF 0xFF 0xFF 0xFF 0xFF 0xFF”验证序列号为 0xEB1C1814 的卡的块 4

注：PLUS CPU 系列的卡的卡号有 4 字节和 7 字节之分，对于 7 字节卡号的卡，只需要将卡号的高 4 字节（等级 2 防碰撞得到的卡号）作为验证的卡号即可。

表 5.48 直接密钥验证命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-------------------------------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 02 | 0046 | 000C |
| Info | | | | | Checksum |
| 60 14 18 1C EB FF FF FF FF FF FF 04 | | | | | F768 |

2. 从机应答

状 态（Status）： 0——成功，其它——失败

信息长度（InfoLength）： 0

信 息（Info）： none

例 如： 验证成功返回的信息

表 5.49 直接密钥验证成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 02 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF4A |

5.2.7 Mifare 卡读（Cmd = G）

该命令对 Mifare 卡进行读操作，读之前必需成功进行密钥验证。

声明： *uint8_t MF_Read(uint8_t nStartBlock, uint8_t nBlockNum, void *pBuf)*

1. 主机命令

命令类型（CmdClass）： 0x02

命令代码（CmdCode）： ‘G’

信息长度（InfoLength）： 0x01

信 息（Info）： 卡块号（1 字节）： S50（0～63）

S70（0～255）

PLUS CPU 2K（0～127）

PLUS CPU 4K（0～255）

例如：读块 4 的数据

表 5.50 Mifare 卡读命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 02 | 0047 | 0001 |
| Info | | | | | Checksum |
| 04 | | | | | FEFF |

2. 从机应答

状态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0x10

信息 (Info): 块数据 (16 字节)

例如：从卡的块 4 读出数据为: “0x7F 0x4B 0xD8 0x37 0xAA 0x99 0xF3 0xE0 0xA5 0xD9 0x93 0x70 0x8F 0x89 0xE2 0x64”

表 5.51 Mifare 读成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|---|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 02 | 0000 | 0010 |
| Info | | | | | Checksum |
| 7F 4B D8 37 AA 99 F3 E0 A5 D9 93 70 8F 89 E2 64 | | | | | F56C |

3. 说明

在验证成功之后, 才能读相应的块数据, 所验证的块号与读块号必须在同一个扇区内, Mifare1 卡从块号 0 开始按顺序每 4 个块 1 个扇区, 若要对一张卡中的多个扇区进行操作, 在对某一扇区操作完毕后, 必须进行一条读命令才能对另一个扇区直接进行验证命令, 否则必须从请求开始操作。

对于 PLUS CPU 卡, 若下一个读扇区的密钥和当前扇区的密钥相同, 则不需要再次验证密钥, 直接读即可。

5.2.8 Mifare 卡写 (Cmd = H)

该命令对 Mifare 卡进行写操作, 写之前必需成功进行密钥验证。

声明: `uint8_t MF_Write(uint8_t nStartBlock, uint8_t nBlockNum, const void *pBuf)`

1. 主机命令

命令类型 (CmdClass): 0x02

命令代码 (CmdCode): 'H'

信息长度 (InfoLength): 0x11

信息 (Info): 卡块号 (1 字节): S50 (0~63)
S70 (0~255)
PLUS CPU 2K (0~127)
PLUS CPU 4K (0~255)

数据 (16 字节)

例如：向块 4 写入 16 字节数据 “0x00 0x01 0x02 0x03 0x04 0x05

0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F”

表 5.52 Mifare 卡写命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|--|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 02 | 0048 | 0011 |
| Info | | | | | Checksum |
| 04 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F | | | | | FE76 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: 数据成功写入卡片模块的回应

表 5.53 Mifare 写成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 02 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF4A |

3. 说明

对卡内某一块进行验证成功后, 即可对同一扇区的各个进行写操作 (只要访问条件允许), 其中包括位于扇区尾的密码块, 这是更改密码的唯一方法。对于 PLUS CPU 卡等级 2、3 的 AES 密钥则是在其他位置修改密钥。

5.2.9 UltraLight 卡写 (Cmd = I)

该命令对 UltraLight 卡进行写操作。

声明: *uint8_t MF0_ULWrite(uint8_t nStartBlock, uint8_t nBlockNum, const void *pBuf)*

1. 主机命令

命令类型 (CmdClass): 0x02
 命令代码 (CmdCode): 'I'
 信息长度 (InfoLength): 0x05
 信 息 (Info): 卡块号 (1 字节): 1~15
 数据 (4 字节)
 例 如: 写块 4 数据

表 5.54 UltraLight 卡写命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|----------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 02 | 0049 | 0005 |
| Info | | | | | Checksum |
| 04 05 05 05 05 | | | | | FEE5 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

| | |
|--------------------|---------------|
| 信息长度 (InfoLength): | 0 |
| 信 息 (Info): | none |
| 例 如: | 数据成功写入卡片模块的响应 |

表 5.55 UltraLight 卡写成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 02 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF4A |

3. 说明

此命令只对 UltraLight 卡有效，对 UltraLight 卡进行读操作与 Mifare1 卡一样。

5.2.10 Mifare 值操作 (Cmd = J)

该命令对 Mifare 卡的值块进行加减操作。

声明: *uint8_t MF1_Value(uint8_t operMode, uint8_t nSourceBlock,*
int32_t nValue, uint8_t nDestinationBlock)

1. 主机命令

| | |
|--------------------|--|
| 命令类型 (CmdClass): | 0x02 |
| 命令代码 (CmdCode): | 'J' |
| 信息长度 (InfoLength): | 0x07 |
| 信 息 (Info): | 模式 (1 字节): 0xC0~减 0xC1~加 卡块号 (1 字节): S50 (0~63) S70 (0~255) PLUS CPU 2K (0~127) PLUS CPU 4K (0~255) |
| | 值 (4 字节有符号数, 低字节在先) |
| | 传输块号 (1 字节) |

例 如： 将块 4 的值减 1，其结果保存到块 5

表 5.56 Mifare 值操作命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|----------------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 02 | 004A | 0007 |
| Info | | | | | Checksum |
| C0 04 01 00 00 00 05 | | | | | FE30 |

2. 从机应答

| | |
|--------------------|--------------|
| 状 态 (Status): | 0——成功，其它——失败 |
| 信息长度 (InfoLength): | 0 |
| 信 息 (Info): | none |

例如：值块操作成功后模块的回应

表 5.57 Mifare 值操作成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 02 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF4A |

3. 说明

要进行此类操作，块数据必须要有值块的格式，可参考 NXP 的相关文档。若卡块号与传输块号相同，则将操作后的结果写入原来的块内；若卡块号与传输块号不相同，则将操作后的结果写入传输块内，结果传输块内的数据被覆盖，原块内的值不变。处于等级 2 的 PLUS CPU 卡不支持值块操作，等级 1、3 支持。

5.2.11 卡复位 (Cmd = L)

该命令是通过将载波信号关闭指定的时间，再开启来实现卡片复位。

声明：void CD_PauseCarrier(uint8_t pause_ms, uint8_t wait_ms)

1. 主机命令

命令类型 (CmdClass): 0x02

命令代码 (CmdCode): 'L'

信息长度 (InfoLength): 0x01

信息 (Info): 时间 (1 字节)，以毫秒为单位，0 为一直关闭

例如：将载波信号关闭 1ms

表 5.58 卡复位命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 02 | 004C | 0001 |
| Info | | | | | Checksum |
| 01 | | | | | FEFD |

2. 从机应答

状态 (Status): 0——成功，其它——失败

信息长度 (InfoLength): 0

信息 (Info): none

例如：执行卡复位成功模块的回应

表 5.59 卡复位成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 02 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF4A |

3. 说明

该命令将天线信号关闭数毫秒，若一直关闭，则等到执行一个请求命令时打开。

5.2.12 卡激活 (Cmd = M)

该命令用于激活卡片，是请求、防碰撞和选择三条命令的组合。

声明: *uint8_t MF_Activate(uint8_t mode, uint8_t reqCode, PiccAResetInfo *pResetInfo)*

1. 主机命令

命令类型 (CmdClass): 0x02
命令代码 (CmdCode): 'M'
信息长度 (InfoLength): 0x02
信 息 (Info): 保留 (1 字节), 设置为 0
请求代码 (1 字节): 0x26~IDLE
0x52~ALL

例 如: 以 IDLE 方式激活卡

表 5.60 卡激活命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 02 | 004D | 0002 |
| Info | | | | | Checksum |
| 00 26 | | | | | FED6 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
信息长度 (InfoLength): Mifare1 S50、S70、Light 卡: 8 字节
Mifare0 UltraLight 卡: 11 字节
Mifare3 Desfire 卡: 11 字节
Plus CPU 卡: 8 字节或 11 字节
信 息 (Info): 请求应答 ATQ (2 字节)
最后一级选择应答 SAK (1 字节)
序列号长度 (1 字节)
序列号 (N 字节, 由序列号长度决定)
例 如: 一张序列号为 0xEB1C1814 的 Mifare1 S50 卡返回的数据

表 5.61 卡激活成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-------------------------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 02 | 0000 | 0008 |
| Info | | | | | Checksum |
| 04 00 08 04 14 18 1C EB | | | | | FDFE |

5.2.13 自动检测 (Cmd = N)

该命令用于卡片的自动检测, 执行该命令成功后, 在 UART 模式下, 模块将主动发送读取到卡片的数据。

声明: *unsigned char PiccAutoDetect(unsigned char ucADMode, unsigned char ucTxMode,*

unsigned char ucReqCode, unsigned char ucAuthMode,

*unsigned char ucKeyType, unsigned char *pKey,
unsigned char ucBlock)*

1. 主机命令

命令类型 (CmdClass): 0x02

命令代码 (CmdCode): 'N'

