

MR800 系列非接触 IC 卡 读写器说明书

(Revision 1.20)

北京金木雨电子有限公司

2011/6/10



在使用本产品前请仔细阅读本说明书，如果有任何疑问，请联系我们，我们会给您详尽的解答



目录

1	简介.....	3
2	技术指标.....	3
3	可读卡型.....	4
3.1	ISO14443A.....	4
3.2	ISO14443B.....	4
3.3	ISO15693.....	4
3.4	ISO7816.....	4
4	接口描述.....	5
4.1	硬件接口.....	5
4.2	上电复位信息(ATR).....	6
5	APDU 操作.....	7
5.1	非接触智能卡 (SmartCard).....	7
5.2	接触智能卡(SAM).....	7
5.3	非接触存储卡.....	7
5.4	非标准 APDU(PC/SC Part3 定义部分).....	7
5.4.1	GetData.....	8
5.4.2	LoadKey.....	9
5.4.3	Authentication.....	10
5.4.4	ReadBinaryBlock.....	11
5.4.5	UpdataBinaryBlock.....	12
5.4.6	ValueBlockOperation.....	14
5.4.7	ReadValueBlock.....	14
5.4.8	RestoreValueBlock.....	16
5.5	非标准 APDU(自定义部分).....	17
5.5.1	Set ISO14443A 寻卡模式.....	18
5.5.2	Halt TypeA 卡片.....	19
5.5.3	MifarePlus 从 Level0 切换到 Level1/3.....	19
5.5.4	Set ISO14443 TypeB 寻卡模式.....	19
5.5.5	Halt TypeB.....	19
5.5.6	AT88F020 Count.....	20
5.5.7	AT88F020 Deslect.....	20
5.5.8	AT88F020Lock.....	20
5.5.9	ISO15693 Inventory.....	20
5.5.10	ISO15693 Stay Quiet.....	21
5.5.11	ISO15693 Select Tag.....	21
5.5.12	IO15693 Reset to Ready.....	22
5.5.13	ISO15693 WriteBlock.....	22
5.5.14	ISO15693 Read Block.....	23
5.5.15	ISO15693 Write AFI.....	23
5.5.16	ISO15693 Lock AFI.....	23
5.5.17	ISO15693 Write DSFID.....	24



5.5.18	ISO15693 Lock DSFID.....	24
5.5.19	ISO15693 Get System info	24
5.5.20	ISO15693 Get M Blk Sec St	25
5.5.21	ISO15693 Lock Block.....	25
5.5.22	设置 SAM 波特率(SetPPS).....	25
5.5.23	设置 SAM 复位波特率	26
5.5.24	切换当前操作智能卡	27
5.5.25	初始化 RTC 时间(仅 MR800/810 支持)	28
5.5.26	读 RTC 时间(仅 MR800/810 支持)	28
5.5.27	设定 RTC 时间显示-时间(仅 MR800 支持)	28
5.5.28	设定 RTC 时间显示-日期(仅 MR800 支持)	29
5.5.29	设定 LCD 显示中文字体类型(仅 MR800 支持)	29
5.5.30	读取 LCD 显示中文字体类型(仅 MR800 支持)	29
5.5.31	LCD 显示指定个数的中文或英文字体(仅 MR800 支持)	30
5.5.32	LCD 显示图片(直接发送图片数据)(仅 MR800 支持).....	30
5.5.33	LCD 擦除行(仅 MR800 支持).....	31
5.5.34	LCD 设定开机画面(仅 MR800 支持).....	31
5.5.35	LCD 设定待机画面(仅 MR800 支持).....	32
5.5.36	LCD 背光控制(仅 MR800 支持).....	32
5.5.37	LCD 显示 Flash 中存储画面(仅 MR800 支持).....	33
5.5.38	读片外 Flash	33
5.5.39	写片外 Flash	34
5.5.40	获取产品序列号.....	34
5.5.41	获取硬件版本和版本号	34
5.5.42	LED 灯控制	35
5.5.43	蜂鸣器控制.....	35
5.5.44	天线状态设置.....	36
5.5.45	卡片加密方法设置.....	36
5.5.46	恢复出厂默认值(系统重新启动)	36
5.5.47	系统重新启动.....	36
6	卡片操作流程.....	38
6.1	Smart 接触和非接触卡.....	39
6.2	存储卡(非智能卡).....	40
附录 A	46



1 简介

MR800 系列 PC/SC 接口读卡器,采用 ARM7 微处理器,运行快速且稳定可靠,外形美观大方。MR800 系列读写器采用 NXP MF RC500 系列射频基站,支持读写的卡片种类丰富,可以读写符合 ISO14443 A、ISO14443B、ISO15683 标准的非接触卡,MR800 还可以读写符合 ISO7816 标准的 T=0 和 T=1 的 SAM 卡(具体支持卡片种类见章节 3)。

用户还可以自由选择是否带有 LCD 显示(MR800 带有 LCD,MR810/790 不带 LCD,MR790),LCD 显示模组为 128x64 点图形,可以支持开机画面,待机画面设定,图片存储等功能。

PC/SC 接口采用 Windows 操作系统自带驱动和 API 函数,用户开发简单且周期短。为了便于开发者的应用,我们提供了 VC、BC、VB、DELPHI 例子程序,开发者可以通过例子程序快速地开展开发工作。



2 技术指标

- 射频基站: RC400、RC500、RC531、RC632
- 电源: USB 供电
- 最大功耗: 150mA (100mA)
- 读写距离: 80mm (Mifare One)
- 外形尺寸: 123 * 88 * 25 (mm)
- 重量: 约 100 克 (不含底座)
- SAM 数量: 2 (支持 T=0/1 协议)
- 存储 Flash: 4M Bytes
- LED: 4 个 LED (红、绿、蓝、黄)
- ISP 功能: 支持
- LCD 分辨率: 128x64 点图形(MR800)
- Demo 软件: PTransWin
- 接口: USB PC/SC
- 备注:
 - ❖ 芯片类型根据客户选择读卡类型有区别。
 - ❖ 不同类型或厂家的卡片支持的读写距离可能有区别。
 - ❖ SAM 卡支持 T=0/1 协议,读卡器会根据复位信息自动选择协议通讯,不需用户设定,且 MR800 /810 支持 2 个 SAM,MR790 支持 3 个 SAM。
 - ❖ MR790 和 MR810 不支持蓝色 LED 灯。
 - ❖ 仅仅 MR800 支持 LCD 显示和 RTC。



3 可读卡型

3.1 ISO14443A

- Mifare One S50
- Mifare One S70
- Mifare Ultra Light
- Mifare Ultra Light C
- Mifare DesFire
- Mifare Plus (全功能支持)
- ISO14443-4 (T=CL) TYPE A 双界面 CPU 卡

3.2 ISO14443B

- AT88RF020
- AT88RF080
- SR176
- SRI512
- SRI1K
- SRI2K
- SRI4K
- SRIX4K
- ISO14443-4 (T=CL) TYPE B 双界面 CPU 卡

3.3 ISO15693

- I.CODE SLI
- Tag-it HF-I
- 其他的符合 ISO15693 标准的标签

3.4 ISO7816

- 符合 ISO7816 的 CPU(SAM)卡，支持 PPSS 操作
- 支持 T=0 和 T=1
- 支持默认任意速率卡片（9600，19200，38400，55800，57600，115200，230400bps）



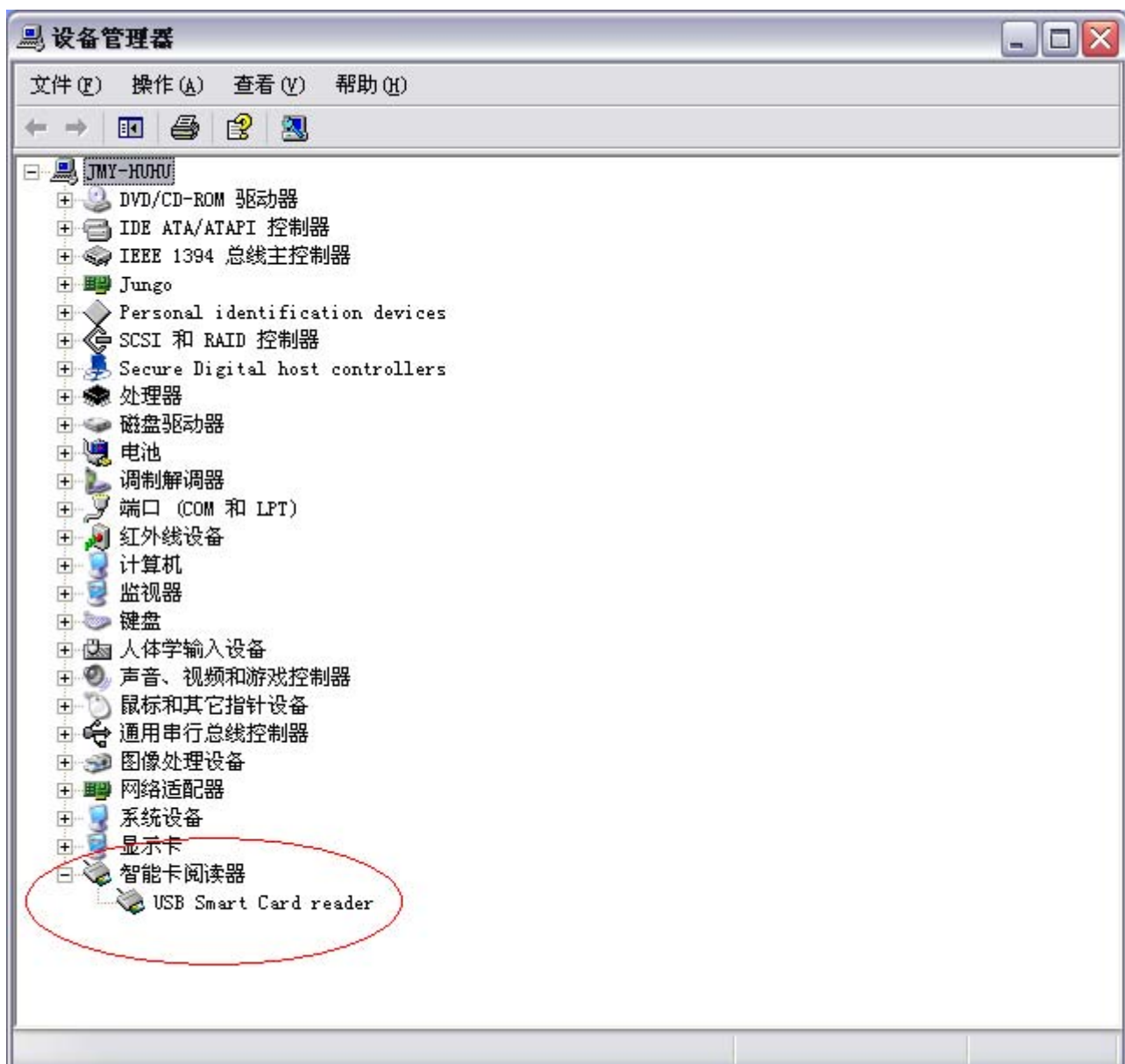
4 接口描述

4.1 硬件接口

MR800 读卡器采用的是 USB 接口，接口描述如下：

管脚	信号	功能
1	VBus	+5V 电源输入
2	D+	数据线+
3	D-	数据线-
4	GND	接地

上电后执行如下步骤可以检测读卡器是否连接好：**我的电脑**→**属性**→**硬件**→**设备管理器**。如下可以看到标注红色部分的 **USB Smart Card Reader**：





