

# Mifare 流程操作说明书

**Version 5.0**

文档版本号: **1.0.7**

RF-UIM 卡

直通电讯有限公司

2008 年 12 月

©2008 Directel corporation. All rights reserved

## 目录

1、应用说明.....	3
1.1、通用应用流程.....	3
1.1.1. 打开 RF.....	4
1.1.2. 询卡.....	4
1.1.3. 断开 RF.....	4
1.2、Logic Mifare 操作流程.....	5
1.2.1、选择 Accounter 应用 (Select) .....	6
1.2.2、Request 寻卡.....	6
1.2.3、Auth 认证.....	6
1.2.4、块操作.....	7
1.2.5、其它.....	8
1.2.6、关闭 (Halt) .....	9
1.3、电子钱包 接口.....	9
1.3.1、选择 Accounter 应用 (Select) .....	9
1.3.2、RF 寻卡.....	10
1.3.3、MifareOne 和 2.4G 的 Mifare 开发说明 .....	11
1.3.4、Mifare cpu 和其他类型卡开发说明 .....	11
(1)、发送 RATS 指令 .....	11
(2)、发送相应卡的标准 Adu 指令 (Tranceive) .....	11
(3)、取消选卡 (Deselect) .....	11
1.4、操作 PSAM 卡.....	12
2、其它应用命令.....	13
3、演示程序.....	13
4、常见问题.....	17
附录 A. 编制历史.....	19

# 1、应用说明

RF-UIM 卡支持多种应用，如 Logic Mifare，ID 卡应用(可定义 ID 号)，RFID 号应用(卡号)，PBOC2.0 应用，磁条卡应用等等。每一种应用的操作模式上稍微有些区别。应用有些指令是通过 RF-POS 加工之后发给 RF-UIM 卡，有些指令是纯粹的通过 RF-POS 来中转。指令收发格式请参阅《3. Mifare 指令集》的第 01 章“帧类别及命令收发”。