信息长度 (InfoLength): 若验证模式 = “E”, 则为 7
若验证模式 = “F”, 则为 12 或 22
若验证模式 = 0, 则为 4

信 息 (Info): 自动检测模式 *ADMode* (1 字节): 该字节内容如表 5.62 所示

表 5.62 *ADMode* 字节位描述

| B7~B4 | B3 | B2 | B1 | B0 |
|-------------|--|------------------------------|---|--|
| RFU 0000 | 执行完一次自动检测后的动作 2: 无动作 3: 最后执行 Halt 命令 | 数据输出后 2: 不继续检测 3: 继续检测 | 当 UART 接口, 检测到有卡时 2: 不产生中断 3: 产生中断, 当串口发送数据完毕后, 中断消失; 当 I ² C 接口时此位无效, 应设置为 0, 因肯定产生中断 | 当 UART 接口, 检测到有卡时 2: 串口不发送 3: 串口主动发送, 发送的数据格式见“检测卡回应格式”。当 I ² C 接口时此位无效, 应设置为 0, 因为是从模式 |

天线驱动方式 *TxMode* (1 字节): 字节描述如表 5.63 所示

表 5.63 *TxMode* 字节位描述

| B7~B2 | B1 | B0 |
|---------------|--|----|
| RFU 000000 | 00: TX1、TX2 交替驱动 01: 仅 TX1 驱动 10: 仅 TX2 驱动 11: TX1、TX2 同时驱动 | |

请求代码 *ReqCode* (1 字节): 0x26~IDLE
0x52~ALL

验证模式 *AuthMode* (1 字节): 'E'~用 E2 密钥验证
'F'~用直接密钥验证
0 ~不验证

密钥 AB *KeyType* (1 字节): 0x60~密钥 A
0x61~密钥 B

密钥 *Key*: 若验证模式为'E', 则为密钥区号 (1 字节)
若验证模式为'F', 则为密钥 (6 或 16 字节)

卡块号 *Block* (1 字节): S50 (0~63)

S70 (0~255)

PLUS CPU 2K (0~127)

PLUS CPU 4K (0~255)

例 如： 设置模块检测到有卡时产生中断，串口输出，以 IDLE 方式激活卡，用直接密码验证密钥 A（密码为 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF），读出第 1 块数据内容

表 5.64 自动检测命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-------------------------------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 02 | 004E | 000C |
| Info | | | | | Checksum |
| 03 03 26 46 60 FF FF FF FF FF FF 01 | | | | | F824 |

2. 从机应答

状 态 (Status): 0——成功，其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如： 模块设置自动检测成功的回应

表 5.65 设置自动检测成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 02 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF4A |

3. 检测卡回应格式

“从机应答”只是说明模块设置成自动检测成功，在串口通信方式下，自动检测模式使能后，若允许串口主动发送（即 *ADMode.0*=1），有卡靠近模块，模块将自动把检测到卡的相应信息按如下的格式发送。

状 态 (Status): 0——成功，其它——失败

信息长度 (InfoLength): 若验证命令不为 0，则为：21+序列号长度
若验证命令为 0，则为：5+序列号长度

信 息 (Info): 天线驱动 *TxDrv* (1 字节): 如表 5.66 所示

表 5.66 *TxDrv* 字节位描述

| B7~B2 | B1 | B0 |
|---------------|--|----|
| RFU 000000 | 00: TX1、TX2 交替驱动 01: 仅 TX1 驱动 10: 仅 TX2 驱动 11: TX1、TX2 同时驱动 | |

请求应答 *ATQ* (2 字节)

选择应答 *SAK* (1 字节)

序列号长度 *UIDLen* (1 字节)

序列号 UID（长度为各种卡序列号的实际长度）

块数据：若验证命令不为 0，则块数据为 16 字节

若验证命令为 0，则块数据为 0 字节

例 如： 检测到序列号为 0xEB1C1814 的 S50 卡，并读出块 1 数据

表 5.67 检测卡成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|--|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 02 | 0000 | 0019 |
| Info | | | | | Checksum |
| 03 04 00 08 04 14 18 1C EB 14 18 1C EB FB 88 04 00 47 C1 24 37 E1 00 11 06 | | | | | F8D6 |

4. 说明

执行自动检测命令成功后，并且读取卡片信息成功返回，整个过程相当于以下命令的组合：请求——防碰撞——选择——验证（若 $AuthMode \neq 0$ ）——读取（若 $AuthMode \neq 0$ ）——挂起（若 $AuthMode.3=1$ ）。当输入的密钥为 6 字节时，模块内部将按 $Key[0:15]=pKey[0:5]pKey[0:5]pKey[0:3]$ 模式扩展。

串口主动发送之后，模块状态由 $ADMode.2$ 位来决定，若 $ADMode.2=1$ ，则自动进入自动检测模式；否则结束自动检测模式，主机可以发送其它任何命令。若 $ADMode.1=1$ ，则模块检测到卡后产生中断信号，可以通过读取自动检测数据命令（ $Cmd = 0$ ）读取。

当为 I²C 接口通信时，因模块为从模式，所以不主动发送数据，但肯定输出中断信号，应使 $ADMode.1=0$ ，产生中断后，主机可以通过以下两种方式读回数据。

- 一是直接读取，这样读取之后的模块状态由 $ADMode.2$ 位来决定：若 $ADMode.2=1$ ，则继续进入自动检测模式；否则结束自动检测模式，主机可以发送其它任何命令。
- 二是通过读取自动检测数据命令（ $Cmd=0$ ）读取数据之后的模块状态由该函数的参数来决定：在自动检测模式期间，主机可以随时发出读取自动检测数据命令，读取自动检测数据、查询自动检测状态、取消或继续自动检测；验证和读命令只对 Mifare1 卡和 PLUS CPU 卡有效。

注：在自动检测期间，若主机发送任何除读自动检测数据外的，且数据长度小于 3（帧长小于 9）的命令，将退出自动检测模式，如请求 $Picca-Request(0)$ 命令，在此期间，模块将不接收数据长度大于 2（帧长大于 8）的命令。

5.2.14 读自动检测数据（ $Cmd = 0$ ）

该命令用于读取自动检测的数据，特别适合于 I²C 通信模式下使用。通过该读取自动检测数据命令，可以决定读取数据后是否继续检测。

声明： $INT8U ReadAutoDetect(INT8U ReadMode);$

1. 主机命令

命令类型（CmdClass）： 0x02

命令代码（CmdCode）： ‘0’

信息长度（InfoLength）： 0x01

信 息（Info）： 读模式（1 字节），该字节内容如表 5.68 所示

表 5.68 读模式字节描述

| B7~B1 | B0 |
|---------------|-------------------------------|
| RFU 000000 | 数据发回之后： 00：取消检测 01：继续检测 |

例 如： 读取自动检测数据之后取消自动检测

表 5.69 读自动检测数据命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 02 | 004F | 0001 |
| Info | | | | | Checksum |
| 00 | | | | | FEFB |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0x19

信 息 (Info): 自动检测读取成功保存的信息

例 如： 读自动检测数据成功的回应

表 5.70 读自动检测数据成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|--|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 02 | 0000 | 0019 |
| Info | | | | | Checksum |
| 03 04 00 08 04 14 18 1C EB 14 18 1C EB FB 88 04 00 47 C1 24 37 E1 00 11 06 | | | | | F8D6 |

5.2.15 设置值块的值 (Cmd = P)

该命令用于设置值块的值。

声明: `uint8_t MF1_SetValue(uint8_t nBlock, int32_t nValue);`

1. 主机命令

命令类型 (CmdClass): 0x02

命令代码 (CmdCode): 'P'

信息长度 (InfoLength): 0x05

信 息 (Info): 块地址 (1 字节): 将要写入数值的块地址
块值 (4 字节): 有符号的 32 位数据, 低字节在前

例 如： 将 0x05 值块地址的值设置为 0x03

表 5.71 设置值块的值命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|----------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 02 | 0050 | 0005 |
| Info | | | | | Checksum |
| 05 03 00 00 00 | | | | | FEEE |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 将 0x05 值块地址的值设置为 0x03 成功后的返回

表 5.72 设置值块的值成功的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 02 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF4A |

5.2.16 获取值块的值 (Cmd = Q)

该命令用于获取值块的值，值块里面的数据只有是按照值格式存储时，才能通过该命令读取成功，否则返回失败。

声明: `uint8_t MFI_GetValue(uint8_t nBlock, int32_t *pValue);`

1. 主机命令

命令类型 (CmdClass): 0x02

命令代码 (CmdCode): 'Q'

信息长度 (InfoLength): 0x01

信 息 (Info): 块地址 (1 字节): 将要读取数值的块地址

例 如: 读 0x06 值块地址的值

表 5.73 获取值块的值命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 02 | 0051 | 0001 |
| Info | | | | | Checksum |
| 06 | | | | | FEF3 |

2. 从机应答

状 态 (Status): 0——成功，其它——失败

信息长度 (InfoLength): 4

信 息 (Info): 块值 (4 字节): 有符号的 32 位数据，低字节在前

例 如: 读 0x06 值块地址的值成功后的返回

表 5.74 获取值块的值成功的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-------------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 02 | 0000 | 0004 |
| Info | | | | | Checksum |
| 01 00 00 00 | | | | | FF45 |

5.2.17 数据交互命令 (Cmd = X)

