JMY600 系列读写卡模块

DESfire 卡操作指南

（Revision 1.12）

北京金木雨电子有限公司

2020/7/8

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 DESfire 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 DESfire 卡的使用和规划。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

- 拥有 CC EAL 4+的安全性能，使用 DES/3DES 加密保护数据的安全
- 容量为 2K/4K/8K 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，
- 每张卡片最多可以有 28 个应用
- 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

DESfire 卡是拥有较高数据安全性的非接触 IC 卡，属于内部带有 CPU 的智能卡，有比较高的安全性，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/8K 字节。

4 存储结构

DESfire 卡的内存管理采用文件系统，我们规划了一个简单的文件系统，用户可以用一张新卡跟随我们去做实验以达到快速掌握 DESfire 卡的目的。

新卡是空白的，默认的主控密钥为 16 字节 0x00，我们规划的文件系统如下：

AID = 0x4A4D59

密钥设置：0F（0x0F），详细内容参阅 DESfire 卡片的 datasheet creat application 部分

密钥数量：0C（0x0C），密钥数量 12 个

在这个 AID 下建立如下文件：

数值（钱包）文件（Value File）ID = 01

通讯方式：03（0x03），密文传送 或是 00 (0x00)，明文传输

文件权限：11 11（0x1111），密钥 1 拥有读写的全部权限

最小值：00 00 00 00（0x00000000），有符号整数

最大值：FF FF FF 7F（0x7FFFFFFF），有符号整数

初始值：78 56 34 12（0x12345678）

有限充值：00（0x00），允许有限充值

标准数据文件（STD DATA FILE）ID = 02

通讯方式：00（0x00），明文传送

文件权限：11 11（0x1111），密钥 1 拥有读写的全部权限

文件大小：00 01 00（0x000100），256 字节

备份数据文件（BACKUP DATA FILE）ID = 03

通讯方式：00（0x00），明文传送

文件权限：11 11（0x1111），密钥 1 拥有读写的全部权限

文件大小：00 01 00（0x000100），256 字节

线性记录文件（LINER RECORD FILE）ID = 04

通讯方式：00（0x00），明文传送

文件权限：11 11（0x1111），密钥 1 拥有读写的全部权限

单个记录长度：10 00 00（0x000010），16 字节

总记录数：10 00 00（0x000010），16 条记录

循环记录文件（LINER RECORD FILE）ID = 04

通讯方式：00（0x00），明文传送

文件权限：11 11（0x1111），密钥 1 拥有读写的全部权限

单个记录长度：10 00 00（0x000010），16 字节

总记录数：10 00 00（0x000010），16 条记录

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 DESfire 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 DESfire 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。

5.2.1 建立文件结构（密文钱包）

取一张新卡，我们上面设计的文件系统进行，请按照顺序做如下操作，可以发送如下命令：

- **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 0E 00 20 BD 32 30 63 35 D8 9F 04 00 08 8C

- **认证卡片主控密钥：**

  TransPort 中输入：90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  实际端口发出指令：00 15 00 90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 85
  实际端口收到：00 15 01 90 00 CC 6C E1 74 46 42 09 8D 1B 78 17 03 49 4C 67 A1 85
  在这里，返回的信息中 CC 6C E1 74 46 42 09 8D 1B 78 17 03 49 4C 67 A1 是本次卡片操作的过程密钥，需要保存。每次认证后得到的过程密钥都是不同的，需要注意。

- **擦除卡片所有应用：**

  TransPort 中输入：99
  实际端口发出指令：00 04 00 99 9D
  实际端口收到：00 05 01 99 00 9D
  此时卡片已经被全部擦除。

- **建应用：**

  TransPort 中输入：95 4A 4D 59 0F 0C
  实际端口发出指令：00 09 00 95 4A 4D 59 0F 0C C1
  实际端口收到：00 05 01 95 00 91

- **选择应用：**

  TransPort 中输入：98 4A 4D 59
  实际端口发出指令：00 07 00 98 4A 4D 59 C1
  实际端口收到：00 05 01 98 00 9C

- **认证应用密钥：**

  TransPort 中输入：90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  实际端口发出指令：00 15 00 90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 85
  实际端口收到：00 15 01 90 00 E1 74 46 42 09 8D 1B CC 6C 78 17 03 49 4C 67 A1 85

