ER301应用编程接口库函数使用说明

1. S50卡的数据结构

S50卡有1k字节（ bytes），也就是8K比特（ bits）。

共16个扇区，每个扇区有4个块，其中第0扇区第0块保存卡序列号以及厂家信息，是只读的，不能写。每个扇区共有4个块，分别是块0，块1，块2，块3。其中，密钥存放在每个扇区的块3。按照绝对块号算的话，16个扇区共有64个块，块号是从0到63逐渐递增。

算存储密钥块的算法是：x=s*4+3;　其中s表示扇区号（0－15）。

下面是绝对块号在不同扇区内的对应图示。

扇区 0	块 0			数据块	0
	块 1			数据块	1
	块 2			数据块	2
	块 3	密码 A	存取控制 密码 B	控制块	3
扇区 1	块 0			数据块	4
	块 1			数据块	5
	块 2			数据块	6
	块 3	密码 A	存取控制 密码 B	控制块	7
		⋮			
扇区 15	0			数据块	60
	1			数据块	61
	2			数据块	62
	3	密码 A	存取控制 密码 B	控制块	63

关于S50卡的进一步介绍，请详细阅读“Mifare1_S50卡中文说明书”。

2. 库函数说明

应用编程接口包含2个DLL动态库文件：

1、MasterRd.dll ：应用程序接口库，后面要介绍的所有函数都包含在这个库文件里面。

2、MasterCom.dll：串口通讯库，由MasterRd调用。

编程时,需要把这2个dll文件包含进所在目录。

[IN]：表示输入

[OUT]：表示输出
在用于ER301这个型号的读卡器时，icdev一律取值为0。

3.1 系统函数

3.1.1 INT WINAPI LIB_VER

功能：获取动态库版本号
原型：int WINAPI lib_ver(unsigned int *pVer)
参数：pVer：[OUT] 动态库版本号
返回：成功返回0

3.1.2 INT WINAPI RF_INIT_COM

功能：初始化串口
原型：int WINAPI rf_init_com (int port, long baud)
参数：port： [IN] 串口号
baud：[IN] 为通讯波特率9600～115200 ，默认为115200bps
返回：成功返回0

3.1.3 INT WINAPI RF_CLOSEPORT

功能：关闭Com端口
原型：int WINAPI rf_ClosePort()
返回：成功返回0

3.1.4 INT WINAPI RF_GET_MODEL

功能：读取读写卡器型号及产品批号
原型：int WINAPI rf_get_model(unsigned short icdev,
unsigned char *pVersion,
unsigned char *pLen)
参数：icdev： [IN] 通讯设备标识符
pVersion： [OUT] 返回的信息
pLen: [OUT] 返回信息的长度
返回：成功返回0

3.1.5 INT WINAPI RF_INIT_DEVICE_NUMBER

功能：指定设备标识
原型：int WINAPI rf_init_device_number(unsigned short icdev)
参数：icdev： [IN] 通讯设备标识符
返回：成功返回0

说明：读卡器只响应设备标识符与本身相符或
　　设备标识符等于0x0000的指令。

3.1.6 INT WINAPI RF_GET_DEVICE_NUMBER

功能：读取设备标识
原型：int WINAPI rf_get_device_number(unsigned short *pIcdev)
参数：pIcdev：[OUT] 返回通讯设备标识符
返回：成功返回0
说明：如果一个串口同时联有两台以上读卡器，不可使用该命令

3.1.7 INT WINAPI RF_INIT_TYPE

功能：设置读写卡器非接触工作方式
原型：int WINAPI rf_init_type(unsigned short icdev, unsigned char type)
参数：icdev：　[IN] 通讯设备标识符
　　type：　[IN] 读写卡器工作方式
返回：成功返回0
说明：只支持单一协议的读卡器此函数无效
　　type = 'A'：设置为TYPE_A方式
　　type = 'B'：设置为TYPE_B方式
　　type = 'r'：设置为AT88RF020卡方式
　　type = '1'：设置为ISO15693方式 （123的1）