该命令用读写器与卡片的数据交互，通过该命令可以实现读写卡器的所有功能。

声明: `uint8_t CD_ExchangeBlock(const ExchangeInputPara *pIn, ExchangeOutputPara *pOut)`

1. 主机命令

命令类型 (CmdClass): 0x02

命令代码 (CmdCode): 'X'

信息长度 (InfoLength): 交互数据块长度+2

信 息 (Info): 交互数据块 (其内容与实际使用的 CPU 卡有关)

WTXM_CRC (1 字节), 该字节内容如表 5.75 所示

表 5.75 WTXM_CRC 字节描述

| B7~B2 | B1 | B0 |
|-------|----------|------------------|
| WTXM | RFU 0 | CRC 禁能 CRC 使能 |

FWI (1 字节): 超时等待时间编码

超时时间 = ((0x01 << FWI) * 302us)

例 如: 向一张已被激活的 Mifare DESFire 卡发送“请求应答以选择 (RATS)”命令, 交互的数据块为该命令的命令帧 (0xE0,0x50), 帧长 2 字节 (不包括 CRC 校验, 其中 0xE0 是 RATS 的命令编码, 0x50 的高半字节为 FSDI, 低半字节为 CID, FSDI=5 表示最大交互帧为 64 字节)

表 5.76 数据交互命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 02 | 0058 | 0004 |
| Info | | | | | Checksum |
| E0 50 01 04 | | | | | FDBA |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0x06

信 息 (Info): ATS

例 如: RATS 命令执行成功的回应

表 5.77 数据交互成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-------------------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 02 | 0000 | 0006 |
| Info | | | | | Checksum |
| 06 77 81 02 80 00 | | | | | FDC4 |

5.3 ISO7816-3 类命令 (CmdClass = 0x05)

ISO7816-3 类命令汇总表如表 5.78 所示。

表 5.78 ISO7816-3 类命令一览表

| 命令码 | 意义 |
|-----|----------------------------------|
| ‘A’ | 接触式 IC 卡复位(自动处理 PPS) |
| ‘B’ | 接触式 IC 卡传输协议 (自动处理 T=0 和 T=1 协议) |
| ‘C’ | 接触式 IC 卡冷复位 |
| ‘D’ | 接触式 IC 卡热复位 |
| ‘E’ | 接触式 IC 卡停活 (关闭电源和时钟) |
| ‘F’ | 接触式 IC 卡 PPS(传输协议协商) |
| ‘G’ | 接触式 IC 卡 T=0 传输协议 |
| ‘H’ | 接触式 IC 卡 T=1 传输协议 |

其中 ‘A’ 和 ‘B’ 命令是组合命令, 根据卡片的实际情况自动调整通信协议; ‘C’、‘D’、‘F’ ~ ‘H’ 命令需要使用者自己根据卡片的情况来调用不同的命令; ‘E’ 命令是停活命令, 执行该命令后, IC 卡处于掉电状态。实际上对用户来说, 只需要执行 ‘A’、‘B’ 命令即可。

注意: ‘D’ 命令没有控制电源, 执行该命令前必须保证该 IC 卡没有处于停活状态。

5.3.1 接触式 IC 卡复位(自动处理 PPS)

该命令是冷复位, 执行成功后会自动根据 IC 卡的复位信息来自动执行 PPS 命令, 然后再选择 ‘B’ 命令使用的传输协议 (T = 0 或 T = 1)。

声明: *uint8_t Cicc_Reset(uint8_t nSlotIndex, uint8_t nResetFD,*

*void *pATRBuf, uint32_t nBufSize, uint32_t *pATRBytes)*

1. 主机命令

命令类型(CmdClass): 0x05

命令代码(CmdCode): ‘A’

信息长度(InfoLength): 0x01

信 息(Info): IC 卡复位时的波特率 (1 字节): 0x11 — 9600bps
0x13 — 38400bps

例 如: 波特率为 38400bps 的接触式 IC 卡复位命令帧见表 5.79

表 5.79 初始波特率为 38400bps 的接触式 IC 卡复位命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 05 | 0041 | 0001 |
| Info | | | | | Checksum |
| 13 | | | | | FEF3 |

2. 从机回应

执行状态 (Status): 0 — 执行成功; 其他 — 警告或失败

信息长度(InfoLength): 16 + (不同的卡回应的字节数不同)

信 息(Info): 保留信息 (16 字节, 该信息为任意值)

接触式 IC 卡复位信息（不同的卡复位信息长度不同）
例 如：接触式 IC 卡复位操作执行成功的回应帧如表 5.80 所示

表 5.80 接触式 IC 卡复位操作执行成功的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|--|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 05 | 0000 | 001D |
| Info | | | | | Checksum |
| 13 00 00 00 00 00 00 00 10 03 00 00 03 03 00 00 3B 69 00 00 57 44 37 51 BA CB 18 18 35 | | | | | FB4D |

注意：表 5.80 中信息字段中的前 16 字节是无效字节，没有任何意义，保留为将来使用，用户不用理会；后 13 字节才是接触式 IC 卡的复位信息。

5.3.2 接触式 IC 卡传输协议（自动处理 T = 0 和 T = 1 协议）

该命令根据接触式 IC 卡的复位信息，自动选择 T = 0 或 T = 1 传输协议，整个过程不需要使用者干预。该命令用于传输 APDU 数据流。

声明：uint8_t Cicc_TPDU(const void *pSendBuf, uint32_t nSendBytes,
void *pRcvBuf, uint32_t nRcvBufSize, uint32_t *pRcvBytes)

1. 主机命令

命令类型(CmdClass): 0x05
命令代码(CmdCode): ‘B’
信息长度(InfoLength): 1~272
信 息(Info): 发送到 IC 卡的数据
例 如：

通过 FID（文件标识符）选择 MF（FID 为：3F00）。选择文件的 APDU 如表 5.81 所示，将其转换为数据流为：00 A4 00 00 02 3F 00 00（不需要区分 APDU 的 4 种情况，‘3F00’在数据流中是以大端模式存放，即高字节在前），该命令能自动处理，其命令帧如错误！未找到引用源。所示

表 5.81 某 CPU 卡选择文件的 APDU

| 代码 | 长度 (字节) | 值 (Hex) | 说明 |
|------|------------|------------|--|
| CLA | 1 | 00 | — |
| INS | 1 | A4 | — |
| P1 | 1 | 00/04 | P1=00，表示按文件标识符选择（P2 必须等于 0），可选择： • 当前目录（DF）下基本文件或子目录文件 • 同级目录文件（DF） P1=04，表示用 DF 名称选择，分如下两种情况： • P2=00，表示第一个或仅有一个 • P2=02，表示下一个 |
| P2 | 1 | 00/02 | — |
| Lc | 1 | xx | — |
| Data | xx | xx…xx | 文件标识符或 DF 名称 |
| Le | 1 | 00 | 对于 DF 而言为卡片自动返回的 FCI 的最大长度 |

注意：在任何情况下均可通过标识符 ‘3F00’ 或目录名称 1PAY.SYS.DDF01 选择 MF。

表 5.82 通过 FID 选择 MF (FID 为 ‘3F00’) 的命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-------------------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 05 | 0042 | 0008 |
| Info | | | | | Checksum |
| 00 A4 00 00 02 3F 00 00 | | | | | FE19 |

2. 从机回应

执行状态 (Status): 0 — 执行成功; 其他 — 警告或失败
 信息长度(InfoLength): 不同的卡回应的字节数不同
 信 息(Info): IC 卡回复的数据
 例 如: 选择 MF 操作执行成功的回应帧如表 5.83 所示

表 5.83 选择 MF 操作执行成功的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|--|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 05 | 0000 | 0019 |
| Info | | | | | Checksum |
| 6F 15 84 0E 31 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 03 88 01 01 90 00 | | | | | F8B1 |

表 5.83 中的前 23 字节为 MF 的 FCI, 最后 2 字节 ‘90 00’ 表示卡片处理成功。需要注意的是 Info 域的最后 2 字节表示卡片执行结果与回应帧中的 ‘Sataus’ 字段表示的不是同一状态, ‘Sataus’ 字段表示是通信链路层的状态; 而 Info 域的最后 2 字节表示卡片执行结果。

5.3.3 接触式 IC 卡冷复位

该命令是冷复位, 执行了接触式 IC 卡上电时序, 执行成功后会自动根据 IC 卡的复位信息来选择 ‘B’ 命令使用的传输协议 (T=0 或 T=1)。与 4.3.1 相比只是没有自动执行 PPS 命令, 需要用户根据复位信息来判断是否使用 `Cicc_PPS()` 来修改协议和参数。

声明: `uint8_t Cicc_ColdReset(uint8_t nSlotIndex, uint8_t nResetFD,`
`void *pATRBuf, uint32_t nBufSize, uint32_t *pATRBytes)`

1. 主机命令

命令类型(CmdClass): 0x05
 命令代码(CmdCode): ‘C’
 信息长度(InfoLength): 0x01
 信 息(Info): IC 卡复位时的波特率 (1 字节): 0x11 — 9600bps
 0x13 — 38400bps

例 如: 初始波特率为 38400bps 的接触式 IC 卡冷复位命令帧和复位的命令帧除了命令码不同, 其他相同, 如表 5.84 所示

表 5.84 初始波特率为 38400bps 的接触式 IC 卡冷复位命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 05 | 0043 | 0001 |
| Info | | | | | Checksum |
| 13 | | | | | FEF1 |

2. 从机回应

执行状态 (Status): 0 — 执行成功; 其他 — 警告或失败

信息长度(InfoLength): 16 + (不同的卡回应的字节数不同)