4.2 上电复位信息(ATR)

按照 PC/SC Part3 协议规定，设备上电返回 SmartCard 复位信息 ATR，为了使读卡器能够阅读更多非接触卡，MR800 采用返回固定的复位信息(未包括卡片信息)，ATR 信息格式如下：

Byte	Value(Hex)	Designation	Description
0	3B	Initial Header	
1	8N	T0	Higher nibble 8 means: no TA1, TB1, TC1 only TD1 is following. Lower nibble N is the number of historical bytes (HistByte 0 to HistByte N-1)
2	80	TD1	Higher nibble 8 means: no TA2, TB2, TC2 only TD2 is following. Lower nibble 0 means T = 0
3	01	TD2	Higher nibble 0 means no TA3, TB3, TC3,TD3 following. Lower nibble 1 means T = 1
4To3+N	80	T1	Category indicator byte, 80 means A status indicator may be present in an optional COMPACT-TLV data object
	4F	Tk	Application identifier Presence Indicator
	0C		Length
	RID		Registered Application Provider Identifier (RID) # A0 00 00 03 06
	SS		Byte for standard
	C0 .. C1		Bytes for card name
	00 00 00 00	RFU	RFU # 00 00 00 00
4+N	UU	TCK	Exclusive-oring of all the bytes T0 to Tk

针对 MR800 读卡器，我们返回 ATR 信息如下：ATR = {3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 00 00 00 00 00 68}



5 APDU操作

MR800 读卡器将 APDU 分为标准 APDU(APDU 中 Class 不是 0xFF)和非标准 APDU(APDU 中 Class 是 0xFF)。为了兼容 PC/SC 标准,对于非接触 SmartCard 和接触式 SAM 卡,除了 GetData 获取卡复位信息外,其余的标准 APDU 可以直接发送到 SmartCard 或 SAM 卡。因 MR800 支持非接触智能卡和接触智能卡(SAM),故在操作前也可以通过切换当前操作智能卡 APDU(APDU:FF 00 FA 00 01 CurSmartCard)切换当前操作智能卡(此处指的是接触和非接触智能卡之间的切换)。卡片操作流程见后面章节。对于存储卡,我们采用的是 Class =FF 非标准 APDU 指令操作,指令描述见后面章节。不论是接触 SmartCard,接触式 SAM 卡还是存储卡,所有对卡片的操作第一个步骤都是通过 GetData APDU 去获取卡片信息。

5.1 非接触智能卡 (SmartCard)

非接触智能卡采用的是标准 APDU 指令,在发送标准 APDU 指令前我们需要通过 GetData 指令获取 SmartCard ATR 数据。若在操作过程中需要读取接触智能卡 SAM,需要通过指令切换到指定的 SAM Slot(APDU:FF 00 FA 00 01 CurSmartCard)去读取相关数据。

5.2 接触智能卡(SAM)

MR800 带有 2 个 SAM 插槽,在发送标准 APDU 指令前我们需要通过 GetData 指令获取 SAM 卡复位信息。若在操作过程中需要读取非接触智能卡,则需要通过切换指令切换到非接触 SmartCard。

如:读卡器所读非接触卡类型,在操作过程中需通过 SAM 数据认证。

5.3 非接触存储卡

MR800 支持如 Mifare one/Ultralight 等存储卡,为了兼容 PC/SC 标准,我们定义了非标准 APDU,在发送非标准 APDU 指令前我们需要通过 GetData 指令去寻卡,获取卡片序列号信息。

5.4 非标准APDU(PC/SC Part3 定义部分)

除了 GetData APDU 既可以对存储卡,也可以 SmartCard/SAM 进行操作外,其它非标准 APDU 主要是用来实现存储类卡片的操作;标准 APDU 主要是用来对 SmartCard/SAM 类卡片的操作。

返回错误(SW1/SW2)状态如下:

结果	SW1	SW2	错误注释
成功	90	00	操作成功
错误	63	00	操作失败
错误	6A	81	功能不支持
错误	6B	00	P1-P2参数错误



5.4.1 GetData

该 APDU 指令是获取卡片序列号或复位信息。在操作一张卡片前，须首先执行该 APDU，因其中包含了对读卡器读卡类型的切换。

发送 APDU 格式：

Command	Class	INS	P1	P2	Le
GetData	FF	CA	CardType	SubCardType	00

CardType 和 SubCardType 定义如下：

ISO	CardType	SubCardType	
ISO14443 Type A	00: ISO14443 A Mifare card	00	
	01: ISO14443 A Smartcard(ISO14443-4)	00	
	02: MIFARE Ultra Light	00	
	03: Mifare Plus	00: Mifare PLUS Level0	
		01: Mifare PLUS Level1	
		02: Mifare PLUS Level2	
		03: Mifare PLUS Level3	
04: Mifare PLUS Level1for switch level			
ISO14443 Type B	20: ISO14443 B Smartcard(ISO14443-4)	00	
	21: SR176	00	
	22: SRIX4K/SRI512	00	
	23: AT88RF020	00	
ISO15693	40: ISO15693 Tag(Only one Tag)	00(NXP/TI Tag)	
ISO7816	60: ISO7816-Contact(T=0/T=1)	00: SAM1	
		01: SAM2	
		02: SAM3	
		03: SAM4	

MIFARE 1K/4K/UltraLight/MifarePlus Level1 (P1 = 00/02/03)应答：

Response	Data Out		
Result	UID Len(1Byte) + UID (LSB- 4/7Byte) + ATQA(2byte) + SAK(1Byte)	SW1	SW2

MIFARE Plus Level0/2/3/1 for switch 和 ISO14443 - 4 TypeA SmartCard (P1 = 03/01)应答：

Response	Data Out		
Result	UID Len(1Byte) + UID (LSB- 4/7Byte) + ATQA(2byte) + SAK(1Byte)+ATQA(nByte)	SW1	SW2

ISO14443 - 4 TypeB SmartCard/AT88F020(P1=20/23) 应答：

Response	Data Out		
Result	ATQB(12Byte)	SW1	SW2

SR176/SRIX4K(SRI512)(P1=21/22) 应答：



Response	Data Out		
Result	CHIPID(1Byte)+UID(8Byte)	SW1	SW2

ISO15693 Tag(P1=40)应答:

Response	Data Out		
Result	DSFID(1Byte)+UID(8Byte)	SW1	SW2

ISO7816 SAM(P1=60)应答:

Response	Data Out		
Result	Reset Info(nByte)	SW1	SW2

例如:

寻 TypeA 卡片:

Send: FF CA 00 00 00

Receive: 04 72 AE A6 9E 04 00 08 90 00

寻 ISO14443 TypeA Smartcard:

Send: FF CA 01 00 00

Receive: 04 50 3D CE EB 08 03 20 11 28 A1 53 43 41 5F 4F 5F 56 31 30 30 5F 54 64
90 00

ISO14443 TypeB SmartCard:

Send: FF CA 20 00 00

Receive: 50 C0 1281 89 54 46 22 08 00 80 A1 90 00

5.4.2 LoadKey

该 APDU 是用来保存卡片授权密钥和密钥传输时加密密钥。装载的密钥可以选择保存还是不保存，不保存的密钥暂时存放在 RAM 中，断电易失；保存的密钥保存于 Flash，断电后不丢失。MR800 最多保存卡片密钥个数是 32，且每个密钥最大长度是 16 字节，若授权密钥小于 16 字节，则取低字节密钥授权。最多保存读卡器密钥个数是 1 条。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
Load Key	FF	82	Key Structure	Key Index	1~16	Key Data (LSB)

Key Structure:

b7	b6	b5	b4	b3	b2	b1	b0	Description
X								0: 卡片密钥 1: 读卡器密钥
	X							0: 明文传输 1: 密文传输
		X						0: 暂时存储 1: 非易失性存储
			X	X	X	X	X	RFU

卡片密钥是用来对卡片授权的密钥，读卡器密钥是对卡片密钥载入时的加密密钥。加密方式是 3DES 加密，所以读卡器密钥必须是 16 字节。所有加密的卡片密钥必须是 8 字节的倍数，不够的在高字节补 00，如 Mifare one 密钥是 FF FF FF FF FF FF 6 字节密钥，假如密钥下载选择密文传输，则先补 0 为 FF FF FF FF FF FF 00 00 (LSB..MSB) 然后再加密。若明文传输则不需要补 0。出厂默认所有密钥都为 0。

**密钥存储结构:**

Key Index	卡片密钥(Byte)	读卡器密钥(Byte)
0	16	16
1	16	-
.....	16	-
31	16	-

(备注: 卡片密钥索引 0~31, 读卡器密钥索引只有 0)

应答:

Response	Data Out	
Result	SW1	SW2

例如:

明文传输 ReaderKey, 不保存:

Send: FF 82 80 00 10 33 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF

Receive: 90 00

5.4.3 Authentication

该 APDU 主要用在带有密钥保护的卡片进行授权。在 GetData 指令后, 若卡片带有密钥保护功能, 则需要通过这个 APDU 对卡片授权后才能对其进行读写操作。一般需要授权的卡片有: Mifare S50/70、MifarePlus、AT88F020。授权可以采用已经存储的密钥或当前下载的密钥授权两种方式中的任意一种。

发送 APDU 格式:

Command	Class	INS	P1	P2	P3	Data
Authenticate	FF	88	High Address	Low Address	Key Type	KeyCofig + KEY

P1/P2:

对于 Mifare S50/70 则卡片块地址。

对于 AT88F020, 则该地址无效(P1=0,P2=0)

对于 MifarePlus Level1/2/3, 则为 AES 密钥存储块的地址(注意 密钥存储块和数据块是一一对应关系, 请参考 MifarePlus 数据手册)。

KeyType: 密钥类型(仅仅在 Mifare S50/S70, 该字节有效: A Key—0x60, B Key—0x61)

KeyConfig:

b7	b6-b0	Meaning
0	XXXXXXXX	XXXXXXXX表示用当前输入密钥KEY的长度, 卡片采用当前密钥授权
1	XXXXXXXX	XXXXXXXX表示存储于读卡器密钥索引, 卡片采用存储的密钥授权

KEY: 若 KeyConfig Bit7 = 0, Key 表示密钥, 密钥长度根据卡片类型的不同而不同; 若 KeyConfig Bit7 = 1, Key 内容不存在。

应答:

Response	Data Out	
Result	SW1	SW2

例如:

寻 Mifare S50 卡片, 并且读第一块:

Send: FF CA 00 00 00



Receive: 04 72 AE A6 9E 04 00 08 90 00
Send: FF 88 00 01 60 06 FF FF FF FF FF FF
Receive: 90 00
Send: FF B0 00 01 10
Receive: 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 90 00

寻 MifarePlus Level3 卡片，并且读数据块 0:

Send: FF CA 03 03 00
Receive: 07 04 8B AD 04 05 06 07 42 00 31 0C 75 77 84 02 4D 46 50 5F 45 4E 47 90
 00
Send: FF 88 40 00 00 10 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF (数
 据块 1 对应的密钥地址是 0x4000 或 0x4001)
Receive: 90 00
Send: FF B0 00 01 10
Receive: 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 90 00

寻 AT88F020 卡片，并且读数据块 9:

Send: FF CA 23 00 00
Receive: 50 00 04 E8 51 00 00 00 00 00 00 41 90 00
Send: FF 88 00 00 00 08 00 00 00 00 00 00 00 00 00
Receive: 90 00
Send: FF B0 00 09 08
Receive: 00 00 00 00 00 00 00 00 90 00

