



RFID 读写器 WEB 开发手册

用户手册 V1.0



2016-01

北京友我科技有限公司

目录

1	概述.....	5
2	技术参数.....	5
3	支持卡片类型.....	6
3.1	ISO14443A	6
3.2	ISO14443B.....	6
3.3	ISO15693	6
3.4	NFC Type1~Type4.....	7
3.5	ISO7816	7
4	支持读写器型号.....	7
5	B/S 结构和读写器控件.....	7
6	读写器控件调用方法.....	8
7	读写器控件属性.....	8
8	读写器控件方法--数据转换类	9
8.1	从数组获取所需格式的卡号字符串	9
8.2	将数组转换成 16 进制的字符串	9
8.3	将数组转换成 ascii 可见字符的字符串.....	10
8.4	将 16 进制字符串转换成 VARIANT 类的数组	10
8.5	将普通字符串转换成 VARIANT 类的数组.....	10
9	读写器控件方法—读写器和标签类.....	11
9.1	读写器相关函数.....	11
9.1.1	读取库函数内部版本号	11
9.1.2	DES 加解密函数	11
9.1.3	3DES 加解密函数	12
9.1.4	带向量的 3DES 加解密函数	12
9.1.5	串口或者 485 读写器，初始化串口	13
9.1.6	串口或者 485 读写器，释放串口	13
9.1.7	USB 无驱读写器，初始化 USB	13
9.1.8	USB 无驱读写器，释放 USB	14
9.1.9	设置设备标识.....	14
9.1.10	查询设备标识.....	14
9.1.11	读取读卡器内部版本号	14
9.1.12	查询读写器产品序列号	15
9.1.13	蜂鸣器控制函数.....	15
9.1.14	LED 指示灯控制.....	16
9.1.15	设置 LED 显示屏显示内容	17
9.1.16	设置天线的状态.....	17
9.1.17	设置寻卡模式.....	18
9.1.18	设置寻卡模式扩展函数.....	18
9.2	ISO7816 标准接触卡	20
9.2.1	SAM 卡复位	20
9.2.2	SAM 卡复位扩展函数	21
9.2.3	SAM 卡执行 COS 命令	21
9.2.4	SAM 卡 PPS 波特率设置	22

9.2.5	SAM 卡 PPS 波特率设置扩展函数.....	23
9.3	ISO14443 TypeA 相关函数.....	25
9.3.1	寻卡.....	26
9.3.2	防碰撞并且选定一张卡.....	26
9.3.3	卡片休眠.....	27
9.3.4	下载密钥到读卡器.....	28
9.3.5	用下载的密钥授权卡片.....	28
9.3.6	用当前密钥授权卡片.....	29
9.3.7	读取一块数据.....	29
9.3.8	写入一块数据.....	30
9.3.9	将某一块初始化为钱包.....	31
9.3.10	读取钱包值.....	31
9.3.11	钱包扣款.....	32
9.3.12	钱包充值.....	32
9.3.13	钱包 Restore 命令.....	32
9.3.14	钱包 Transfer 命令.....	33
9.3.15	读取多块数据.....	33
9.3.16	写多块数据.....	34
9.4	ISO14443 TYPEA UltraLight 卡操作函数.....	35
9.4.1	UltraLight 卡读块数据.....	35
9.4.2	UltraLight 卡写块数据.....	36
9.5	ISO14443-4 Type A CPU 卡操作函数.....	37
9.5.1	ISO14443-4 TypeA CPU 卡复位.....	37
9.5.2	ISO14443-4 TypeA CPU 卡执行 COS 命令.....	38
9.6	Mifare Plus 卡操作函数.....	39
9.6.1	Mifare Plus 卡 Level 0 级写数据.....	42
9.6.2	Mifare Plus 卡从 Level 0 级向 Level 1 或 3 级切换.....	43
9.6.3	Mifare Plus 卡从 Level1 或 Level2 向高级切换.....	43
9.6.4	Mifare Plus 卡 Level 3 级授权.....	44
9.6.5	Mifare Plus 卡 Level 3 级读块数据.....	44
9.6.6	Mifare Plus 卡 Level 3 级写块数据.....	45
9.6.7	Mifare Plus 卡 Level 3 级将某一块初始化为钱包.....	45
9.6.8	Mifare Plus 卡 Level 3 级读取钱包值.....	46
9.6.9	Mifare Plus 卡 Level 3 级钱包充值.....	47
9.6.10	Mifare Plus 卡 Level 3 级钱包扣款.....	47
9.6.11	Mifare Plus 卡 Level 3 级备份钱包.....	47
9.6.12	Mifare Plus 卡 Level 3 级通用读写第一次授权.....	48
9.6.13	Mifare Plus 卡 Level 3 级通用读写第二次授权.....	49
9.6.14	Mifare Plus 卡 Level 3 级通用读块.....	49
9.6.15	Mifare Plus 卡 Level 3 级通用写块.....	50
9.7	ISO14443 TypeB.....	51
9.7.1	ISO14443 Type B 卡复位.....	53
9.7.2	ISO14443 Type B 卡复位（扩展函数）.....	53
9.7.3	ISO14443 Type B 卡 COS 命令.....	54

9.7.4	ISO14443 Type B 卡 Halt	55
9.7.5	获取中国居民身份证卡号	55
9.7.6	AT88RF020/080 密钥认证	55
9.7.7	AT88RF020/080 读数据块	56
9.7.8	AT88RF020/080 写数据块	56
9.7.9	AT88RF020/080 锁数据块	57
9.7.10	AT88RF020/080 读取计数	57
9.7.11	AT88RF020/080 Deslect 当前卡片	58
9.7.12	AT88RF020/080 读取多块	58
9.7.13	AT88RF020/080 写多块	59
9.7.14	激活 S T 系列卡	59
9.7.15	S T 卡休眠	60
9.7.16	SR176 卡获取 UID	60
9.7.17	SR176 卡读数据块	60
9.7.18	SR176 卡写数据块	61
9.7.19	SR176 卡锁数据块	61
9.7.20	SR176 卡获取锁块状态	62
9.7.21	SR176 卡读取多个数据块	62
9.7.22	SR176 卡写多个数据块	63
9.7.23	SRI/SLI 卡获取 UID	63
9.7.24	SRI/SLI 卡读数据块	64
9.7.25	SRI/SLI 卡写数据块	64
9.7.26	SRI/SLI 卡锁数据块	65
9.7.27	SRI/SLI 卡获取锁块状态	65
9.7.28	SRI/SLI 卡读多块	66
9.7.29	SRI/SLI 卡写多块	66
9.7.30	SR1X4K 卡密钥认证	67
9.8	ISO15693 标签	68
9.8.1	ISO15693 标签 Inventory	69
9.8.2	ISO15693 标签 Stay Quiet	69
9.8.3	ISO15693 标签选卡	70
9.8.4	ISO15693 标签复位	70
9.8.5	ISO15693 标签读块	71
9.8.6	ISO15693 标签写块	72
9.8.7	ISO15693 标签锁块	73
9.8.8	ISO15693 标签写 AFI	74
9.8.9	ISO15693 标签锁 AFI	75
9.8.10	ISO15693 标签写 DSFID	76
9.8.11	ISO15693 标签锁 DSFID	77
9.8.12	获取 ISO15693 标签系统信息	77
9.8.13	获取 ISO15693 标签块安全信息	78
9.8.14	ISO15693 标签防冲突寻多张卡	79
9.9	NFC-Type1 TOPAZ 标签	81
9.9.1	TOPAZ 卡片获取卡号	81

