JMY600 系列读写卡模块

MIFARE Plus 卡操作指南

（Revision 1.02）

北京金木雨电子有限公司

2013/12/27

在使用本产品前请详细阅读本说明书，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 概述

本文详细介绍了使用JMY600 系列读卡模块操作MIFARE Plus卡的操作方法和顺序以及基本卡片功能设计，您可以通过阅读本手册很快速地掌握MIFARE Plus卡的使用。本手册的使用对象为使用JMY600 系列RFID模块的程序员，我们也有通讯协议的例子代码，可以在随货的产品光盘上找到，也可以在金木雨的网站上找到。如果在编写程序中依然有任何的问题，请随时联系我们的技术支持。或发送电子邮件到：jinmuyu@vip.sina.com 我们会给您满意的答复。

2 主要性能指标

- 卡片的操作方法类似 MIFARE 1
- 拥有比 MIFARE 1 更高的安全性能，每个扇区使用 AES（2*128bit）加密保护数据的安全
- 容量为 2K/4K 字节
- 每张卡有唯一序列号，为 7 字节
- 具有防冲突机制，支持多卡操作
- 存储器使用文件系统，设计使用非常灵活，
- 数据保存期为 10 年，可改写 10 万次，读无限次
- 工作温度：-20℃~50℃(湿度为 90%)
- 工作频率：13.56MHZ
- 通信速率：106 KBPS
- 读写距离：大于 10 cm

3 卡片功能

MIFARE Plus 卡是用来代替 MIFARE 1 卡的，是拥有较高数据安全性的非接触 IC 卡，内部数据的存储格式与 MIFARE 1 相同，操作方式类似，但使用 AES 加密保护数据安全，拥有 CC EAL 4+的安全性能，安全级别比较高，价格低廉，非常适合用电子钱包使用。也可以当作数据存储使用，但空间只有 2K/4K 字节。

4 存储结构

MIFARE Plus 卡片内部的存储器以 16 字节为单位的格式存储，称为一个块，每个块的存储内容如下表。

Command	HEX Address	Description
块和数据区		
MIFARE 数据块 MIFARE 块尾	00 00h 到 00 7Fh	扇区 0 到扇区 31
MIFARE 数据块 MIFARE 块尾	00 80h 到 00 FFh	扇区 32 到扇区 39
MIFARE Plus 配置块	B0 00h	卡片功能配置
安装标识	B0 01h	安装标识用于虚拟卡片概念，详细信息请联系 NXP
复位信息 ATS	B0 02h	卡片的复位信息
行业配置块	B0 03h	用于行业配置信息
密钥区		
扇区 AES 密钥	40 00h 到 40 3Fh	扇区 0 到 31 的 AES 密钥 KEY A：扇区号×2 KEY B：扇区号×2+1
扇区 AES 密钥	40 40h 到 40 4Fh	扇区 32 到 39 的 AES 密钥 KEY A：扇区号×2 KEY B：扇区号×2+1
验真密钥	80 00h	用于验证 NXP 真品。这个密钥已经被 NXP 初始化并且不能修改，NXP 并不公开这个密钥，认证这个密钥只能使用 NXP 提供的一个 SAM 卡
卡片主控密钥	90 00h	用于修改层级切换密钥和卡片配置密钥
卡片配置密钥	90 01h	用于修改区域配置块
层级 2 切换密钥	90 02h	从层级 1 切换到层级 2 的密钥
层级 3 切换密钥	90 03h	从层级 2 切换到层级 3 的密钥
层级 1 卡片认证密钥	90 04h	在层级 1 时做额外的 AES 认证的密钥
虚拟卡片选择密钥	A0 00h	用于选择虚拟卡片
接近检查密钥	A0 01h	用于验证卡片接近检查
虚拟卡片轮训加密密码	A0 80h	虚拟卡片轮训加密密码
虚拟卡片轮训 MAC 密码	A0 81h	虚拟卡片轮训 MAC 密码

MIFARE Plus 卡片有 4 个层级，层级 0 是工厂生产出来的默认模式，用于配置；层级 1 完全兼容于 MIFARE 1；层级 2 在 JMY600 中不完全支持，需要用户使用 ISO14443-4 通道进行自主研发；层级 3 是 MIFARE Plus 的最高安全级别，JMY600 完全支持，全部以加密和双向带 MAC 方式传送数据，并提供非常简单的接口以最大限度支持对 MIFARE Plus 的开发应用。

5 卡片操作

5.1 主动读卡模式

主动读卡模式只读 UID，在这里用 MIFARE Plus 来做 UID 识别性价比不高，因此不做介绍。

5.2 被动读卡模式

在操作 MIFARE Plus 的卡片的过程中不能使用自动寻卡功能，必须关闭。多卡功能，可以根据用户的要求自己选择。我们在此做的实验是基于层级 3 的，切换到层级 3 之前需要做一些准备工作，切换到层级 3 之后，就可以体验 MIFARE Plus 带来的高安全性了。

将一张新的 MIFARE Plus 卡写入默认值并切换到层级 3，可以使用我公司产品 TransWin 配合 MR780 桌面读写器通过非常简单的操作将卡片做好，这部分请查看 MR780 桌面读写器。

取一张出厂默认格式的新卡放到天线上，使用 TransPost 软件做这些实验，请按照顺序发送如下命令：

- 按照 EMV 或 PBOC 的规则寻卡并复位：

  TransPort 中输入：32

  实际端口发出指令：00 04 00 32 36

  实际端口收到：00 1C 01 32 41 07 04 0C 4D 0A D7 2C 80 42 00 20 0C 75 77 80 02 C1 05 2F 2F 01 BC D6 1C