信 息(Info): 保留信息 (16 字节, 该信息为任意值)

接触式 IC 卡复位信息 (不同的卡复位信息长度不同)

例 如: 接触式 IC 卡复位操作执行成功的回应帧如表 5.85 所示。和复位操作成功的回应帧相同, 见表 5.80

表 5.85 接触式 IC 卡冷复位操作执行成功的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|---|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 05 | 0000 | 001D |
| Info | | | | | Checksum |
| 13 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 03 3B 69 00 00 57 44 37 51 BA CB 18 18 35 | | | | | F985 |

注意: 表 5.80 中信息字段中的前 16 字节是无效字节, 没有任何意义, 保留为将来使用, 用户不用理会; 后 17 字节才是接触式 IC 卡的复位信息。

5.3.4 接触式 IC 卡热复位

该命令是热复位, 没有执行了接触式 IC 卡上电时序, 执行成功后会自动根据 IC 卡的复位信息来选择 ‘B’ 命令使用的传输协议 (T = 0 或 T = 1)。该命令和 4.3.3 比较只是没有执行 IC 卡上电操作。需要用户根据复位信息来判断是否使用 Cicc_PPS() 来修改协议和参数。该命令必须在 IC 卡时钟和电源均有效的情况下才能执行。

声明: `uint8_t Cicc_WarmReset(uint8_t nSlotIndex, uint8_t nResetFD,`
`void *pATRBuf, uint32_t nBufSize, uint32_t *pATRBytes)`

1. 主机命令

命令类型(CmdClass): 0x05

命令代码(CmdCode): ‘D’

信息长度(InfoLength): 0x01

信 息(Info): IC 卡复位时的波特率 (1 字节): 0x11 — 9600bps
0x13 — 38400bps

例 如: 初始波特率为 38400bps 的接触式 IC 卡冷复位命令帧和复位的命令帧除了命令码不同, 其他相同, 如表 5.86 所示

表 5.86 初始波特率为 38400bps 的接触式 IC 卡热复位命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 05 | 0044 | 0001 |
| Info | | | | | Checksum |
| 13 | | | | | FEF0 |

2. 从机回应

执行状态 (Status): 0 — 执行成功; 其他 — 警告或失败

信息长度(InfoLength): 16 + (不同的卡回应的字节数不同)

信 息(Info): 保留信息（16 字节，该信息为任意值）
 接触式 IC 卡复位信息（不同的卡复位信息长度不同）
例 如: 接触式 IC 卡复位操作执行成功的回应帧如表 5.87 所示。和
 复位操作成功的回应帧相同，见表 5.80

表 5.87 接触式 IC 卡热复位操作执行成功的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|---|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 05 | 0000 | 001D |
| Info | | | | | Checksum |
| 13 53 59 53 2E 44 44 46 30 31 A5 03 3B 69 00 00 3B 69 00 00 57 44 37 51 BA CB 18 18 35 | | | | | F7BE |

注意：表 5.80 中信息字段中的前 16 字节是无效字节，没有任何意义，保留为将来使用，用户不用理会；后 17 字节才是接触式 IC 卡的复位信息。

5.3.5 接触式 IC 卡停活

该命令是关闭接触式 IC 卡的电源和时钟。

声明: *uint8_t Cicc_Deactivation(void)*

1. 主机命令

命令类型(CmdClass): 0x05
命令代码(CmdCode): ‘E’
信息长度(InfoLength): 0
信 息(Info): none
例 如: 关闭接触式 IC 卡电源和时钟的命令帧如表 5.88 所示

表 5.88 接触式 IC 卡停活命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 05 | 0045 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF03 |

2. 从机回应

执行状态 (Status): 0 — 执行成功； 其他 — 警告或失败
信息长度(InfoLength): 0
信 息(Info): none
例 如: 接触式 IC 卡停活操作执行成功的回应帧如表 5.89 所示

表 5.89 接触式 IC 卡停活操作执行成功的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 05 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF47 |

5.3.6 接触式 IC 卡协议和参数选择 (PPS)

该命令是冷复位或热复位之后且必须首先执行（协商模式下需要执行，专用模式不需要执行）。若对接触式 IC 卡不了解，建议使用 `Cicc_Reset()`（该命令自动处理了 PPS 命令），而不要使用 `Cicc_ColdReset() + Cicc_PPS()` 或 `Cicc_WarmReset() + Cicc_PPS()`。

注意：PPS 命令是修改通信协议和参数，必须在 IC 卡复位之后首先执行。修改的参数必须是 IC 卡支持的才可以。

声明：`uint8_t Cicc_PPS(const uint8_t *pPPS)`

1. 主机命令

命令类型(CmdClass): 0x05
命令代码(CmdCode): 'F'
信息长度(InfoLength): 0x04
信 息(Info): PPS 参数（4 字节）

PPS[0] — 指定是否存在 PPS1、PPS2、PPS3
PPS[0].3:0 — 保留
PPS[0].4 = 1 — PPS1 存在；0 — PPS1 不存在
PPS[0].5 = 1 — PPS2 存在；0 — PPS2 不存在
PPS[0].6 = 1 — PPS3 存在；0 — PPS3 不存在
PPS[0].7 — 保留
PPS[1] — F/D
PPS[2] — N
PPS[3] — 待定

例 如：将接触式 IC 卡通信波特率改为 115200bps(Fi 为 1;Di 为 8)，其他的不修改，命令帧如表 5.90 所示

表 5.90 接触式 IC 卡协议和参数选择命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 05 | 0046 | 0004 |
| Info | | | | | Checksum |
| 10 18 00 00 | | | | | FED6 |

2. 从机回应

执行状态 (Status): 0 — 执行成功； 其他 — 警告或失败
信息长度(InfoLength): 0
信 息(Info): none

例 如：接触式 IC 卡协议和参数选择操作执行成功的回应帧如表 5.91 所示

表 5.91 接触式 IC 卡协议和参数选择操作执行成功的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 05 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF47 |

5.3.7 接触式 IC 卡传输协议 (T = 0)

该命令用于 T = 0 传输协议。若接触式 IC 卡的传输协议为 T = 0，该命令等同于 `Cicc_TPDU()`。

声明：`uint8_t Cicc_TP0(const void *pSendBuf, uint32_t nSendBytes, void *pRcvBuf, uint32_t nRcvBufSize, uint32_t *pRcvBytes)`

1. 主机命令

命令类型(CmdClass): 0x05
命令代码(CmdCode): 'G'
信息长度(InfoLength): 1~272
信 息(Info): 发送到 IC 卡的数据
例 如: 通过 FID (文件标识符) 选择 MF (FID 为: 3F00)。选择文件的 APDU 如表 5.81 所示, 将其转换为数据流为: 00 A4 00 00 02 3F 00 00 (不需要区分 APDU 的 4 种情况, '3F00' 在数据流中是以大端模式存放, 即高字节在前), 该命令能自动处理, 其命令帧如表 5.92 所示

表 5.92 通过 FID 选择 MF (FID 为 '3F00', T=0) 的命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-------------------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 05 | 0047 | 0008 |
| Info | | | | | Checksum |
| 00 A4 00 00 02 3F 00 00 | | | | | FE14 |

2. 从机回应

执行状态 (Status): 0 — 执行成功; 其他 — 警告或失败
信息长度(InfoLength): 不同的卡回应的字节数不同
信 息(Info): IC 卡回复的数据
例 如: 选择 MF 操作执行成功的回应帧如表 5.93 所示

表 5.93 选择 MF (T=0) 操作执行成功的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|--|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 05 | 0000 | 0019 |
| Info | | | | | Checksum |
| 6F 15 84 0E 31 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 03 88 01 01 90 00 | | | | | F8B1 |

表 5.93 中的前 23 字节为 MF 的 FCI, 最后 2 字节 '90 00' 表示卡片处理成功。需要注意的是 Info 域的最后 2 字节表示卡片执行结果与回应帧中的 'Sataus' 字段表示的不是同一状态, 'Sataus' 字段表示是通信链路层的状态; 而 Info 域的最后 2 字节表示卡片执行结果。

5.3.8 接触式 IC 卡传输协议 (T = 1)

该命令用于 T=1 传输协议。若接触式 IC 卡的传输协议为 T=1, 其等同于 `Cicc_TPDU()`。

声明：`uint8_t Cicc_TPDU(const void *pSendBuf, uint32_t nSendBytes, void *pRcvBuf, uint32_t nRcvBufSize, uint32_t *pRcvBytes)`

1. 主机命令

命令类型(CmdClass): 0x05
命令代码(CmdCode): 'H'
信息长度(InfoLength): 1~272
信 息(Info): 发送到 IC 卡的数据
例 如: 通过 FID (文件标识符) 选择 MF (FID 为: 3F00)。选择文件的 APDU 如表 5.81 所示, 将其转换为数据流为: 00 A4 00 00 02 3F 00 00 (不需要区分 APDU 的 4 种情况, '3F00' 在数据流中是以大端模式存放, 即高字节在前), 该命令能自动处理, 其命令帧如表 5.94 所示

表 5.94 通过 FID 选择 MF (FID 为 '3F00', T=1) 的命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-------------------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 05 | 0048 | 0008 |
| Info | | | | | Checksum |
| 00 A4 00 00 02 3F 00 00 | | | | | FE13 |

2. 从机回应

执行状态 (Status): 0 — 执行成功; 其他 — 警告或失败
信息长度(InfoLength): 不同的卡回应的字节数不同
信 息(Info): IC 卡回复的数据
例 如: 选择 MF 操作执行成功的回应帧如表 5.95 所示

