



# FM8301 双界面读卡机具

*用户操作手册*

---

**2013.7**



## 版本列表

日期	版本号	简要描述
2011-10-26	0.10	1. 初始版本
2013-4-15	1.00	1.修改 APDU 描述，注意事项增加 64 位动态库。
2013-7-17	1.01	1.补充 FM_API_SearchHidDevice 函数使用限制。



# 目 录

<b>1</b>	<b>简介</b>	<b>6</b>
1.1	图示	6
1.2	说明	6
1.3	读卡机工作流程	7
<b>2</b>	<b>上位机动态库函数</b>	<b>7</b>
2.1	FM8301V2 设备 API	7
2.1.1	常量定义	7
2.1.2	FM8301V2 接口函数定义（使用芯片 FM1715）	7
<b>3</b>	<b>DEMO 代码说明</b>	<b>31</b>
<b>4</b>	<b>注意</b>	<b>31</b>
<b>5</b>	<b>附件</b>	<b>32</b>
5.1	错误代码定义	32
5.1.1	说明	32
5.1.2	设备底层错误代码定义	33
5.1.3	API 错误代码定义	35

FM8301 是上海复旦微电子集团在原有 FM8201SD 桌面非接触 IC 卡读写机的基础上，结合 USB 端口以及接触卡读卡机要求升级的一款产品。该产品是非接触读卡机和接触卡读卡机结合的一款产品，产品提供 USB2.0 全速接口，采用 HID1.1 协议，windows 操作系统使用无需安装 USB 驱动。（FM8301 暂不提供该型号串口接口产品，如用户有特殊需要可另外向公司专门订货）

### 外形：

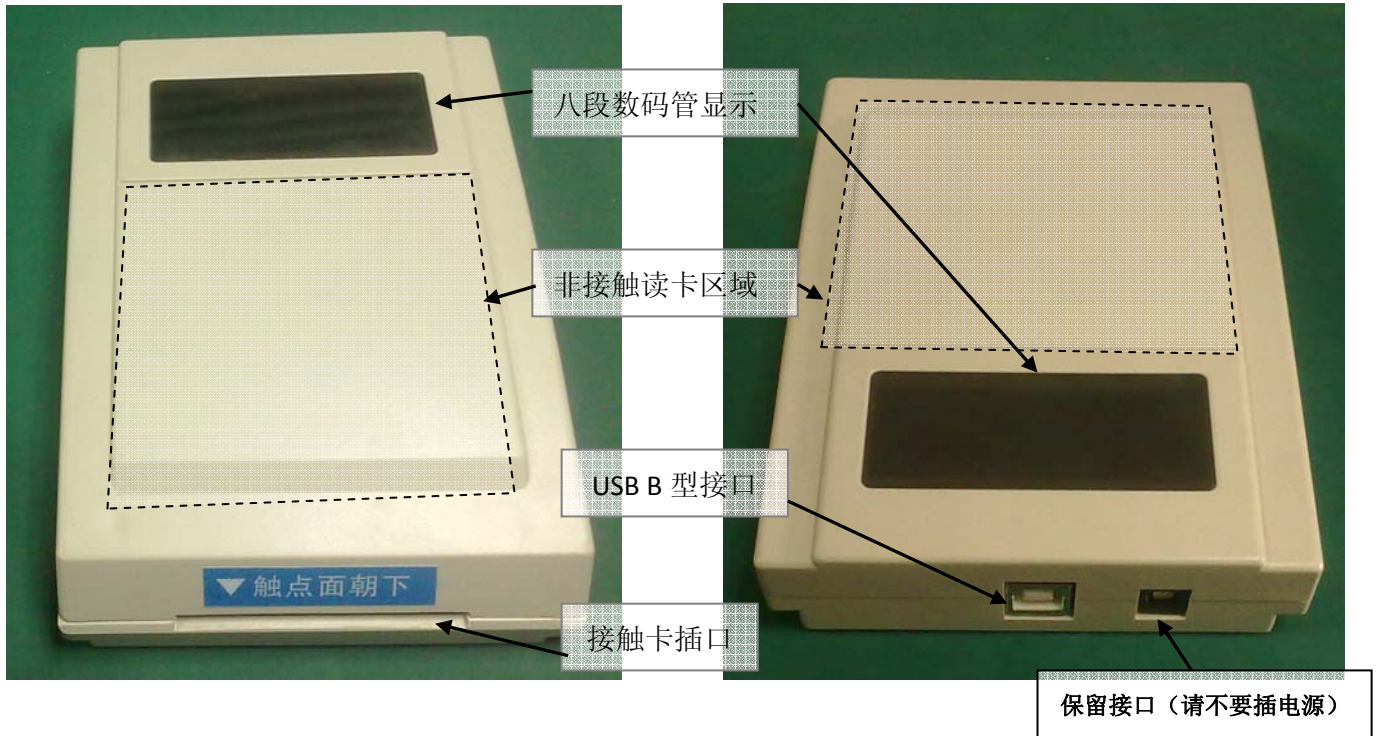


## 术语与缩略语

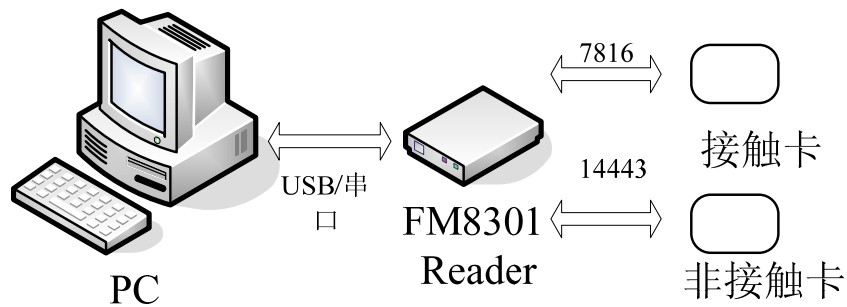
术语、缩略语	定 义
APDU	应用协议数据单元 (Application Protocol Data Unit)
API	应用程序编程接口(Application Programming Interface)
PPS	协议参数选择
PTS	协议类型选择
RFU	预留 (Reserved for Future Use)
RATS	选择应答请求
TPDU	传送协议数据单元(Transport Protocol Data Unit)
TLV	标记长度值
XOR	异或
WTX	扩展等待时间