5.4.4 ReadBinaryBlock

该 APDU 主要是根据 GetData APDU 指定的寻卡类型来读取卡片存储块的内容。若卡片带有密码保护，则读取卡片块内容前，先对卡片进行授权(APDU: Authentication)。

发送 APDU 格式:

Command	Class	INS	P1	P2	Le
Read Binary	FF	B0	High Address	Low Address	Data Len

P1/P2: 所读块地址

DataLen: 所读数据长度(所有数据都是低字节在前)

MIFARE 1K/4K 16字节
 MifarePlus 16字节(Level3支持多块读)
 MIFARE Ultralight 每块4字节，但是一次能读出4块 = 16字节
 SR176 2字节
 SR512 2字节
 SR1X4K 2字节
 AT88RF020 8字节
 ISO15693 Tag 4字节(支持多块读)

该 APDU 支持读多块指令(注意: 卡片也必须支持多块读)。若读 ISO15693Tag 连续 2 块，那么 DataLen = 4x2 = 8。注意该 APDU 对 ISO15693 Tag 的读操作是对最后一次寻到的 Tag 操作，若对选择或指定 UID 的 tag 操作请参考 3.5 章节非标准 APDU(自定义部分)。

应答:



Response	Data Out		
Result	Data	SW1	SW2

例如:

读 SR176 卡片第 10 块:

Send: FF CA 21 00 00
 Receive: 20 42 2F 69 18 08 92 D0 02 90 00
 Send: FF B0 00 0A 02
 Receive: 00 00 90 00

读 MIFARE Ultralight 第 10 块:

Send: FF CA 02 00 00
 Receive: 07 04 24 A2 E1 BF 02 80 44 00 00 90 00
 Send: FF B0 00 0A 10
 Receive: 11 22 33 44 00 00 00 00 00 00 00 00 00 00 90 00

读 ISO15693 Tag 从第 10 块开始的 2 块(即第 10、11 块):

Send: FF CA 40 00 00
 Receive: 00 3D 3D 08 17 00 01 04 E0 90 00
 Send: FF B0 00 0A 08
 Receive: 00 00 00 00 00 00 00 00 90 00

5.4.5 UdataBinaryBlock

写块操作会根据 GetData APDU 指定的寻卡类型来对其写操作。若卡片带有密码保护, 则写卡片块内容前, 先对卡片进行授权(APDU: Authentication)。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
UdataBinary	FF	D6	High Address	Low Address	Data Len	Data

P1/P2: 所写块地址

DataLen: 所写数据长度(所有数据都是低字节在前)

MIFARE 1K/4K 16字节
 MifarePlus 16字节(Level3支持多块写)
 MIFARE Ultralight 4字节
 SR176 2字节
 SR512 4字节
 SR1X4K 2字节
 AT88RF020 8字节
 ISO15693 Tag 4字节

该 APDU 支持写多块指令(注意: 卡片也必须支持多块写)。若写 ISO15693Tag 连续 2 块, 则 DataLen = 4x2 = 8。注意该 APDU 对 ISO15693 Tag 的读操作是对最后一次寻到的 Tag 操作, 若对选择或指定 UID 的 tag 操作请参考 3.5 章节非标准 APDU(自定义部分)。

应答:

Response	Data Out	
Result	SW1	SW2

例如:

**寻 Mifare S50 卡片，并且写读第一块：**

Send: FF CA 00 00 00
Receive: 04 72 AE A6 9E 04 00 08 90 00
Send: FF 88 00 01 60 06 FF FF FF FF FF FF
Receive: 90 00
Send: FF D6 00 01 10 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 00
Receive: 90 00
Send: FF B0 00 01 10
Receive: 01 10 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 00

寻 MifarePlus Level1 卡片，并读写第 4 块：

Send: FF CA 03 01 00
Receive: 04 72 AE A6 9E 04 00 08 90 00
Send: FF 88 00 04 60 06 FF FF FF FF FF FF
Receive: 90 00
Send: FF D6 00 04 10 00 00 00 04 05 06 07 08 09 0A 0B 0C 0D 0E 01 00
Receive: 90 00
Send: FF B0 00 04 10
Receive: FF D6 00 04 10 00 00 00 04 05 06 07 08 09 0A 0B 0C 0D 0E 01 00

读写 MIFARE Ultralight 第 10 块：

Send: FF CA 02 00 00
Receive: 07 04 24 A2 E1 BF 02 80 44 00 00 90 00
Send: FF D6 00 0A 04 00 01 02 03
Receive: 90 00
Send: FF B0 00 0A 10
Receive: 00 01 02 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 90 00

读写 MifarePlus Level3 第 1 块：

Send: FF CA 03 03 00
Receive: 07 04 8B AD 04 05 06 07 42 00 31 0C 75 77 84 02 4D 46 50 5F 45 4E 47 90 00
Send: FF 88 40 00 00 10 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF (数据块 1 对应的密钥地址是 0x4000 或 0x4001)
Receive: 90 00
Send: FF D6 00 01 10 00 00 00 04 05 06 07 08 09 0A 0B 0C 0D 0E 01 00
Receive: 90 00
Send: FF B0 00 01 10
Receive: 00 00 00 04 05 06 07 08 09 0A 0B 0C 0D 0E 01 00 90 00

写读 SR176 卡片第 10 块：

Send: FF CA 21 00 00
Receive: 20 42 2F 69 18 08 92 D0 02 90 00
Send: FF D6 00 0A 02 00 01
Receive: 90 00
Send: FF B0 00 0A 02
Receive: 00 01 90 00

寻 AT88F020 卡片，并且读数据块 9：



Send: FF CA 23 00 00
Receive: 50 00 04 E8 51 00 00 00 00 00 00 41 90 00
Send: FF 88 00 00 00 08 00 00 00 00 00 00 00
Receive: 90 00
Send: FF D6 00 09 08 00 01 02 03 04 05 06 07
Receive: 90 00
Send: FF B0 00 09 08
Receive: 00 01 02 03 04 05 06 07 90 00

读 ISO15693 Tag 从第 10 块开始的 2 块(即第 10、11 块):

Send: FF CA 40 00 00
Receive: 00 3D 3D 08 17 00 01 04 E0 90 00
Send: FF D6 00 0A 04 00 01 02 03
Receive: 90 00
Send: FF B0 00 0A 04
Receive: 00 01 02 03 90 00

5.4.6 ValueBlockOperation

值块操作仅限于带有钱包功能的卡片，如：Mifare S50/70, MifarePlus Level1/3. 值块操作包括：初始化钱包、充值、扣款。若卡片带有密码保护，则操作卡片块内容前，先对卡片进行授权(APDU: Authentication)。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
ValueBlock	FF	D7	HighAddress	LowAddress	05	VB_OP+VB_Value

P1/P2: 块地址

VB_OP(1Byte): 0x00-初始化钱包
 0x01-充值
 0x02-扣款

VB_Value(4Byte): 钱包值，低字节在前。

应答:

Response	Data Out	
Result	SW1	SW2

5.4.7 ReadValueBlock

读钱包操作仅仅限于带有钱包功能的卡片，如：Mifare S50/70, MifarePlus Level1/3. 若卡片带有密码保护，则读卡片块内容前，先对卡片进行授权(APDU: Authentication)。

发送 APDU 格式:

Command	Class	INS	P1	P2	Le
ReadValueBlock	FF	B1	HighAddress	LowAddress	04

P1/P2: 所读块地址

应答:

Response	Data Out
----------	----------



Result	Value(4Byte)	SW1	SW2
--------	--------------	-----	-----

例如:

Mifare S50 初始化钱包, 充值, 扣款, 读钱包:

Send: FF CA 00 00 00
 Receive: 04 72 AE A6 9E 04 00 08 90 00
 Send: FF 88 00 01 60 06 FF FF FF FF FF FF
 Receive: 90 00
 Send: FF D7 00 01 05 00 00 00 00 01
 Receive: 90 00
 Send: FF B1 00 01 04
 Receive: 00 00 00 01 90 00
 Send: FF D7 00 01 05 01 00 00 00 02
 Receive: 90 00
 Send: FF B1 00 01 04
 Receive: 00 00 00 03 90 00
 Send: FF D7 00 01 05 02 00 00 00 01
 Receive: 90 00
 Send: FF B1 00 01 04
 Receive: 00 00 00 02 90 00

MifarePlus Level1 初始化钱包, 充值, 扣款, 读钱包:

Send: FF CA 03 01 00
 Receive: 07 04 8C AF 04 05 06 07 42 00 18 90 00
 Send: FF 88 00 04 60 06 FF FF FF FF FF FF
 Receive: 90 00
 Send: FF D7 00 04 05 00 00 00 00 01
 Receive: 90 00
 Send: FF B1 00 04 04
 Receive: 00 00 00 01 90 00
 Send: FF D7 00 04 05 01 00 00 00 02
 Receive: 90 00
 Send: FF B1 00 04 04
 Receive: 00 00 00 03 90 00
 Send: FF D7 00 04 05 02 00 00 00 01
 Receive: 90 00
 Send: FF B1 00 04 04
 Receive: 00 00 00 02 90 00

MifarePlus Level3 初始化钱包, 充值, 扣款, 读钱包(Block =0x01):

Send; FF CA 03 03 00
 Receive: 07 04 8B AD 04 05 06 07 42 00 31 0C 75 77 84 02 4D 46 50 5F 45 4E 47 90 00
 Send: FF 88 40 00 00 10 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
 Receive: 90 00
 Send: FF D7 00 01 05 00 00 00 00 01
 Receive: 90 00
 Send: FF B1 00 01 04



Receive: 00 00 00 01 90 00
Send: FF D7 00 01 05 01 00 00 00 02
Receive: 90 00
Send: FF B1 00 01 04
Receive: 00 00 00 03 90 00
Send: FF D7 00 01 05 02 00 00 00 01
Receive: 90 00
Send: FF B1 00 01 04
Receive: 00 00 00 02 90 00

5.4.8 RestoreValueBlock

备份值块操作仅仅限于带有钱包功能的卡片，如：Mifare S50/70, MifarePlus Level1/3。备份值块操作时，目标值块和源值块需在同一个扇区。若卡片带有密码保护，则操作卡片块内容前，先对卡片进行授权(APDU: Authentication)。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
Restore Value Block	FF	D7	Source High Address	Source Low Address	03	03+TargetAddress

P1/P2: 源块地址

TargetAddress: 目标地址(2Byte, 高地址在前)

应答:

Response	Data Out	
Result	SW1	SW2



5.5 非标准APDU(自定义部分)

非标准 APDU(自定义部分)是对 PC/SC Part3 定义的非标准 APDU 功能的扩展。该部分指令是通过 FF 类指令 INS = 00 进行扩展。该部分指令可以实现当前操作智能卡切换、LCD 显示、Beep/LED 控制等。具体内容见下列表：