表 5.95 选择 MF (T=1) 操作执行成功的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|--|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 05 | 0000 | 0019 |
| Info | | | | | Checksum |
| 6F 15 84 0E 31 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 03 88 01 01 90 00 | | | | | F8B1 |

表 5.95 中的前 23 字节为 MF 的 FCI, 最后 2 字节 '90 00' 表示卡片处理成功。需要注意的是 Info 域的最后 2 字节表示卡片执行结果与回应帧中的 'Sataus' 字段表示的不是同一状态, 'Sataus' 字段表示是通信链路层的状态; 而 Info 域的最后 2 字节表示卡片执行结果。

5.4 ISO14443 (PICC) 卡类命令 (CmdClass = 0x06)

ISO14443 (PICC) 卡类命令总汇如表 5.96 所示。

表 5.96 ISO14443 (PICC) 卡类命令一览表

| 命令码 | 意义 |
|-----|----------------------------|
| ‘A’ | A 型卡请求 |
| ‘B’ | A 型卡防碰撞 |
| ‘C’ | A 型卡选择 |
| ‘D’ | A 型卡挂起 |
| ‘E’ | A 型卡 RATS |
| ‘F’ | A 型卡 PPS |
| ‘G’ | A 型卡解除激活 |
| ‘H’ | T=CL |
| ‘J’ | 数据交换 |
| ‘L’ | A 型卡复位 |
| ‘M’ | A 型卡激活 |
| ‘N’ | B 型卡激活 |
| ‘O’ | B 型卡复位 |
| ‘P’ | B 型卡请求 |
| ‘Q’ | B 型卡防碰撞 |
| ‘R’ | B 型卡修改传输属性 |
| ‘S’ | B 型卡挂起 |

前 4 条命令 (命令 A~D) 是 ISO14443-3A 标准定义的命令, 只要符合该标准的卡都应能发出响应; 中间 4 条命令 (命令 E~H) 为是 ISO14443-4 标准定义的命令。其中 A~D 命令和 Mifare S50/S70 卡类命令的 A~D 命令完全相同

5.4.1 A 型卡请求 (Cmd = A)

该命令用于 A 型卡的请求操作, 该命令的操作与 Mifare S50/S70 卡类的请求 (Cmd = A) 命令一样。

例 如: 请求天线范围内所有的 A 型卡。

主机命令: B2 00 00 06 41 00 01 00 52 B3 FE。

5.4.2 A 型卡防碰撞 (Cmd = B)

该命令用于 A 型卡的防碰撞, 该命令的操作与 Mifare S50/S70 卡类的防碰撞 (Cmd = B) 命令一样。

例 如: 第一级防碰撞。

主机命令: B2 00 00 06 42 00 02 00 93 00 70 FE。

5.4.3 A 型卡选择 (Cmd = C)

该命令用于 A 型卡的选择, 该命令的操作与 Mifare S50/S70 卡类的卡选择 (Cmd = C) 命令一样。

例 如: 第一级选择, UID 为 0xEB1C1814。

主机命令：B2 00 00 06 43 00 05 00 93 14 18 1C EB 39 FD。

5.4.4 A 型卡挂起 (Cmd = D)

该命令用于 A 型卡的挂起，该命令的操作与 Mifare S50/S70 卡类的卡挂起 (Cmd = D) 命令一样。

例 如：将已激活的卡挂起，使之不响应请求空闲卡命令。

主机命令：B2 00 00 06 44 00 00 00 03 FF。

5.4.5 A 型卡 RATS (Cmd = E)

RATS (request for answer to select) 是 ISO14443-4 协议的命令，模块发送 RATS，卡片发出 ATS (answer to select) 作为 RATS 的应答，在执行该命令前，必需先进行一次卡选择操作，且执行过一次 RATS 命令后，想再次执行 RATS 命令，必需先解除激活。

声明：uint8_t PiccA_RATS(uint8_t CID, void *pRATS,

uint32_t nRATSBufSize, uint32_t *pRATSBytes)

1. 主机命令

命令类型 (CmdClass): 0x06

命令代码 (CmdCode): 'E'

信息长度 (InfoLength): 0x01

信 息 (Info): CID (1 字节): 卡标识符 (card Identifier, 取值范围 0x00~0x0E)

例 如：向 PLUS CPU 卡发送 RATS 命令，CID 设备为 0x0A

表 5.97 A 型卡 RATS 命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 06 | 0045 | 0001 |
| Info | | | | | Checksum |
| 0A | | | | | FEF7 |

2. 从机应答

状 态 (Status): 0——成功，其它——失败

信息长度 (InfoLength): 0x0C (不同的卡，ATS 的字节数不同)

信 息 (Info): ATS

例 如：一张 SL3 的 PLUS CPU 卡会回应的 ATS

表 5.98 A 型卡响应 RATS 的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-------------------------------------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 06 | 0000 | 000C |
| Info | | | | | Checksum |
| 0C 75 77 80 02 C1 05 2F 2F 01 BC D6 | | | | | FB09 |

5.4.6 A 型卡 PPS (Cmd = F)

PPS (protocol and parameter selection) 是 ISO14443-4 协议的命令，用于改变有关的专用协议参数，该命令不是必需的，命令只支持默认参数，即该命令的参数设置为 0 即可。在

执行该命令前，必需先成功执行一次 RATS 命令。

声明： *uint8_t PiccA_PPS(uint8_t DSI_DRI)*

1. 主机命令

命令类型 (CmdClass): 0x06

命令代码 (CmdCode): 'F'

信息长度 (InfoLength): 0x01

信 息 (Info): DSI_DRI (1 字节): 模块与卡通信波特率，设置为 0 (106Kb/s)

例 如: 将 PLUS CPU 卡与模块间的通信波特率设置为 106Kb/s

表 5.99 A 型卡 PPS 命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 06 | 0046 | 0001 |
| Info | | | | | Checksum |
| 00 | | | | | FF00 |

2. 从机应答

状 态 (Status): 0——成功，其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: PLUS CPU 卡执行 PPS 成功后的回应

表 5.100 A 型卡响应 PPS 的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 06 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF46 |

5.4.7 A 型卡解除激活 (Cmd = G)

该命令是 ISO14443-4 协议的命令,用于将卡片置为挂起(HALT)状态,处于挂起(HALT)状态的卡可以用“请求”命令(请求代码为 ALL)来重新激活卡,只有执行“RATS”命令的卡才用该命令。

声明: *uint8_t PiccA_DeSelect(void)*

1. 主机命令

命令类型 (CmdClass): 0x06

命令代码 (CmdCode): 'G'

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 将激活的卡置为挂起状态

表 5.101 A 型卡解除激活命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 06 | 0047 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF00 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: PLUS CPU 卡执行 PPS 成功后的回应

表 5.102 A 型卡响应解除激活的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 06 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF46 |

5.4.8 T=CL (Cmd = H)

T=CL 是半双工分组传输协议, ISO14443-4 协议命令, 用于读写器与卡片之间的数据交互, 一般符合 ISO14443 协议的 CPU 卡均用该协议与读写器通信。调用该命令时只需要将 CPU 卡 COS 命令的数据作为输入即可, 其他的如分组类型、卡标识符 CID、帧等待时间 FWT、等待时间扩展倍增因子 WTXM (waiting time extensionmultiplier), 等等由该命令自动完成。

声明: *uint8_t Picc_TPCL(const void *pSBuf, uint32_t nSBytes,*

*void *pRBuf, uint32_t nRBufSize, uint32_t *pRealBytes)*

1. 主机命令

命令类型 (CmdClass): 0x06
 命令代码 (CmdCode): 'H'
 信息长度 (InfoLength): COS 命令的长度
 信 息 (Info): COS 命令
 例 如: 选择 FM1208 的 MF 标识符为 3F00, 选择 COS 命令如下

表 5.103 FM1208 选择 MF 的命令编码

| 代码 | CLA | INS | P1 | P2 | Lc | Data | Le |
|----|-----|-----|----|----|----|-------|----|
| 值 | 00 | A4 | 00 | 00 | 02 | 3F 00 | — |

表 5.104 A 型卡 T=CL 命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|----------------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 06 | 0048 | 0007 |
| Info | | | | | Checksum |
| 00 A4 00 00 02 3F 00 | | | | | FE13 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): COS 命令回应数据长度
 信 息 (Info): COS 命令回应数据
 例 如: FM1208 选择 MF 时响应的数据为嵌套的 TLV 格式的变长记录, 其意义请参考《FMCOS 用户手册》

表 5.105 A 型卡响应 T=CL 的回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|---|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 06 | 0000 | 0016 |
| Info | | | | | Checksum |
| 6F 15 84 0E 31 50 41 59 53 2E 44 44 46 30 31 A5 03 88 01 01 90 00 | | | | | F98D |

5.4.9 数据交换 (Cmd = J)

该命令用读写器与卡片的数据交互, 通过该命令可以实现读写卡器的所有功能。

声明: `uint8_t CD_ExchangeBlock(const ExchangeInputPara *pIn, ExchangeOutputPara *pOut)`

1. 主机命令

命令类型 (CmdClass): 0x06
 命令代码 (CmdCode): 'J'
 信息长度 (InfoLength): 交互数据块长度+2
 信 息 (Info): 交互数据块 (其内容与实际使用的 CPU 卡有关)
 WTXM_CRC (1 字节), 该字节内容如表 5.106 所示