- Write Perso：

  这一步是写入切换到层级 3 之后需要使用的各个密钥和配置

  ◆ 写入密钥：

  密钥的地址范围是 0x4000～0x403F，0x4040～0x404F（MIFARE Plus 4K），0x9000～0x9003，这些值的含义请看上面的表格。请按步骤分别写入这些值，或使用我公司产品 TransWin 配合 MR780 桌面读写器一次性写入这些值。

  TransPort 中输入：33 40 00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

  实际端口发出指令：00 16 00 33 40 00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 65

  实际端口收到：00 05 01 33 90 A7

  在这条指令里，发送的地址是 0x4000，请循环执行这条指令，并将地址递增为 0x4001，0x4002 直到 0x404F 逐个写入，然后写入 0x9000，0x9001，0x9002 和 0x9003，密钥值都可以写入 16 字节 0xFF 以方便实验。

  ◆ 写入 MIFARE Plus 配置块：

  TransPort 中输入：33 B0 00 0F FF FF FF FF FF FF FF FF FF FF FF FF FF FF

  实际端口发出指令：00 16 00 33 B0 00 00 0F FF FF FF FF FF FF FF FF FF FF FF FF FF FF 9A

  实际端口收到：00 05 01 33 90 A7

- Commit Perso：

  当以上信息写入无误后，执行本指令，将卡片的层级切换到 1 或者 3，切换到的层级与卡片有关，您需要询问供应商这个卡片是层级 1 或层级 3 的，层级 1 意味着执行本

指令后将切换到层级 1，层级 3 意味着执行本指令后将切换到层级 3。

TransPort 中输入：34

实际端口发出指令：00 04 00 34 30

实际端口收到：00 05 01 34 90 A0

此时，卡片已经切换了层级，但切换到的层级依靠卡片出厂设置。

- 切换到层级 3：

如果您的卡片是层级1的，那么您还需要执行一步层级切换，才能切换到层级3。因为层级1是MIFARE 1兼容的，在回复的SAK中并不指示支持ISO14443-4，因此，需要执行“ISO14443 TYPE A寻卡”和“ISO14443-4 TYPE A卡复位(RATS)”，然后继续执行：

- ISO14443 TYPE A 寻卡：

TransPort 中输入：20 00

实际端口发出指令：00 05 00 20 00 25

实际端口收到：00 0E 01 20 04 34 5F 0A D7 2C 80 42 00 18 6B

- ISO14443-4 TYPE A 卡复位(RATS)：

TransPort 中输入：30

实际端口发出指令：00 04 00 30 34

实际端口收到：00 10 01 30 0C 75 77 80 02 C1 05 2F 2F 01 BC D6 02

- 切换到层级 3：

TransPort 中输入：35 03 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

实际端口发出指令：00 15 00 35 03 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 23

实际端口收到：00 05 01 35 90 A1

此时，卡片已经切换到了层级 3，将卡片移开再重新放到天线上。

- 按照 EMV 或 PBOC 的规则寻卡并复位：

上面已有介绍，在此不再重复。

- 授权数据块：

TransPort 中输入：36 00 00 04 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

实际端口发出指令：00 17 00 36 00 00 04 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 25

实际端口收到：00 05 01 36 90 A2

- 写数据块：

TransPort 中输入：38 00 05 01 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

实际端口发出指令：00 17 00 38 00 04 01 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 2B

实际端口收到：00 05 01 38 90 AC

- 读数据块：

TransPort 中输入：37 00 05 01

实际端口发出指令：00 07 00 37 00 05 01 34

实际端口收到：00 15 01 37 90 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F B3

- 创建钱包：

TransPort 中输入：39 00 04 00 01 00 00

实际端口发出指令：00 0A 00 39 00 04 00 01 00 00 36

实际端口收到：00 05 01 39 90 AD

- 读钱包余额：

TransPort 中输入：3A 00 04
实际端口发出指令：00 06 00 3A 00 04 38
实际端口收到：00 09 01 3A 90 00 01 00 00 A3

- 钱包充值：

  TransPort 中输入：3B 00 04 00 01 00 00
  实际端口发出指令：00 0A 00 3B 00 04 00 01 00 00 34
  实际端口收到：00 05 01 3B 90 AF

- 再读钱包余额：

  TransPort 中输入：3A 00 04
  实际端口发出指令：00 06 00 3A 00 04 38
  实际端口收到：00 09 01 3A 90 00 02 00 00 A0

- 钱包扣款：

  TransPort 中输入：3C 00 04 00 01 00 00
  实际端口发出指令：00 0A 00 3C 00 04 00 01 00 00 33
  实际端口收到：00 05 01 3C 90 A8

- 再读钱包余额：

  TransPort 中输入：3A 00 04
  实际端口发出指令：00 06 00 3A 00 04 38
  实际端口收到：00 09 01 3A 90 00 01 00 00 A3

- 备份钱包：

  TransPort 中输入：3D 00 04 00 05
  实际端口发出指令：00 08 00 3D 00 04 00 05 34
  实际端口收到：00 05 01 3D 90 A9

- 读备份的钱包余额：

  TransPort 中输入：3A 00 05
  实际端口发出指令：00 06 00 3A 00 05 39
  实际端口收到：00 09 01 3A 90 00 01 00 00 A3