## 1 简介

### 1.1 图示



### 1.2 说明



FM8301 读卡机配合 PC 使用，使用者请将读卡机通过 USB 线连接到 PC 机，读卡机由 USB 供电，无需外接电源。读卡机支持 USB2.0 全速 HID 协议，无需另外安装驱动。使用者可以通过提供的动态链接库使用 API 函数操作读卡机进行接触卡(ISO7816 class B T=0 协议)或非接触卡(ISO14443 TYPEA)的读写。接触卡使用时请注意插入方向，触点面向下插入。

上海复旦微电子集团股份有限公司  
Shanghai Fudan Microelectronics Group Company Limited

用户操作手册

## 1.3 读卡机工作流程

FM8301 读卡机在主动询卡过程中会自动寻找 ISO14443 typeA 协议非接触卡，如果有卡片进入非接触卡操作区域则显示卡片的 UID（8 位 16 进制），无卡显示“hello”。当收到 pc 通信要求后，会终止这一过程进入通信及命令处理流程，此时使用者可以通过 PC 对读卡机进行操作。

## 2 上位机动态库函数

### 2.1 FM8301V2 设备 API

#### 2.1.1 常量定义

```
#define FM8301V2_API_RF_SLOT1      0      //RF卡槽
#define FM8301V2_API_CT_SLOT1     1      //接触式卡槽1(小卡槽)
#define FM8301V2_API_CT_SLOT2     2      //接触式卡槽2(大卡槽)
```

#### 2.1.2 FM8301V2 接口函数定义（使用芯片 FM1715）

##### 2.1.2.1 FM\_API\_SearchHidDevice（调用时候请注意说明）

```
int FM_API_SearchHidDevice(unsigned char* SN,unsigned int snBufferSize);
```

功能：搜索设备

说明：支持HID端口

入口：snBufferSize, SN 缓存大小

出口：SN, 返回1个或多个产品序列号(LV格式, L为1字节(0-255), V是以0结尾的字符串)

返回：<0, 表示函数执行失败, =0, 表示没有搜索到指定要求的设备, >0, 表示搜索到的设备个数

说明：多进程调用动态库时候，在设备通信过程中不能调用FM\_API\_SearchHidDevice函数，改函数会造成设备通信通道和控制通道冲突引起长时间等待或设备复位，在多进程时候建议搜索设备不要使用，或者避开设备通信过程调用改函数。

### 2.1.2.2 FM\_API\_OpenHidDevice

```
int FM_API_OpenHidDevice(const unsigned char *SN=NULL);
```

功能：打开设备

说明：支持HID端口；

入口：SN, 产品序列号, 以0结尾的字符串指针; SN=NULL时默认打开第一个设备

出口：无

返回：小于0表示函数执行错误, 否则返回端口句柄参数

### 2.1.2.3 FM\_API\_OpenUartDevice (专供使用串口接口读卡机)

```
FM_API_OpenUartDevice(unsigned int port, unsigned long rate);
```

功能：打开串口 (Windows)

说明：只支持UART接口

入口：port: 端口号;

rate: 端口通讯波特率

出口：无

返回：>0, 返回句柄；<0, 错误；=0: 预留

### 2.1.2.4 FM\_API\_Close

```
int FM_API_Close(unsigned int handle);
```

功能：关闭设备

入口：handle, 句柄

出口：无

返回：0表示成功，其它表示失败

### 2.1.2.5 FM\_API\_GetEditionInfo

```
int FM_API_GetEditionInfo(unsigned int handle, unsigned char* info, unsigned int* infoLen, unsigned int rBufSize);
```

上海复旦微电子集团股份有限公司  
Shanghai Fudan Microelectronics Group Company Limited



功能：获取程序版本信息

入口：handle, 句柄;

rBufSize, 返回程序版本信息info缓存大小

出口：info, 返回程序版本信息

infoLen, 返回程序版本信息长度

返回：0表示成功，其它表示失败

### 2.1.2.6 FM\_API\_Beep

int FM\_API\_Beep(unsigned int handle, unsigned int msec);

功能：蜂鸣

入口：handle, 句柄

\_Msec: 蜂鸣时间, 典型时间为0x7f

出口：无

返回：0表示成功，其它表示失败

### 2.1.2.7 FM\_API\_Display

int FM\_API\_Display(unsigned int handle, const unsigned char\* str, unsigned int strlen);

功能：屏幕显示

入口：handle, 句柄

str: 待显示字符, HEX码表示, 长度小于或等于4字节

strlen: 待显示字符长度

出口：无

返回：0表示成功，其它表示失败

### 2.1.2.8 FM\_API\_Request

int FM\_API\_Request(unsigned int handle, unsigned char mode, unsigned short int\* tagType);

上海复旦微电子集团股份有限公司  
Shanghai Fudan Microelectronics Group Company Limited

功能：寻卡

入口：handle, 句柄

mode:0:表示IDLE模式, 只选择天线范围内IDLE模式的卡片;1表示ALL模式, 选择天线范围内所有卡片

出口：tagType, 返回卡片tagType值

返回：0表示成功, 其它表示失败

### 2.1.2.9 FM\_API\_Anticoll

FM\_API\_Anticoll(unsigned int handle, unsigned char sel, unsigned char nvb, unsigned long\* snr,  
 unsigned char &bcc, unsigned char &collisionPos)

功能：防冲突, 用于防冲突操作, 轮询卡片

说明：支持windows和linux系统

入口：sel(目前该参数固定为 0x93)定义如下：

b8	b7	b6	b5	b4	b3	b2	b1	说明
1	0	0	1	0	0	1	1	'93': 选择串联级别 1
1	0	0	1	所有其他				RFU

nvb, 有效位数, 目前该参数固定为0x20

其他, RFU

出口：snr, 卡片UID, 此处仅考虑4个字节的UID;

bcc, BCC校验码, 目前该参数无效

collisionPos, 冲突位, 目前该参数无效

返回：0表示成功, 其它表示失败, 具体请见后续结果代码定义

### 2.1.2.10 FM\_API\_Select

int FM\_API\_Select(unsigned int handle, unsigned char sel, unsigned char nvb, unsigned long snr,

```
unsigned char &SAK);
```