## 1.1、通用应用流程

适合应用：ID 卡应用(可定义 ID 号)，RFID 号应用(卡号)，PBOC2.0 应用，磁条卡应用。操作流程请参照图 1。

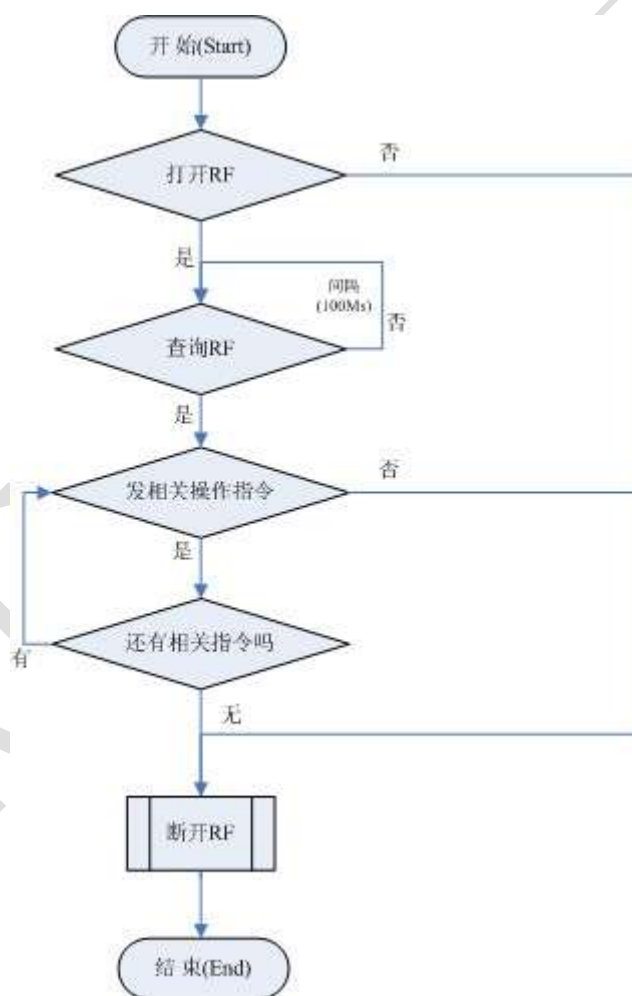


图 1

**特别说明：Logic Mifare 不适合本流程，Mifare 请参考 1.2 操作流程。**

### 1.1.1. 打开 RF

发送指令：80 05 90 B0 01 00 00

返回

成功：90 02 90 00

失败：其它错误

说明：该指令返回成功后，Reader 开始寻卡，进入等待刷卡状态。

### 1.1.2. 询卡

#### 1. 查询式询卡。

打开 RF 功能后，通过定时发送询卡指令去查询是否有卡连接上。

发送指令：80 05 90 B0 04 00 00

卡已连上：90 02 9C 02

未连上：90 02 9C 03

说明：建议上位机大于或等于 100 毫秒查询一次。

#### 2. 读卡 ID。查询到连接成功后，可通过指令取卡 ID 值：

发送：80 05 90 B0 05 00 00

成功：90 42 8 字节 TermID 值，8 字节随机数，40 字节保留值，8 字节卡 ID 值，90 00

失败：其它错误

### 1.1.3. 断开 RF

断开本次 RF 连接或关闭 Reader 寻卡功能

发送指令：80 05 90 B0 00 00 00

返回：

成功：90 02 90 00

失败：其它错误

## 1.2、Logic Mifare 操作流程

具体电子钱包指令收发格式请参阅文档《3. Mifare 指令集》的第 5 章“Mifare 接口”。在读卡器应用已选择 Mifare 接口,且读卡器和卡已正确初始化的情况下,可直接执行 Mifare 核心命令如下。

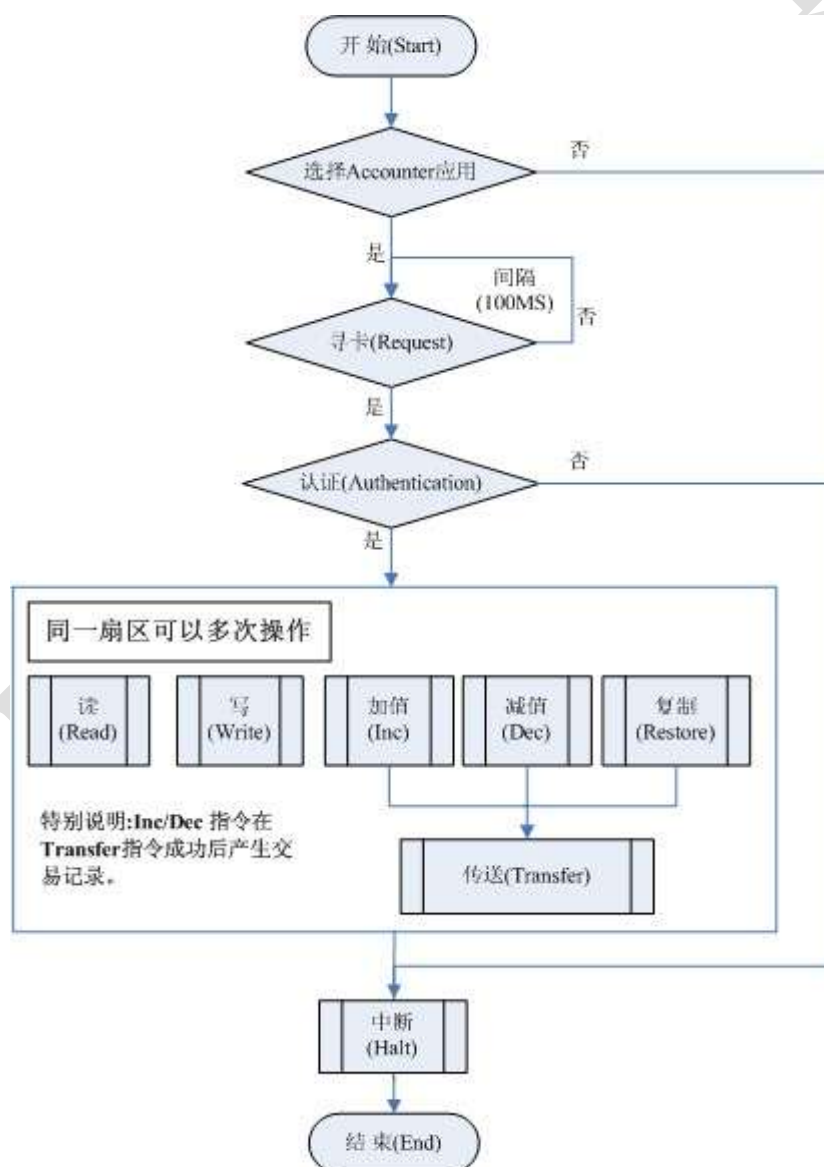
特别注意:

1)、执行 Logic Mifare 应用时,请使用 Request 来寻卡,断开 RF 请使用 Halt 指令。本流程为基本流程,每次操作都需要按照这个流程执行。

2)、默认的电子钱包卡头设置了交易记录存放在第 10 扇区之后的连续 3 个扇区(10, 11, 12)里,总共 9 笔交易记录(调用 Transfer 指令之后自动产生)。

3)、出厂时默认电子钱包放在 1 扇区 0 块(系统集成商可以删除掉)。

4)、建设部标准的 Mifare 和各省规范的 Mifare(2, 3)两条不适合。



Logic Mifare 应用典型流程图

### 1.2.1、选择 Accounter 应用 (Select)

发送: 80 15 A0 A4 04 00 10 (Accounter App) 41 63 63 6F 75 6E 74 65 72  
20 41 70 70 00 00 00

返回:

成功: 90 02 90 00

失败: 其它错误

说明:

### 1.2.2、Request 寻卡

发送: 80 05 40 0A 00 00 00

返回:

电信集团规范:

成功: 90 28 Data 90 00

失败: 90 02 9C 00

Data:

内容	长度 (byte)	备注
卡号	4	Mifare 卡号
黑名单	1	公共信息区的黑名单标志
保留	1	
发行数据	32	发行区 0、1 块数据

普通的卡:

成功: 90 12 (0 扇区 0 块的 16 字节数据) 90 00

失败: 90 02 9C 00

说明: 如果返回 0x9C03, 那么需要延时 100MS 再重发寻卡指令。

注明: 只有读头版本是 05.05.19 (包括 05.05.19) 以后的版本才会有这种区别, 否则全部返回 (90 12 (0 扇区 0 块的 16 字节数据) 90 00) 的数据

读卡 ID。查询到连接成功后, 可通过指令取卡 ID 值:

发送: 80 05 90 B0 05 00 00

成功: 90 42 8 字节【RF-POS】ID 值+ 8 字节随机数+40 字节保留值+8 字节【RF-UIM 卡】ID 值, 90 00

失败: 其它错误

### 1.2.3、Auth 认证

A. 用读卡器内部密钥认证

发送: 80 05 40 08 01 (扇区号) 00 (00:B 密钥, 01:A 密钥) 00 (用读卡器内部密钥)

返回:

成功: 90 02 90 00

失败: 90 02 9A 07 (认证失败) 或其它错误

B. 用命令所带密钥认证

发送: 80 0B 40 08 01 (扇区号) 00 (00:B 密钥, 01:A 密钥) 06 FF  
FF FF FF FF (具体的密钥值)

返回:

成功: 90 02 90 00

失败: 90 02 9A 07 (认证失败) 或其它错误

## 1.2.4、块操作

### 1. Read 块

发送: 80 05 40 02 01 00 (块号) 00

返回:

成功: 90 12 ReturnData (16byte) 90 00

失败: 其他状态字

ReturnData 如下表

内容	长度 (byte)	备注
所读块号数据	16	相对扇区块号

### 2. Write 块

发送: 80 15 40 02 02 00 (块号) 10 00 00 00 00 FF FF FF FF 00 00 00 00  
00 FF 00 FF (16 个字节的块数据)

返回:

成功: 90 02 90 00

失败: 其它错误

## 钱包块格式说明:

(只有按钱包格式的值, 才能正确的冲值、消费和通过 stk 显示);