注：ER301型号的读卡器不需要用到这个函数。

3.1.8 INT WINAPI RF_ANTENNA_STA

功能：设置读写卡器天线状态
原型：int WINAPI rf_antenna_sta(unsigned short icdev, unsigned char model)
参数：icdev：　[IN] 通讯设备标识符
　　model：　[IN] 天线状态
返回：成功返回0
说明：model = 0：关闭天线
　　model = 1：开启天线
关闭天线可以节约功耗，或者也可以通过关闭天线后开启天线来激活放在读卡器上并且被休眠的卡。

3.1.9 INT WINAPI RF_LIGHT

功能：设置LED指示灯颜色

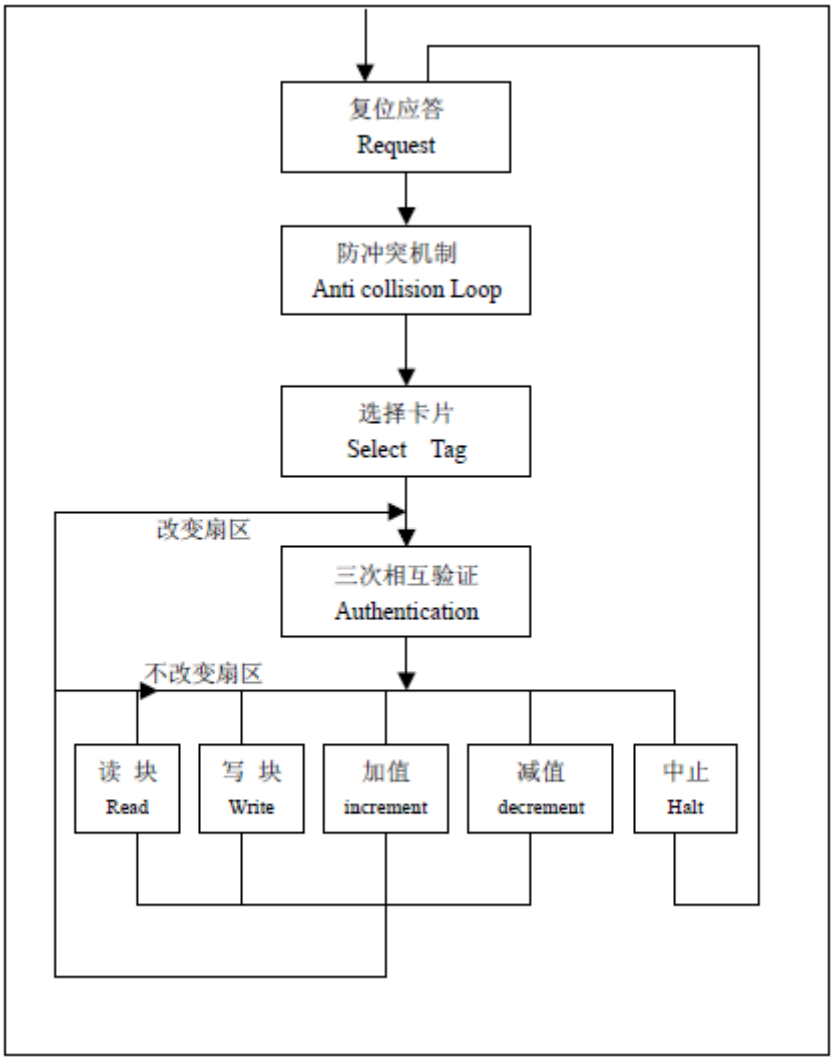
原型：int WINAPI rf_light(unsigned short icdev, unsigned char color)
参数：icdev：[IN] 通讯设备标识符
color：[IN] 0 = LED熄灭
1 = 蓝光LED亮
2 = 红光LED亮
返回：成功返回0