功能：选卡

说明：支持windows和linux系统；

入口：handle, 句柄

Sel (目前该参数固定为 0x93) 定义如下：

b8	b7	b6	b5	b4	b3	b2	b1	说明
1	0	0	1	0	0	1	1	'93': 选择串联级别 1
1	0	0	1	所有其他				RFU

nvb, 有效位数, 目前该参数固定为0x70

snr: 卡片系列号

出口：SAK: 返回SAK值

返回：0表示成功, 其它表示失败

### 2.1.2.11 FM\_API\_LoadKey

```
int FM_API_LoadKey(unsigned int handle,unsigned char mode,unsigned char sector,
```

```
const unsigned char* key, unsigned int keyLen);
```

功能：装载密钥

入口：handle, 句柄

mode: 装入密码模式, 0~3 KeyA , 4~7 KeyB;

sector: 扇区号 (0~15);

key: 密钥值

keyLen: 密钥长度

出口：无

上海复旦微电子集团股份有限公司  
Shanghai Fudan Microelectronics Group Company Limited

FM8301

版本 1.01

用户操作手册

11



返回：0表示成功，其它表示失败

### 2.1.2.12 FM\_API\_Authentication

`int FM_API_Authentication(unsigned int handle,unsigned char mode,unsigned char sector, unsigned long snr=0);`

功能：密钥认证

入口：handle, 句柄

mode:装入密码模式, 0~3 KeyA, 4~7 KeyB;

sector:扇区号(0~15);

snr:卡片序列号, 若snr等于0, 由设备默认卡片序列号

出口：无

返回：0表示成功，其它表示失败

### 2.1.2.13 FM\_API\_Read

`int FM_API_Read(unsigned int handle,unsigned char block,unsigned char* rdata,unsigned int* rdataLen,unsigned int rBufSize);`

功能：读块

入口：handle, 句柄

block:块号;

rBufSize, 返回数据rdata缓存大小

出口：rdata, 返回块数据

rdataLen, 返回数据长度

返回：0表示成功，其它表示失败

### 2.1.2.14 FM\_API\_Write

`int FM_API_Write(unsigned int handle,unsigned char block,const unsigned char* data,unsigned int dataLen);`

功能：写块

入口：handle, 句柄

上海复旦微电子集团股份有限公司  
Shanghai Fudan Microelectronics Group Company Limited

block:块号;

出口: data, 待写入数据

dataLen, 待写入数据长度

返回: 0表示成功, 其它表示失败

### 2.1.2.15 FM\_API\_HighRead

```
int FM_API_HighRead(unsigned int handle,unsigned char mode,unsigned char block,unsigned char* rdata,  
                    unsigned int* rdataLen,unsigned int rBufSize,unsigned long* snr);
```

功能: 高级读

说明: 支持windows和linux系统;支持FM8201(新)、FM8201(老)和FM274

入口: handle, 句柄

mode:密钥认证模式, 0~3 KeyA , 4~7 KeyB;

block:块号

rBufSize, 返回数据rdata缓存大小

出口: rdata:返回块数据;

rdataLen:返回块数据长度;

snr:返回卡片系列号

返回: 0表示成功, 其它表示失败

### 2.1.2.16 FM\_API\_HighWrite

```
int FM_API_HighWrite(unsigned int handle,unsigned char mode,unsigned char block,  
                     const unsigned char* data,unsigned int dataLen,unsigned long* snr);
```

功能: 高级写

入口: handle, 句柄

mode: 密钥认证模式, 0~3 KeyA , 4~7 KeyB;

block: 块号

data: 待写入数据;

dataLen, 待写入数据长度

出口: snr: 返回卡片系列号

返回: 0表示成功, 其它表示失败

### 2.1.2.17 FM\_API\_Initval

```
int FM_API_Initval(unsigned int handle, unsigned char block, unsigned long value);
```

功能: 钱包初始化

入口: handle, 句柄

block: 块号;

value: 钱包初值

出口: 无

返回: 0表示成功, 其它表示失败

### 2.1.2.18 FM\_API\_HighInitval

```
int FM_API_HighInitval(unsigned int handle, unsigned char mode, unsigned char block,  
  
                        unsigned long value, unsigned long* snr);
```

功能: 高级初始化操作

入口: handle, 句柄

mode: 密钥认证模式, 0~3 KeyA , 4~7 KeyB;

block: 块号;

value: 钱包初值



出口: snr:返回卡片系列号

返回: 0表示成功, 其它表示失败

#### 2.1.2.19 FM\_API\_Increment

```
int FM_API_Increment(unsigned int handle, unsigned char block, unsigned long value);
```

功能: 钱包增值

入口: handle, 句柄

block:块号;

value:待增值数

出口: 无

返回: 0表示成功, 其它表示失败

#### 2.1.2.20 FM\_API\_HighIncrement

```
int FM_API_HighIncrement(unsigned int handle, unsigned char mode, unsigned char block,
                          unsigned long value, unsigned long* rvalue, unsigned long* snr);
```

功能: 高级增值

入口: handle, 句柄

mode:密钥认证模式, 0~3 KeyA , 4~7 KeyB;

block:块号;

value: 增值数

出口: rvalue:返回加值后钱包值

snr:返回卡片系列号;

返回: 0表示成功, 其它表示失败

#### 2.1.2.21 FM\_API\_Decrement

```
int FM_API_Decrement(unsigned int handle, unsigned char block, unsigned long value);
```

上海复旦微电子集团股份有限公司  
Shanghai Fudan Microelectronics Group Company Limited



功能：钱包减值

入口：handle, 句柄

block:块号;

value:待减值数

出口：无

返回：0表示成功，其它表示失败

### 2.1.2.22 FM\_API\_HighDecrement

```
int FM_API_HighDecrement(unsigned int handle, unsigned char mode, unsigned char block,  
                           unsigned long value, unsigned long* rvalue, unsigned long* snr);
```

功能：高级减值

入口：handle, 句柄

mode:密钥认证模式, 0~3 KeyA , 4~7 KeyB;

block:块号;

value: 减值数

出口：rvalue:返回充值后钱包值

snr:返回卡片系列号;

返回：0表示成功，其它表示失败

### 2.1.2.23 FM\_API\_Readval

```
int FM_API_Readval(unsigned int handle, unsigned char block, unsigned long* value);
```

功能：读钱包值

入口：handle, 句柄

block:块号;





出口: value:返回钱包值

返回: 0表示成功, 其它表示失败

#### 2.1.2.24 FM\_API\_HighReadval

```
int FM_API_HighReadval(unsigned int handle, unsigned char mode, unsigned char block,  
  
                        unsigned long* value, unsigned long* snr);
```