1、钱包格式表示一个带符号4 字节值, 钱包格式的最低一个字节保存在最低的地址中, 接下去以取反的字节保存第二次, 在一次带符号4 字节值保存第三次; 为了保证数据的正确性和保密性值被保存了3 次, 两次不取反保存, 一次取反保存;

2、Adr1 Adr2 Adr3 Adr4 默认为00 FF 00 FF, 如果有需要, 可以自行存储相应的数值;

3、一个扇区建议只建一个钱包; 除钱包块为外还有钱包名称, 钱包名称必须放在钱包块的下一块(或者说UTK自动会寻找钱包块的下一块是否满足钱包名称的格式要求); 如果没有写钱包名称, 默认为(钱包+序号); 钱包名称可以参考<Mifare指令集 05.15. >

4、建钱包用写块的方法, 可以参考以1、2两点的说明进行写块操作就可以, 如:

80 15 40 02 02 00 (块号) 10 00 00 00 00 (带符号4 字节值) FF FF FF FF  
(带符号4 字节值取反) 00 00 00 00 (带符号4 字节值) 00 FF 00 FF (16

个字节的块数据)

以上就是在认证扇区相应的块0，建一个钱包，金额为0；

5、删除钱包用写块的方法，把16字节的数据全部替换成00，就可以了；

如：

80 15 40 02 02 00(块号) 10 00 00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00

以上就把认证扇区相应的钱包块删除；

Byte Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Description	Value				Value				Value				Adr	Adr	Adr	Adr

6、默认控制块（每个扇区的最后一块）的数据为 FF FF FF FF FF FF(A 密钥) FF 07 80 69(控制字) FF FF FF FF FF FF(B 密钥)，可以修改。

### 1.2.5、其它

1. Inc 增值(请参考 3.Mifare 指令集 增值部分)

发送：80 15 40 02 03 00(块号) 10 64 00 00 00(金额)+12 字节其它内容

“**HEX 格式，低字节在前**”

返回：

成功：90 02 90 00

失败：其它错误

**说明：**增值完需要 (Transfer 传送)，才能真正实现增值。

2. Dec 减值(请参考 3.Mifare 指令集 减值部分)

发送：80 15 40 02 04 00(块号) 10 64 00 00 00(金额)+12 字节其它内容

“**HEX 格式，低字节在前**”

返回：

成功：90 02 90 00

失败：其它错误

**说明：**减值完需要 (Transfer 传送)，才能真正实现减值。

3. Restore 复制(只能在钱包块之间使用)

发送：80 05 40 02 05 00(块号) 00

返回：

成功：90 02 90 00

失败：其它错误

**说明：**本条指令可以实现同一扇区的不同块进行数据拷贝，从而实现了备份的功能。

例如：从块 0 备份到块 2；

80 05 40 02 05 00(块号) 00(拷贝块 0)

80 05 40 02 06 02(块号) 00(复制到块 2)

4. Transfer 传送

带时间，用于交易记录，共 7 字节 BCD 码，依次为秒，分，时，星期(未用)，日，月，年(减 08)

发送：80 0C 40 02 06 00(块号) 07 30 58 10 XX 01 09 00 (命令中时



间为 2008-9-1 10:58:30)

返回:

成功: 90 12 Block\_Data(16byte) (传送块号的数据) 90 00

失败: 其它错误

说明: 增值、减值、复制完需要 (Transfer 传送)。

### 1.2.6、关闭 (Halt)

断开本次 RF 连接或关闭 Reader 寻卡功功能

发送: 80 05 40 04 80 00 00

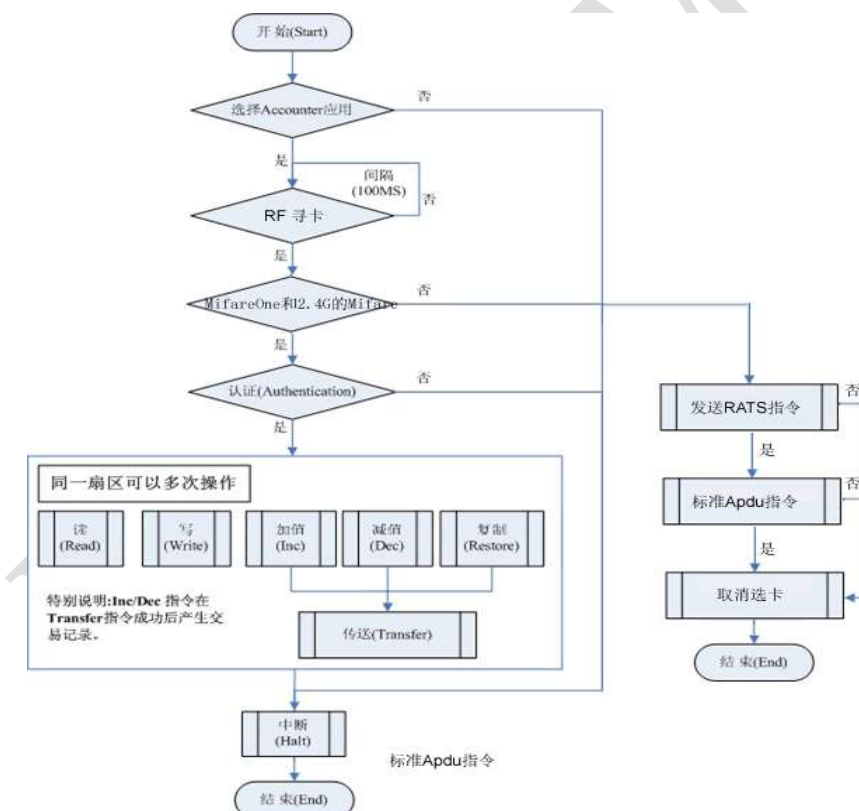
返回:

成功: 90 02 90 00

失败: 其它错误

## 1.3、电子钱包 接口

说明: 这些指令只有在盛华双频读头, 型号(SHRM202)的读头上, 才可以正确的执行。



### 1.3.1、选择 Accounter 应用 (Select)

发送: 80 15 A0 A4 04 00 10(Accounter App) 41 63 63 6F 75 6E 74 65 72  
20 41 70 70 00 00 00

返回:

成功: 90 02 90 00

失败: 其它错误

说明:

### 1.3.2、RF 寻卡

发送: 80 05 90 B0 04 XX 00

返回:

成功: 90 0C ReturnData(见下面说明) 9C 02 (状态字)

失败: 90 02 9C 00

XX: 对于双频读头可以用来设置询卡方式:

0x52: 寻卡模式要求操作的卡在读写完成后卡片只需关闭后又可读写

0x26: 寻卡模式要求操作的卡在读写完成后要离开感应区才能再读写

说明: **如果返回 0x9C03, 那么需要延时 100MS 再重发寻卡指令。**

读卡 ID。查询到连接成功后, 也可通过该指令取卡 ID 值:

发送: 80 05 90 B0 05 00 00

成功: 90 42 8 字节【RF-POS】ID 值+ 8 字节随机数+40 字节保留值+8 字节【RF-UIM 卡】ID 值, 90 00

失败: 其它错误

说明: **本条指令只支持 2.4G 卡; 读取相应的连接参数。**

ReturnData 如下表:

用户可以通过第 3 个字节高 4 位来判读是 2.4G 或者 Mifare, 如果是 Mifare, 可以通过第 10 个字节来判断是 MifareOne 或者 Mifare cpu;

1、2.4G 卡:

内容	长度 (byte)	备注
状态字	2	这个状态字是兼容老的直接读第一个返回数据的程序
卡类型和数据长度	1	通道类型: 高 4bit = 1; 低 4bit = 9; //8 个字节的卡号+通道本身长度 1
卡号 (UID)	8	8 字节【RF-UIM 卡】ID 值

2、其他卡片

内容	长度 (byte)	备注
状态字	2	这个状态字是兼容老的直接读第一个返回数据的程序
卡类型和数据长度	1	通道类型: 高 4bit = 4: //Mifare (iso1443a)(读头上必须 13.56M 的硬件) 低 4bit = 5: 表示卡号字节数+通道本身长度 1:
卡号 (UID)	4	4 字节【Mifare 卡】ID 值
ATQA	2	卡类型 0x0400 = Mifare_One(Mifare One S50) 0x0200 = Mifare_One(S70) 0x0800 = Mifare_Pro(X) 0x4400 = Mifare_UltraLight 0x4403 = Mifare_DESFire <b>这个数据由 Mifare 卡给出, 请参见 Mifare 卡的芯片资料</b>