扩展命令列表：

Class	Ins	P1		P2	Le/Lc	功能	
FF	00	ISO14443 Type A (0x00~0x1F)	MifareClass (0x00)	00		设定TypeA寻卡模式	
					01		HaltA卡片
			MifarePlus (0x03)	00			从Level0切换到Level1/3
		ISO14443 TypeB (0x20~0x3F)	ISO14443SMARTB (0x20)	00			TypeB寻卡模式
				01			HaltB
			AT88F020 (0x23)	00			AT88F020 COUNT
				01			AT88F020 Deselect
				02			AT88F020 Lock block
				00			MultiTag Inventory
		ISO15693 (0x40~0x5F)	Tag (0x40)	01			Stay Quiet
				02			Select Tag
				03			Reset to Ready
				04			Read Block
				05			Write Block
				06			Write AFI
				07			Lock AFI
				08			Write DSFID
				09			Lock DSFID
				0A			Get System info
				0B			Get M Blk Sec St
				0C			Lock Block
		ISO7816 (0x60~0x6F)	0x60	00			设置SAM1 PPSBaud
				01			设置SAM2 PPSBaud
				02			设置SAM3 PPSBaud
				03			设置SAM4 PPSBaud
				04			设置SAM1 RSTBaud
				05			设置SAM2 RSTBaud
				06			设置SAM3 RSTBaud
				07			设置SAM4 RSTBaud
		SYSTEM (0xE0~0xFF)	智能卡切换 (0xFA)	00			智能卡操作类别切换(非接触和接触)



			RTC 操作 (0xFB)	00	初始化时间
				01	读时间
				02	设定LCD显示时间
				03	设定LCD显示日期
			LCD&&LED数码管操作 (0xFC)	00	设置显示字体类型
				01	读取显示字体类型
				02	显示指定个数字符
				03	显示图片(直接下载数据)
				04	擦除LCD
				05	设定开机图片
				06	设定待机界面
				07	LCD背光控制
			Flash操作 (字体下载0xFD)	00	读Flash
				01	写Flash
			RFU(0xFE)	-	系统保留指令
			系统指令 (0xFF)	00	获取序列号
				01	获取版本号(硬件&&软件)
				02	LED 灯控制
				03	蜂鸣器操作
				04	天线状态设置
				05	设置卡片加密标准
				06	恢复出厂默认值
				07	Reader重新启动

5.5.1 Set ISO14443A 寻卡模式

设置 ISO14443 TypeA 寻卡模式。ISO14443 TypeA 寻卡模式上电默认值是 REQA(0x26)。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
SetRequestModeA	FF	00	00	00	01	RequestMode

Request Mode:

0x26- REQA

0x52- WUPA

应答:

Response	Data Out	
Result	SW1	SW2



5.5.2 Halt TypeA 卡片

使符合 ISO14443 TypeA 卡片进入休眠模式。

发送 APDU 格式:

Command	Class	INS	P1	P2	Le
Halt A	FF	00	00	01	00

应答:

Response	Data Out	
Result	SW1	SW2

5.5.3 MifarePlus 从Level0 切换到Level1/3

在 Level0 初始化完毕后, 可以通过该 APDU 从 Level0 切换到 Level1 或 Level3。切换到的目标层级依据卡片类型而定。注意, 在 MifarePlus 卡片出厂时, 默认层级是 Level0, 在切换到其它 Level 前需要通过 WriteBinary APDU 写入一些块参数(如: 切换前必须写入 0x9000/0x9001/0x9002/0x9003 地址设置值)。

发送 APDU 格式:

Command	Class	INS	P1	P2	Le
SwitchLevel	FF	00	01	00	00

应答:

Response	Data Out	
Result	SW1	SW2

5.5.4 Set ISO14443 TypeB 寻卡模式

设置 ISO14443 TypeB 寻卡模式。ISO14443 TypeB 寻卡模式上电默认值是 REQB (0x00)。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
SetRequestModeB	FF	00	20	00	01	RequestMode

RequestMode:

0x00- REQB

0x01- WUPB

应答:

Response	Data Out	
Result	SW1	SW2

5.5.5 Halt TypeB

使符合 ISO14443 TypeB 卡片进入休眠模式。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
---------	-------	-----	----	----	----	------



HaltB	FF	00	20	01	04	PUPI
--------------	----	----	----	----	----	------

PUPI: TypeB 卡片序号

应答:

Response	Data Out	
Result	SW1	SW2

5.5.6 AT88F020 Count

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
AT88F020Count	FF	00	23	00	06	Signature

Signature: 6 字节

应答:

Response	Data Out	
Result	SW1	SW2

5.5.7 AT88F020 Deslect

发送 APDU 格式:

Command	Class	INS	P1	P2	Le
AT88F020Deselect	FF	00	23	01	00

应答:

Response	Data Out	
Result	SW1	SW2

5.5.8 AT88F020Lock

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
AT88F020Lock	FF	00	23	02	04	LockData

应答:

Response	Data Out	
Result	SW1	SW2

5.5.9 ISO15693 Inventory

除了通过 GetData 获取 Tag 标签 UID 外, 也可以通过该 APDU 实现寻单张或多张 Tag 标签, 标签的数量要看天线承载能力。注意该指令和 GetData APDU 同样具有切换寻卡类型的功能, 使用该 APDU, 寻卡类型自动切换到 ISO15693。

发送 APDU 格式:



Command	Class	INS	P1	P2	Lc	Data
Inventory	FF	00	40	00	03	Type+Flag+AFI

Type:

0x00—寻一张标签(如Flag = x26)

0x01—寻多张标签

Flag: 见 ISO15693 标准**AFI:** 指定所寻标签的应用标识符(AFI)**应答:**

Response	Data Out		
Result	((DSFID(1Byte)+UID(8Byte))*n	SW1	SW2

例如:**寻 ISO15693 单张 Tag:****Send:** FF 00 40 00 03 00 26 00**Receive:** 00 3D 3D 08 17 00 01 04 E0 90 00**Send:** FF 00 40 01 09 22 3D 3D 08 17 00 01 04 E0(休眠)**Receive:** 90 00**Send:** FF 00 40 00 03 00 26 00**Receive:** 63 00**Send:** FF 00 40 03 09 22 3D 3D 08 17 00 01 04 E0**Receive:** 00 3D 3D 08 17 00 01 04 E0 90 00**Send:** FF 00 40 00 03 00 26 00**Receive:** 00 3D 3D 08 17 00 01 04 E0 90 00

5.5.10 ISO15693 Stay Quiet

ISO15693 Tag 休眠。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
Stayquiet	FF	00	40	01	09	Flag+UID

Flag: 见 ISO15693 标准(如: Flag =0x22)**UID:** 待休眠卡片 UID(8Byte)**应答:**

Response	Data Out	
Result	SW1	SW2

5.5.11 ISO15693 Select Tag

ISO15693 Tag 选卡操作。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
SelectTag	FF	00	40	02	09	Flag+UID

Flag: 见 ISO15693 标准(如: Flag = 0x22)



UID: 卡片 UID(8Byte)

应答:

Response	Data Out	
Result	SW1	SW2

例如:

选择一张卡片, 并进行读写操作:

Send: F 00 40 00 03 00 26 00

Receive: 00 3D 3D 08 17 00 01 04 E0 90 00

Send: F 00 40 02 09 22 3D 3D 08 17 00 01 04 E0

Receive: 9000

Send: F 00 40 05 0E 12 00 00 00 00 00 00 00 0A 11 22 33 44

Receive: 9000

Send: F 00 40 04 0B 12 00 00 00 00 00 00 00 00 0A 01

Receive: 11 22 33 44 90 00

5.5.12 IO15693 Reset to Ready

ISO15693 Tag 从 Halt 到 Ready 状态。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
ResetToReady	FF	00	40	03	09	Flag+UID

Flag: 见 ISO15693 标准(如: Flag = x22)

UID: 卡片 UID(8Byte)

应答:

Response	Data Out	
Result	SW1	SW2

5.5.13 ISO15693 WriteBlock

ISO15693 Tag 写块。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
WriteBlock	FF	00	40	05	0E	Flag + UID + BlockAddr + BlockData

Flag: 见 ISO15693 标准(如: Flag = 0x22 或 0x12(Selected tag))

UID: 卡片 UID(8Byte)

BlockAddr: 起始块地址(1Byte)

BlockData: 块数据(4 Byte)

应答:

Response	Data Out	
Result	SW1	SW2



5.5.14 ISO15693 Read Block

ISO15693 Tag 读块。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
ReadBlock	FF	00	40	04	0B	Flag + UID + BlockAddr + BlockNum

Flag: 见 ISO15693 标准(如: Flag = 0x22 或 0x12(Selected tag))

UID: 卡片 UID(8Byte)

BlockAddr: 起始块地址

BlockNum: 根据不同的卡片支持读取块数不同(最小是 0)

应答:

Response	Data Out	
Result	SW1	SW2

5.5.15 ISO15693 Write AFI

写 ISO15693 Tag AFI。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
Write AFI	FF	00	40	06	0A	Flag+UID+AFI

Flag: 见 ISO15693 标准(如: Flag = 0x22 或 0x12(Selected tag))

UID: 卡片 UID(8Byte)

AFI: 新的 AFI

应答:

Response	Data Out	
Result	SW1	SW2

例如:

写 AFI Send: FF 00 40 06 0A 22 3D 3D 08 17 00 01 04 E0 00

Receive: 90 00

5.5.16 ISO15693 Lock AFI

锁 ISO15693 Tag AFI。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
LockAFI	FF	00	40	07	09	Flag+UID

Flag: 见 ISO15693 标准(如: Flag = 0x22 或 0x12(Selected tag))

UID: 卡片 UID(8Byte)

应答:

Response	Data Out	
Result	SW1	SW2



Result	SW1	SW2
--------	-----	-----

5.5.17 ISO15693 Write DSFID

写 ISO15693 Tag DSFID。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
WriteDSFID	FF	00	40	08	0A	Flag+UID(8Byte)+DSFID

Flag: 见 ISO15693 标准(如: Flag = 0x22 或 0x12(Selected tag))

UID: 卡片 UID(8Byte)

DSFID: 新的 DSFID

应答:

Response	Data Out	
Result	SW1	SW2

例如:

写 DSFID:

Send: FF 00 40 08 0A 22 3D 3D 08 17 00 01 04 E0 00

Receive: 90 00

5.5.18 ISO15693 Lock DSFID

锁 ISO15693 Tag DSFID。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
LockDSFID	FF	00	40	09	09	Flag+UID

Flag: 见 ISO15693 标准(如: Flag = 0x22 或 0x12(Selected tag))

UID: 卡片 UID(8Byte)

应答:

Response	Data Out	
Result	SW1	SW2

5.5.19 ISO15693 Get System info

获取 ISO15693 Tag 系统信息

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
GetSysInfo	FF	00	40	0A	09	Flag+UID

Flag: 见 ISO15693 标准(如: Flag = 0x22(需要带 UID),Flag = 0x02(可以不带 UID))

UID: 卡片 UID(8Byte)

应答:

Response	Data Out		
Result	System Info	SW1	SW2



SystemInfo: InfoFlag(1Byte)+UID(8Byte)+DSFID(1Byte)+AFI(1Byte)+Other(nByte)

例如:

Get system information:

Send: FF 00 40 0A 09 22 3D 3D 08 17 00 01 04 E0

Receive: 0F 3D 3D 08 17 00 01 04 E0 01 00 1B 03 01 90 00

5.5.20 ISO15693 Get M Blk Sec St

获取 ISO15693 Tag 块安全状态

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
GetMultiBlkSecSt	FF	00	40	0B	0B	Flag + UID + StartAddr+Num

Flag: 见 ISO15693 标准(如: Flag = 0x22)

UID: 卡片 UID(8Byte)

StartAddr: 开始块(1Byte)

Num: 块数(最小 0)

应答:

Response	Data Out		
Result	BlockSecSta ×Num	SW1	SW2

例如:

获取 ISO15693 第 10、11、12 块安全状态:

Send: FF 00 40 0B 09 22 3D 3D 08 17 00 01 04 E0 0A 02

Receive: 00 00 00 90 00

5.5.21 ISO15693 Lock Block

锁 ISO15693 Tag DSFID。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
LockDSFID	FF	00	40	0C	0A	Flag+UID+BlockNO