功能: 高级读值

入口: handle, 句柄

mode: 密钥认证模式, 0~3 KeyA, 4~7 KeyB;

block: 块号;

出口: value: 返回钱包值

snr: 返回卡片系列号

返回: 0表示成功, 其它表示失败

#### 2.1.2.25 FM\_API\_Restore

```
int FM_API_Restore(unsigned int handle, unsigned char block);
```

功能: 回传, 将EEPROM中的内容传入卡的内部寄存器

入口: handle, 句柄

block: 块号;

出口: 无

返回: 0表示成功, 其它表示失败

#### 2.1.2.26 FM\_API\_Transfer

```
int FM_API_Transfer(unsigned int handle, unsigned char block);
```

功能: 传送, 将寄存器的内容传送到EEPROM中

入口: handle, 句柄

上海复旦微电子集团股份有限公司  
Shanghai Fudan Microelectronics Group Company Limited

block:块号;

出口: 无

返回: 0表示成功, 其它表示失败

### 2.1.2.27 FM\_API\_SetCardType

```
int FM_API_SetCardType(unsigned int handle, unsigned char type);
```

功能: 设置卡类型

入口: handle, 句柄

type: 卡类别编号 0: 表示TypeA卡; 2: 上海标准卡; 出口: 无

返回: 0表示成功, 其它表示失败

### 2.1.2.28 FM\_API\_Halt

```
int FM_API_Halt(unsigned int handle);
```

功能: 中止卡操作

入口: handle, 句柄

出口: 无

返回: 0表示成功, 其它表示失败

### 2.1.2.29 FM\_API\_UltralightReqA

```
int FM_API_UltralightReqA(unsigned int handle, unsigned char* resp, unsigned int* rlen, unsigned int rBufSize);
```

功能: 寻卡 (ultralight卡片)

入口: handle, 句柄

rBufSize, 返回数据resp缓存大小

出口: resp: 寻卡返回数据

rlen, 返回数据长度

返回: 0表示成功, 其它表示失败

上海复旦微电子集团股份有限公司  
Shanghai Fudan Microelectronics Group Company Limited

### 2.1.2.30 FM\_API\_UltralightWupA

```
int FM_API_UltralightWupA(unsigned int handle, unsigned char* resp, unsigned int* rlen, unsigned int rBufSize);
```

功能：唤醒(ultralight卡片)

入口：handle, 句柄

rBufSize, 返回数据resp缓存大小

出口：resp: 唤醒返回数据

rlen, 返回数据长度

返回：0表示成功，其它表示失败

### 2.1.2.31 FM\_API\_UltralightAnticoll1st

```
int FM_API_UltralightAnticoll1st(unsigned int handle, unsigned char* firstPartUid,  
                                  unsigned int* uidLen, unsigned int rBufSize);
```

功能：防冲突步骤1(ultralight卡片)

入口：handle, 句柄

rBufSize, 返回数据firstPartUid缓存大小

出口：firstPartUid: 返回UID前半部分

uidLen, 返回前半部分UID长度

返回：0表示成功，其它表示失败

### 2.1.2.32 FM\_API\_UltralightSelect1st

```
int FM_API_UltralightSelect1st(unsigned int handle, const unsigned char* firstPartUid,  
                                 unsigned int uidLen, unsigned char* resp, unsigned int* rlen, unsigned int rBufSize);
```

功能：选卡步骤1(ultralight卡片)

入口：handle, 句柄

firstPartUid: UID前半部分

上海复旦微电子集团股份有限公司  
Shanghai Fudan Microelectronics Group Company Limited

uidLen, 前半部分UID长度

rBufSize, 返回数据缓存大小

出口: resp: 选卡返回数据

rLen, 返回数据长度

返回: 0表示成功, 其它表示失败

### 2.1.2.33 FM\_API\_UltralightAnticoll2nd

```
int FM_API_UltralightAnticoll2nd(unsigned int handle, unsigned char* lasPartUid,  
  
                                unsigned int* uidLen, unsigned int rBufSize);
```

功能: 防冲突步骤2(ultralight卡片)

入口: handle, 句柄

rBufSize, 返回数据lasPartUid缓存大小

出口: lasPartUid: 返回UID后半部分

uidLen, 返回后半部分UID长度

返回: 0表示成功, 其它表示失败

### 2.1.2.34 FM\_API\_UltralightSelect2nd

```
int FM_API_UltralightSelect2nd(unsigned int handle, const unsigned char* lasPartUid,  
  
                                unsigned int uidLen, unsigned char* resp, unsigned int* rLen, unsigned int rBufSize);
```

功能: 选卡步骤2(ultralight卡片)

入口: handle, 句柄

lastPartUid: UID后半部分

uidLen, 后半部分UID长度

rBufSize, 返回数据resp缓存大小

出口: resp:选卡返回数据

rLen, 返回数据长度

返回: 0表示成功, 其它表示失败

### 2.1.2.35 FM\_API\_UltralightRead

```
int FM_API_UltralightRead(unsigned int handle, unsigned char block, unsigned char* rdata,  
  
                          unsigned int* rdataLen, unsigned int rBufSize);
```

功能: 读卡 (ultralight卡片)

入口: handle, 句柄

block:块号;

rBufSize, 返回数据rdata缓存大小

出口: rdata:返回块数据, 一次操作返回16个字节(4块), 当从数据块后部读取时, 不足16字节的数据从卡片前部补足(类似循环buffer)

rdataLen, 返回数据长度

返回: 0表示成功, 其它表示失败

### 2.1.2.36 FM\_API\_UltralightWrite

```
int FM_API_UltralightWrite(unsigned int handle, unsigned char block, const unsigned char* data,  
  
                           unsigned int dataLen);
```

功能: 写卡 (ultralight卡片)

入口: handle, 句柄

block:块号;

data, 待写入数据, 一次操作写4个字节

dataLen, 待写入数据长度

出口: 无

上海复旦微电子集团股份有限公司  
Shanghai Fudan Microelectronics Group Company Limited

FM8301

版本 1.01

用户操作手册

21

返回：0表示成功，其它表示失败

### 2.1.2.37 FM\_API-TokenHighReadNoauth

```
int FM_API-TokenHighReadNoauth(unsigned int handle, unsigned char block, unsigned char* rdata,  
                                unsigned int* rdataLen, unsigned int rBufSize, unsigned long* snr);
```

功能：token卡不带认证高级读(用于读0-7块，无须密钥)