9.9.2	TOPAZ 卡片读所有数据	82
9.9.3	TOPAZ 卡片读单个字节	83
9.9.4	TOPAZ 卡片先擦除再写单个字节	83
9.9.5	TOPAZ 卡片不带擦除写单个字节	84
9.9.6	TOPAZ 卡片读 Segment	85
9.9.7	TOPAZ 卡片读块	85
9.9.8	TOPAZ 卡片先擦除再写块	86
9.9.9	TOPAZ 卡片不带擦除写块	87
9.10	NFC-Type2 NTAG 系列标签	88
9.10.1	NTAG 卡片获取版本信息	89
9.10.2	NTAG 卡片读数据	89
9.10.3	NTAG 卡片快速读数据	90
9.10.4	NTAG 卡片写数据	90
9.10.5	NTAG 卡片兼容写数据	91
9.10.6	NTAG 卡片读计数器	91
9.10.7	NTAG 卡片密码授权	92
9.10.8	NTAG 卡片读签名	92
9.11	NFC-Type3 Felica 卡	94
9.11.1	Felica 卡片寻卡	94
9.11.2	Felica 卡片不加密读操作	95
9.11.3	Felica 卡片不加密写操作	96
10	订购方式	98
附录 A		99

1 概述

友我科技RFID读写器是采用13.56M非接触射频技术设计而成的通用型NFC读卡器。该读写器一系列原装芯片，确保读写性能稳定可靠。部分读写器带有2个SAM卡插槽和一个接触式IC卡读写插槽，可以很方便对需要进行安全认证的卡片进行读写操作。读卡器采用USB(HID)或者RS232，RS485，网络TCP/IP与计算机相连接，且不需要安装读卡器驱动，使计算机二次开发变得更为简单。

该读卡器支持ISO14443 TypeA，ISO14443 TypeB和ISO15693协议卡片；支持NFC定义的Type1(ISO14443 TYPEA、TOPAZ)，Type2(ISO14443 TYPEA、MIFARE Ultralight、NTAG203/210/212/213/216等)，Type3(Sony Felica)，Type4(ISO14443 TYPEA/B、MIFARE DESFire等)。该读卡器广泛应用于非接触智能水、电、气三表、交通一卡通读写器，桌面发卡器，办公/商场/洗浴中心储物箱的安全控制，各种防伪系统及生产过程控制，数据采集等。

2 技术参数

- ◆ **工作频率：**13.56MHZ
- ◆ **支持协议：**ISO14443A、ISO14443B、ISO15693、ISO7816 及其 NFC Type1~4
- ◆ **读写距离：**5~10cm，一般的正常标准卡大小能达到 8cm 左右
- ◆ **SAM 卡：**2 个卡槽(支持 ISO7816 T0/T1、A/B 类卡,Max 230400bps)
- ◆ **接触卡：**1 个卡槽(支持 ISO7816 T0/T1、A/B 类卡,Max 230400bps)
- ◆ **蜂鸣器及指示：**1 个可控蜂鸣器及 2 色可控 LED
- ◆ **接口：**USB(HID)，免驱动安装
- ◆ **电源：**USB DC5V ± 10%
- ◆ **功耗：**1W 左右
- ◆ **操作温度：**-20 ~ +70°C
- ◆ **存储温度：**-40 ~ +125°C
- ◆ **尺寸：**128 * 86 * 27 (mm)

- ◆ **重量:** 大约 150g
- ◆ **DLL 及软件支持:** DLL API 支持,同时 VC、VB、Delphi、C++Builder、C#.net、WEB 客户端等各种开发语言的例程
- ◆ **远程更新:** 支持在线升级功能,方便程序更新操作

3 支持卡片类型

3.1 ISO14443A

- ◆ Mifare One S50/70
- ◆ Mifare One Mini
- ◆ Mifare Ultra Light
- ◆ Mifare Desfire
- ◆ Mifare Plus(支持安全层级: Level0~level3)
- ◆ Jcop 41 only the (Mifare 1K & 4K compatible)
- ◆ Jewel
- ◆ ISO14443-4 (T=CL) TYPE A dual Long erface CPU Card

3.2 ISO14443B

- ◆ AT88RF020
- ◆ AT88RF080
- ◆ SR176
- ◆ SRI512/1k/2k/4k
- ◆ SLIX4K
- ◆ ISO14443-4 (T=CL) TYPE B dual Long erface CPU Card

3.3 ISO15693

- ◆ I.CODE SLI
- ◆ Tag-it HF-I

- ◆ SRF55VxxP/ SRF55VxxS
- ◆ LRI12/64/128/2k
- ◆ EM4135
- ◆ 其它兼容的 ISO15693 标签

3.4 NFC Type1~Type4

- ◆ Type1 :ISO14443 TypeA, Topaz96/512
- ◆ Type2 :ISO14443 TypeA, Mifare ultralight, NTAG203/210/212/213/216
- ◆ Type3 :Sony Felica
- ◆ Type 4 :ISO14443 TYPEA/B, Mifare desfire 等

3.5 ISO7816

- ◆ ISO7816 T=0/T=1 接触非接触卡，支持最大速度 230400bps 和最大 APDU FSD = 256 字节。支持A类卡和B类卡操作。

4 支持读写器型号

- ◆ SDT-HA
- ◆ YW-605 系列
- ◆ YW-608 系列
- ◆ YW-610 系列
- ◆ YW-620 系列
- ◆ YW-607 系列

5 B/S 结构和读写器控件

为了适应越来越多的B/S结构下要操作RFID读写器，我们开发了基于ActiveX技术的ocx文件供web浏览器调用，此ocx支持32位的windows和64位的

windows。

控件文件名称: yw60x.ocx

32位控件: x86\yw60x.ocx

64位控件: x64\yw60x.ocx

用户可以根据B/S客户端的环境配置相应的ocx文件，也可以采用安装包 yw60xocxSetup.exe自动判断客户端的环境并且自动选择适合的ocx进行安装并注册。

6 读写器控件调用方法

读写器控件yw60x.ocx类名: YW60X.yw60xCtrl.1

读写器控件yw60x.ocx的classid: 167E1838-7388-4A24-86DE-985B91F0FFBF

在web中可以采用静态调用和动态调用的方法来初始化控件

动态调用方法:

```
var obj = new ActiveXObject("YW60X.yw60xCtrl.1");
```

静态调用方法:

```
<object id="yoworfidreader"  
    classid="clsid:167E1838-7388-4A24-86DE-985B91F0FFBF">  
</object>
```

7 读写器控件属性

yw60x.ocx具有13个输出性的属性，如下表所示

属性名称	类型	含义	备注
LastResult	Long	最后一个函数执行后的返回值，与函数本身的返回值相同	判断函数的返回值与判断LastResult值是一样的
OutData1	Variant	一个Byte数组，函数执行要输出的首要内容	Byte数组可以在js和vbs里面自由转换成数组使用，或者使用控件自带的转换函数进行转换
OutData1Size	Long	OutData1数组的长度，字节数	
OutData2	Variant	一个Byte数组，函数执行要输出的首要内容	

OutData2Size	Long	OutData2数组的长度，字节数	
OutData3	Variant	一个Byte数组，函数执行要输出的首要内容	
OutData3Size	Long	OutData3数组的长度，字节数	

8 读写器控件方法--数据转换类

8.1 从数组获取所需格式的卡号字符串

函数原形： BSTR ConvertCardNo(VARIANT vCardNo, LONG ConvertIndex)

参数列表：

参数	类型	方向	含义
vCardNo	VARIANT	IN	卡号的VARIANT变量
ConvertIndex	LONG	IN	卡号字符串输出类型： 0：16进制正序输出 1：16进制倒序输出 2：如果卡号长度为4，则输出为10位10进制正序 3：如果卡号长度为4，则输出为10位10进制倒序 其他：16进制正序输出

返回值： 转换后的字符串

8.2 将数组转换成 16 进制的字符串

函数原形： BSTR GetHexStr(VARIANT vData);

参数列表：

参数	类型	方向	含义
vData	VARIANT	IN	数据的数组

返回值：转换后的16进制字符串

8.3 将数组转换成 ascii 可见字符的字符串

函数原形：BSTR GetAlphaStr(VARIANT vData);