Flag: 见 ISO15693 标准(如: Flag = 0x22 或 0x12(Selected tag))

UID: 卡片 UID(8Byte)

BlockNO: 待锁块号

应答:

Response	Data Out	
Result	SW1	SW2

5.5.22 设置SAM波特率(SetPPS)

该功能主要是设置 SAM 卡通讯波特率。每个读卡器支持的 SAM 个数可能不同,详情请参考读卡器说明书(MR800 支持 2 个 SAM)。在发送 GetData APDU 复位 SAM 卡后,若修改 SAM 卡波特率(注:该 SAM 卡必须支持所设置波特率),可发送该 APDU 去设置(PPS)。

**发送 APDU 格式:**

Command	Class	INS	P1	P2	Lc	Data
SetSamBaud	FF	00	60	SAMPPS	01	Baudrate

SAMPPS:

- 0- SAM0 SetPPS
- 1- SAM1 SetPPS
- 2- SAM2 SetPPS
- 3- SAM3 SetPPS

Baudrate:

- 0- 9600 (默认)
- 1- 19200
- 2- 38400
- 3- 55800
- 4- 57600
- 5- 115200

应答:

Response	Data Out	
Result	SW1	SW2

5.5.23 设置SAM 复位波特率

该功能主要是设置 SAM 复位时采用的波特率。每个读卡器支持的 SAM 个数可能不同，详情请参考读卡器说明书(MR800 支持 2 个 SAM)。一般默认情况下，SAM 卡默认复位波特率是 9600，若想修改 SAM 复位波特率，在发送 GetData APDU 复位 SAM 卡前，可发送该 APDU 去设置 SAM 复位波特率 (注：该 SAM 卡必须支持所设置的复位波特率)。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
SetRstSamBaud	FF	00	60	SAMRestBaudNO	01	Baudrate

SAMRestBaudNO:

- 4- SAM0 Reset Baudrate
- 5- SAM1 Reset Baudrate
- 6- SAM2 Reset Baudrate
- 7- SAM3 Reset Baudrate

Baudrate:

- 0- 9600 (默认)
- 1- 19200
- 2- 38400
- 3- 55800
- 4- 57600
- 5- 115200

应答:

Response	Data Out	
Result	SW1	SW2



5.5.24 切换当前操作智能卡

该功能主要实现非接触 SmartCard 和 接触的 SAM 之间切换。因为非接触 SmartCard 和 SAM 卡除了寻卡和复位使用非标准 APDU(GetData)外, 其余都是发送标准的 APDU 指令。为了区分当前操作的是 SmartCard 还是 SAM 卡, 通过此指令可以实现切换。在实际应用中, 有时已经通过 GetData 寻到 smartcard 后, 需要通过 SAM 卡进行认证, 那么需要通过该 APDU 暂时将对智能卡的操作对象切换到 SAM, 操作完毕后需再切换到 SmartCard。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
SwitchSmartCard	FF	00	FA	00	01	CurSmartCard

CurSmartCard:

- 0- 非接触智能卡
- 1- SAM1 卡
- 2- SAM2 卡
- 3- SAM3 卡
- 4- SAM4 卡

应答:

Response	Data Out	
Result	SW1	SW2

例如:

同时操作 SmartCard 和 SAM 卡, 操作流程图:





5.5.25 初始化RTC时间(仅MR800/810 支持)

该功能实现对读卡器内部时钟初始化操作。若需要时间能掉电保持，需要配备电池。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
InitialRTC	FF	00	FB	00	08	Time

Time: 年(High Byte)+年(Low Byte) 十月(Month)+日(Date) 十时(Hour) 十分(Minute)+秒(Second)+星期(Week)

如: 2010-4-12 12:01:00 星期一 时间数据是: 07 DA 04 0C 0C 01 00 01

应答:

Response	Data Out	
Result	SW1	SW2

例如:

设置时间并读出时间:

Send: FF 00 FB 00 08 07 DA 04 0C 0C 01 00 01

Receive: 90 00

Send: FF 00 FB 01 08

Receive: 07 DA 04 0C 0C 03 15 01 90 00

5.5.26 读RTC时间(仅MR800/810 支持)

该功能实现读取读卡器内部时钟。若需要时间能掉电保持，需要配备电池。

发送 APDU 格式:

Command	Class	INS	P1	P2	Le
ReadRTC	FF	00	FB	01	08

应答:

Response	Data Out		
Result	Time	SW1	SW2

Time: 年(High Byte)+年(Low Byte) 十月(Month)+日(Date) 十时(Hour) 十分(Minute)+秒(Second)+星期(Week)

如: 2010-4-12 12:01:00 星期一 时间数据是: 07 DA 04 0C 0C 01 00 01

5.5.27 设定RTC时间显示-时间(仅MR800 支持)

该功能主要是设置时间在 LCD 上的显示模式。若需要时间能掉电保持，需要配备电池。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
DisTime	FF	00	FB	02	03	EnableFag+Line+Column

EnableFag: 日期显示使能(0-Disable, 1-Enable)

Line: 显示起始行(0~7)



Column: 显示起始列(0~127)

如显示 12 点 10 分 10 秒, 显示为: 12:10:10

应答:

Response	Data Out	
Result	SW1	SW2

5.5.28 设定RTC时间显示-日期(仅MR800 支持)

该功能主要是设置日期在 LCD 上的显示模式。若需要时间能掉电保持, 需要配备电池。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
DisDate	FF	00	FB	03	03	EnableFag+Line+Column

EnableFag: 日期显示使能(0-Disable, 1-Enable)

Line: 显示起始行(0~7)

Column: 显示起始列(0~127)

如显示 2010-04-16, 显示为: 10/04/16

应答:

Response	Data Out	
Result	SW1	SW2

5.5.29 设定LCD显示中文字体类型(仅MR800 支持)

MR800 支持 **简体**和 **繁体**两类中文字体。通过该指令可实现简体和繁体字库切换。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
SetFontType	FF	00	FC	00	01	ChineseFontType

ChineseFontType: 简体中文(默认)和繁体中文

应答:

Response	Data Out	
Result	SW1	SW2

5.5.30 读取LCD显示中文字体类型(仅MR800 支持)

MR800 支持简体和繁体中文字体。通过该指令可获知当前显示的中文字体类型。

发送 APDU 格式:

Command	Class	INS	P1	P2	Le
ReadFontType	FF	00	FC	01	01

应答:

Response	Data Out		
Result	ChineseFontType	SW1	SW2

ChineseFontType: 简体中文(默认)和繁体中文



5.5.31 LCD显示指定个数的中文或英文字体(仅MR800 支持)

MR800 支持**简体**和**繁体**两类中文字体。该指令显示指定个数的字符(包括英文或中文)。注意一个中文字体占 2Byte，英文字体占 1Byte，LCD 一行最多显示 16Byte 内容。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
Display Font	FF	00	FC	02	nByte	Configure + Row + Column + Display Data

Configure:

Bit0(NegativeDis): 正反显示 0-正显, 1-反显

Bit2~1:

00-显示画面前不清屏幕

01-显示画面前只清除显示画面的行

10-显示画面前全部清屏

Bit3(BackLight): 0-背光不亮, 1-背光亮

Bit4~7: RFU

Row(1Row = 16 dot High): 0~7

Column: 0~127

DisplayData: 显示内容(汉字相当与 2Byte)，一行最多显示 16Byte。中文字体是 16x16，英文是 8x16

应答:

Response	Data Out	
Result	SW1	SW2

5.5.32 LCD显示图片(直接发送图片数据) (仅MR800 支持)

该功能实现显示规定大小的图片，大的图片可以分多次显示。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
Display Picture	FF	00	FC	03	FF	Configure + Row + Column + PictureWidth + PictureHigh+ Display Data

Configure:

Bit0(Negative Dis): 正反显示 0-正显, 1-反显

Bit2~1:

00-显示画面前不清屏幕

01-显示画面前只清除显示画面的行

10-显示画面前全部清屏

Bit3(Back Light): 0-背光不亮, 1-背光亮

Bit4~7: RFU

Row(1row = 8 dot High): 0~7 (开始列)



Column: 0~127(开始行)

Picture Width: 1~128, 图片宽度

Picture High: 1~8, 图片高度

Display Data: 显示图片内容(字节数= 宽度 x 高度)

应答:

Response	Data Out	
Result	SW1	SW2

5.5.33 LCD擦除行(仅MR800 支持)

MR800 为了方便清屏, 用户可以分行清除字体或图片。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
EraseLCD	FF	00	FC	04	1	Row

Row(1row = 8 dot High): Bit0~Bit7 分别代表 0~7 行(0-保持不变, 1-擦除)

应答:

Response	Data Out	
Result	SW1	SW2

5.5.34 LCD设定开机画面(仅MR800 支持)

该功能实现默认开机画面设置。若没有设置, 则开机默认显示金木雨开机画面。所有显示画面都保存于读卡器内 AT45DB321 内。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
PowerOnPIC	FF	00	FC	05	08	Enable+SaveAddr+Width+High+StartLine+StartColumn+Time

Enable(1Byte): 0-禁止显示开机画面, 1-显示开机画面

SaveAddr(2Byte): 开机画面保存于 *Flash(AT45DB321)* 中, 地址低字节在前

Width(1Byte): 图片宽度(1~128)

High(1Byte): 图片高度(1~8)

StartLine(1Byte): 显示开始行(0~7)

StartColumn(1Byte): 显示开始列(0~127)

Time: 设定显示启动画面时间(单位: S)

应答:

Response	Data Out	
Result	SW1	SW2

备注:

- ❖ 若设置开机画面禁止, 则后面参数无效。
- ❖ 开机画面保存在读卡器片外Flash中, 字库占据开始的1303块(0~1302), 用户不能进行擦写, 供用户使用的块号是1303~8191, 每块大小是512字节。
- ❖ 在使能开机画面前, 需用FlashWrite APDU 写入画面数据到Flash SaveAddr地址中,



否则显示画面为不确定，若画面大于512字节，则多余字节写入紧接的第2块。

- ❖ 画面大小=Width*High。

5.5.35 LCD设定待机画面(仅MR800 支持)

该功能实现待机画面设置，若没有设置，则显示完毕用户界面后不会回到待机画面。所有显示画面都保存于读卡器内 AT45DB321 内。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
IdlePIC	FF	00	FC	06	08	Configure +SaveAddr+ Width+ High+StartLine+StartColumn+Time

Configure (1Byte):

Bit0: 0-禁止显示开机画面, 1-显示开机画面

Bit2~1:

00-显示画面前不清屏幕

01-显示画面前只清除显示画面的行

10-显示画面前全部清屏

Bit3(BackLight): 0-背光不亮, 1-背光亮

Bit4~7: RFU

SaveAddr(2Byte): 开机画面保存于 *Flash(AT45DB321)*中,地址低字节在前。

Width(1Byte): 图片宽度(1~128)

High(1Byte): 图片高度(1~8)

StartLine(1Byte): 显示开始行(0~7)

StartColumn(1Byte): 显示开始列(0~127)

Time: 设定多长时间未操作 LCD, 进入待机画面(单位: S)

应答:

Response	Data Out	
Result	SW1	SW2

备注:

- ❖ 若设置待机画面禁止，则后面参数无效。
- ❖ 待机画面保存在读卡器片外Flash中，字库占据开始的1303块(0~1302)，用户不能进行擦写，供用户使用的块号是1303~8191，每块大小是512字节。
- ❖ 在使能待机画面前，需用FlashWrite APDU 写入画面数据到Flash SaveAddr地址中，否则显示画面为不确定，若画面大于512字节，则多余字节写入紧接的第2块。
- ❖ 画面大小=Width*High。