入口：handle, 句柄

block:块号;

rBufSize, 返回数据rdata缓存大小

出口：rdata, 返回数据

rdataLen, 返回数据长度

snr, 返回卡片SN号

返回：0表示成功，其它表示失败

### 2.1.2.38 FM\_API-TokenHighRead

```
int FM_API-TokenHighRead(unsigned int handle, unsigned char block, unsigned char* rdata,  
                          unsigned int* rdataLen, unsigned int rBufSize, unsigned long* snr);
```

功能：token卡高级读函数

入口：handle, 句柄

block:块号;

rBufSize, 返回数据rdata缓存大小

出口：rdata, 返回块数据

rdataLen, 返回数据长度

snr, 返回卡片序列号

返回：0表示成功，其它表示失败

### 2.1.2.39 FM\_API-TokenHighWrite

```
int FM_API-TokenHighWrite(unsigned int handle, unsigned char block, const unsigned char* data,
                           unsigned int dataLen);
```

功能：token卡高级写函数

入口：handle, 句柄

block:块号;

data, 待写入数据

dataLen, 待写入数据长度

出口：无

返回：0表示成功，其它表示失败

### 2.1.2.40 FM\_API\_Fieldon

```
int FM_API_Fieldon(unsigned int handle, unsigned char mode, unsigned char FSDI_CID,
                   unsigned short int* tagType, unsigned char &SAK,
                   unsigned char* snrBuffer, unsigned int* snrLength, unsigned int snrBufferSize,
                   unsigned char* ratsBuffer, unsigned int* ratsLength, unsigned int ratsBufferSize);
```

入口：handle, 句柄

mode, 定义如下:

b	b	b	b	b	b	b	b	说明
7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	b	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	0	b	0	执行 RATS
0	0	0	0	0	b	0	0	0: 外卡
0	0	0	0	b	0	0	0	1: 上海标准 0: 标准 14443-3
b	0	0	0	0	0	0	0	1: Request 前延迟 0: 无需延迟

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

snrBufferSize, snrBuffer缓存大小

ratsBufferSize, ratsBuffer缓存大小

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

SAK, 返回SAK;

snr, 返回卡系列号(允许多级UID情况)

snrLength, 返回卡系列号长度

ratsData, rats应答数据, 不含SW1和SW2

#### 2.1.2.41 FM\_API\_Rats

```
int FM_API_Rats(unsigned int handle, unsigned char FSDI_CID, unsigned char* resp,
                unsigned int* rlen, unsigned int rBufSize);
```

功能: 选择应答请求

上海复旦微电子集团股份有限公司  
Shanghai Fudan Microelectronics Group Company Limited

FM8301

版本 1.01

用户操作手册

24





入口: handle, 句柄

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低半字节:表示已编址的PICC的逻辑号

出口: resp:应答数据;

返回: 0表示成功, 其它表示失败

#### 2.1.2.42 FM\_API\_PPS

```
int FM_API_PPS(unsigned int handle, unsigned char pps0, unsigned char pps1);
```

功能: 协议参数选择

入口: handle, 句柄

pps0, 协议参数选择0

Pps1, 协议参数选择1

出口: 无

返回: 0表示成功, 其它表示失败

#### 2.1.2.43 FM\_API\_Reset

```
int FM_API_Reset(unsigned int handle, unsigned char flag, unsigned char* resp, unsigned int* rlen,
                 unsigned int rBufSize, unsigned int slot=FM8201_API_CT_SLOT1);
```

功能: 接触卡复位

说明: 冷复位, 设备会进行上电操作;

热复位, 设备不进行上电操作;

执行该命令前必须先执行FM\_API\_FM17InReset

入口: handle, 句柄

flag:高半字节为表示0热复位,高半字节为1表示冷复位;低半字节为0表示需要回送结果;低半字节为1表示不需要回送结果(注意如果不返回结果ATR,PTS指令无法执行)

rBufSize, 返回数据resp缓存大小

slot:默认接触卡槽1

出口: resp, 返回复位数据

rLen, 返回复位数据长度

返回: 0表示成功, 其它表示失败

#### 2.1.2.44 FM\_API\_PTS

```
int FM_API_PTS(unsigned int handle,const unsigned char* data,unsigned int dataLen,unsigned char* resp,  
  
               unsigned int* rLen,unsigned int rBufSize,unsigned int slot=FM8201_API_CT_SLOT1);
```

功能: 协议类型选择

入口: handle, 句柄

data:协议类型选择参数(具体请见IS07816协议)

参考数据: 9600: FF 10 11 FE

19200: FF 10 12 FD

38400: FF 10 13 FC

76800: FF 10 14 FB

115200:FF 10 18 F7

dataLen:协议类型选择参数数据长度

rBufSize, 返回数据resp缓存大小

slot:默认接触卡槽1

出口: resp, 返回数据

rLen, 返回复位数据长度

返回: 0表示成功, 其它表示失败

#### 2.1.2.45 FM\_API\_DisActive

```
int FM_API_DisActive(unsigned int handle,unsigned int slot=FM8201_API_CT_SLOT2);
```

功能: 接触卡下电

说明: 只支持大卡槽

入口: handle, 句柄

slot:默认接触卡槽2

出口: 无

返回: 0表示成功, 其它表示失败

#### 2.1.2.46 FM\_API\_APDU

```
int FM_API_APDU(unsigned int handle,unsigned char flag,const unsigned char * cmd,unsigned int cmdLen,
```

```
unsigned char *resp,unsigned int* rLen,unsigned int rBufSize,unsigned int timeOut=0);
```

功能: 应用协议数据单元

入口: handle, 句柄

flag定义如下:

b7	b6	b5	b4	b3	b2	b1	b0	说明
0	0	0						<b>b0~b2:表示卡通道号</b>  <b>0: 表示 FM8301 非接触通道 1</b>  <b>1: 表示 FM8301 接触通道 1(小卡槽)</b>  <b>2: 表示 FM8301 接触通道 2(大卡槽)</b>
0	0	0						
0	0	0						