参数列表：

参数	类型	方向	含义
vData	VARIANT	IN	数据的数组，转换以0为结束符

返回值：转换后的可见字符字符串

8.4 将 16 进制字符串转换成 VARIANT 类的数组

函数原形：VARIANT GetVariantFromHexStr(BSTR sHexStr);

参数列表：

参数	类型	方向	含义
sHexStr	BSTR	IN	16进制字符串，只能出现16进制的字符，并且字符长度必须是2的整数倍

返回值：转换后的数组

8.5 将普通字符串转换成 VARIANT 类的数组

函数原形：VARIANT GetVariantFromAlphaStr(BSTR sAlphaStr);

参数列表：

参数	类型	方向	含义
sHexStr	BSTR	IN	普通的字符串

返回值：转换后的数组

9 读写器控件方法—读写器和标签类

9.1 读写器相关函数

9.1.1 读取库函数内部版本号

函数原形: Long YW_GetDLLVersion();

参数列表: 无

返回值: 大于0为版本号, <=0为错误

9.1.2 DES 加解密函数

函数原形: Long DES (BYTE cModel, VARIANT pkey, VARIANT in);

参数列表:

参数	类型	可接受类型	含义
cModel	BYTE	数字	加解密模式选择: 0->加密 1->解密
pkey	VARIANT	byte数组, 字符串	加解密密钥, 8个字节
in	VARIANT	byte数组, 字符串	原始数据, 8个字节

输出数据列表:

属性	类型	含义
OutData1	VARIANT , BYTE数组	加密后的数据, 8个字节的BYTE数组
OutData1Size	Long	OutData1的字节长度

返回值: 1成功, <=0失败

9.1.3 3DES 加解密函数

函数原形: Long DES3 (BYTE cModel, VARIANT pKey, VARIANT In, VARIANT Out);

参数列表:

参数	类型	可接受类型	含义
cModel	BYTE	数字	加解密模式选择: 0x00->加密 0x01->解密
pkey	VARIANT	byte数组, 字符串	加解密密钥, 16个字节
in	VARIANT	byte数组, 字符串	原始数据, 8个字节

输出数据列表:

属性	类型	含义
OutData1	VARIANT , BYTE 数组	加密后的数据, 8个字节的BYTE数组
OutData1Size	Long	OutData1的字节长度

返回值: 1成功, <=0失败

9.1.4 带向量的 3DES 加解密函数

函数原形: Long DES3_CBC (BYTE cModel, VARIANT pKey, VARIANT In, VARIANT pIV);

参数列表:

参数	类型	可接受类型	含义
cModel	BYTE	数字	加解密模式选择: 0x00->加密 0x01->解密
pkey	VARIANT	byte数组, 字符串	加解密密钥, 16个字节

in	VARIANT	byte数组, 字符串	原始数据, 8个字节
pIV	VARIANT	byte数组, 字符串	加解密向量, 8个字节

输出数据列表:

属性	类型	含义
OutData1	VARIANT , BYTE 数组	加密后的数据, 8个字节的BYTE数组
OutData1Size	Long	OutData1的字节长度

返回值: 1成功, <=0失败

9.1.5 串口或者 485 读写器, 初始化串口

函数原形: `Long YW_ComInitial(LONG PortIndex, LONG Baud);`

参数列表:

参数	类型	可接受类型	含义
PortIndex	LONG	数字	串口号
Baud	LONG	数字	通信波特率

返回值: 1成功, <=0失败

9.1.6 串口或者 485 读写器, 释放串口

函数原形: `Long YW_ComFree();`

参数列表: 无

返回值: 1成功, <=0失败

9.1.7 USB 无驱读写器, 初始化 USB

函数原形: `Long YW_USBHIDInitial(void);`

参数列表: 无

返回值: 1成功, <=0失败

9.1.8 USB 无驱读写器，释放 USB

函数原形：Long YW_USBHIDFree(void);

参数列表：无

返回值：1成功，<=0失败

9.1.9 设置设备标识

函数原形：Long YW_SetReaderID(Long OldID, Long NewID);

参数列表：

参数	类型	可接受类型	含义
OldID	Long	数字	旧的设备标示ID，范围0x0000-0xFFFF
NewID	Long	数字	修改成新的设备标示ID，范围0x0000-0xFFFF

返回值：1成功，<=0失败

9.1.10 查询设备标识

函数原形：Long YW_GetReaderID(Long ReaderID);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0

返回值：>=0成功，并且为所获取的设备标示，<0失败

9.1.11 读取读卡器内部版本号

函数原形：Long YW_GetReaderVersion(Long ReaderID);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围 0x0000-0xFFFF，如果未知，则 ReaderID=0

返回值：大于0为版本号，<=0为错误

9.1.12 查询读写器产品序列号

函数原形：Long YW_GetReaderSerial(Long ReaderID)

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围 0x0000-0xFFFF，如果未知，则ReaderID=0

输出数据列表：

属性	类型	含义
OutData1	VARIANT ， BYTE 数组	读取的产品序列号，8个字节的BYTE 数组
OutData1Size	Long	OutData1的字节长度

返回值：大于0为成功，<=0为失败

9.1.13 蜂鸣器控制函数

函数原形：Long YW_Buzzer(Long ReaderID, Long Time_ON, Long
Time_OFF, Long Cycle);

参数列表：

参数	类型	可接受类型	含义

ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
Time_ON	Long	数字	蜂鸣器鸣叫时间，单位：100毫秒
Time_OFF	Long	数字	蜂鸣器静音时间，单位：100毫秒
Cycle	Long	数字	把Time_ON和Time_OFF作为一个周期，则此参数为执行此周期的次数。

返回值：大于0为命令发送成功，≤0为命令发送失败

9.1.14 LED 指示灯控制

函数原形：Long YW_Led(Long ReaderID, Long LEDIndex, Long Time_ON, Long Time_OFF, Long Cycle, Long LedIndexOn);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
LEDIndex	Long	数字	LED灯序号 0x01：红灯 0x02：绿灯
Time_ON	Long	数字	LED灯亮时间，单位：100毫秒
Time_OFF	Long	数字	LED灯灭时间，单位：100毫秒
Cycle	Long	数字	把Time_ON和Time_OFF作为一个周期，则此参数为执行此周期的次数。
LedIndexOn	Long	数字	最后要亮的灯： 0x00：全灭 0x01：红灯 0x02：绿灯

返回值：大于0为命令发送成功，≤0为命令发送失败

9.1.15 设置 LED 显示屏显示内容

函数原形：LONGYW_LEDDisplay (LONG ReaderID, LONG Alignment, VARIANT LEDText)

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
Alignment	Long	数字	显示时的对齐方式： 1：左对齐 2：居中对齐 3：右对齐
LEDText	VARIANT	字符串	要显示的内容，最多8位

返回值：大于0为命令发送成功，<=0为命令发送失败

9.1.16 设置天线的状态

函数原形：Long YW_AntennaStatus (Long ReaderID, bool Status);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
Status	bool	数字, 布尔	天线状态： 0x01->开天线 0x00->关天线

返回值：大于0为命令发送成功，<=0为命令发送失败

9.1.17 设置寻卡模式

函数原形：Long YW_SearchCardMode(Long ReaderID, Long SearchMode);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围 0x0000-0xFFFF，如果未知，则ReaderID=0
SearchMode	Long	数字	卡类型： 0x41 ('A') -> ISO14443 TypeA 包括如Mifare Classic卡，如S50、S70、Mifare mini、Mifare ultralight、MifarePlus、Mifare Desfire、Jcop 41、NTAG203/210/212/213/216、Jewel及其ISO14443-4 TYPEA CPU卡等。 0x42 ('B') -> ISO14443 TypeB 包括如AT88RF020、AT88RF080 、SR176、SRI512/1k/2k/4k、SLIX4K 、ISO14443-4 (T=CL) TYPE B CPU卡 0x31 ('1') -> ISO15693 包括如 I.CODE SLI 、Tag-it HF-I 、EM4135、SRF55VxxP、SRF55VxxS、LRI12/64/128/2k、等其它兼容卡片