5.5.36 LCD背光控制(仅MR800 支持)

该功能对 LCD 的背光进行控制。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
LCDBackLight	FF	00	FC	07	2	Mode+Time

Mode:



- 00-灭
- 01-常亮
- 02-规定时间亮(Time内容有效)

Time: 仅仅在 Mode =2 才有效(单位: S)

应答:

Response	Data Out	
Result	SW1	SW2

5.5.37 LCD显示Flash中存储画面(仅MR800 支持)

该功能实现保存画面显示。所有显示画面都保存于读卡器内 AT45DB321 内。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
IdlePIC	FF	00	FC	08	09	Configure +DisAddr +Width+ High + StartLine+StartColumn

Configure (1Byte):

Bit0: RFU

Bit2~1:

- 00-显示画面前不清屏幕
- 01-显示画面前只清除显示画面的行
- 10-显示画面前全部清屏

Bit3(BackLight): 0-背光不亮, 1-背光亮

Bit4~7: RFU

DisAddr(2Byte): 显示画面保存于 *Flash(AT45DB321)*中, 地址低字节在前

Width(1Byte): 图片宽度(1~128)

High(1Byte): 图片高度(1~8)

StartLine(1Byte): 显示开始行(0~7)

StartColumn(1Byte): 显示开始列(0~127)

应答:

Response	Data Out	
Result	SW1	SW2

备注:

- ❖ 显示画面保存在读卡器片外Flash中, 字库占据开始的1303块(0~1302), 用户不能进行擦写, 供用户使用的块号是1303~8191, 每块大小是512字节。
- ❖ 在显示画面前, 需用FlashWrite APDU 写入画面数据到Flash SaveAddr地址中, 否则显示画面为不确定, 若画面大于512字节, 则多余字节写入紧接的第2块。
- ❖ 画面大小=Width*High。

5.5.38 读片外Flash

MR800 片外 Flash 采用的是 AT45DB321, 其中 0~1302 为字体保存块, 所以不要读写这些块。

发送 APDU 格式:



Command	Class	INS	P1	P2	Lc	Data
ReadFlash	FF	00	FD	00	06	BlockAddr+ ByteAddr+ Len

BlockAddr: 块地址(2Byte, 高字节在前)

ByteAddr: 块内字节起始地址(2Byte, 高字节在前)

Len: 所读字节长度(2Byte, 高字节在前)

应答:

Response	Data Out		
Result	Flash Data	SW1	SW2

5.5.39 写片外Flash

MR800 片外 Flash 采用的是 AT45DB321, 其中 0~1302 为字体保存块, 所以不要写该块。

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
WriteFlash	FF	00	FD	01	04+n	BlockAddr+ ByteAddr+Data(n)

BlockAddr: 块地址(2Byte, 高字节在前)

ByteAddr: 块内字节起始地址(2Byte, 高字节在前)

Data: 所写数据

应答:

Response	Data Out	
Result	SW1	SW2

5.5.40 获取产品序列号

发送 APDU 格式:

Command	Class	INS	P1	P2	Le
GetSNR	FF	00	FF	00	0A

应答:

Response	Data Out		
Result	Product SNR	SW1	SW2

5.5.41 获取硬件版本和版本号

发送 APDU 格式:

Command	Class	INS	P1	P2	Le
GetVer	FF	00	FF	01	04

应答:

Response	Data Out		
Result	Hardware ver(2Byte)+Software ver(2Byte)	SW1	SW2



5.5.42 LED 灯控制

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
LEDCtr	FF	00	FF	02	05	LED state + state Mask +T1 duration + T2 Duration + Number

LED Status:

BIT0 = 红灯最终状态(1-ON, 0-OFF)

BIT1 = 绿灯最终状态(1-ON, 0-OFF)

BIT2 = 蓝灯最终状态(1-ON, 0-OFF)

BIT3 = 黄灯最终状态(1-ON, 0-OFF)

BIT4 = 红灯闪动初始状态(1-ON, 0-OFF)

BIT5 = 绿灯闪动初始状态(1-ON, 0-OFF)

BIT6 = 蓝灯闪动初始状态(1-ON, 0-OFF)

BIT7 = 黄灯闪动初始状态(1-ON, 0-OFF)

LED Status Mask:

BIT0 = 红灯状态更新掩码(1-更新, 0-不改变)

BIT1 = 绿灯状态更新掩码 (1-更新, 0-不改变)

BIT2 = 蓝灯状态更新掩码 (1-更新, 0-不改变)

BIT3 = 黄灯状态更新掩码 (1-更新, 0-不改变)

BIT4~7 RFU

T1/T2: T1,T2 时间(单位: 100ms) , T=T1+T2

Number: 次数

应答:

Response	Data Out	
Result	SW1	SW2

5.5.43 蜂鸣器控制

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
BuzzerCtr	FF	00	FF	03	05	Beep state + state Mask +T1 duration + T2 Duration + Number

BEEP Status:

BIT0 = BEEP最终状态(1-ON, 0-OFF)

BIT4 = BEEP闪动初始状态(1-ON, 0-OFF)

Status Mask:

BIT0 = Buzzer状态更新掩码(1-更新, 0-不改变)

BIT4~7 RFU

T1/T2: T1,T2 时间(单位: 100ms) , T=T1+T2



Number: 次数

应答:

Response	Data Out	
Result	SW1	SW2

5.5.44 天线状态设置

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
AntennaCtr	FF	00	FF	04	01	Antena status

Antena status:

00-关闭

01-打开

应答:

Response	Data Out	
Result	SW1	SW2

5.5.45 卡片加密方法设置

发送 APDU 格式:

Command	Class	INS	P1	P2	Lc	Data
EncrMode	FF	00	FF	05	01	Encrypt Standard

Encrypt Standard:

0x00-Philips

0x01-上海标准

应答:

Response	Data Out	
Result	SW1	SW2

5.5.46 恢复出厂默认值(系统重新启动)

发送 APDU 格式:

Command	Class	INS	P1	P2	Le
FactoryDefault	FF	00	FF	06	00

应答:

Response	Data Out	
Result	SW1	SW2

5.5.47 系统重新启动

发送 APDU 格式:



Command	Class	INS	P1	P2	Le
Reboot	FF	00	FF	07	00

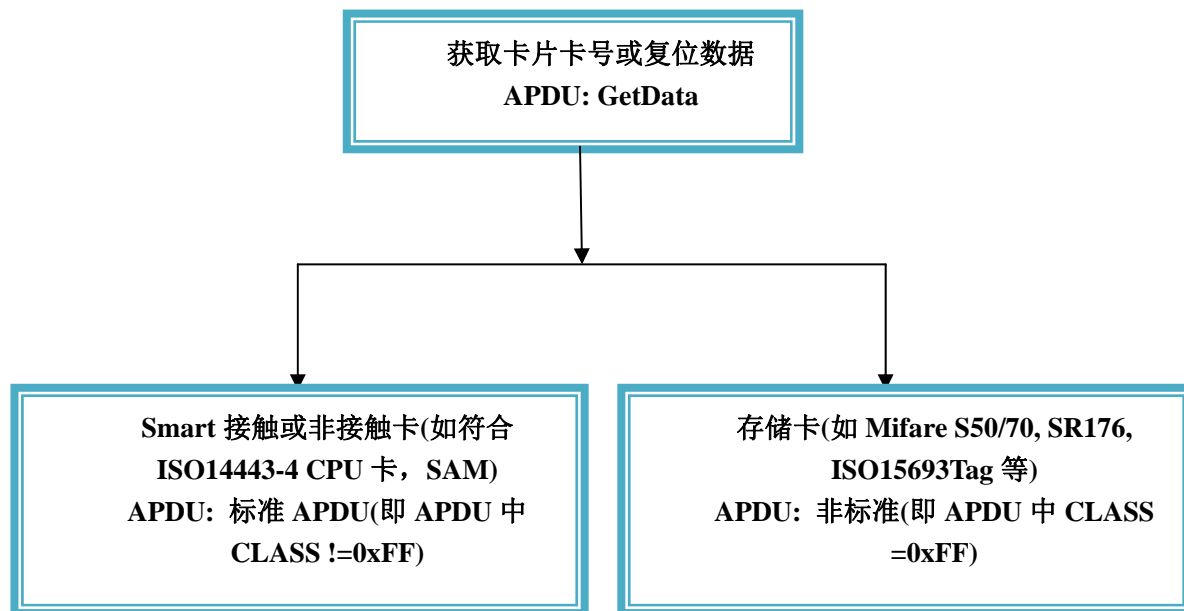
应答:

Response	Data Out	
Result	SW1	SW2



6 卡片操作流程

各种卡片操作基本流程如下：

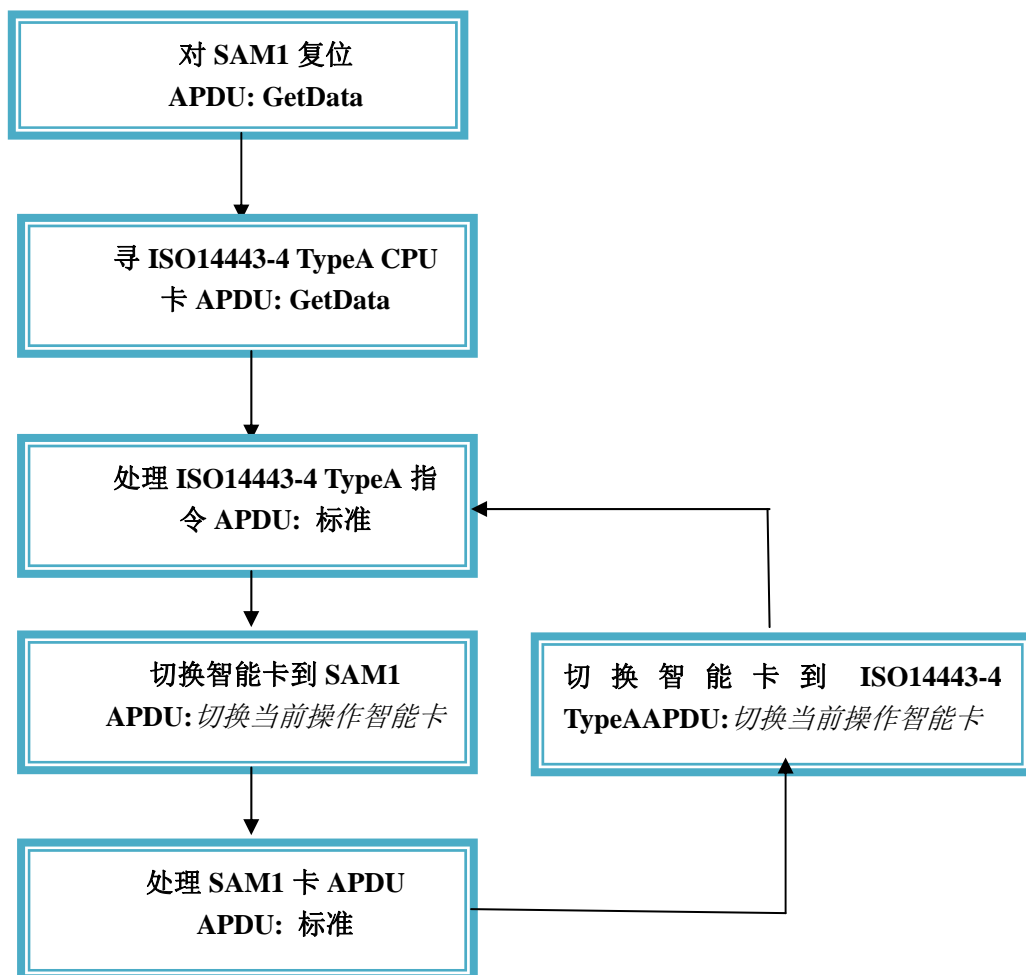


在操作任何卡片前需要执行 GetData APDU 获取卡片基本信息(包括卡序列号, 复位信息等), GetData 包含了读卡类型的切换, 所以在对任何卡片执行操作前需执行该 APDU, 获取卡片信息的同时, 读卡器读卡类型也切换到这个类型上。