3.1.10 INT WINAPI RF_BEEP

功能：控制蜂鸣器响
原型：int WINAPI rf_beep(unsigned short icdev, unsigned char msec)
参数：icdev： [IN] 通讯设备标识符
msec： [IN] 蜂鸣时间，单位是10毫秒
返回：成功返回0

3.3 ISO14443A 函数

3.3.2 Mifare_Std



3.3.3.1 INT WINAPI RF_REQUEST

功能：寻TYPE_A卡

原型：int WINAPI rf_request(unsigned short icdev,
unsigned char model,
unsigned short *pTagType)

参数：icdev: [IN] 通讯设备标识符
model: [IN] 寻卡模式
pTagType: [OUT] 返回卡类型值

返回：成功返回0

说明：mode = 0x26：寻未进入休眠状态的卡
mode = 0x52：寻所有状态的卡 ，包括被休眠的卡
pTagType代码的含义：
0x4400 = Ultra_light
0x0400 = Mifare_One(S50)

0x0200 = Mifare_One(S70)
0x4403 = Mifare_DESFire
0x0800 = Mifare_Pro
0x0403 = Mifare_ProX
0x0033 = SHC1102

3.3.3.2 INT WINAPI RF_ANTICOLL

功能：TYPE_A卡防冲撞
原型：int WINAPI rf_anticoll(unsigned short icdev,
unsigned char bcnt,
unsigned char *pSnr,
unsigned char *pLen)
参数：icdev： [IN] 通讯设备标识符
bcnt： [IN] 取值4
pSnr： [OUT] 返回的卡序列号
pLen： [OUT] 返回序列号的长度
返回：成功返回0
说明：如果同时有多张卡在读卡器附近，则系统会随机获取到其中一张卡的序列号。

3.3.3.3 INT WINAPI RF_SELECT

功能：激活TYPE_A卡
原型：int WINAPI rf_select(unsigned short icdev,
unsigned char *pSnr,
unsigned char snrLen,
unsigned char *pSize)
参数：icdev： [IN] 通讯设备标识符
pSnr： [IN] 卡序列号
snrLen： [IN] 卡序列号长度
pSize： [OUT] 返回卡容量
返回：成功返回0
说明：卡片接收到该命令后被锁定并进入激活状态，在一个感应区域内的同一时刻只可有一张TYPE_A卡处于激活状态
以上3个步骤可以实现完全锁定某张卡，锁定后的卡才能进行下面的操作。

3.3.3.4 INT WINAPI RF_M1_AUTHENTICATION2

功能：验证Mifare_Std卡密钥
原型：int WINAPI rf_M1_authentication2(unsigned short icdev,

```
                                    unsigned char  model,
                                    unsigned char  block,
                                    unsigned char  *pKey)
```

参数：icdev：［IN］通讯设备标识符
Model：［IN］密钥验证模式
block：　［IN］要验证密钥的绝对块号
pKey：　［IN］密钥内容，6 字节
返回：成功返回0
说明：model = 0x60：验证A密钥
model = 0x61：验证B密钥
M1卡的初始密钥是12个F，B密钥出厂一般没有激活，所以一般都用A密钥，密钥只能写入。
如果验证密钥失败的话需要重新执行rf_request ，int WINAPI rf_anticoll，int WINAPI rf_select 这3个函数以便再次锁定这张卡。同理，如果后面的读块或者写块以及钱包操作失败，都需要从头开始执行这些函数，也就是说，要严格根据s50卡的操作流程图来执行。在同一个扇区内，只要验证密钥成功后，可以重复执行读写等操作直到更换扇区或者失败。如果更换扇区则只需要再次执行对应扇区的密钥块的密钥验证操作，如果验证失败，则需要从request开始执行。
每个扇区的密钥都存放在每个扇区的块3（每个扇区含有块0到块3共4个块）。