返回值：大于0为命令发送成功，<=0为命令发送失败

9.1.18 设置寻卡模式扩展函数

函数原形：Long YW_SearchCardModeEx(Long ReaderID, Long SearchMode, Long Param);

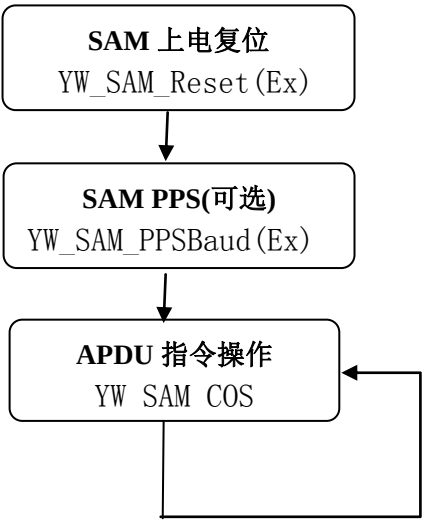
参数列表：

参数	类型	可接受	含义
----	----	-----	----

		类型	
ReaderID	Long	数字	所要获取的设备标示ID，范围 0x0000-0xFFFF，如果未知，则ReaderID=0
SearchMode	Long	数字	卡类型： 0x41('A')->ISO14443 TypeA 包括如Mifare Classic卡，如S50、S70、Mifare mini、 Mifare ultralight、MifarePlus、Mifare Desfire、Jcop 41、 NTAG203/210/212/213/216、Jewel及其ISO14443-4 TYPEA CPU卡等。 0x42('B')->ISO14443 TypeB 包括如AT88RF020、AT88RF080 、SR176、 SRI512/1k/2k/4k、SLIX4K 、ISO14443-4 (T=CL) TYPE B CPU卡 0x31('1')->ISO15693 包括如 I.CODE SLI 、Tag-it HF-I 、EM4135、 SRF55VxxP、SRF55VxxS、LRI12/64/128/2k、等其它兼 容卡片 0x54('T')->Topaz 0x46('F')->Felica
Param	Long	数字	此参数与SearchMode参数相关，具体设置如 下： 当SearchMode = 0x46 Felica 0x01 ->424k bps Felica 0x00 ->212k bps Felica 其他SearchMode该参数暂未定义

返回值：大于0为命令发送成功，<=0为命令发送失败

9.2 ISO7816 标准接触卡



(图-0 ISO7816 卡片操作流程图)

9.2.1 SAM 卡复位

函数原形: `Long YW_SAM_Reset (Long ReaderID, Long SAMIndex);`
参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围 0x0000-0xFFFF，如果未知，则 ReaderID=0
SAMIndex	Long	数字	ISO7816 卡序号 0x00-> SAM1 0x01-> SAM2 0x02> IC卡插槽

输出数据列表:

属性	类型	含义
----	----	----

OutData1	VARIANT , BYTE数 组	SAM卡复位返回的数据
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为成功, <=0为失败

9.2.2 SAM 卡复位扩展函数

函数原形: Long YW_SAM_ResetEx(Long ReaderID, Long SAMIndex, BYTE SAMVoltage);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
SAMIndex	Long	数字	IS07816 卡序号 0x00-> SAM1 0x01-> SAM2 0x02> IC卡插槽
SAMVoltage	BYTE	数字	SAM卡供电电压, 0为5V, 1为3.3V

输出数据列表:

属性	类型	含义
OutData1	VARIANT , BYTE 数组	SAM卡复位返回的数据
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为成功, <=0为失败

9.2.3 SAM 卡执行 COS 命令

函数原形: Long YW_SAM_COS(Long ReaderID, Long SAMIndex, Long LenCOS, VARIANT Com_COS);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
SAMIndex	Long	数字	ISO7816 卡序号 0x00-> SAM1 0x01-> SAM2 0x02> IC卡插槽
LenCOS	Long	数字	向SAM卡要发送的COS命令的长度
Com_COS	VARIANT	byte数组	向SAM卡要发送的COS命令

输出数据列表:

属性	类型	含义
OutData1	VARIANT , BYTE 数组	SAM执行COS命令后返回的数据
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为成功, <=0为失败

9.2.4 SAM 卡 PPS 波特率设置

函数原形: Long YW_SAM_PPSBaud(Long ReaderID, Long SAMIndex, Long BaudIndex);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
SAMIndex	Long	数字	ISO7816 卡序号 0x00-> SAM1 0x01-> SAM2

			0x02> IC卡插槽
BaudIndex	Long	数字	IS07816 卡速率： 0x00->9600、0x01->19200、 0x02->38400、 0x03->55800、0x04->57600、 0x05->115200、 0x06->223200、0x07->230400、 0x08->446400、 0x09->460800

返回值：大于0为成功，<=0为失败

9.2.5 SAM 卡 PPS 波特率设置扩展函数

函数原形：Long YW_SAM_PPSBaudEx(Long ReaderID, Long SAMIndex, Long BaudIndex, Long Protocol);

参数列表：

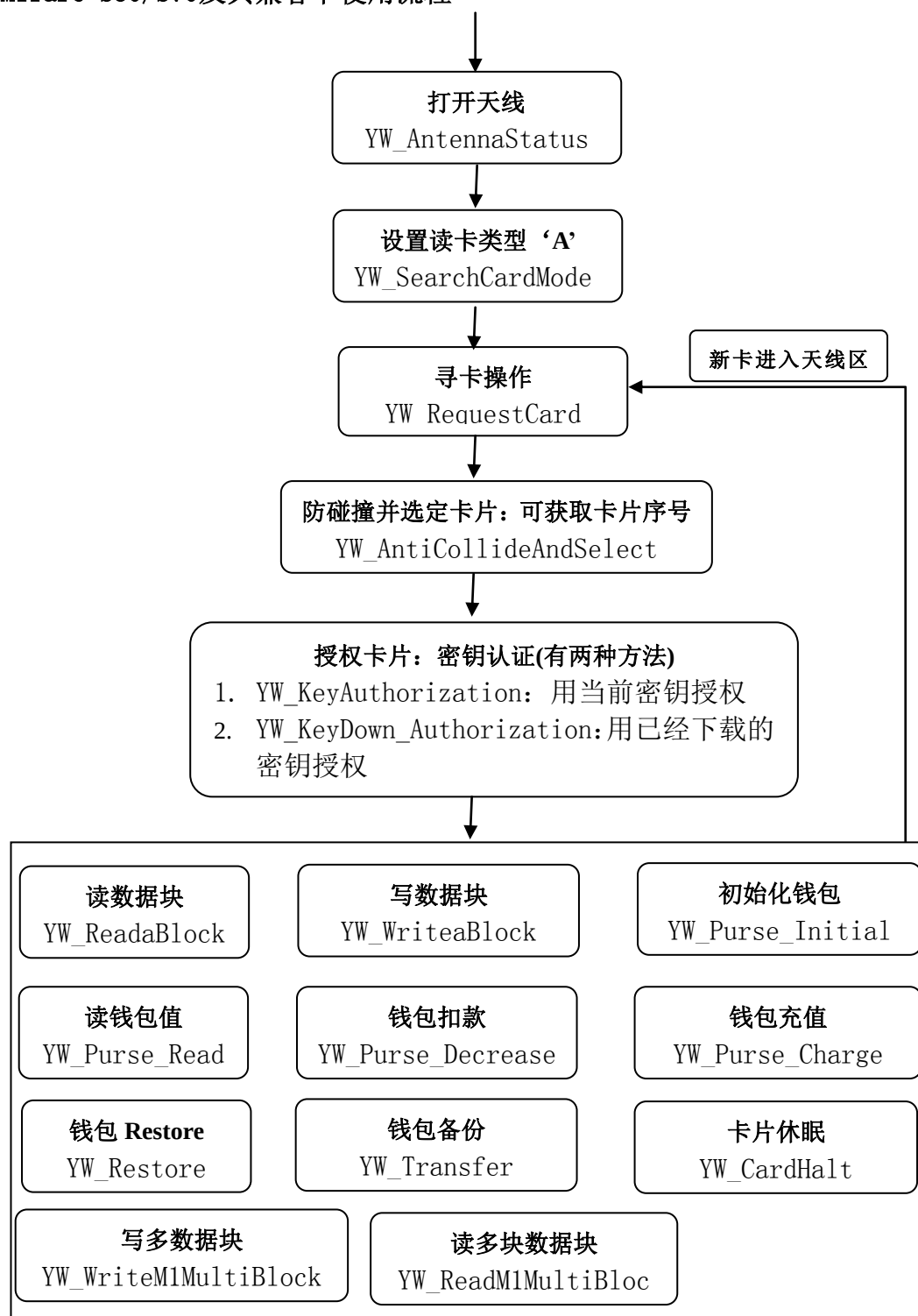
参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围 0x0000-0xFFFF，如果未知，则ReaderID=0
SAMIndex	Long	数字	IS07816 卡序号 0x00-> SAM1 0x01-> SAM2 0x02> IC卡插槽
BaudIndex	Long	数字	IS07816 卡速率： 0x00->9600、0x01->19200、0x02->38400、 0x03->55800、0x04->57600、 0x05->115200、 0x06->223200、0x07->230400、 0x08->446400、