表 5.106 WTXM_CRC 字节描述

| B7~B2 | B1 | B0 |
|-------|-----|--------|
| WTXM | RFU | CRC 禁能 |
| | 0 | CRC 使能 |

FWI (1 字节): 超时等待时间编码

超时时间 = ((0x01 << FWI) * 302us)

例 如: 向一张已被激活的 Mifare DESFire 卡发送“请求应答以选择 (RATS)”命令, 交互的数据块为该命令的命令帧 (0xE0, 0x50), 帧长 2 字节 (不包括 CRC 校验, 其中 0xE0 是 RATS 的命令编码, 0x50 的高半字节为 FSDI, 低半字节为 CID, FSDI=5 表示最大交互帧为 64 字节)

表 5.107 数据交互命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 06 | 004A | 0004 |
| Info | | | | | Checksum |
| E0 50 01 04 | | | | | FDC4 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): 0x06

信 息 (Info): ATS
例 如: RATS 命令执行成功的回应

表 5.108 数据交互成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-------------------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 06 | 0000 | 0006 |
| Info | | | | | Checksum |
| 06 77 81 02 80 00 | | | | | FDC0 |

5.4.10 A 型卡复位 (Cmd = L)

该命令是通过将载波信号关闭指定的时间，再开启来实现卡片复位。

声明: *void CD_PauseCarrier(uint8_t pause_ms, uint8_t wait_ms)*

1. 主机命令

命令类型 (CmdClass): 0x06
命令代码 (CmdCode): 'L'
信息长度 (InfoLength): 0x01
信 息 (Info): 时间 (1 字节)，以毫秒为单位，0 为一直关闭
例 如: 将载波信号关闭 1ms

表 5.109 卡复位命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 06 | 004C | 0001 |
| Info | | | | | Checksum |
| 01 | | | | | FEF9 |

2. 从机应答

状 态 (Status): 0——成功，其它——失败
信息长度 (InfoLength): 0
信 息 (Info): none
例 如: 执行卡复位成功模块的回应

表 5.110 卡复位成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 06 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF46 |

3. 说明

该命令将天线信号关闭数毫秒，若一直关闭，则等到执行一个请求命令时打开。

5.4.11A 型卡激活 (Cmd = M)

该命令用于激活卡片，是请求、防碰撞和选择三条命令的组合。

声明: *uint8_t MF_Activate(uint8_t mode, uint8_t reqCode, PiccAResetInfo *pResetInfo)*

1. 主机命令

命令类型 (CmdClass): 0x06
 命令代码 (CmdCode): 'M'
 信息长度 (InfoLength): 0x02
 信 息 (Info): 保留 (1 字节), 设置为 0
 请求代码 (1 字节): 0x26~IDLE
 0x52~ALL

例 如: 以 IDLE 方式激活卡

表 5.111 卡激活命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 06 | 004D | 0002 |
| Info | | | | | Checksum |
| 00 26 | | | | | FED2 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): Mifare1 S50、S70、Light 卡: 8 字节
 Mifare0 UltraLight 卡: 11 字节
 Mifare3 Desfire 卡: 11 字节
 Plus CPU 卡: 8 字节或 11 字节
 信 息 (Info): 请求应答 ATQ (2 字节)
 最后一级选择应答 SAK (1 字节)
 序列号长度 (1 字节)
 序列号 (N 字节, 由序列号长度决定)
 例 如: 一张序列号为 0xEB1C1814 的 Mifare1 S50 卡返回的数据

表 5.112 卡激活成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-------------------------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 06 | 0000 | 0008 |
| Info | | | | | Checksum |
| 04 00 08 04 14 18 1C EB | | | | | FDFB |

5.4.12 B 型卡激活 (Cmd = N)

该命令用于激活 B 型卡片, 在调用该命令前, 需要先执行设备控制类的“设置 IC 卡接口协议 (工作模式) (Cmd = D)”, 把模块先配置成 TypeB 模式。

声明: `uint8_t PiccB_Activate(uint8_t reqCode, uint8_t AFI, PiccBResetInfo *pResetInfo)`

1. 主机命令

命令类型 (CmdClass): 0x06
 命令代码 (CmdCode): 'N'
 信息长度 (InfoLength): 0x02

表 5.113 卡激活命令帧

| | | | | | |
|-----------|-----------|--------|----------|---------|------------|
| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
| B2 | 00 | 00 | 06 | 004E | 0002 |
| Info | | | | | Checksum |
| 00 00 | | | | | FEF7 |

2. 从机应答

表 5.114 卡激活成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-------------------------------------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 06 | 0000 | 000C |
| Info | | | | | Checksum |
| 70 05 34 07 00 00 00 00 00 81 C1 00 | | | | | FD48 |

5.4.13 B 型卡复位 (Cmd = 0)

声明: `void CD PauseCarrier(uint8 t pause ms, uint8 t wait ms)`

1. 主机命令

表 5.115 卡复位命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 06 | 004F | 0001 |
| Info | | | | | Checksum |
| 01 | | | | | FEF6 |

2. 从机应答

| | |
|--------------------|--------------|
| 信息长度 (InfoLength): | 0 |
| 信 息 (Info): | none |
| 例 如: | 执行卡复位成功模块的回应 |

表 5.116 卡复位成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 06 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF46 |

5.4.14 B 型卡请求 (Cmd = P)

该命令用于 B 型卡请求。

声明: `uint8_t PiccB_Request(uint8_t reqCode, uint8_t AFI, uint8_t N, PiccBResetInfo *pResetInfo)`

1. 主机命令

| | |
|--------------------|--|
| 命令类型 (CmdClass): | 0x06 |
| 命令代码 (CmdCode): | 'P' |
| 信息长度 (InfoLength): | 0x03 |
| 信 息 (Info): | 请求代码 (1 字节): 0x00~IDLE 0x08~ALL 应用标识 (1 字节): 默认为 0x00 时隙总数 (1 字节): 范围 0~4 |
| 例 如: | 以 IDLE 方式请求卡片 |

表 5.117 卡请求命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 06 | 0050 | 0003 |
| Info | | | | | Checksum |
| 00 00 00 | | | | | FEF4 |

2. 从机应答

| | |
|--------------------|----------------------|
| 状 态 (Status): | 0——成功, 其它——失败 |
| 信息长度 (InfoLength): | 0x0C |
| 信 息 (Info): | UID 相关信息 |
| 例 如: | 一张 TypeB 卡请求成功后返回的数据 |

表 5.118 卡请求成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-------------------------------------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 06 | 0000 | 000C |
| Info | | | | | Checksum |
| 70 05 34 07 00 00 00 00 00 81 C1 00 | | | | | FD48 |

5.4.15 B 型卡防碰撞 (Cmd = Q)

该命令用于 B 型卡的防碰撞，一般在请求成功结束后执行。

注：该命令是保留命令，本模块暂时不处理该命令。

声明： `uint8_t PiccB_SlotMarker(uint8_t N, PiccBResetInfo *pResetInfo);`

1. 主机命令

命令类型 (CmdClass): 0x06

命令代码 (CmdCode): 'Q'

信息长度 (InfoLength): 0x01

信 息 (Info): 时隙标记 (1 字节): 范围 2~16, 该参数值与请求命令的时隙总数有关系, 假如请求命令的时隙总数为 n, 侧该时隙标记 $N < 2^n$

例 如: 时隙标记为 4 防碰撞

表 5.119 卡防碰撞命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 06 | 0051 | 0001 |
| Info | | | | | Checksum |
| 04 | | | | | FEF1 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0x0C

信 息 (Info): UID 相关信息

例 如: 一张 TypeB 卡防碰撞成功后返回的数据

表 5.120 卡防碰撞成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-------------------------------------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 06 | 0000 | 000C |
| Info | | | | | Checksum |
| 70 05 34 07 00 00 00 00 00 81 C1 00 | | | | | FD48 |

5.4.16 B 型卡修改传输属性 (Cmd = R)

该命令用于 B 型卡修改传输属性 (卡选择)。

声明： `uint8_t PiccB_Attrib(const void *pPUI, uint8_t CID, uint8_t proType, void *pRBuf, uint32_t nRBufSize, uint32_t *pRealBytes);`

1. 主机命令

命令类型 (CmdClass): 0x06

命令代码 (CmdCode): 'R'

信息长度 (InfoLength): 0x06

信 息 (Info): PUI (4 字节): 卡片标识符

CID (1 字节): 取值范围为 0 - 14, 若不支持 CID, 则设置为 0

proType (1 字节): 支持的协议, 由请求回应中的 ProtocolType 指定

proType.3: PCD 与 PICC 是否继续通信

1~PCD 中止与 PICC 继续通信

0~PCD 与 PICC 继续通信

proType.2:1: PICC EOF 和 PCD SOF 间的最小延迟

11~10 etu + 512 / fs

10~10 etu + 256 / fs

01~10 etu + 128 / fs

00~10 etu + 32 / fs

proType.0: 是否遵循 ISO14443-4

1~遵循 ISO14443-4;

0~不遵循 ISO14443-4. (二代证必须为 1)

例 如: 选择 PUPI 为 0x07340570

表 5.121 卡选择命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-------------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 06 | 0052 | 0006 |
| Info | | | | | Checksum |
| 70 05 34 07 00 01 | | | | | FE3E |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 卡选择成功的回应

表 5.122 卡选择成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 06 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF46 |

5.4.17 B 型卡挂起 (Cmd = S)

该命令用于 B 型卡挂起, 在执行挂起命令前, 必需先执行成功过一次请求命令。执行挂起命令成功后, 卡片处于挂起状态, 模块必需通过 ALL 方式请求卡片, 而不能用 IDLE 方式请求。