3.3.3.5 INT WINAPI RF_M1_READ

功能：读取MifareOne卡某个指定块的数据
原型：

```
int WINAPI rf_M1_read (unsigned short icdev,
                       unsigned char  block,
                       unsigned char  *pData,
                       unsigned char  *pLen)
```

参数：icdev：［IN］ 通讯设备标识符
block：　［IN］ M1卡绝对块号
pData：　［OUT］返回的数据
pLen：　［OUT］返回数据的长度
返回：成功返回0
说明：S50卡的每个块有16个字节数据，数据以十六进制格式存放，每次读取都是固定16个字节长度。

3.3.3.6 INT WINAPI RF_M1_WRITE

功能：写入Mifare_One卡一个块的数据
原型：

```
int WINAPI rf_M1_write (unsigned short icdev,
```

unsigned char block,
unsigned char *pData)

参数：icdev: [IN] 通讯设备标识符
block: [IN] M1卡绝对块号
pData: [IN] 写入的数据，16 字节十六进制数据

返回：成功返回0

说明：S50卡的每个块是固定16个字节为单位的，每次写入也必须是按照16个字节为单位写入，哪怕只需要写入一个字节，也需要填充完16个字节一并写入，也就是说，多余的部分可以用0填充。 块里存放的数据是以十六进制方式存储的，所以，如果用户界面输入的是字符串，需要转换成十六进制后再传给写函数。

比如，界面用户输入是“北京CBD123”，那么转换成ascii十六进制代码是“B1B1BEA9434244313233”，一个汉字对应2个字节十六进制字符（北：B1B1），英文字母和数字对应1个字节十六进制字符（C:43），转换后是10个字节长度，要写入某个块，需要增加到16个字节，这时候，可以在后面用6个字节的0填充，所以 最后可以变成“B1B1BEA94342443132330000000000000”。当用读块函数读取的时候，利用相应的转换函数就可以自动转换成“北京CBD123”。同理，如果超过16个字节长度，就需要分多个块甚至多个扇区来处理，最后不足16个字节的处理方式同上。

修改S50卡的某个扇区的密钥也是通过写函数来实现的。

算存储密钥块的算法是：x=s×4+3; 其中s表示扇区号（0－15）。

密钥块是有6个字节的密钥A和4个字节的控制字以及6个字节的密钥B组成的16个字节的数据，4个字节的控制字初始值是ff 07 80 69,密钥B是12个F，也就是FFFFFFFFFFFF，假设现在要把密钥A的12个F改为“303132333435”，其余不变，那么可以在指定扇区的密钥块写入如下数据：“303132333435FF078069FFFFFFFFFFFF”，如果写入成功后，密钥A就变成了“303132333435”，以后再次读写这个扇区就需要用新的密钥来验证了。

注意：关于控制字的操作相对比较复杂，用户在完全熟悉之前不建议修改，否则可能会造成无法恢复，有兴趣的用户可以自行研究相关资料，这里不再赘述。

3.3.3.7 INT WINAPI RF_M1_INITVAL

功能：将Mifare One 卡某一块初始化为钱包

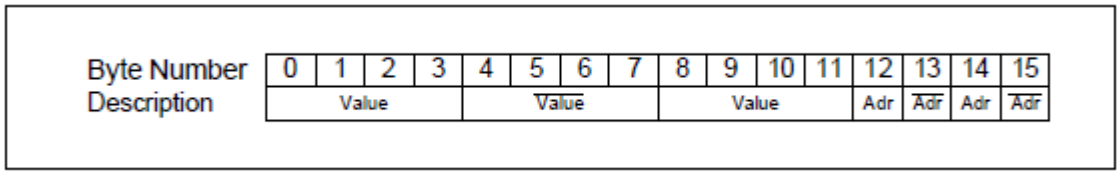
原型：int WINAPI rf_M1_initval(unsigned short icdev,
unsigned char block,
long value)