			0x09->460800
Protocol	Long	数字	重新定义协议类型:(必须卡片支持此协议) 0x00->T0 0x01->T1 其它->保持原有协议不变

返回值: 大于0为成功, <=0为失败

9.3 ISO14443 TypeA 相关函数

Mifare S50/S70及其兼容卡使用流程



(图-1 ISO14443 TYPEA Mifare S50/S70操作流程图)

注: 1. YW_RequestCard 和 YW_AntiCollideAndSelect 两步骤可以用 YW_RequestAntiandSelect 代替。

9.3.1 寻卡

函数原形: `Long YW_RequestCard(Long ReaderID, BYTE RequestMode);`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则ReaderID=0
RequestMode	BYTE	数字	寻卡的模式: 0x52->所有卡 0x26->未休眠卡

输出数据列表:

属性	类型	含义
OutData1	VARIANT , short 数据	卡的类型: 0x4400->Ultralight/UltraLight C /MifarePlus (7Byte UID) 0x0400-> Mifare Mini/Mifare 1K (S50) /MifarePlus (4Byte UID) 0x0200-> Mifare_4K (S70) / MifarePlus (4Byte UID) 0x0800-> Mifare_Pro 0x0403-> Mifare_ProX 0x4403->Mifare_DESFire 0x4200-> MifarePlus (7Byte UID)
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, <=0为命令发送失败

9.3.2 防碰撞并且选定一张卡

函数原形: `Long YW_AntiCollideAndSelect(Long ReaderID, BYTE
MultiCardMode);`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
MultiCardMode	BYTE	数字	对多张卡的处理方式: 0x00->多张卡返回错误 0x01->返回一张卡号

输出数据列表:

属性	类型	含义
OutData1	VARIANT , BYTE数组	输出卡的序列号(4/7/10字节)
OutData1Size	Long	OutData1的字节长度
OutData2	VARIANT , BYTE数据	卡片容量代码:SAK(1字节)
OutData2Size	Long	OutData2的字节长度

返回值: 大于0为命令发送成功, <=0为命令发送失败

9.3.3 卡片休眠

函数原形: Long YW_CardHalt(Long ReaderID);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0

返回值: 大于0为命令发送成功, <=0为命令发送失败

9.3.4 下载密钥到读卡器

函数原形: `Long YW_DownloadKey(Long ReaderID, Long KeyIndex, VARIANT Key);`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
KeyIndex	Long	数字	待写入密钥存储的序号 (0~31), 共可存储32个密钥, 该密钥是先写入读卡器保存, 到卡片验证密钥的时候可以指定序号去验证。
Key	VARIANT	BYTE数组或者字符串	密钥, 每个密钥6个字节

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.3.5 用下载的密钥授权卡片

函数原形: `Long YW_KeyDown_Authorization (Long ReaderID, BYTE KeyMode, Long BlockAddr, Long KeyIndex);`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
KeyMode	BYTE	数字	验证密钥类别: 0x60->A密钥

			0x61-→B密钥
BlockAddr	Long	数字	要授权卡片的绝对块号地址
KeyIndex	Long	数字	授权密钥的序号 (0~31) 通过YW_DownloadKey下载保存的密钥序号。

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.3.6 用当前密钥授权卡片

函数原形: Long YW_KeyAuthorization (Long ReaderID, char KeyMode, Long BlockAddr, VARIANT Key);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
KeyMode	char	数字	验证密钥类别: 0x60-→A密钥 0x61-→B密钥
BlockAddr	Long	数字	要授权卡片的绝对块号地址
Key	VARIANT	BYTE 数组 或者 字符串	当前密钥字节 (共6个字节)

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.3.7 读取一块数据

函数原形: Long YW_ReadaBlock (Long ReaderID, Long BlockAddr, Long LenData);

参数列表:

参数	类型	可接受类型	含义
----	----	-------	----

ReaderID	Long	数字	所要获取的设备标示ID，范围 0x0000-0xFFFF，如果未知，则 ReaderID=0
BlockAddr	Long	数字	绝对地址块号
LenData	Long	数字	要读出的数据的字节数，Mifare S50/S70为16字节 一般固定为：0x10(16字节)

输出数据列表：

属性	类型	含义
OutData1	VARIANT ， BYTE数 组	读到的块的数据，16字节
OutData1Size	Long	OutData1的字节长度

返回值：大于0为命令发送成功，小于0为命令发送失败

9.3.8 写入一块数据

函数原形：Long YW_WriteaBlock (Long ReaderID, Long BlockAddr, Long LenData, VARIANT Data);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围 0x0000-0xFFFF，如果未知，则 ReaderID=0
BlockAddr	Long	数字	绝对块号地址
LenData	Long	数字	要写入的数据的字节数，Mifare S50/S70固定为16字节 一般固定为：0x10(16字节)
Data	VARIANT	BYTE 数组 或 者字符串	要写入的块的数据，16字节

返回值：大于0为命令发送成功，小于0为命令发送失败

9.3.9 将某一块初始化为钱包

函数原形: Long YW_Purse_Initial (Long ReaderID, Long BlockAddr, Long IniMoney);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
BlockAddr	Long	数字	绝对块号地址
IniMoney	Long	数字	初始化钱包时的初始值

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.3.10 读取钱包值

函数原形: Long YW_Purse_Read (Long ReaderID, Long BlockAddr);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
BlockAddr	Long	数字	绝对块号地址

输出数据列表:

属性	类型	含义
OutData1	VARIANT , Long数据	钱包的当前值
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.3.11 钱包扣款

函数原形：Long YW_Purse_Decrease (Long ReaderID, Long BlockAddr, Long Decrement);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockAddr	Long	数字	绝对块号地址
Decrement	Long	数字	扣款值

返回值：大于0为命令发送成功，小于0为命令发送失败

9.3.12 钱包充值

函数原形：Long YW_Purse_Charge (Long ReaderID, Long BlockAddr, Long Charge);

参数列表：

参数	类型	可接受类型	方向	含义
ReaderID	Long	数字	IN	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockAddr	Long	数字	IN	绝对块号地址
Charge	Long	数字	IN	钱包中要充值的值

返回值：大于0为命令发送成功，小于0为命令发送失败

9.3.13 钱包 Restore 命令

函数原形：Long YW_Restore (Long ReaderID, Long BlockAddr);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
BlockAddr	Long	数字	待备份钱包的绝对块号地址