- **建数值文件：**

  TransPort 中输入：A0 01 03 11 11 00 00 00 00 FF FF FF 7F 78 56 34 12 00

实际端口发出指令：00 15 00 A0 01 03 11 11 00 00 00 00 FF FF FF 7F 78 56 34 12 00 3F

实际端口收到：00 05 01 A0 00 A4

- **建标准数据文件：**

  TransPort 中输入：9E 02 00 11 11 00 01 00

  实际端口发出指令：00 0B 00 9E 02 00 11 11 00 01 00 96

  实际端口收到：00 05 01 9E 00 9A

- **建备份数据文件：**

  TransPort 中输入：9F 03 00 11 11 00 01 00

  实际端口发出指令：00 0B 00 9F 03 00 11 11 00 01 00 96

  实际端口收到：00 05 01 9F 00 9B

- **建线性记录文件：**

  TransPort 中输入：A1 04 00 11 11 10 00 00 10 00 00

  实际端口发出指令：00 0E 00 A1 04 00 11 11 10 00 00 10 00 00 AB

  实际端口收到：00 05 01 A1 00 A5

- **建循环记录文件：**

  TransPort 中输入：A2 05 00 11 11 10 00 00 10 00 00

  实际端口发出指令：00 0E 00 A2 05 00 11 11 10 00 00 10 00 00 A9

  实际端口收到：00 05 01 A2 00 A6

5.2.2 应用实验（密文钱包）

- **按照 EMV 或 PBOC 的规则寻卡并复位：**

  见上面操作

- **选择应用：**

  TransPort 中输入：98 4A 4D 59

  实际端口发出指令：00 07 00 98 4A 4D 59 C1

  实际端口收到：00 05 01 98 00 9C

- **认证应用密钥 1（密钥 1 是我们的建立的文件的管控密钥）：**

  TransPort 中输入：90 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

  实际端口发出指令：00 15 00 90 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 84

  实际端口收到：00 15 01 90 00 B0 11 F1 7C CA 4F 17 95 1B 5C 40 53 FE 51 9B 6C A3

- **读取钱包余额：**

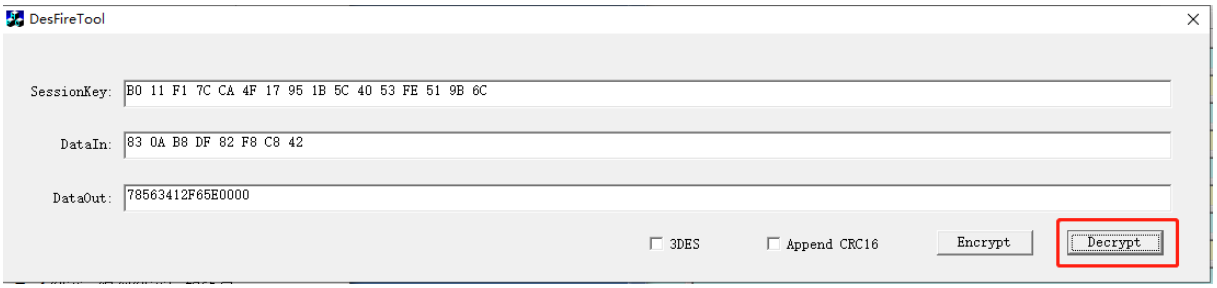
  TransPort 中输入：A6 01

  实际端口发出指令：00 05 00 A6 01 A2

  实际端口收到：00 0D 01 A6 00 83 0A B8 DF 82 F8 C8 42 B4

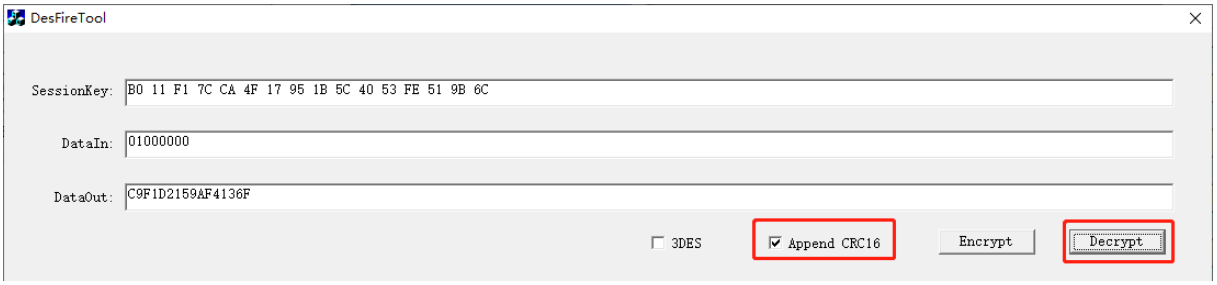
  钱包操作过程，如果钱包文件是需要认证的，那么通讯过程都使用加密数据，在此，

使用过程密钥（B0 11 F1 7C CA 4F 17 95 1B 5C 40 53 FE 51 9B 6C）对收到的加密数据（83 0A B8 DF 82 F8 C8 42）使用我们提供的工具软件 DESfire Tool 解密，得到解密后的钱包余额（78563412F65E0000），前 4 字节是钱包余额，78 56 34 12（0x12345678），F6 5E 为 CRC 校验，如图：