0	0	0			<p><b>B3~b4:</b>表示卡应用平台</p> <p><b>0:</b> 表示默认平台</p> <p><b>1:</b> 表示 java 平台</p>
1	0	0			<p><b>b7:</b> 协议处理标志位</p> <p><b>1:</b> 表示 14443-4 或 7816-4 协议由驱动处理</p> <p><b>0:</b> 表示 14443-4 或 7816-4 协议由设备处理</p> <p>附：建议 7816-4 协议由设备处理，速度会更快；建议 14443-4 协议由驱动处理，因设备处理 14443-4 协议未经长时间测试</p>

cmd, 发送命令

cmdLen, 发送命令长度

rBufSize, 应答数据resp缓存大小

timeOut:超时时间，单位毫秒，0表示由系统默认超时时间

出口： resp, 应答数据

rLen, 应答数据长度

返回： 0表示成功，其它表示失败，具体请见后续结果代码定义

### 2.1.2.47 FM\_API\_TPDU

```
int FM_API_APDU(unsigned int handle, unsigned char flag, const unsigned char * cmd, unsigned int cmdLen,
```

```
    unsigned char *resp, unsigned int* rlen, unsigned int rBufSize, unsigned int timeOut=0);
```

功能： 传输协议数据单元

入口： handle, 句柄

flag定义如下：

b7	b6	b5	b4	b3	b2	b1	b0	说明
----	----	----	----	----	----	----	----	----

0	0	0	0	0		b0~b2:表示卡通道号
0	0	0	0	0		0: 表示 FM8301 非接触通道 1
0	0	0	0	0		1: 表示 FM8301 接触通道 1

cmd, 发送命令

cmdLen, 发送命令长度

rBufSize, 应答数据resp缓存大小

timeOut:超时时间, 单位毫秒, 0表示由系统默认超时时间

出口: resp, 应答数据

rLen, 应答数据长度

返回: 0表示成功, 其它表示失败, 具体请见后续结果代码定义

#### 2.1.2.48 FM\_API\_Deselect

`int FM_API_Deselect(unsigned int handle);`

功能: 中止卡操作(CPU卡)

入口: handle, 句柄

出口: 无

返回: 0表示成功, 其它表示失败

#### 2.1.2.49 FM\_API\_RadioFrequencyReset

`int FM_API_RadioFrequencyReset(unsigned int handle, unsigned int _Msec);`

功能: 射频读写模块复位

入口: handle, 句柄

\_Msec: 复位持续时间, 0~250有效; 典型值0x7F

出口: 无

返回：0表示成功，其它表示失败

#### 2.1.2.50 FM\_API\_FM17InReset

```
int FM_API_FM17InReset(unsigned int handle);
```

功能：使FM17XX一直处于复位状态

说明：支持windows和linux系统；

入口：无

出口：无

返回：0表示成功，其它表示失败

#### 2.1.2.51 FM\_API\_SetBaudrate

```
int FM_API_SetBaudrate(unsigned int handle,unsigned long rate,unsigned int slot=FM8301_API_CT_SLOT2);
```

功能：设置接触卡通讯波特率

入口：handle, 句柄

rate, 波特率

slot, 默认接触卡2

出口：无

返回：0表示成功，其它表示失败

#### 2.1.2.52 FM\_API\_TimeConfigure

```
int FM_API_TimeConfigure(unsigned int handle,unsigned char flag,unsigned short int byteTime,
```

```
unsigned short int streamTime,unsigned int slot=FM8201V2_API_CT_SLOT2);
```

功能：时间参数配置

说明：只支持大卡槽

入口：handle, 句柄

flag, 1, 表示配置字节间延时, 此时参数streamTime无意义;

上海复旦微电子集团股份有限公司  
Shanghai Fudan Microelectronics Group Company Limited

2, 表示配置数据流方向改变时的延时时间, 此时参数2无意义;

3, 表示配置字节间延时和数据方向改变时的延时时间

其它预留

byteTime, 表示字节间延时时间, 以0.25us为计数单位, 比如1000代表250us该值(字节间延时时间)可被接触卡冷热复位、PTS命令、配置波特率命令改变。

streamTime, 表示数据流方向改变时的延时时间;意义同上

slot, 默认卡槽2

出口: 无

返回: 0表示成功, 其它表示失败

### 3 DEMO 代码说明

默认路径为\DLL\_and\_demo\, 该目录下有以下文件

\demo\TestEnv\FM8301Demo.exe 读卡器 DEMO 软件

\demo\ VC2008 语言范例的源代码目录

\api\ 外部使用 API 接口函数目录

具体操作卡片流程, 请查阅相关卡片说明书。

### 4 注意

- 目前发布 API 为 cdecl/stdcall 两种调用方式动态库, X86 为 32 位操作系统版本, x64 为 64 位操作系统版本。
- 用户可向公司索取后续版本更新

## 5 附件

### 5.1 错误代码定义

#### 5.1.1 说明

注：为统一规划错误代码，约定各层错误代码定义如下：

1. 0x8000xxxx表示通用错误代码

2. 0x8001xxxx表示设备返回的错误代码, 其中最后字节xx是由设备返回的应答状态码, 具体含义请参见具体设备错误代码定义

0x800100xx表示设备执行命令返回的状态字节

0x800101xx表示设备接收APP协议层返回错误代码

0x800102xx表示设备接收HID协议层返回错误代码

0x800140xx表示FM83xx系列读卡器返回统一错误状态码

其它x8000xxxx表示卡片返回状态字SW1和SW2

3. 0x8003xxxx表示工具代码库返回的错误代码

0x800301xx表示端口通讯工具代码库返回的错误代码

4. 0x8004xxxx表示驱动代码库返回的错误代码

0x800401xx表示设备驱动模块返回的错误代码

0x800402xx表示设备协议模块返回的错误代码

0x800403xx表示通讯协议模块返回的错误代码

0x800404xx表示卡片协议模块返回的错误代码

5. 0x8005xxxx表示应用接口层返回的错误代码

0x800501xx表示API接口错误

0x800503xx表示通用读卡器接口错误



其它:预留

### 5.1.2 设备底层错误代码定义

#### FM8301 系列设备错误代码定义

1. HID 协议层状态返回值 0x800102xx, xx 见表格中 HID\_ERRFLAG 定义

名称 (name)		Sta	定 义 (description)
HID_ERRFLAG	USB_RIDERR	0x01	REPORT ID 错误错误
	USB_FRAMEERR	0x02	帧长度域错误
	FM222_BUFFERERR	0x03	数据缓冲区溢出
	FM222_OVERTIME	0x04	超时错误