SAK	1	08: MifareOne, 28: Mifare cpu 卡 这个数据也是 Mifare 卡给的, 请参见 Mifare 卡的芯片资料
-----	---	--

### 1.3.3、MifareOne 和 2.4G 的 Mifare 开发说明

对于 MifareOne 和 2.4G 的 Mifare 开发流程可以参考

<1.2、Logic Mifare 操作流程——1.2.3、Auth 认证>流程说明书;

### 1.3.4、Mifare cpu 和其他类型卡开发说明

#### (1)、发送 RATS 指令

发送: c8 05 40 50 00 00 00

返回:

成功: 90 02+(卡的响应数据长度) (卡的响应数据) + 90 00

失败: 90 02 9A 12 (执行超时没有得到响应)

说明: RATS 指令只能发一次, 不能重复发, 直到取消选卡(Deselect)完, 然后从第一步开始往下执行, 方可;

#### (2)、发送相应卡的标准 Adu 指令 (Tranceive)

发送: c8 XX (数据总长度) 40 53 00 00 XX (Adu 指令长度) XX XX XX XX  
XX XX ..... (Adu 指令数据)

返回:

成功: 90 02+(卡的响应数据长度) (卡的响应数据) + 90 00

失败: 90 02 9A 12 (执行超时没有得到响应)

说明: 此指令可以做为相应卡的指令的转发通道;

#### (3)、取消选卡 (Deselect)

发送: c8 05 40 51 00 00 00

返回:

成功: 90 02 90 00

失败: 90 02 9A 12 (执行超时没有得到响应)

说明: 执行完卡的相关操作, 必须调用这条指令, 才可以重新寻卡。

## 1.4、操作 PSAM 卡

说明：本部分属于 SHR600 型读卡器，外接 PSAM 卡的使用说明

1)、先发 ATR 复位：E0 03 41 54 52 (ATR) 注：ATR 大写的 ASCII 码，PSAM 槽不支持热插拔。

2)、然后按<3.Mifare 指令集.doc>这份文档第 3 页的发送数据格式给读头 PSAM 卡的指令，

## 2、其它应用命令

请参考<<3. Mifare 指令集>>.

## 3、演示程序

以下是 pc 程序的消费程序，供参考：

fnTransmit 说明：

本函数主要是往串口发送数据，同时接收返回的数据和长度。

// Value2BCD 函数说明

```
unsigned char Value2BCD(int nTemp)
```

```
{
```

```
    unsigned char nRet;
```

```
    nRet=((nTemp/10)<<4)+((nTemp%10)&0x0F);
```

```
    return nRet;
```

```
}
```

```
#include <time.h>
```

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
    unsigned char RevBuff[255]; //数据接收包
```

```
    unsigned char SendBuff[264]; //数据发送包
```

```
    unsigned long RevLen; //接收长度
```

```
    unsigned char sCurCardID[8]/* 【 RF-UIM 卡 】 ID*/, sCurTermID[8]/*
```

```
    【 RF-POS 】 ID*/;
```

```
    int nResult;
```

```
int iCurDelay=30;//可以自己根据实际情况，重新设置寻卡次数
bool bConnect =false;
int nBlock, nSectorCs; //电子钱包所在扇区号，块号；

unsigned int nValue; //消费金额,以分为单位
unsigned int nMoney; //最后余额
unsigned short iyear, imonth, iday, ihour, imin, isecnd, iweek;
unsigned char DayInfo[7];

time_t timer;
struct tm *tblock;

nValue = 100;//100 分

//默认电子钱包在 1 扇区 0 块;
nSectorCs = 1;
nBlock    = 0;

char sTemp[255];

//打开串口，自行修改
nResult =fnOpenUart("Com6");
if (nResult !=0x9000)
    goto Exit;