- **钱包加值：**

  我们要对钱包加值 0x00000001，那么，我们在 DESfire Tool 的 DataIn 框中输入：01000000（低字节在前），不选 3DES，勾选 CRC16，点击“解密”按键，得到 C9F1D2159AF4136F，如图：



TransPort 中输入：A7 01 C9 F1 D2 15 9A F4 13 6F
实际端口发出指令：00 0D 00 A7 01 C9 F1 D2 15 9A F4 13 6F 46
实际端口收到：00 05 01 A7 00 A3

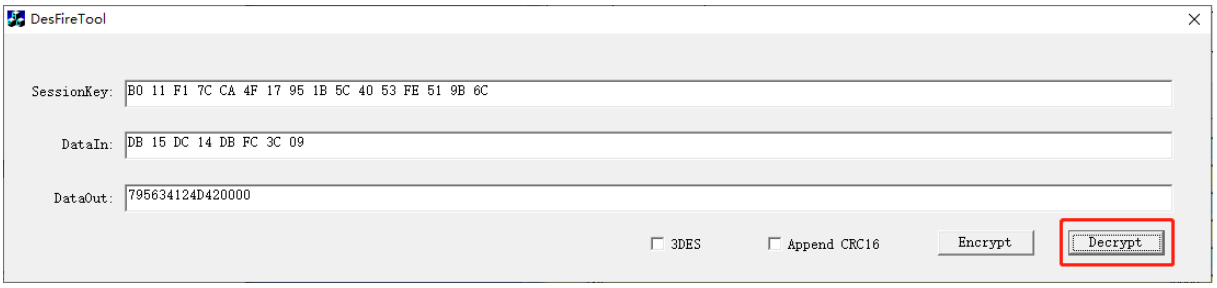
- **使钱包文件改动生效（Commit）：**

TransPort 中输入：AD
实际端口发出指令：00 04 00 AD A9
实际端口收到：00 05 01 AD 00 A9

- **再次读取钱包余额：**

TransPort 中输入：A6 01
实际端口发出指令：00 05 00 A6 01 A2
实际端口收到：00 0D 01 A6 00 DB 15 DC 14 DB FC 3C 09 BE

使用过程密钥（B0 11 F1 7C CA 4F 17 95 1B 5C 40 53 FE 51 9B 6C）对收到的加密数据（DB 15 DC 14 DB FC 3C 09）使用我们提供的工具软件 DESfire Tool 解密，得到解密后的钱包余额（795634124D420000），前 4 字节是钱包余额，79 56 34 12（0x12345679），4D 42 为 CRC 校验，如图：



- **写入备份数据文件 03：**

  TransPort 中输入：A5 03 00 00 00 08 00 00 01 02 03 04 05 06 07 08

  实际端口发出指令：00 13 00 A5 03 00 00 00 08 00 00 01 02 03 04 05 06 07 08 B5

  实际端口收到：00 05 01 A5 00 A1

- **读出备份数据文件 03：**

  TransPort 中输入：A4 03 00 00 00 08 00 00

  实际端口发出指令：00 0B 00 A4 03 00 00 00 08 00 00 A4

  实际端口收到：00 0D 01 A4 00 00 00 00 00 00 00 00 00 A8

  读出内容与写入内容不等，改动未生效

- **使备份数据文件改动生效（Commit）：**

  TransPort 中输入：AD

  实际端口发出指令：00 04 00 AD A9

  实际端口收到：00 05 01 AD 00 A9

- **再次读出备份数据文件 03：**

  TransPort 中输入：A4 03 00 00 00 08 00 00

  实际端口发出指令：00 0B 00 A4 03 00 00 00 08 00 00 A4

  实际端口收到：00 0D 01 A4 00 01 02 03 04 05 06 07 08 A0

  可以看到，改动已经被生效

5.2.3 建立文件结构（明文钱包）

取一张新卡，我们上面设计的文件系统进行，请按照顺序做如下操作，可以发送如下命令：

- **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 0E 00 20 BD 32 30 63 35 D8 9F 04 00 08 8C

- **认证卡片主控密钥：**

  TransPort 中输入：90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

  实际端口发出指令：00 15 00 90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 85

  实际端口收到：00 15 01 90 00 CC 6C E1 74 46 42 09 8D 1B 78 17 03 49 4C 67 A1 85

  在这里，返回的信息中 CC 6C E1 74 46 42 09 8D 1B 78 17 03 49 4C 67 A1 是本次卡片操作的过程密钥，需要保存。每次认证后得到的过程密钥都是不同的，需要注意。