2. APP 协议层状态返回值 0x800101xx, xx 见表格中 NEW\_ERRFLAG 定义

名称 (name)		Sta	定 义 (description)
NEW_ERRFLAG	FM222_OVERTIME	0x01	超时
	FM222_CLA_ERR	0x10	CLA 错误
	FM222_CLA_NOT	0x11	CLA 不支持
	FM222_LEN_ERR	0x12	长度错误
	FM222_INS_ERR	0x15	INS 错误
	FM222_PIPE_ERR	0x30	管道错误

3. 设备状态返回值 0x800140xx, xx 见表格中 NEW\_ERRFLAG\_DATA 定义



名称 (name)	DATA[0]	定 义 (description)	
NEW_ERFLAG_DATA	FM222_NOTAGERR	0x01	在有效区域内没有卡
	FM222_CRCERR	0x02	从卡中接收到了错误的 CRC 校验和
	FM222_NOEMPTY	0x03	值溢出
	FM222_AUTHERR	0x04	验证不成功
	FM222_PARITYERR	0x05	从卡中接收到了错误的校验位
	FM222_CODEERR	0x06	通信错误
	FM222_OVEERTIME	0x07	超时
	FM222_SERNRERR	0x08	在防冲突时读到了错误的串行码
	FM222_SELECTERR	0x09	卡片数据长度字节错误(SELECT 错误)
	FM222_NOTAUTHERR	0x0a	卡片没有通过验证
	FM222_BITCOUNTERR	0x0b	从卡片接收到的位数错误
	FM222_BYTECOUNTERR	0x0c	从卡片接收到的字节数错误
	FM222_RESTERR	0x0d	调用 restore 函数出错
	FM222_TRANSERR	0x0e	调用 transfer 函数出错
	FM222_WRITEERR	0x0f	调用 write 函数出错
FM222_INCRERR	0x10	调用 increment 函数出错	

FM222_DECRERR	0x11	调用 decrement 函数出错
FM222_READERR	0x12	调用 read 函数出错
FM222_LOADKEYERR	0x13	调用 loadkey 函数出错
FM222_FRAMINGERR	0x14	FM17 帧错误
FM222_REQERR	0x15	调用 req 函数出错
FM222_SELERR	0x16	调用 sel 函数出错
FM222_ANTICOLLERR	0x17	调用 anticoll 函数出错
FM222_INTIVALERR	0x18	调用初始化函数出错
FM222_READVALERR	0x19	调用高级读块值函数出错
FM222_DESELECTERR	0x1a	type b deselect 命令出错
FM222_BADPROTOCOL	0x1b	坏的协议
FM222_CMD_ERR	0x1c	命令错误
FM222_HARD_ERR	0x1d	硬件错误

### 5.1.3 API 错误代码定义

```
typedef enum
```

```
{
    FM_OK=0x00, //函数调用成功

    /******. 0x8003xxxx表示工具代码库返回的错误代码******/

    /*0x800301xx表示端口通讯工具代码库返回的错误代码

    FMPORTC_PORTTYPE_ERROR=0x80030101, //端口类型错误
```

上海复旦微电子集团股份有限公司  
Shanghai Fudan Microelectronics Group Company Limited

用户操作手册

```

FMPORTC_BUFFER_LACK=0x80030102, //缓存不足

FMPORTC_IOCTL_FAIL=0x80030103, //ioctl 执行失败

FMPORTC_SENMLEN_ERROR=0x80030104, //向端口发送数据长度错误

FMPORTC_PATA_ERROR=0x80030105, //参数错误

FMPORTC_PORT_NOTINIT=0x80030106, //端口未初始化

FMPORTC_TIMEOUTERR_ERROR=0x80030107, //从端口接收数据超时

FMPORTC_PORTPATA_ERROR=0x80030108, //端口参数无效或端口类型错误

FMPORTC_PORT_UNKNOWN_ERROR=0x80030109, //端口未知错误

FMPORTC_HANALE_INVALID=0x8003010a, //端口句柄无效

FMPORTC_SETSPEED_FAIL=0x8003010b, //波特率设置失败

FMPORTC_SETPARITY_FAIL=0x8003010c, //奇偶校验设置失败

FMPORTC_OPEN_FAIL=0x8003010d, //打开端口失败

FMPORTC_SELECT_FAIL=0x8003010e, //查询端口数据失败

FMPORTC_PATAINIT_ERROR=0x8003010f, //参数初始化错误或未进行初始化

FMPORTC_HIDWRITE_ERROR=0x80030110, //HID写数据失败

FMPORTC_HIDREAD_ERROR=0x80030111, //HID读数据失败

FMPORTC_UARTWRITE_ERROR=0x80030112, //UART写数据失败

FMPORTC_INTERFACE_NONSUPPORT=0x80030113, //接口不支持

FMPORTC_RFIDPOWER_FAIL=0x80030114, //RFID上下电失败

////////////////////////////////////

/*****0x8004xxxx表示驱动代码库返回的错误代码*****/