6.1 Smart接触和非接触卡

Smart 接触或非接触卡可以直接发送标准的 APDU 至卡片，假如需要同时操作非接触和接触的 Smart 卡(如：ISO14443-4 TypeA CPU 卡和 SAM1 卡) 卡片操作如下：

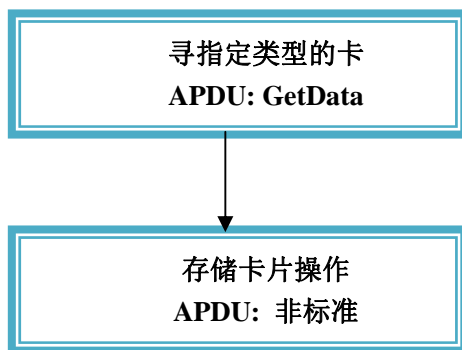


因为智能非接触和接触卡都采用的是标准 APDU，在对 SAM 卡复位后，若需要再对 SAM 进行操作，需要通过 *切换智能卡类别* 指令去切换当前操作智能卡，以保证数据是发送到指定类型的智能卡。若是智能卡和存储卡不需要切换，则执行完毕 GetData 后，当前操作类型就是 GetData 操作的卡片类型。

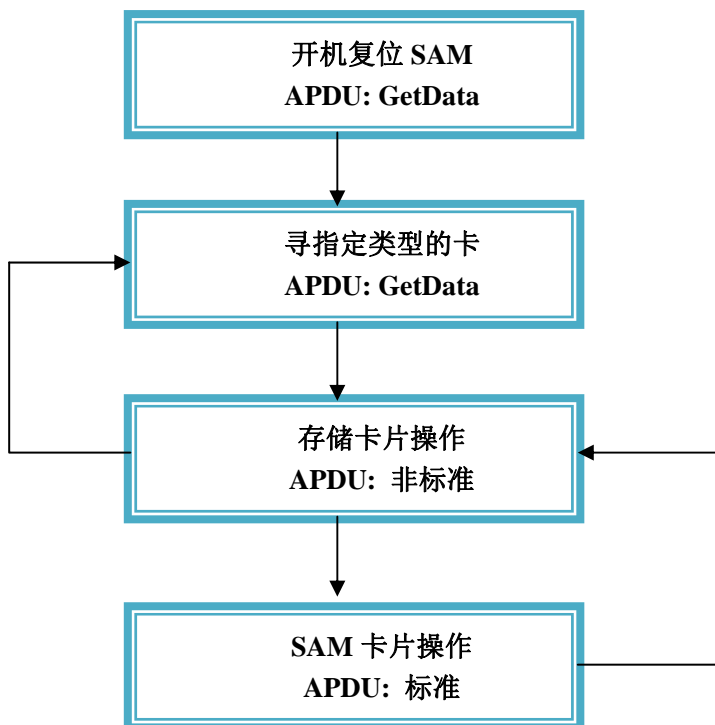


6.2 存储卡(非智能卡)

存储卡片的操作都是通过非标准的 APDU 来操作，主要操作如下：



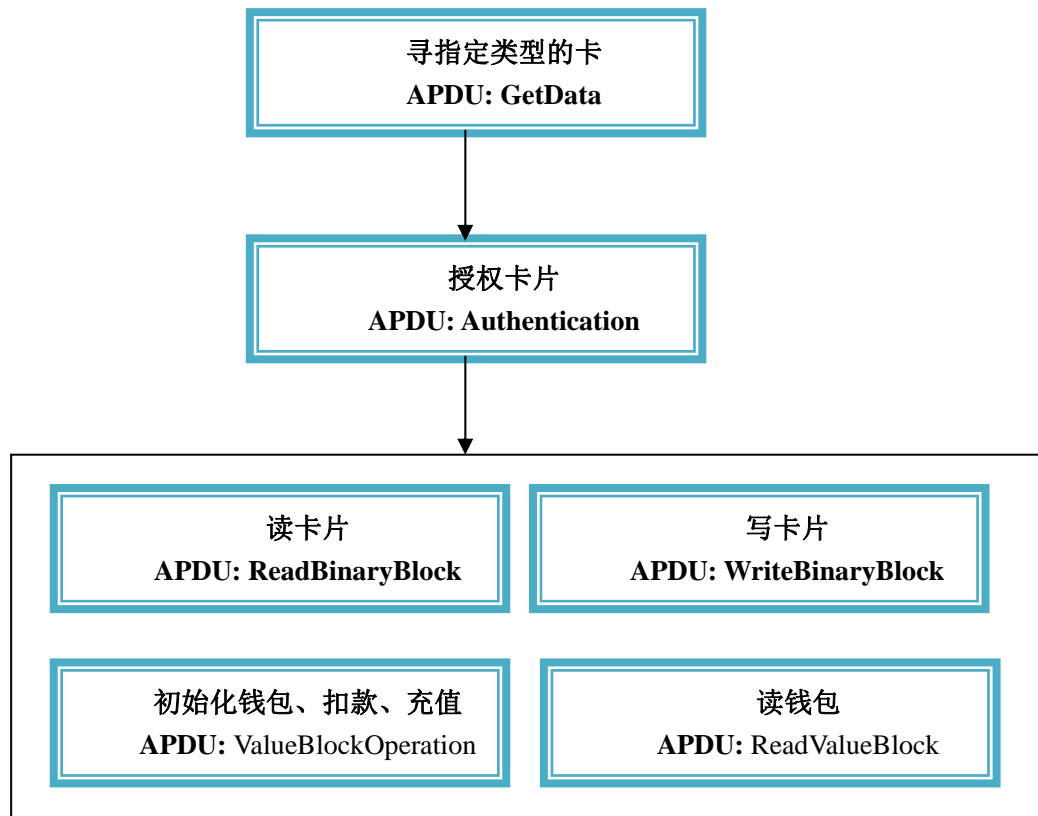
存储卡操作需要带 SAM 操作流程如下：



存储卡和单一 SAM 操作不需切换，若需要对多个 SAM 卡操作，则在操作这个 SAM 卡之前，需 *切换智能卡类别* 去切换指定 SAM 卡。



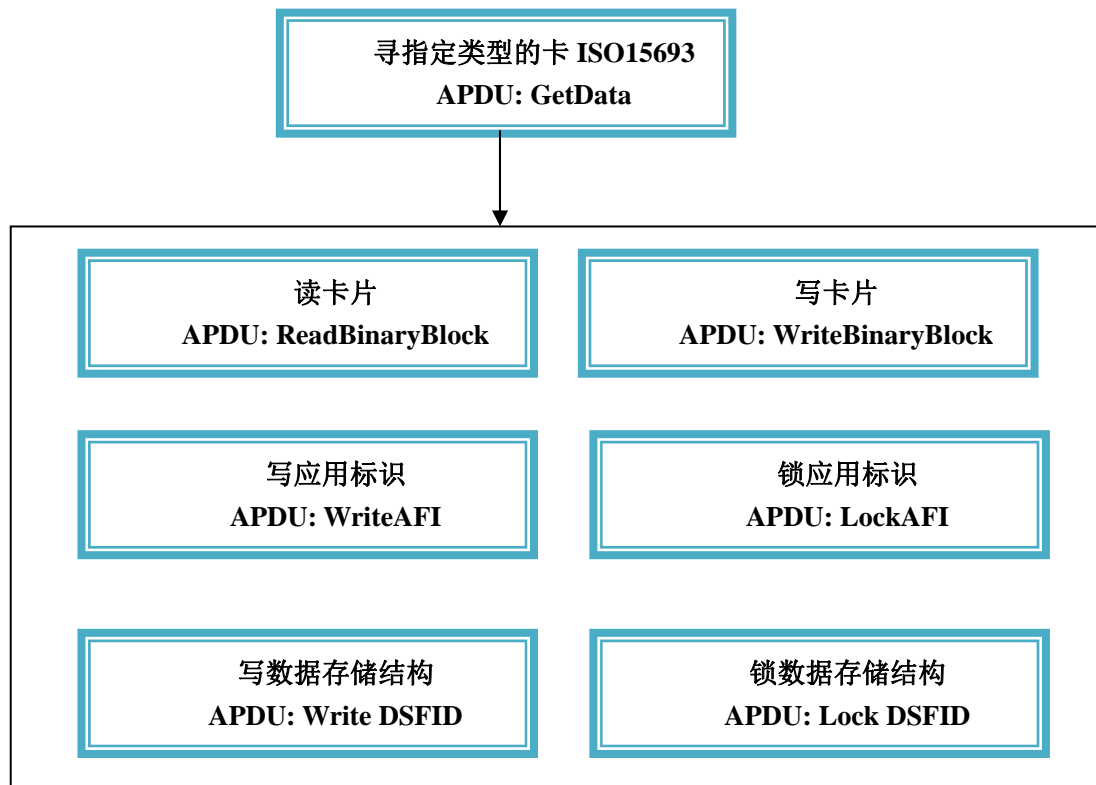
如常见的 Mifare S50/70 卡片操作：



以上操作不带 SAM，若带 SAM 卡操作，见上面流程。



如 ISO15693Tag 操作:



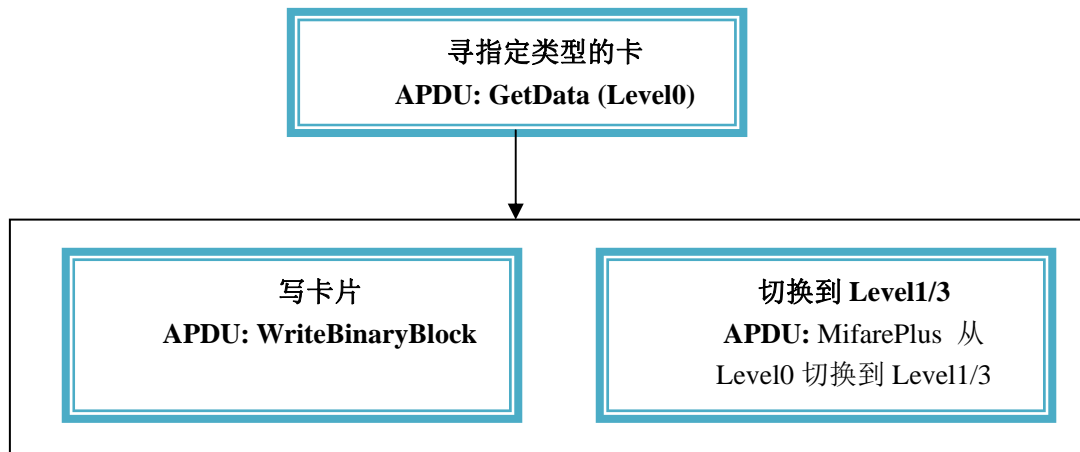
ISO15693 Tag 操作通过 ReadBinaryBlock 和 WriteBinaryBlock 仅仅针对最后寻到的一张 Tag, 若需要对指定 UID 的一个 Tag 操作, 可以参考非标准 APDU(自定义部分)。