- **擦除卡片所有应用：**

  TransPort 中输入：99

  实际端口发出指令：00 04 00 99 9D

  实际端口收到：00 05 01 99 00 9D

  此时卡片已经被全部擦除。

- **建应用：**

  TransPort 中输入：95 4A 4D 59 0F 0C

  实际端口发出指令：00 09 00 95 4A 4D 59 0F 0C C1

  实际端口收到：00 05 01 95 00 91

- **选择应用：**

  TransPort 中输入：98 4A 4D 59

  实际端口发出指令：00 07 00 98 4A 4D 59 C1

  实际端口收到：00 05 01 98 00 9C

- **认证应用密钥：**

  TransPort 中输入：90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

  实际端口发出指令：00 15 00 90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 85

  实际端口收到：00 15 01 90 00 E1 74 46 42 09 8D 1B CC 6C 78 17 03 49 4C 67 A1 85

- **建数值文件：**

  TransPort 中输入：A0 01 00 11 11 00 00 00 00 FF FF FF 7F 78 56 34 12 00

  实际端口发出指令：00 15 00 A0 01 00 11 11 00 00 00 00 FF FF FF 7F 78 56 34 12 00 3C

  实际端口收到：00 05 01 A0 00 A4

- **建标准数据文件：**

TransPort 中输入：9E 02 00 11 11 00 01 00
实际端口发出指令：00 0B 00 9E 02 00 11 11 00 01 00 96
实际端口收到：00 05 01 9E 00 9A

- **建备份数据文件：**
  TransPort 中输入：9F 03 00 11 11 00 01 00
  实际端口发出指令：00 0B 00 9F 03 00 11 11 00 01 00 96
  实际端口收到：00 05 01 9F 00 9B

- **建线性记录文件：**
  TransPort 中输入：A1 04 00 11 11 10 00 00 10 00 00
  实际端口发出指令：00 0E 00 A1 04 00 11 11 10 00 00 10 00 00 AB
  实际端口收到：00 05 01 A1 00 A5

- **建循环记录文件：**
  TransPort 中输入：A2 05 00 11 11 10 00 00 10 00 00
  实际端口发出指令：00 0E 00 A2 05 00 11 11 10 00 00 10 00 00 A9
  实际端口收到：00 05 01 A2 00 A6

5.2.4 应用实验（明文钱包）

- **按照 EMV 或 PBOC 的规则寻卡并复位：**
  见上面操作

- **选择应用：**
  TransPort 中输入：98 4A 4D 59
  实际端口发出指令：00 07 00 98 4A 4D 59 C1
  实际端口收到：00 05 01 98 00 9C

- **认证应用密钥 1（密钥 1 是我们的建立的文件的管控密钥）：**
  TransPort 中输入：90 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  实际端口发出指令：00 15 00 90 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 84
  实际端口收到：00 15 01 90 00 F9 5A B1 20 3D 75 D5 06 47 07 61 77 FC A3 0C 2E 06

- **读取钱包余额：**
  TransPort 中输入：A6 01
  实际端口发出指令：00 05 00 A6 01 A2
  实际端口收到：00 09 01 A6 00 78 56 34 12 A6
  钱包余额为 78 56 34 12（0x12345678）。

- **钱包加值：**
  钱包加值 0x00000001
  TransPort 中输入：A7 01 01 00 00 00
  实际端口发出指令：00 09 00 A7 01 01 00 00 00 AE

实际端口收到：00 05 01 A7 00 A3

- **使钱包文件改动生效（Commit）：**

  TransPort 中输入：AD
  实际端口发出指令：00 04 00 AD A9
  实际端口收到：00 05 01 AD 00 A9

- **再次读取钱包余额：**

  TransPort 中输入：A6 01
  实际端口发出指令：00 05 00 A6 01 A2
  实际端口收到：00 09 01 A6 00 79 56 34 12 A7
  钱包余额为：79 56 34 12 (0x12345679）

- **写入备份数据文件 03：**

  TransPort 中输入：A5 03 00 00 00 08 00 00 01 02 03 04 05 06 07 08
  实际端口发出指令：00 13 00 A5 03 00 00 00 08 00 00 01 02 03 04 05 06 07 08 B5
  实际端口收到：00 05 01 A5 00 A1

- **读出备份数据文件 03：**

  TransPort 中输入：A4 03 00 00 00 08 00 00
  实际端口发出指令：00 0B 00 A4 03 00 00 00 08 00 00 A4
  实际端口收到：00 0D 01 A4 00 00 00 00 00 00 00 00 00 A8
  读出内容与写入内容不等，改动未生效

- **使备份数据文件改动生效（Commit）：**

  TransPort 中输入：AD
  实际端口发出指令：00 04 00 AD A9
  实际端口收到：00 05 01 AD 00 A9

- **再次读出备份数据文件 03：**

  TransPort 中输入：A4 03 00 00 00 08 00 00
  实际端口发出指令：00 0B 00 A4 03 00 00 00 08 00 00 A4
  实际端口收到：00 0D 01 A4 00 01 02 03 04 05 06 07 08 A0
  可以看到，改动已经被生效