//1、...设置读头为电子钱包应用
memset(RevBuff, 0x00, sizeof(RevBuff));
memset(SendBuff, 0x00, sizeof(SendBuff));
//指令头
memcpy(SendBuff, "\x80\x15\xA0\xA4\x04\x00\x10", 7);
//数据
memcpy(SendBuff+7, "Accounter App", sizeof("Accounter App"));
nResult = fnTransmit(SendBuff, RevBuff, RevLen);
if (nResult !=0x9000)
    goto Exit;

//2、...连接 RF-SIM 卡
for(int i=0; i<iCurDelay; i++)
{
//....寻卡
    memset(RevBuff, 0x00, sizeof(RevBuff));
    memset(SendBuff, 0x00, sizeof(SendBuff));
    //指令头
```

```
memcpy(SendBuff, "\x80\x05\x40\x0A\x00\x00\x00", 7);
nResult = fnTransmit(SendBuff, RevBuff, RevLen);
if ( nResult == 0x9A03) /*串口通信出错，此状态字，根据各自的串口通信
出错状态字修改*/
{
    break;
}
else if( nResult == 0x9000)
{
    bConnect =true;
    memset(sCurCardID, 0, sizeof(sCurCardID));
    memset(sCurTermID, 0, sizeof(sCurTermID));
//....取卡号
    memset(RevBuff, 0x00, sizeof(RevBuff));
    memset(SendBuff, 0x00, sizeof(SendBuff));
    //指令头
    memcpy(SendBuff, "\x80\x05\x90\xB0\x05\x00\x00", 7);
    fnTransmit(SendBuff, RevBuff, RevLen);

    memcpy(sCurCardID, RevBuff+2+8+8+40, 8);
    memcpy(sCurTermID, RevBuff+2, 8);
    break;
}
Sleep(100); //如果返回 0x9C03，那么需要延时 100MS 再重发寻卡指令
}

if(!bConnect)
    goto Exit;

//3、....认证
memset(RevBuff, 0x00, sizeof(RevBuff));
memset(SendBuff, 0x00, sizeof(SendBuff));
//、指令头
memcpy(SendBuff, "\x80\x05\x40\x08\x01\x00\x00", 7);
//、扇区
SendBuff[4] = nSectorCs; //根据你钱包所在的块号，自行设置，默认电子钱包
放在 0
if(fnTransmit(SendBuff, RevBuff, RevLen) != 0x9000) //认证的扇区
{
    sprintf(sTemp, "电子钱包认证失败！");
    goto Exit;
}
```

```
//4、...消费
memset(RevBuff, 0x00, sizeof(RevBuff));
memset(SendBuff, 0x00, sizeof(SendBuff));
//指令头
memcpy(SendBuff, "\x80\x15\x40\x02\x04\x00\x10", 7);
//块号
SendBuff[5] = nBlock;
//指令数据，以下数据可参考指令集の説明
//金额
memcpy(SendBuff+7, &nValue, 4);
// 交易时间
/* gets time of day */
timer = time(NULL);
/* converts date/time to a structure */
tblock = localtime(&timer);
memset(DayInfo, 0x00, sizeof(DayInfo));
DayInfo[0] = Value2BCD(tblock->tm_mday);
DayInfo[1] = Value2BCD(tblock->tm_hour);
DayInfo[2] = Value2BCD(tblock->tm_min);
DayInfo[3] = Value2BCD(tblock->tm_sec);
memcpy(&SendBuff[7+4], DayInfo, 4);
//aTermid
memcpy(&SendBuff[7+4+4], &sCurTermID[4], 4);
SendBuff[7+4+4+4] = 0x01;
SendBuff[7+4+4+4+1] = 0x01;
nResult = fnTransmit(SendBuff, RevBuff, RevLen);
if (nResult != 0x9000)
{
    sprintf(sTemp, "电子钱包消费失败！");
    goto Exit;
}

//5、...传送
memset(RevBuff, 0x00, sizeof(RevBuff));
memset(SendBuff, 0x00, sizeof(SendBuff));
//指令头
memcpy(SendBuff, "\x80\x0C\x40\x02\x06\x00\x07", 7);
//块号
SendBuff[5] = nBlock;