```

//0x800401xx表示设备驱动模块返回的错误代码

FMDRIVER\_PORTTYPE\_NONSUPPORT=0x80040100, //端口类型不支持

FMDRIVER\_PATA\_ILLEGAL=0x80040101, //参数非法

FMDRIVER\_DATALEN\_ILLEGAL=0x80040102, //数据长度非法

FMDRIVER\_UNKNOWN\_ERROR=0x80040103, //未知错误

FMDRIVER\_SLOT\_NONSUPPORT=0x80040104, //卡槽不支持

FMDRIVER\_CARDPRO\_NONSUPPORT=0x80040105, //卡协议类型不支持

FMDRIVER\_RECVLEN\_ERROR=0x80040106, //接收长度错误

FMDRIVER\_PURSEFORMAT\_ERROR=0x80040107, //钱包格式错误

FMDRIVER\_SELPIPE\_FAIL=0x80040108, //卡槽选择失败

FMDRIVER\_FUN\_NONSUPPORT=0x80040109, //功能不支持

FMDRIVER\_DEVTYPE\_NONSUPPORT=0x8004010a, //设备类型不支持

FMDRIVER\_CARDTYPE\_ERROR=0x8004010b, //卡类型错误

FMDRIVER\_SYSTEM\_ERROR=0x8004010c, //系统错误

FMDRIVER\_CARDCHANNEL\_NOTFIND=0x8004010d, //当前卡通道未找到

FMDRIVER\_HIDDEVICE\_NOFIND=0x8004010e, //HID设备未找到

FMDRIVER\_FIELDONLEN\_ERROR=0x8004010f, //FIELDON应答长度错误

FMDRIVER\_UID\_NONSUPPORT=0x80040110, //UID级别不支持

FMDRIVER\_RATSLEN\_ERROR=0x80040111, //RATS应答长度错误

FMDRIVER\_PORT\_INITED=0x80040112, //端口已初始化

FMDRIVER\_PORT\_NOTINIT=0x80040113, //端口未初始化

FMDRIVER\_PORT\_NOTFINDED=0x80040114, //没有找到指定序号的端口

FMDRIVER\_COLLISION\_ERROR=0x80040115, //选卡冲突

//0x800402xx表示设备协议模块返回的错误代码

FMDEVICEPRO\_SYSTEM\_ERROR=0x80040200, //系统错误

FMDEVICEPRO\_FORMAT\_ERROR=0x80040201, //数据帧格式错误

FMDEVICEPRO\_PORT\_NONSUPPORT=0x80040202, //端口类型不支持

FMDEVICEPRO\_DEVTYPE\_NONSUPPORT=0x80040203, //设备类型不支持

FMDEVICEPRO\_FUN\_NONSUPPORT=0x80040204, //功能不支持

FMDEVICEPRO\_HANDSHAKE\_FAIL=0x80040205, //与设备握手失败

FMDEVICEPRO\_DEVPRO\_ERROR=0x80040206, //设备协议不存在

FMDEVICEPRO\_TIMEOUT\_ERROR=0x80040207, //超时错误

FMDEVICEPRO\_TRAN\_NONSUPPORT=0x80040208, //传输类型不支持

FMDEVICEPRO\_FRAME\_NONSUPPORT=0x80040209, //帧类型不支持

FMDEVICEPRO\_INPUTDATALEN\_ERROR=0x8004020a, //输入数据长度错误

FMDEVICEPRO\_RECVDATALEN\_ERROR=0x8004020b, //接收数据长度错误

FMDEVICEPRO\_COMMSN\_ERROR=0x8004020c, //通讯系列号错误

FMDEVICEPRO\_CTRL\_ERROR=0x8004020d, //协议控制字(帧类型)错误

FMDEVICEPRO\_MAXLINK\_ERROR=0x8004020e, //超出最大帧链接数

//0x800403xx表示通讯协议模块返回的错误代码

FMCOMMPRO\_BUFFER\_LACK=0x80040300, //缓存不足

FMCOMMPRO\_NOT\_SUPPORT=0x80040301, //暂不支持

FMCOMMPRO\_CHECKLEN\_ERROR=0x80040302, //数据校验长度错误

FMCOMMPRO\_REPORTID\_ERROR=0x80040303, //数据帧格式错误

FMCOMMPRO\_VER\_ERROR=0x80040304, //协议版本错误

FMCOMMPRO\_PATA\_ERROR=0x80040305, //参数错误

FMCOMMPRO\_COMMSN\_ERROR=0x80040306, //通讯系列号错误

FMCOMMPRO\_DATALEN\_INVALID=0x80040307, //数据长度无效

FMCOMMPRO\_FORMAT\_ERROR=0x80040308, //数据帧格式错误

FMCOMMPRO\_SYSTEM\_ERROR=0x80040309, //系统错误

FMCOMMPRO\_FUN\_NONSUPPORT=0x8004030a, //功能不支持

FMCOMMPRO\_BCC\_FAIL=0x8004030b, //BCC校验失败

FMCOMMPRO\_CRC\_FAIL=0x8004030c, //CRC校验失败

//0x800404xx表示卡片协议模块返回的错误代码

FMCARDPRO\_PATA\_ERROR=0x80040400, //参数错误

FMCARDPRO\_SYSTEM\_ERROR=0x80040401, //系统错误

FMCARDPRO\_UNKNOWN\_ERROR=0x80040402, //未知错误

FMCARDPRO\_BUFFER\_LACK=0x80040403, //缓存不足

FMCARDPRO\_DATALEN\_ILLEGAL=0x80040404, //数据长度非法

FMCARDPRO\_TIMEOUT\_ERROR=0x80040407, //超时错误

////////////////////////////////////

/\*\*\*\*\*0x8005xxxx表示应用接口层返回的错误代码\*\*\*\*\*/

//0x800501xx表示API接口错误

FMAPI_GETFACTORY_ERROR=0x80050101,	// 获取产类失败
FMAPI_GETREADER_ERROR=0x80050102,	// 获取读卡器实例失败
FMAPI_INVALID_HANDLE=0x80050103,	// 无效Handle
FMAPI_COMM_ERROR=0x80050104,	// 通讯错误
FMAPI_BUFFER_LACK=0x80050105,	// 缓冲不足
FMAPI_SLOT_ERROR=0x80050106,	// 卡槽参数错误
FMAPI_PATA_ERROR=0x80050107,	// 参数错误
FMAPI_SYSTEM_ERROR=0x80050108,	// 系统错误
FMAPI_HANDLE_INVALID=0x80050109,	// 句柄无效
FMAPI_PORT_NOTINIT=0x8005010a,	// 端口未初始化
FMAPI_DIRECTORY_ERROR=0x8005010b,	// 目录长度超出范围
FMAPI_HIDDEVICE_NOFIND=0x8005010c,	// 未找到HID设备
FMAPI_TTYLEN_ERROR=0x8005010d,	// 串口描述符长度超出范围(linux)
FMAPI_SENLEN_INVALID=0x8005010e,	// 发送长度无效
FMAPI_SNLEN_INVALID=0x8005010f,	// SN长度无效
//0x800503xx表示通用读卡器接口错误	
FMRH_BUFFER_LACK=0x80050301,	// 缓存不足
FMRH_LOAD_FAIL=0x80050302,	// DLL加载失败
FMRH_NOT_SUPPORT=0x80050303,	// 暂不支持
FMRH_READERTYPE_NONSUPPORT=0x80050304,	// 读卡器类型不支持
FMRH_SNLEN_SPILL=0x80050305,	// SN长度超出范围





```
FMRH_HANDLE_ERROR=0x80050306,           // 输入句柄错误
FMRH_NOT_OPENED=0x80050307,             // 设备未被打开过
FMRH_PATA_ERROR=0x80050308,            // 参数错误
FMRH_API_ERROR=0x80050309,            // 找不到API方法
FMRH_NO_DEVICE=0x8005030a,            // 未找到设备
FMRH_NONSUPPORT=0x8005030b,          // 功能不支持
}FM_ERROR_CODE;
```