参数：icdev: [IN] 通讯设备标识符
block: [IN] M1卡绝对块号
pValue: [IN] 初始金额，16进制，低字节在前 ，比如十进制的100，变成十六进制是0x64,这里需要3个字节，所以是64000000,64是低字节，放在前面，在块里的对应的顺序是b0b1b2b3。

返回：成功返回0

注意：所有数据块在第一次作为电子钱包功能使用之前，都必须执行初始化操作，通过这个操作才能转换为适合钱包操作的数据格式，钱包数据格式采用专门的算法来保障数据的可靠性。在此基础上才能开始进行充值和扣款以及余额查询等操作。

下图是关于钱包数据的存储格式。



关于钱包内金额的定义可以自己根据需要处理成若干个小数，但是最终数据还是以整数来存取的，这点要注意。

3.3.3.8 INT WINAPI RF_M1_READVAL

功能：读取Mifare One卡钱包值

原型：int WINAPI rf_M1_readval(unsigned short icdev,
unsigned char block,
long *pValue)

参数：icdev: [IN] 通讯设备标识符
block: [IN] M1卡绝对块号
pValue: [OUT] 返回的值，16进制，低字节在前

返回：成功返回0

3.3.3.9 INT WINAPI RF_M1_INCREMENT

功能：Mifare One卡充值

原型：int WINAPI rf_M1_increment (unsigned short icdev,
unsigned char block,
long value)

参数：icdev: [IN] 通讯设备标识符
block: [IN] M1卡绝对块号
value: [IN] 要增加的值，16进制，低字节在前

返回：成功返回0

3.3.3.10 INT WINAPI RF_M1_DECREMENT

功能：Mifare One卡扣款

原型：int WINAPI rf_M1_decrement (unsigned short icdev,
unsigned char block,
long value)

参数：icdev：　[IN]　通讯设备标识符
　　block：　[IN]　M1卡绝对块号
　　value：　[IN]　要减掉的值，16进制，低字节在前
返回：成功返回0

3.3.3.13 INT WINAPI RF_HALT

功能：命令已激活的TYPE_A卡进入HALT状态
原型：int WINAPI rf_halt(unsigned short icdev)
参数：icdev：　[IN] 通讯设备标识符
返回：成功返回0
说明：卡片接收到此命令后退出激活状态 ，需要把卡移出感应区后再次进入才能激活。

4. 关于函数的返回代码

所有函数执行成功后返回0，错误返回非零，主要的错误代码请参考下表所示。

错误代码	含义
1	波特率错误
2	串口号错误或者连接断开
10	通用错误
11	不支持该命令
12	指令参数错误
13	无卡
20	寻卡失败
21	卡复位错误
22	卡密钥验证错误
23	读卡错误
24	写卡错误
25	写地址错误
26	读地址错误

5. 通信协议

用户可以根据以下通信协议自行开发用于其它系统的软件。

5.1 串口通信协议　（以下未说明皆为16进制）

- ▲ 异步半双工，1 位起始位＋8 位数据位＋1 位停止位
- ▲ 波特率：115200（默认）
- ▲ 主机发送数据格式：
  命令头＋ 长度字＋ 节点ID ＋ 命令字＋ 数据域＋ 校验字

序号	1	2	3	4	5	6
数据帧格式	命令头	长度字	节点ID	命令字	数据域	校验字
长度（字节）	2	2	2	2	x	1
说明		低字节在前	00 00			

说明：

- 命令头：固定为AABB，若后续数据中包含AA 则随后补充一字节00 以区分命令头但长度字不增加！
- 长度字：指明从节点ID到校验字最后一字节的字节数；
- 命令字：本条命令的含义；
- 数据域：此项可以为空；
- 校验字：从节点ID到数据域最后一字节的逐字节异或值。

节点ID这里都设置为0000

▲ **读卡器**返回数据格式：

序号	1	2	3	4	5	6	7
数据帧格式	命令头	长度	节点ID	命令	状态	数据域	校验
长度(字节)	2	2	2	2	1	x	1
说明		低字节在前	AdrL,adrH				