//指令数据
/* gets time of day */
timer = time(NULL);
```



```
/* converts date/time to a structure */
tblock = localtime(&timer);
iyear = tblock->tm_year + 1900 - 2008;
imonth = tblock->tm_mon + 1; /* 月份（从一月开始，0 代表一月） - 取值区间
                               为[0,11], 所以正常显示需加 1 */

iweek = tblock->tm_wday + 1;
iday = tblock->tm_mday;
ihour = tblock->tm_hour;
imin = tblock->tm_min;
isecond = tblock->tm_sec;
memset(DayInfo, 0x00, sizeof(DayInfo));
sprintf(DayInfo, "%.2d%.2d%.2d%.2d%.2d%.2d%.2d", isecond, imin, ihour,
iweek, iday, imonth, iyear);
ToBCD(DayInfo, SendBuff+7);
nResult = fnTransmit(SendBuff, RevBuff, RevLen);
if (nResult!=0x9000)
{
    sprintf(sTemp, "电子钱包传递失败！");
    goto Exit;
}
memcpy(&nMoney, RevBuff+2, 4);
Exit:
//6....最后关闭 RF
memset(RevBuff, 0x00, sizeof(RevBuff));
memset(SendBuff, 0x00, sizeof(SendBuff));
//、指令头
memcpy(SendBuff, "\x80\x05\x40\x04\x80\x00\x00", 7);
fnTransmit(SendBuff, RevBuff, RevLen);
//关闭串口
fnCloseUart();
```

## 4、常见问题

返回状态	常见情况
6E 00	1. 如果超出 5 秒种没有向读头发指令，RF 会自动断掉了；因此在发任何操作卡的指令，会出现这种情况；
	2. 没有选择应用，而直接发任何操作卡的指令，也会出现这种情况；
	3. 刷卡的时候一定要用手机的背面(电池那一面)去靠近读卡器，因为手机的设计原理是正面基本上屏蔽信号。

	<p>4. 手机电量超低时可能无法刷卡。</p>
	<p>5. 如不能正常刷卡，请进入手机菜单调整。菜单位置在<b>天翼-&gt;UIM 卡应用-&gt;RF-UIM 卡</b>中的<b>【设置】-&gt;【基本设置】-&gt;【距离】-&gt;【近距离】-&gt;【灵敏度】</b>和<b>【校正】</b>（进入菜单需要输入 PIN：默认为 1234 ）查看菜单<b>【灵敏度】</b>是否为 1，<b>【校正】</b>是否为 2，若不是请调整，一般可正常刷卡。建议缺省的<b>【近距离】</b>选项参数为<b>【灵敏度：1】</b>，<b>【校正：2】</b>，请谨慎修改近距离参数。一定不能直接将参数设为最大值，以免刷卡距离过大导致误刷卡。</p> <p>5.1 若以上设置如不能正常刷卡请调整灵敏度和校正，方法如下：</p> <p>5.1. a. 先固定<b>【灵敏度】</b>当前菜单下参数不变，把<b>【校正】</b>菜单距离参数调弱或调强一个级别，然后再测试距离是否满意，否则继续调节校正的参数，直到参数值使用完。</p> <p><b>校正：</b>包含 0（稍弱），1（弱），2（弱强），3（强），4（最强）</p> <p>5.1. b. 如果第一步的校正参数使用完，还是不满意，就把灵敏度参数调弱或调强一个级别，然后重复<b>【5.1. a】</b>的操作步骤，直到刷卡距离满意为止。</p> <p><b>灵敏度：</b>0（稍弱），1（弱），2（弱强），3（强）四个选项。选择其中一项就表示确认。选项前出现“*”，表示此选项功能被开启。</p>

## 附录 A. 编制历史

版本号	更新时间	修改人	主要内容或重大修改
V1.0.0	2008-01-24	simon	编制规范
V1.0.1	2008-02-26	simon	修改
V1.0.2	2008-05-06	xicy	修改
V1.0.3	2008-07-16	xicy	修改
V1.0.4	2009-05-06	huangzn	修改
V1.0.5	2009-12-05	huhq	修改
V1.0.6	2009-12-09	huhq	修改
V1.0.7	2010-03-18	huangzn	修改