



FM8303

API 说明手册

时间：2016-9

版本：0.10



日期	版本号	简要描述
2016-9-12	0.10	1. 初始版本

上海复旦微电子集团股份有限公司

中国 上海

目 录

1	FM8303 驱动简介	4
2	驱动 API 功能及描述	5
2.1	API 描述	5
2.1.1	FM_Reader_Factory	5
2.1.1.1	getInstance	5
2.1.1.2	getReader	5
2.1.2	FM_Reader	5
2.1.2.1	open	5
2.1.2.2	close	6
2.1.2.3	setCardType	6
2.1.2.4	closeRF	6
2.1.2.5	fieldon	6
2.1.2.6	reset	8
2.1.2.7	dsinactive	8
2.1.2.8	setDefaultPPS4CT	8
2.1.2.9	setPPS4CT	8
2.1.2.10	setPPS4RF	8
2.1.2.11	getAppInfo	8
2.1.2.12	ReadAppCrc	9
2.1.3	IsoDepReader	9
2.1.3.1	apdu	9
2.1.3.2	tpdu	9
2.1.4	MifareReader	10
2.1.4.1	authenticateSectorWithKeyA	10
2.1.4.2	authenticateSectorWithKeyB	10
2.1.4.3	decrement	11
2.1.4.4	increment	11
2.1.4.5	transfer	11
2.1.4.6	initval	12
2.1.4.7	readval	12
2.1.4.8	readBlock	12
2.1.4.9	writeBlock	13



1 驱动简介

该驱动为 FM8303 实现读/写卡功能提供了一套应用层接口，使 Java 应用程序无需关心底层驱动程序实现细节。

2 驱动 API 功能及描述

2.1 API 描述

2.1.1 FM_Reader_Factory

根据读卡器设备实例的厂类。

2.1.1.1 getInstance

```
public FM_Reader_Factory getInstance()
```

功能：用于获取场类的实例。

2.1.1.2 getReader

```
public FM_Reader getReader()
```

功能：获取不同类型的卡读写器。

2.1.2 FM_Reader

8303 读写器对象。

2.1.2.1 open

```
public int open(String dev) throws Driver8303Exception
```

功能：终端打开 8303 读卡器，该操作不会触发打开天线动作。

参数说明：

dev: 设备的标识信息

返回值:

0: 表示成功, 其它表示失败, 具体请见后续结果代码定义

2.1.2.2 close

```
public int close () throws Driver8303Exception
```

功能: 关闭当前打开的设备, 该操作不会触发打开天线动作。

2.1.2.3 setCardType

```
public int setCardType (byte mode) throws Driver8303Exception
```

功能: 设置 8303 读卡器的类型。

参数说明:

mode: 设置非接触工作模式: 0x01 TYPE A (缺省); 0x02: TYPE B

返回值:

0: 表示成功, 其它表示失败, 具体请见后续结果代码定义

2.1.2.4 closeRF

```
public int closeRF () throws Driver8303Exception
```

功能: FM8303 关场。

返回值:

0: 表示成功, 其它表示失败, 具体请见后续结果代码定义

2.1.2.5 fieldon

```
public int fieldon(byte mode,  
                  byte FSDI_CID,  
                  FM_Bytes TagType,
```

```
FM_Bytes SAK,
FM_Bytes snr,
FM_Bytes ratsData) throws Driver8303Exception
```

功能：激活卡进入到 **active** 状态，完成 **request anticoll select** 全部过程，**rats** 可选；

参数说明：

mode,定义如下：

b6	b5	b4	b3	b2	b1	b0	说明
0	0	0	0	0	0	1	(1)ALL 模式,选择天线范围内所有卡片 0x52 = Wakeup (0)IDLE 模式,只选择天线范围内 IDLE 模式的卡片 0x26 = ReqA
0	0	0	0	0	1	0	执行 RATS
0	0	0	0	1	0	0	(1) 内卡 (0) 外卡
0	0	0	0	0	0	0	(0) 标准 14443—3

FSDI_CID 定义如下：

FSDI_CID 高半字节:表示 FSDI,FSDI 的取值决定了 FSD(近场通讯帧长度),FSDI 对应 FSD

如下：

FSDI	0	1	2	3	4	5	6	7	8	9-F
FSD	16	24	32	40	48	64	96	128	256	>256(RFU)

FSDI_CID 低半字：表示已编址的 PICC 的逻辑号

出口：TagType,返回卡类型值，0x0004 为 M1 卡，0x0010 为 ML 卡

SAK,返回 SAK;

snr,返回卡系列号(允许多级 UID 情况)

ratsData, rats 应答数据，不含 SW1 和 SW2

返回值：

0 表示成功，其它表示失败，具体请见后续结果代码定义

2.1.2.6 reset

```
public byte[] reset(int type,boolean isOutResponse) throws Driver8303Exception
```