声明: *uint8_t PiccB_Halt(uint8_t *pPUPI);*

1. 主机命令

命令类型 (CmdClass): 0x06

命令代码 (CmdCode): 'S'

信息长度 (InfoLength): 0x04

信 息 (Info): PUPI (4 字节): 4 字节标识符
例 如: 挂起 PUPI 为: 0x38492295 的卡片

表 5.123 卡挂起命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 06 | 0053 | 0004 |
| Info | | | | | Checksum |
| 95 22 49 38 | | | | | FDB8 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
信息长度 (InfoLength): 0
信 息 (Info): none
例 如: 卡挂起成功的回应

表 5.124 卡挂起成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 06 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF46 |

5.5 PLUS CPU 卡类命令（CmdClass = 0x07）

PLUS CPU 卡类命令总汇如表 4.125 所示，该命令集包括了 PLUS CPU 卡 SL0（Security Level 0）、SL3 的命令，其中等级 1 的命令与 Mifare S50/S70 卡（M1）相同，所以不在本命令集中。

表 5.125 PLUS CPU 卡类命令一览表

| 命令码 | 意义 |
|-----|--|
| ‘B’ | SL0 个人化更新数据 |
| ‘C’ | SL0 提交个人化 |
| ‘J’ | SL3 首次验证（直接密钥验证） |
| ‘K’ | SL3 首次验证（E² 密钥验证） |
| ‘L’ | SL3 跟随验证（直接密钥验证） |
| ‘M’ | SL3 跟随验证（E² 密钥验证） |
| ‘N’ | SL3 复位验证 |
| ‘O’ | SL3 读数据块 |
| ‘P’ | SL3 写数据块 |
| ‘S’ | SL3 值块操作 |

卡片激活后，只有通过“首次验证”之后才能使用“跟随验证”，卡片激活后，则只需要第一次验证命令使用“首次验证”命令，之后的验证命令都可以使用“跟随验证”，当然也可以都是用“首次验证”；若执行“复位验证”，则“复位验证”之后的第一次验证，也必须使用“首次验证”命令。两种验证的区别在于使用的时机不同，“首次验证”所需要的时间比“跟随验证”的时间要长些。

PLUS CPU 卡的密钥 A/B 是通过地址的奇偶数来区分，AES 的密钥地址与数据块的扇区关系对应如下。

- 密钥 A 地址=0x4000 + 扇区 × 2
- 密钥 B 地址=0x4000 + 扇区 × 2 + 1

除扇区密钥外，其它密钥不分密钥 A/B，详细的 PLUS CPU 卡地址分配请参阅 PLUS CPU 卡的数据手册。

5.5.1 SL0 个人化更新数据（Cmd = B）

该命令用于 SL0（Security Level 0，安全等级 0）的 PLUS CPU 卡个人化，PLUS CPU 卡出厂时的安全等级为 SL0，该等级下，不需要任何验证就可以向卡里写数据，写入的数据是作为其它安全等级的初始值，例如：

向 SL0 的 0x0003 块写入 0xA0 0xA1 0xA2 0xA3 0xA4 0xA5 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF，当卡片升级到 SL1 后，扇区 0 的 A 密钥为 0xA0 0xA1 0xA2 0xA3 0xA4 0xA5，而不是默认的 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF，即可以在 SL0 修改卡片的默认数据和密钥。

注意：PLUS CPU 卡在 SL0 的存储器地址均为 2 字节，其中地址 0x0000 ~ 0x00FF 为用户数据块，与 Mifare S50/S70 卡的数据/密钥块一一对应，该命令是 ISO14443-4 的命令。

声明：*uint8_t PLUS_WritePersoTCL(uint32_t usBNr, uint8_t *pBuf)*

1. 主机命令

命令类型 (CmdClass): 0x07
 命令代码 (CmdCode): 'B'
 信息长度 (InfoLength): 0x12
 信 息 (Info): BNr (2 字节): PLUS CPU 卡存储器地址
 Data (16 字节): 数据/AES 密钥/配置字
 例 如: 更改 PLUS CPU 卡的主控密钥 (地址为 0x9000)

表 5.126 SL0 个人化更新数据命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|--|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 07 | 0042 | 0012 |
| Info | | | | | CheckSum |
| 00 90 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF | | | | | EE72 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: 更改主控密钥成功的回应

表 5.127 SL0 个人化更新数据成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 07 | 0000 | 0000 |
| Info | | | | | CheckSum |
| none | | | | | FF45 |

5.5.2 SL0 提交个人化 (Cmd = C)

该命令用于 SL0 (Security Level 0, 安全等级 0) 的 PLUS CPU 卡提交个人化数据, 命令“SL0 个人化更新数据”只是更新卡中的数据, 但该数据还未生效, 用户还不能直接使用。“SL0 提交个人化”使更新的个人化数据生效。执行该命令后, PLUS CPU 卡的安全等级提高到 SL1 或者 SL3 (若是支持 SL1 的卡, 则执行该命令后卡片安全等级提高到 SL1; 若是只支持 SL0 和 SL3 的卡, 则执行该命令后卡片安全等级提高到 SL3)。

注意: 在 SL0 的 PLUS CPU 卡, 只有修改了以下地址才能执行“SL0 提交个人化”命令:

- 0x9000 (主控密钥)
- 0x9001 (配置块密钥)
- 0x9002 (SL2 提升密钥, 只有支持 SL2 的卡才有该密钥)
- 0x9003 (SL3 主控密钥, 只有支持 SL3 的卡才有该密钥)

该命令是 ISO14443-4 的命令

声明: *uint8_t PLUS_CommitPersoTCL(void)*

1. 主机命令

命令类型 (CmdClass): 0x07
 命令代码 (CmdCode): 'C'
 信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 将已修改主控密钥、配置块密钥、SL2 提升密钥和 SL3 主控密钥卡的安全等级提高到 SL1

表 5.128 SL0 提交个人化命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 07 | 0043 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF 03 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 更改主控密钥成功的回应

表 5.129 SL0 提交个人化成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 07 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF45 |

5.5.3 SL3 首次验证 (直接密钥验证) (Cmd = J)

该命令用于 SL3 PLUS CPU 卡的密钥验证, 验证的密钥通过该命令的参数输入。

声明: *uint8_t PLUS_SL3FirstAuth(uint32_t uiKNr, const uint8_t *pKey)*

1. 主机命令

命令类型 (CmdClass): 0x07

命令代码 (CmdCode): 'J'

信息长度 (InfoLength): 0x12

信 息 (Info): AES 密钥地址 (2 字节)

AES 密钥 (16 字节)

例 如: 用密钥“FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF”

验证 1 扇区的 AES 密钥 A (1 扇区的 AES 密钥 A 对应的密钥地址为 0x4002)

表 5.130 SL3 首次验证 (直接密钥验证) 命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|--|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 07 | 004A | 0012 |
| Info | | | | | Checksum |
| 02 40 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF | | | | | EEB8 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none
例 如: 验证密钥成功的回应

表 5.131 SL3 首次验证（直接密钥验证）成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 07 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF45 |

5.5.4 SL3 首次验证（E² 密钥验证）（Cmd = K）

该命令也是用于 SL3 PLUS CPU 卡的密钥验证，验证的密钥来自模块内部，掉电不丢失的数据。

声明: *uint8_t PLUS_SL3FirstAuthE2(uint32_t uiKNr, uint8_t KeySector)*

1. 主机命令

命令类型 (CmdClass): 0x07
命令代码 (CmdCode): 'K'
信息长度 (InfoLength): 0x03
信 息 (Info): AES 密钥地址 (2 字节)
密钥区号 (1 字节)

例 如: 用密钥 1 区的密钥验证 1 扇区的 AES 密钥 A (密钥地址为 0x4002)

表 5.132 SL3 首次验证（E² 密钥验证）命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 07 | 004B | 0003 |
| Info | | | | | Checksum |
| 02 40 01 | | | | | FEB5 |

2. 从机应答

状 态 (Status): 0——成功，其它——失败
信息长度 (InfoLength): 0
信 息 (Info): none
例 如: 验证密钥成功的回应

表 5.133 SL3 首次验证（E² 密钥验证）成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 07 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF45 |

5.5.5 SL3 跟随验证（直接密钥验证）（Cmd = L）

该命令用于 SL3 PLUS CPU 卡的跟随密钥验证，验证的密钥来自命令参数，只有执行过

“首次验证”命令成功后才能使用该命令。

声明: *uint8_t PLUS_SL3FollowingAuth(uint32_t uiKNr, const uint8_t *pKey)*

1. 主机命令

命令类型 (CmdClass): 0x07

命令代码 (CmdCode): 'L'

信息长度 (InfoLength): 0x12

信 息 (Info): AES 密钥地址 (2 字节)

AES 密钥 (16 字节)

例 如: 用密钥“FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF”
验证 1 扇区的 AES 密钥 A (1 扇区的 AES 密钥 A 对应的密钥地址为 0x4002)

表 5.134 SL3 跟随验证 (直接密钥验证) 命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|--|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 07 | 004C | 0012 |
| Info | | | | | CheckSum |
| 02 40 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF | | | | | EEB6 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 验证密钥成功的回应

表 5.135 SL3 跟随验证 (直接密钥验证) 成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 07 | 0000 | 0000 |
| Info | | | | | CheckSum |
| none | | | | | FF45 |

5.5.6 SL3 跟随验证 (E² 密钥验证) (Cmd = M)

该命令用于 SL3 PLUS CPU 卡的跟随密钥验证, 验证的密钥来自模块内部掉电不丢失的数据, 只有执行过“首次验证”命令成功后才能使用该命令。

声明: *uint8_t PLUS_SL3FollowingAuth(uint32_t uiKNr, const uint8_t *pKey)*

1. 主机命令

命令类型 (CmdClass): 0x07

命令代码 (CmdCode): 'M'