如 MifarePlus 卡片操作如下:

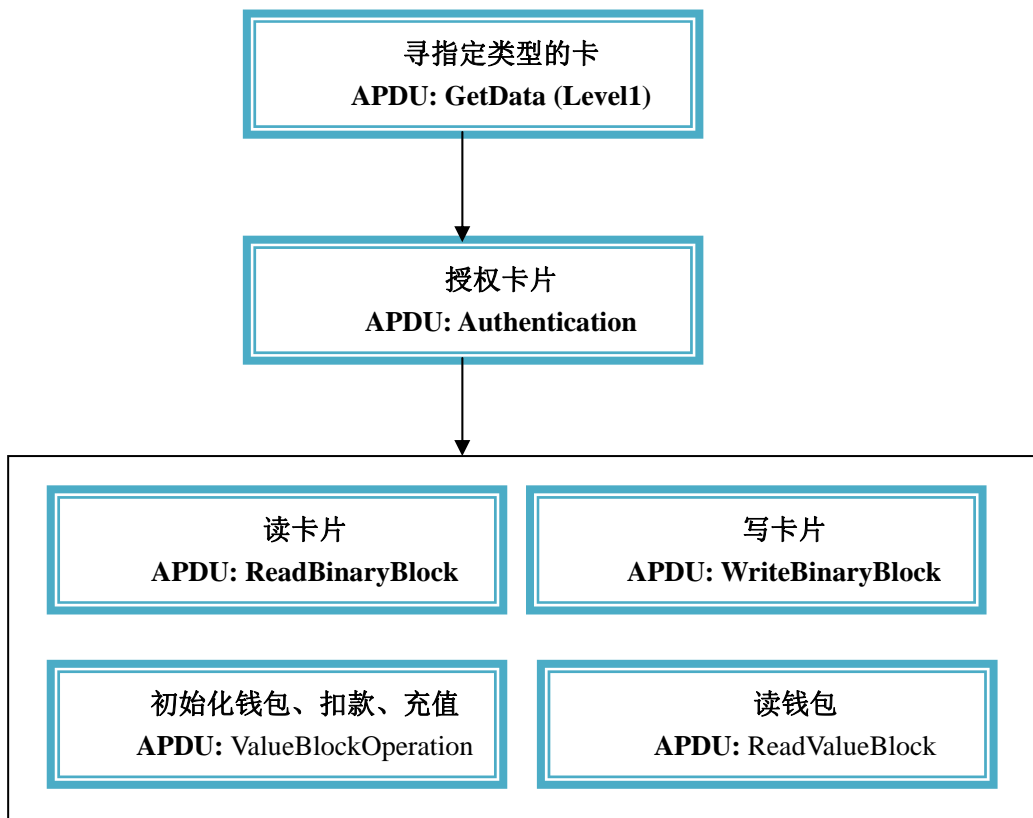
MifarePlus 卡片结构见附录，在 GetData 中针对 MifarePlus 有不同的 GetData 指令，是因为 MifarePlus 分为 4 个安全级别(Level0~Level3)，不同的安全级别对寻卡操作不同，有的只需要寻卡片序号，有的需要寻卡后需要对卡片进行复位操作。其中 MifarePlus Level1 兼容原来的 Mifareone，所有操作同 Mifareone。

Level0 操作如下:

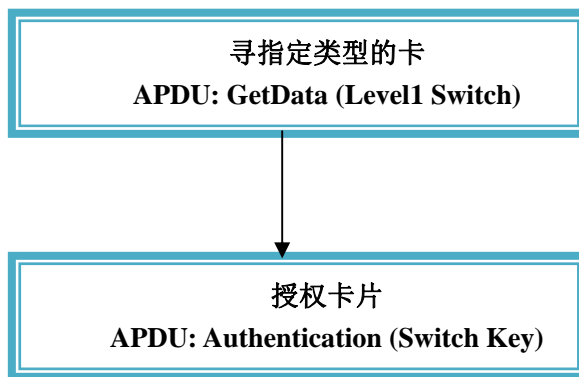




Level1 操作:



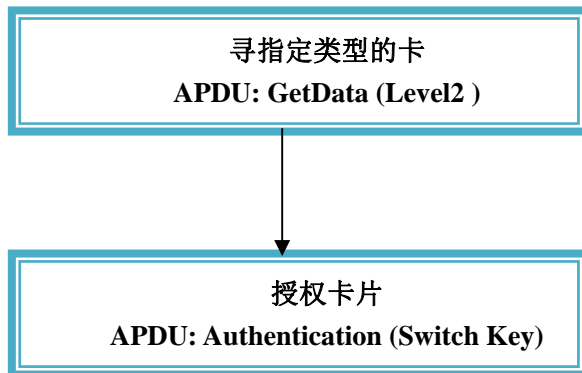
Level1 Switch 操作:



注意从 Level1 切换到其它 Level, GetData 寻卡类型有区别, 假如想从 Level1 切换到 Level2, 那么 Switch Key 就用 Switch Key2。

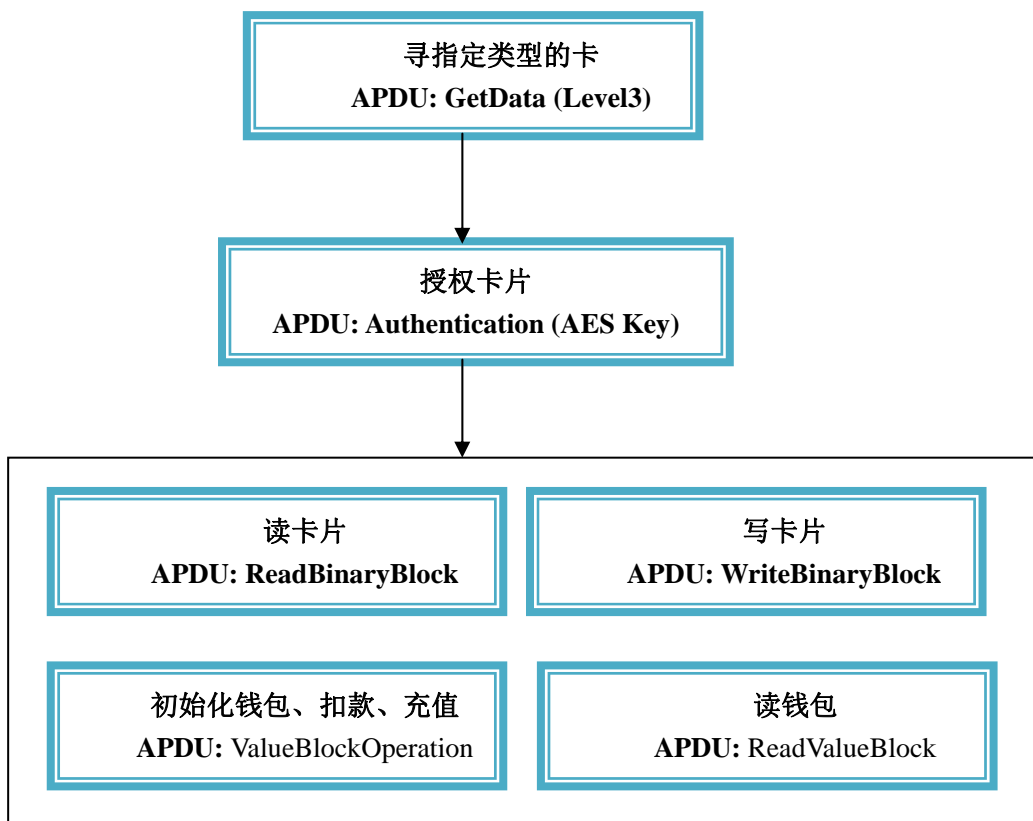


Level2 操作:



假如想从 Level2 切换到 Level3, 那么 Switch Key 就用 Switch Key3。

Level3 操作:



其他类别卡片操作基本类似, 基本都是用到 GetData、ReadBinaryBlock、WriteBinaryBlock 指令操作, 若需要对寻卡参数进行设定, 请参考非标准 APDU(自定义部分)。

对于 LCD 操作、时钟操作、当前智能卡操作切换、SAM 复位 baudrate、LED、蜂鸣器等操作请参考非标准 APDU(自定义部分)。



附录A

MifarePlus Level3 的数据及密钥存储结构和 Mifareone 有所区别，结构如下：

块相对地址		块地址	对应密钥块地址
Sector0			
Block0	数据块	0x0000	A 密 钥 : 0x4000 B 密 钥 : 0x4001
Block1	数据块	0x0001	
Block2	数据块	0x0002	
Block3	数据块	0x0003	
Sector1			
Block0	数据块	0x0004	A 密 钥 : 0x4002 B 密 钥 : 0x4003
Block1	数据块	0x0005	
Block2	数据块	0x0006	
Block3	数据块	0x0007	
....			
Sector31			
Block0	数据块	0x007C	A 密 钥 : 0x403E B 密 钥 : 0x403F
Block1	数据块	0x007D	
Block2	数据块	0x007E	
Block3	数据块	0x007F	
配置块			
	MFP Configuration Block	0xB000	
	Installation Identifier	0xB001	
	ATS Information	0xB002	
	Field Configuration Block	0xB003	
Key 块			
	AES Sector Keys	0x4000~0x403F	
	AES Sector Keys	0x4040~0x404F	
	Originality Key	0x8000	
	Card Master Key	0x9000	
	Card Configuration Key	0x9001	
	Level2 switch Key	0x9002	
	Level3 switch Key	0x9003	
	SL1 Card Authentication Key	0x9004	
	Select VC Key	0xA000	
	Proximity Check Key	0xA001	
	VC Polling ENC Key	0xA080	
	VC Polling MAC Key	0xA081	

注意：

- 1、蓝色和黄色部分是关联部分。即数据区和密钥区对应部分(仅仅是在 Level 2/3 才对应，因只有级别 2/3 才使用到 AES 密钥认证)。



- 2、在安全级别 Level 1, 是和 Mifare classic 兼容的, 每个扇区最后一块为密钥和配置块。
- 3、AES 密钥分为 A/B 密钥是人为划分, 是为了同 Mifare classic 概念相同。在 PLUS 内部一个扇区是对应地址连续的 AES 密钥块。
- 4、主要掌握如下 key:

AES Sector Keys:

在 Level2/3 中对数据的授权采用 AES Key 授权。该密钥可以在 Level0 写入, 或者通过 AES Sector Keys 对卡片授权而修改 AES Key。

CardMasterKey:

通过对该 Key 的授权, 可以改变 *Card Configuration Key* 和 *Level2/3 switch Key*

Card Configuration Key:

通过对该 key 的授权, 可以改变 MFP Configuration Block 配置块内容。

Level2 switch Key:

通过对该 key 的授权, 可以从 Level1 切换 Level2。

Level3 switch Key:

通过对该 key 的授权, 可以从 Level2 切换 Level3, 或从 Level1 切换到 Level3。

- 5、在 Level0, 除了出厂写入的用户不能修改的密钥外, 都可以以明文方式写入, 一般在 Level0 做初始化操作。注意, 必须在该安全级别写入 0x9000~0x9003 块。
- 6、Level3 级别支持明文、AES 加密、加密且带 MAC 方式读写方式。本读卡器采用的是最保密的方式读写 MifarePlus 块: *加密且带 MAC 方式*。