功能：接触卡复位，包括冷复位和热复位，冷复位：设备会进行上电操作；热复位：设备不进行上电操作。

参数说明：

type: 复位类型 0: 冷复位； 1: 热复位

isOutResponse: 复位结果是否输出, false: 不输出; true: 输出

返回：

应答数据（如果不输出，则返回 0 长度的字节数组）

2.1.2.7 dsinactive

```
public int dsinactive () throws Driver8303Exception
```

功能：取消选卡(接触式)

返回：成功则返回 0,其它详见错误代码表

2.1.2.8 setDefaultPPS4CT

2.1.2.9 setPPS4CT

2.1.2.10 setPPS4RF

2.1.2.11 getAppInfo

```
public byte[] getAppInfo () throws Driver8303Exception
```

功能：获取程序版本信息。

上海复旦微电子股份有限公司
SHANGHAI FUDAN MICROELECTRONICS CO., LTD.

用户操作手册

FM8303

版本 0.10

8

返回:

程序版本信息

2.1.2.12 ReadAppCrc

2.1.3 IsoDepReader

CPU 卡读写器。

2.1.3.1 apdu

```
public byte[] apdu(byte flag ,byte[] request ,int timeout) throws Driver8303Exception
```

功能: CPU 卡指令执行;

参数说明:

request: 上位机向卡片发送的数据

timeout: 超时时间, 单位毫秒, 0 表示由系统默认超时时间

返回值:

卡片向上位机应答的数据。

2.1.3.2 tpdu

```
public byte[] tpdu(byte flag ,byte[] request ,int timeout) throws Driver8303Exception
```

功能: CPU 卡指令执行;

参数说明:

flag:

b7	b6	b5	b4	b3	b2	b1	b0	说明
0		0	0	0	0	0	0	非接通道
0		0	0	0	0	0	1	接触式通道

1		0	0	0	1	0	0	14443-4 或 7816-4 协议由驱动处理
0		0	0	0	0	0	0	14443-4 或 7816-4 协议由设备处理

request: 上位机向卡片发送的数据

timeout: 超时时间, 单位毫秒, 0 表示由系统默认超时时间

返回值:

卡片向上位机应答的数据。

2.1.4 MifareReader

M1 卡的读写器。

2.1.4.1 authenticateSectorWithKeyA

```
public boolean authenticateSectorWithKeyA(int sector,byte[] key)
    throws Driver8303Exception
```

功能: keyA 的密钥认证, 密钥的长度是 6 个字节

参数说明:

sector: 扇区号(0~63)

key: 密钥

返回:

true: 密钥验证通过

false: 密钥验证失败

2.1.4.2 authenticateSectorWithKeyB

```
public boolean authenticateSectorWithKeyA(int sector,byte[] key)
    throws Driver8303Exception
```

功能: keyA 的密钥认证, 密钥的长度是 6 个字节

参数说明:

sector: 扇区号(0~63)

key: ,密钥

返回:

true: 密钥验证通过

false: 密钥验证失败

2.1.4.3 decrement

public void increment(byte block,int value) throws Driver8303Exception

功能: 增值操作,用于将卡中的数值块加上某一数值 (value) 并把结果存于卡内的寄存器。

注意: 执行此操作时, 结果存放在卡内的寄存器中, 并没有对卡片的 EEPROM 写操作。需要再执行 Transfer 操作。

参数说明:

block: 待增值块号

value: 待增加金额

2.1.4.4 increment

public void decrement(byte block,long value) throws Driver8303Exception

功能: 减值操作, 用于将卡中的数值块减去某一数值 (value) 并把结果存于卡内的寄存器。

注意: 执行此操作时, 结果存放在卡内的寄存器中, 并没有对卡片的 EEPROM 写操作。需要再执行 Transfer 操作。

参数说明:

block: 待扣减块号

value: 待扣减金额

2.1.4.5 transfer

public void transfer(byte block) throws Driver8303Exception

功能: 传送, 将寄存器的内容传送到 EEPROM 中, 用于将卡内寄存器中的内容写入块中。

此操作只能在执行 increment、decrement 和 restore 操作后才可以直接调用;

参数说明:

block,钱包块地址

2.1.4.6 initval

```
public void initval(byte block,long value) throws Driver8303Exception
```

功能: 给卡片中的某一块区赋予初值(value), 以供加值和减值。实际上执行的是 Write 命令;

参数说明:

block: 钱包块地址

value: 初始化钱包数值

2.1.4.7 readval

```
public byte[] readval(int block) throws Driver8303Exception
```

功能: 读取钱包块金额, 实际上执行的是 read 命令;

参数说明:

block: 钱包块地址

返回:

钱包金额

2.1.4.8 readBlock

```
public byte[] read(byte block) throws Driver8303Exception
```

功能: 读出卡中某一块的 16 个字节数;

参数说明:

block: 待读出数据的块号

返回:

读出的数据

2.1.4.9 writeBlock

```
public void write(byte block, byte[] data) throws Driver8303Exception
```

功能：将数据写入卡中的某一块；

参数说明：

block: 待写入数据块号

data: 待写入块数据,16 字节