信息长度 (InfoLength): 0x03

信 息 (Info): AES 密钥地址 (2 字节)

密钥区号 (1 字节)

例 如: 用密钥 1 区的密钥验证 1 扇区的 AES 密钥 A (密钥地址为 0x4002)

表 5.136 SL3 跟随验证 (E² 密钥验证) 命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 07 | 004D | 0003 |
| Info | | | | | Checksum |
| 02 40 01 | | | | | FEB3 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: 验证密钥成功的回应

表 5.137 SL3 跟随验证 (E² 密钥验证) 成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 07 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF45 |

5.5.7 SL3 复位验证 (Cmd = N)

该命令用于 PLUS CPU 卡通过首次验证后的使用过程中, 复位读写计数器和验证等信息。

声明: *uint8_t PLUS_SL3ResetAuth(void)*

1. 主机命令

命令类型 (CmdClass): 0x07
 命令代码 (CmdCode): 'N'
 信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: 复位验证卡片的验证信息

表 5.138 SL3 复位验证命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-----------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 07 | 004E | 0000 |
| Info | | | | | Checksum |
| none | | | | | FEF8 |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败
 信息长度 (InfoLength): 0
 信 息 (Info): none
 例 如: 验证密钥成功的回应

表 5.139 SL3 复位验证成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 07 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF45 |

3. 说明

若执行“复位验证”命令，读写计数器和所有的认证信息都将清空，若还需要对卡片进行操作，则必需使用“首次验证”命令或者将卡片重新激活。

5.5.8 SL3 读数据块（Cmd = 0）

该命令用于读取 SL3 的数据块，在读数据块之前必需成功执行一次密钥验证。

声明： *uint8_t PLUS_SL3Read(uint8_t ucMode, uint32_t usBNr, uint8_t ucExt, uint8_t *pBuf)*

1. 主机命令

命令类型（CmdClass）： 0x07

命令代码（CmdCode）： ‘0’

信息长度（InfoLength）： 0x04

信 息（Info）： 读模式（1 字节）： 0x30～命令有 MAC；数据密文；回应无 MAC
0x31～命令有 MAC；数据密文；回应有 MAC
0x32～命令有 MAC；数据明文；回应无 MAC
0x33～命令有 MAC；数据明文；回应有 MAC
0x34～命令无 MAC；数据密文；回应无 MAC
0x35～命令无 MAC；数据密文；回应有 MAC
0x36～命令无 MAC；数据明文；回应无 MAC
0x37～命令无 MAC；数据明文；回应有 MAC

起始块号（2 字节）

读的块数（1 字节）： 范围 1～3

例 如： 从块 4 开始以“命令有 MAC，数据明文，回应有 MAC”的方式读 1 块数据

表 5.140 SL3 读数据块命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|-------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 07 | 004F | 0004 |
| Info | | | | | Checksum |
| 33 04 00 01 | | | | | FEBB |

2. 从机应答

状 态（Status）： 0——成功，其它——失败

信息长度（InfoLength）： 0x10

信 息（Info）： 数据（16 字节）

例 如： 从卡中读出的数据为“05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05”

05 05 05”

表 5.141 SL3 读数据块成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|--|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 07 | 0000 | 0010 |
| Info | | | | | Checksum |
| 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 | | | | | FEE5 |

3. 说明

在验证成功之后，才能读相应的块数据，若不同扇区的密钥相同，则所验证的块号与读块号不必在同一个扇区内。当读的块数不为 1 时，且读的块包含了区尾块（密钥/配置块），则该读操作会自动跳过区尾块读到下一个扇区的数据，若需要对区尾块进行访问时，则需要将读的起始地址设为区尾块的地址，读的块数设置为 1 即可。

PLUS CPU 卡的数据块、区尾块和 Mifare S50/70 卡分配相同，只是将块地址扩展为 2 字节。读命令可以根据需要设置如下不同的安全模式。

- 0x30～命令有 MAC；数据密文；回应无 MAC
- 0x31～命令有 MAC；数据密文；回应无 MAC
- 0x32～命令有 MAC；数据明文；回应无 MAC
- 0x33～命令有 MAC；数据明文；回应无 MAC
- 0x34～命令无 MAC；数据密文；回应无 MAC
- 0x35～命令无 MAC；数据密文；回应无 MAC
- 0x36～命令无 MAC；数据明文；回应无 MAC
- 0x37～命令无 MAC；数据明文；回应无 MAC

注意：PLUS S 系列的卡只支持“命令有 MAC，数据明文，回应有 MAC”这一种模式，数据是否加密是指——读写模块与卡之间的数据通信是否加密，而不是模块与主控制器间的数据是否加密。

5.5.9 SL3 写数据块（Cmd = P）

该命令用于写 SL3 的数据块，在写数据块之前必需成功执行一次密钥验证。

声明：*uint8_t PLUS_SL3Write(uint8_t ucMode, uint32_t usBNr, uint8_t ucExt, const uint8_t *pBuf)*

1. 主机命令

命令类型（CmdClass）： 0x07

命令代码（CmdCode）： ‘P’

信息长度（InfoLength）： 写的块数×16+4

信 息（Info）： 写模式（1 字节）： 0xA0～命令有 MAC；数据密文；回应无 MAC
0xA1～命令有 MAC；数据密文；回应无 MAC
0xA2～命令有 MAC；数据明文；回应无 MAC
0xA3～命令有 MAC；数据明文；回应无 MAC

起始块号（2 字节）

写的块数（1 字节）： 范围 1～3

写入的数据（写的块数×16 字节）

例如：将“05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05”用“命令有 MAC，数据明文，回应无 MAC”的方式写到第 0x0004 块。

表 5.142 SL3 写数据块命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|---|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 07 | 0050 | 0014 |
| Info | | | | | Checksum |
| A3 04 00 01 05 05 05 05 05 05 05 05 05 05 05 05 | | | | | FDEA |

2. 从机应答

状态 (Status): 0——成功，其它——失败
 信息长度 (InfoLength): 16
 信息 (Info): 数据 (16 字节)
 例如：从卡中读出的数据为“05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05”

表 5.143 SL3 写数据块成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 07 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF45 |

3. 说明

在验证成功之后，才能写相应的块数据，若不同扇区的密钥相同，则所验证的块号与写块号不必在同一个扇区内。当写的块数不为 1 时，且写的块包含了区尾块（密钥/配置块），则该写操作会自动跳过区尾块写到下一个扇区的数据，若需要对区尾块进行访问时，则需要将写的起始地址设为区尾块的地址，写的块数设置为 1 即可。

PLUS CPU 卡的数据块、区尾块和 Mifare S50/70 卡分配相同，只是将块地址扩展为 2 字节。写命令可以根据需要设置如下不同的安全模式。

- 0xA0~命令有 MAC；数据密文；回应无 MAC
- 0xA1~命令有 MAC；数据密文；回应无 MAC
- 0xA2~命令有 MAC；数据明文；回应无 MAC
- 0xA3~命令有 MAC；数据明文；回应无 MAC

注意：PLUS S 系列的卡只支持“命令有 MAC，数据明文，回应无 MAC”这一种模式，数据是否加密是指——读写模块与卡之间的数据通信是否加密，而不是模块与主控制器间的数据是否加密。

5.5.10 SL3 值块操作 (Cmd = S)

该命令用于写 SL3 的数据块，在写数据块之前必需成功执行一次密钥验证。

声明：*uint8_t PLUS_SL3ValueOperTran(uint8_t ucMode, uint32_t usSBNr, uint32_t usDBNr, long lValue)*

1. 主机命令

命令类型 (CmdClass): 0x07
 命令代码 (CmdCode): 'S'

信息长度 (InfoLength): 0x09

信 息 (Info): 值操作模式 (1 字节): 0xB7~加值
0xB9~减值

源块号 (2 字节)

目的块号 (2 字节)

值数据 (4 字节): 4 字节有符号数, 低字节在前, 高字节的符号位被忽略

例 如: 将 0x0004 块的值用“加值传输模式, 回应 MAC”方式加上 0x01234567 其结果存放到 0x0005。

表 5.144 SL3 值块操作命令帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | CmdCode | InfoLength |
|----------------------------|-----------|--------|----------|---------|------------|
| B2 | 00 | 00 | 07 | 0053 | 0009 |
| Info | | | | | Checksum |
| B7 04 00 05 00 67 45 23 01 | | | | | FD5A |

2. 从机应答

状 态 (Status): 0——成功, 其它——失败

信息长度 (InfoLength): 0

信 息 (Info): none

例 如: 加值成功模块的回应

表 5.145 SL3 值块操作成功回应帧

| LocalAddr | SlotIndex | SMCSeq | CmdClass | Status | InfoLength |
|-----------|-----------|--------|----------|--------|------------|
| B3 | 00 | 00 | 07 | 0000 | 0000 |
| Info | | | | | Checksum |
| none | | | | | FF45 |

注意: PLUS S 系列卡不支持该命令。

6. 免责声明

广州致远电子股份有限公司所提供的服务内容旨在协助客户加速产品的研发进度，在服务过程中所提供的任何程序、文档、测试结果、方案、支持等资料和信息，都仅供参考，客户有权不使用或自行参考修改，本公司不提供任何的完整性、可靠性等保证，若在客户使用过程中因任何原因造成的特别的、偶然的或间接的损失，本公司不承担任何责任。