说明：

- 命令头：固定为AABB，若后续数据中包含AA 则随后补充一字节00 以区分命令头但长度字不增加！
- 长度字：指明从节点ID到校验字最后一字节的字节数；
- 命令字：本条命令的含义；
- 状态字：成功返回00，否则为错误；
- 数据域：此项可以为空；
- 校验字：从节点ID到数据域最后一字节的逐字节异或值。

5.2. 命令列表

序号	名称	代码
1、	**Initialize port**	**0x0101**
2、	**Set device node number**	**0x0102**
3、	**Read device node number**	**0x0103**
4、	**Read device Mode**	**0x0104**
5、	**Set buzzer beep**	**0x0106**
6、	**Set Led color**	**0x0107**

7、	**RFU**	**0x0108**
8、	**Set antenna status**	**0x010c**
9、	**Mifare Request**	**0x0201**
10、	**Mifare anticollision**	**0x0202**
11、	**Mifare Select**	**0x0203**
12、	**Mifare Hlta**	**0x0204**
13、	**RFU**	**0x0206**
14、	**Mifare Authentication2**	**0x0207**
15、	**Mifare Read**	**0x0208**
16、	**Mifare Write**	**0x0209**
17、	**Mifare Initval**	**0x020A**
18、	**Mifare Read Balance**	**0x020B**
19、	**Mifare Decrement**	**0x020C**
20、	**Mifare Increment**	**0x020D**

5.3.1. Initialize port：0x0101

Function：Set baud rate

Format： AA BB 06 00 00 01 01 “Baud_parameter” “xor Chk”

Baud_parameter：

0 = 4800;
1 = 9600;
2 = 14400;
3 = 19200;
4 = 28800;
5 = 38400;
6 = 57600;
7 = 115200;

Host Send to Reader Example：

Send： AA BB 06 00 00 00 01 01 03 03 //Set Baud Rate as 19200

Respond：AA BB 06 00 bf ff 01 01 00 40

5.3.2. Set device node number：0x0102

Host Send to Reader Example：

Send： AA BB 07 00 00 00 02 01 00 00 03 //Set device node number = 0x00 00

5.3.3. Read device node number：0x0103

Host Send to Reader Example：

Send: AA BB 05 00 00 00 03 01 02 //Read device node number

5.3.4. Read device Mode: 0x0104

Function: Read device mode and version

Host Send to Reader Example:

Send: AA BB 05 00 00 00 04 01 05

Respond: AA BB 12 00 52 51 04 01 00 *59 48 59 36 33 32 41 2D 31 32 30 33* 11

"*59 48 59 36 33 32 41 2D 31 32 30 33*" is "YHY632A-1203"

5.3.5. Set buzzer beep: 0x0106

Function: Beep

Format: AA BB 06 00 00 00 06 01 Delay XOR

Delay*10ms beep time, XOR is xor check

Host Send to Reader Example:

发送: AA BB 0600 0000 0601 6463 (0600—长度，表示6个字节，红色部分就是)

返回: AA BB 0600 0000 0601 *00* 04 （返回成功）

5.3.6. Set Led color: 0x0107

Host Send to Reader Example:

Send: AA BB 06 00 00 00 07 01 03 05 //Set Red&green LED on

Respond: AA BB 06 00 bf bf 07 01 00 06

Tenth data is LED parameter, function as below:

00 = LED_RED Off, LED_GREEN Off

01 = LED_BLUE On, LED_RED Off

02 = LED_BLUE Off, LED_RED On

5.3.8. Antenna status: 0x010c

Host Send to Reader Example:

Send: AA BB 06 00 00 00 0c 01 *00* 0D //Set antenna off 。

"*00*" is Antenna status parameter:

00 = Close Filed, 01= Open Filed

5.3.9. Mifare Request: 0x0201

Function: Request Type a Card

Format: AA BB 06 00 00 00 01 02 req_code XOR

req_code: Request mode:

req_code: 0x52: request all Type A card In filed

req_code: 0x26: request idle card （寻卡，被休眠的卡不被激活）

Host Send to Reader Example:

Send: AA BB 06 00 000001 0252 51

Respond: AA BB 0800 52 51 01 02 00 *04 00* 04

TagType: 0x4400 = ultra_light

0x0400 = Mifare_One(S50)

0x0200 = Mifare_One(S70)

0x4403 = Mifare_DESFire

0x0800 = Mifare_Pro

0x0403 = Mifare_ProX

5.3.10. Mifare anticollision：0x0202

Function：Card anticollision

Format： AA BB 05 00 00 00 02 02 00

Respond：AA BB 0a0052 51 02 02 00 *46 ff a6 b8* a4

"*46 ff a6b8*" is card serial number ，卡号，低位到高位

5.3.11. Mifare Select：0x0203

Function：Select card

Format： AA BB 09 00 00 00 03 02 xx xx xx xx XOR

Ninth to twelfth is card serial number 。

Host Send to Reader Example：

Send： AA BB 09 00 00 00 03 02 46 ff a6 b8 a6

Respond：AA BB 07 00 52 51 03 02 00 08 0a

5.3.12. Mifare Hlta：0x0204 卡休眠，休眠后不响应读卡指令

Function：Hlta card

Host Send to Reader Example：

Send： AA BB 05 00 0000 04 02 06

Respond：AA BB 06 00 52 51 04 02 00 05

5.3.14. Mifare Authentication2：0x0207 验证密码，如果失败请从request重新执行。

Function：Authenticate Card

Format： AA BB xx 00 00 00 07 02 Auth_mode Block xx xx xx xx xx xx XOR

Auth_mode: Authenticate mode, 0x60: KEY A, 0x61: KEY B

Block: Authenticate block

Host Send to Reader Example：

Send： AA BB 0d 00 00 00 07 02 60 04 ff ff ff ff ff ff 61

Authenticate Block 4, Key A = "FF FF FF FF FF FF"

Respond：AA BB 0600 52 51 07 02 00 06

5.3.15. Mifare Read：0x0208

Function：Read block

Format： AA BB 06 00 00 00 08 02Block XOR

Block = which block want read

Host Send to Reader Example：

Send： AA BB 06 00 00 0008 02 040e

Respond：AA BB 16 00 52 51 08 02 00 00 00 00 00 00 00 00 00 00 00 00 00 12 34 56 78 01

Tenth to sixteenth byte is Data

5.3.16. Mifare Write：0x0209

Function：Write block

Format： AA BB 16 00 00 00 0902 Block D0 D1 D2 D3 D4 D5 D6D7 D8 D9 Da Db Dc Dd De Df XOR

Sample：Write data to Block4

Host Send to Reader Example：
Send： AA BB 16 00 00 00 09 02 04 00 00 00 00 00 00 00 00 00 00 00 00 12 34 78 56 07
Respond：AA BB 06 00 52 51 09 02 00 08

5.3.17．Mifare Initval：0x020A

Function：Initialize purse
Format： AA BB 0a 00 00 00 0a 02 Block V0V1V2V3 XOR

5.3.18．Mifare Read Balance：0x020B

Function：Read balance
Format： AA BB 06 00 00 00 0B 02 Block XOR Return four byte balance

5.3.19．Mifare Decrement：0x020C

Function：Decrease balance
Format： AA BB 0a 00 00 00 0c 02 Block V0V1V2V3 XOR

5.3.20．Mifare Increment：0x020D

Function：Increase balance
Format： AA BB 0a 00 00 00 0D02 Block V0V1V2V3 XOR

联系信息	
公司名称	北京易火眼科技有限公司
电话：	010-59870151， 59754725
邮箱：	info@ehuoyan.com
网址：	http://www.ehuoyan.com/