返回值: 大于0为命令发送成功, 小于0为命令发送失败

注意: Restore指令是将钱包值暂存于卡片Data缓冲区, 已备Transfer指令备份到不同的钱包块中, 这两个指令是结合起来使用, 使用方法: 1. 先使用Restore将待备份的钱包值写入卡片Data缓冲区; 2. 使用Transfer将Data缓冲区中的钱包值备份到指定钱包的块中。

9.3.14 钱包 Transfer 命令

函数原形: Long YW_Transfer (Long ReaderID, Long BlockAddr);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
BlockAddr	Long	数字	备份到钱包的绝对块号地址

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.3.15 读取多块数据

函数原形: Long YW_ReadM1MultiBlock(Long ReaderID, Long StartBlock, Long BlockNums);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
StartBlock	Long	数字	绝对地址开始块号
BlockNums	Long	数字	要读块的块数

输出数据列表:

属性	类型	含义
OutData1	VARIANT , BYTE数组	实际读到的块的数据
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.3.16 写多块数据

函数原形: `Long YW_WriteM1MultiBlock(Long ReaderID, Long StartBlock, Long BlockNums, Long LenData, char *pData);`

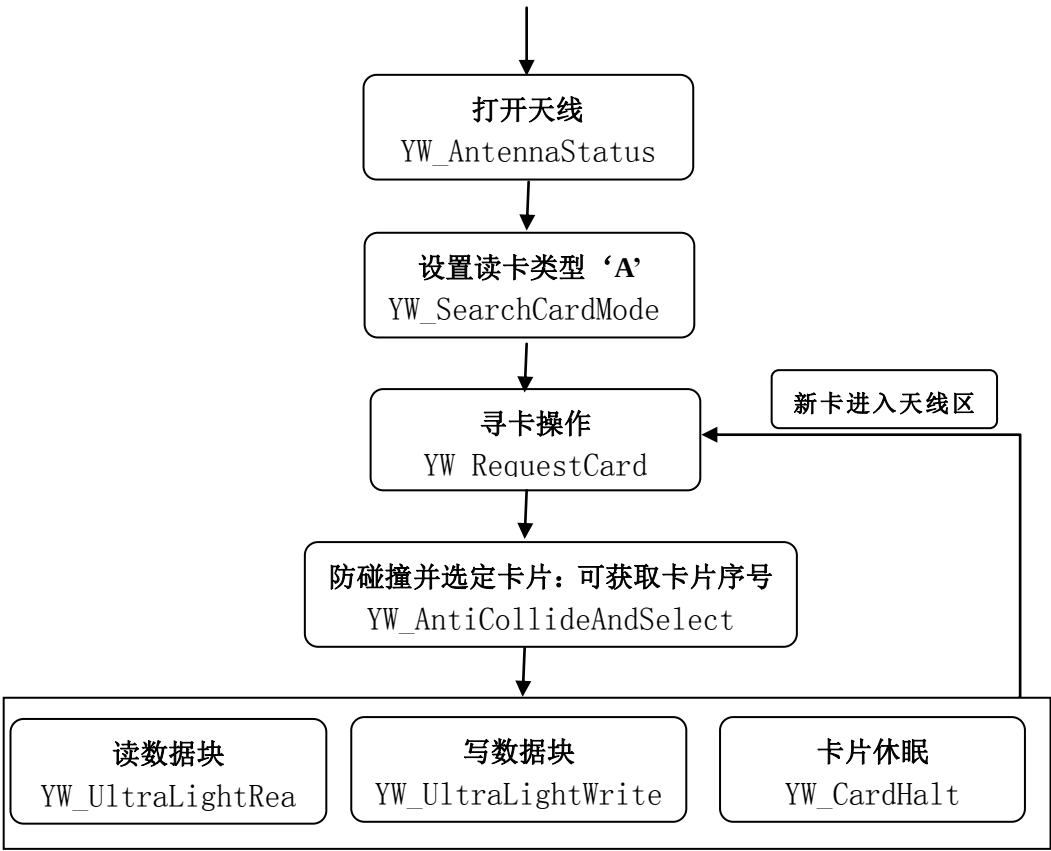
参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
StartBlock	Long	数字	写入块数据的绝对地址开始块号
BlockNums	Long	数字	写入块的数量
LenData	Long	数字	要写入的数据的字节数, Mifare One 为16* BlockNums个字节
Data	VARIANT	byte数组, 字符串	写入的块的数据

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.4 ISO14443 TYPEA UltraLight 卡操作函数

Mifare Ultralight及其兼容卡使用流程



(图-2 ISO14443 TYPEA Ultralight卡操作流程)

- 注：1. YW_RequestCard和YW_AntiCollideAndSelect 两步骤可以用 YW_RequestAntiandSelect 代替。
2. YW_UltraLightRead 和 YW_ReadaBlock功能一致。

9.4.1 UltraLight 卡读块数据

函数原形: Long YW_UltraLightRead(Long ReaderID, Long BlockID);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则ReaderID=0

BlockID	Long	数字	起始绝对地址块号
---------	------	----	----------

输出数据列表:

属性	类型	含义
OutData1	VARIANT , BYTE数组	输出读到的块的数据, 16字节, Ultralight一次读出从起始块的4块数据, 所以4*4=16
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.4.2 UltraLight 卡写块数据

函数原形: `Long YW_UltraLightWrite(Long ReaderID, Long BlockID, VARIANT pData);`

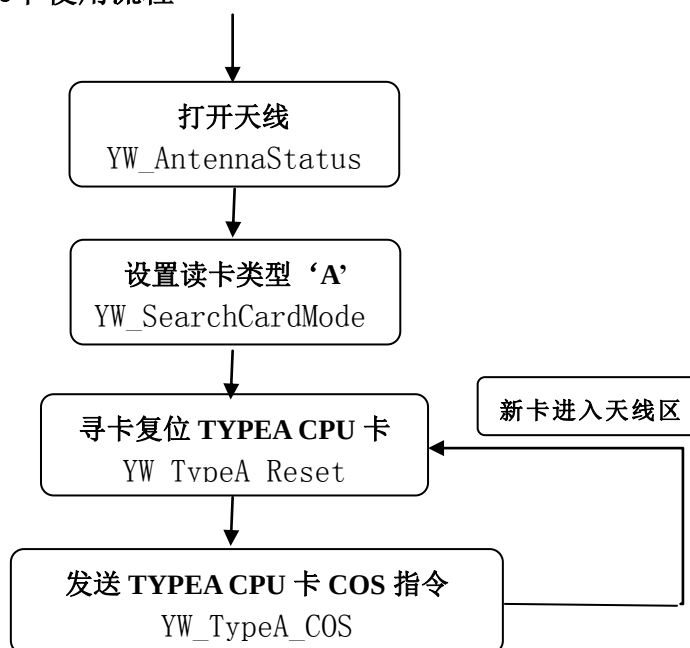
参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
BlockID	Long	数字	绝对地址块号
pData	VARIANT	byte数组, 字符串	要写入的块的数据, 4字节

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.5 ISO14443-4 Type A CPU 卡操作函数

ISO14443-4 TypeA CPU卡使用流程



(图-3 ISO14443-4 TYPEA CPU 操作流程图)

注：1. YW_TypeA_COS 其中包括 A 卡的寻卡，选卡操作，同时多出复位动作。

9.5.1 ISO14443-4 TypeA CPU 卡复位

函数原形： Long YW_TypeA_Reset (Long ReaderID, BYTE Mode, BYTE MultiMode, Long *rtLen, VARIANT pData);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围 0x0000-0xFFFF，如果未知，则 ReaderID=0
Mode	BYTE	数字	寻卡的模式： 0x52->所有的卡 0x26->未休眠的卡
MultiMode	BYTE	数字	对多张卡的处理方式：

			0x00->多张卡返回错误 0x01->返回一张卡复位信息
--	--	--	----------------------------------

输出数据列表:

属性	类型	含义
OutData1	VARIANT , BYTE数组	复位信息内容
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.5.2 ISO14443-4 TypeA CPU 卡执行 COS 命令

函数原形: Long YW_TypeA_COS(Long ReaderID, Long LenCOS, VARIANT Com_COS);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
LenCOS	BYTE	数字	输入的 C O S 命令的长度
Com_COS	VARIANT	byte数组, 字符串	C O S 命令

输出数据列表:

属性	类型	含义
OutData1	VARIANT , BYTE数组	执行命令结果
OutData1Size	Long	OutData1的字节长度

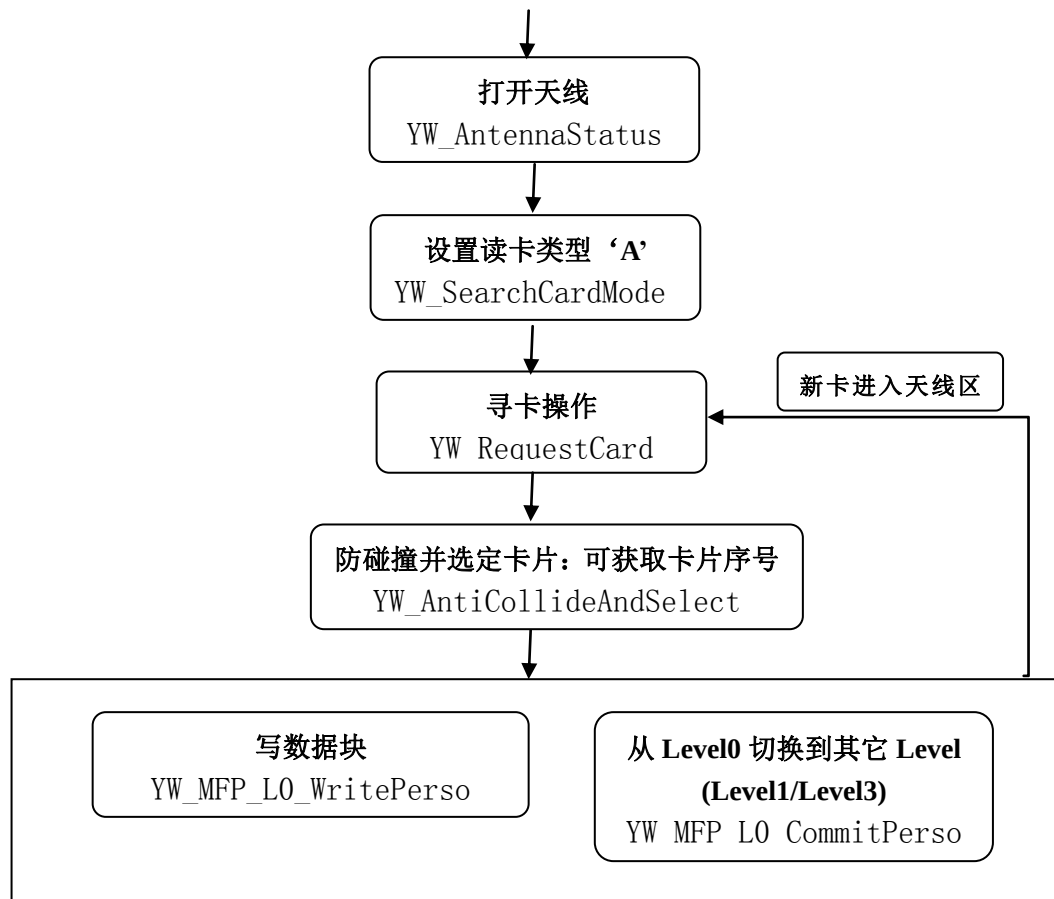
返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.6 Mifare Plus 卡操作函数

MifarePlus 卡片结构见附录，MifarePlus 分为 4 个安全级别(Level0~Level3)，不同的安全级别对寻卡操作不同，有的只需要寻卡片序号，有的需要寻卡后对卡片进行复位操作。其中 MifarePlus Level1 兼容原来的 mifareone，所有操作同 Mifareone。

Level0 操作如下：

Mifare Plus Level0 是出厂层级，在该层级主要完成一些密钥和数据初始化，然后切换到其它层级，切换到的 level 依据卡不同而不同，常见是切换到 level1。在该层级必须初始化的参数是：0x9000~0x9003。也可以写入 AES 的初始密钥，具体见 MifarePlus 数据手册。Level0 操作流程如下：



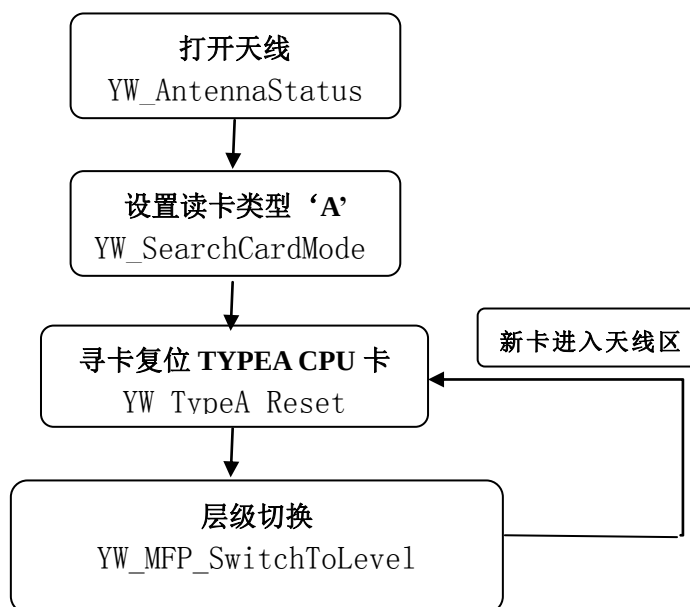
(图-4 MifarePlus Level0 卡操作流程图)

Level1 块操作如下：

Mifare Plus level1所有的块操作兼容mifare S50/S70, 操作流程见 图-1

Level1 Switch 操作:

Level1 层级切换，可以切换到 Level2 或 Level3，只能切换到高层级，不能向低的层级切换。



(图-5 MifarePlus Level1 层级切换操作流程)

注意：从 Level1 切换到其它 Level，假如想从 Level1 切换到 Level2，那么 Switch Key 就用 Switch Key2。切换到 Level3，那么 Switch Key 就用 Switch Key2

Level2 操作:

Level2 是该读卡器仅提供过度层级，只支持切换到 Level3 操作，操作流程见图-5，切换 key 用 Switch Key3。

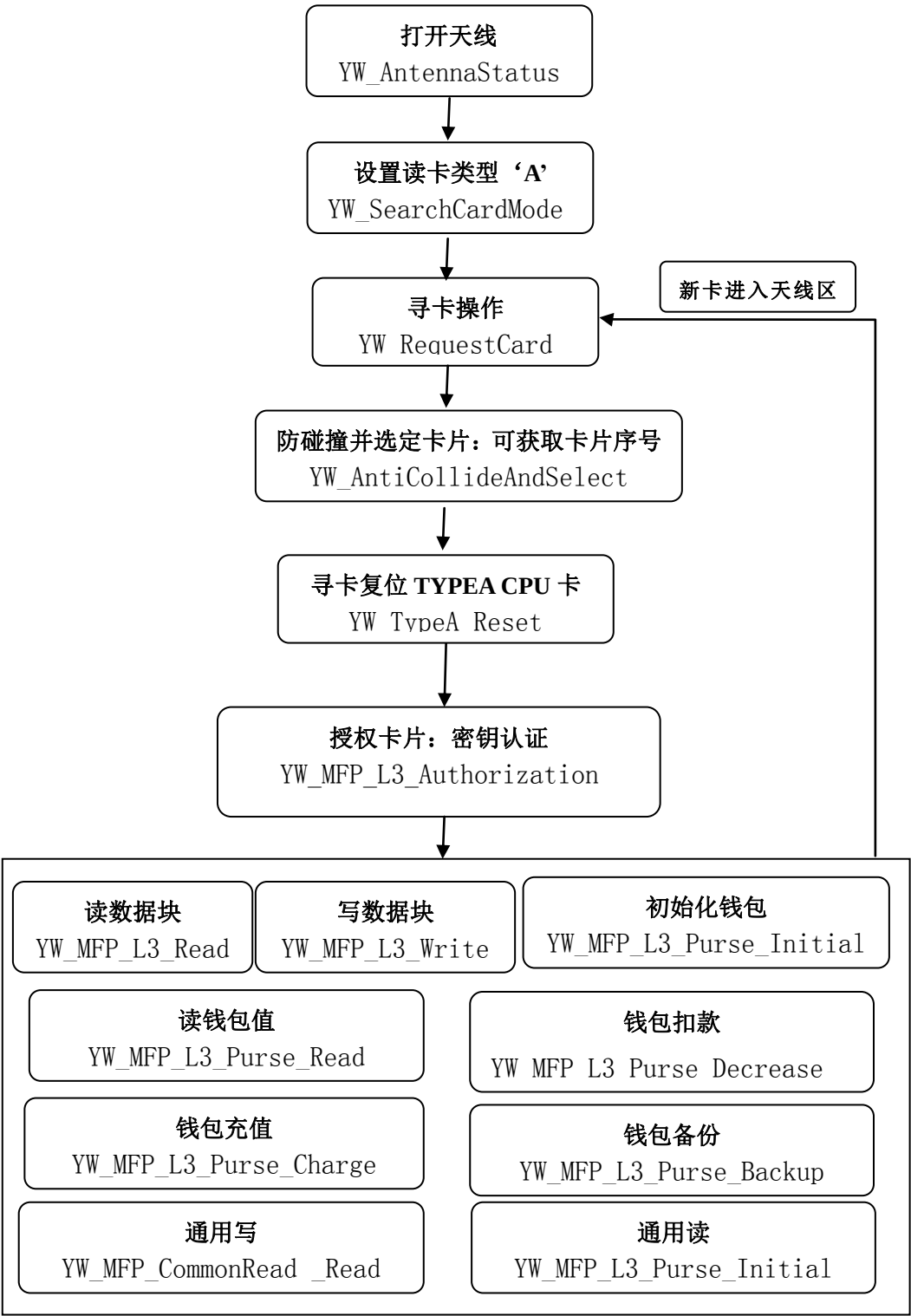
假如想从 Level2 切换到 Level3，那么 Switch Key 就用 Switch Key3。

Level3 操作:

Level3 操作分两种模式：类似 Mifare S50/S70 模式，块地址用 1 个字节（因为高字节是 00）；通用模式，该模式所有地址都是 2 个字节，完全按 MifarePlus 数据手册定义操作。对于高位地址不是 0x00 的块，如 0x9000 块，则可以通过通用模式指令读写，但是相应的授权需根据 MifarePlus 手册定义的选择用

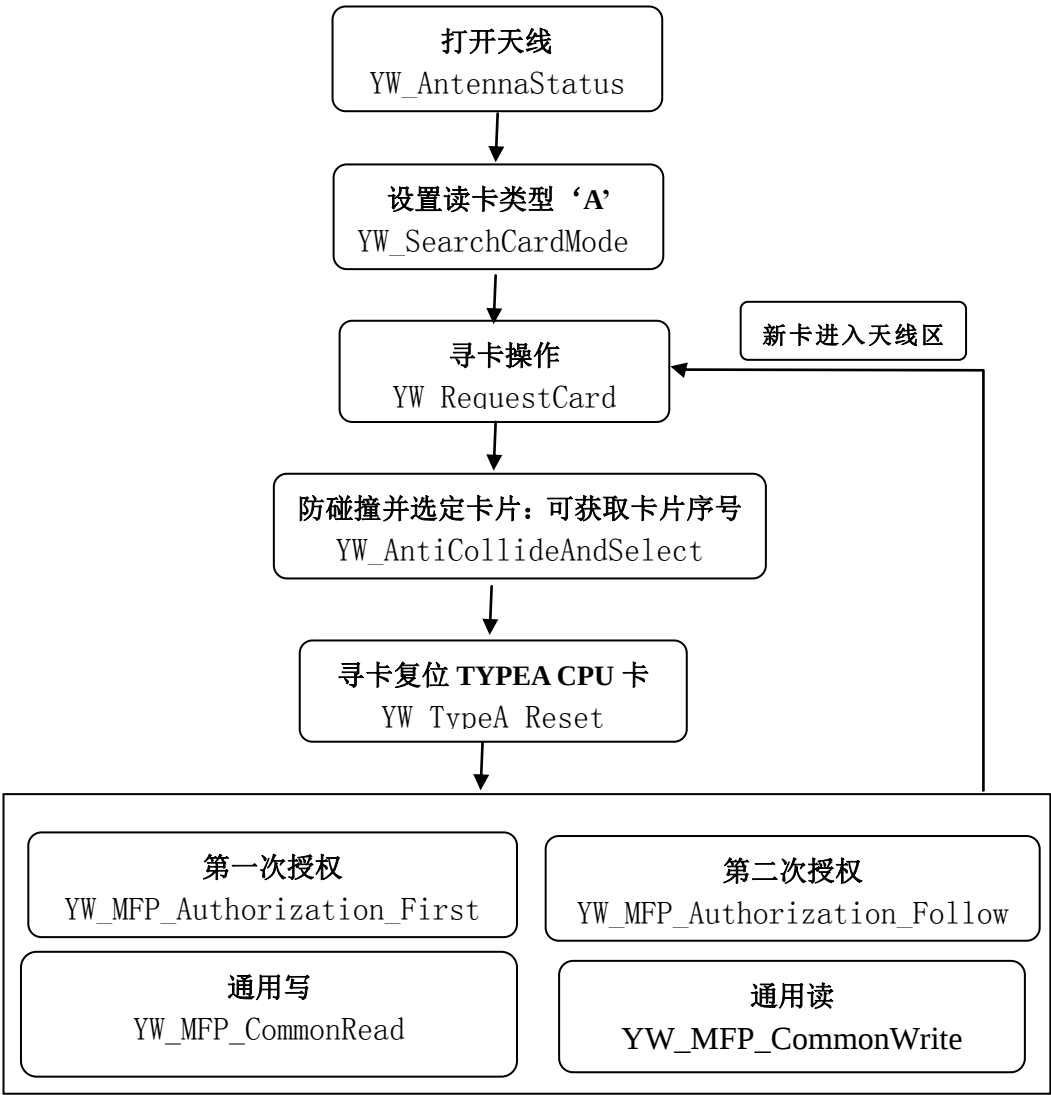
YW_MFP_Authorization_First 或者 YW_MFP_Authorization_Follow 进行授权。

类似 Mifare S50/S70 模式:



(图-6 Mifare Plus Level3 模式1操作流程图)

Level3 通用模式:



(图-7 Mifare Plus Level3 模式2操作流程图)

9.6.1 Mifare Plus 卡 Level 0 级写数据

函数原形: `Long YW_MFP_L0_WritePerso(Long ReaderID, Long Address, VARIANT pData);`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0

Address	Long	数字	要写入数据的地址
pData	VARIANT	byte数组，字符串	要写入的数据，16字节

返回值：大于0为命令发送成功，小于0为命令发送失败

9.6.2 Mifare Plus 卡从 Level 0 级向 Level 1 或 3 级切换

函数原形：Long YW_MFP_L0_CommitPerso(Long ReaderID);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0

返回值：大于0为命令发送成功，小于0为命令发送失败

注意：切换到的层级依据不同的卡而不同

9.6.3 Mifare Plus 卡从 Level1 或 Level2 向高级切换

函数原形：Long YW_MFP_SwitchToLevel(Long ReaderID, Long DesLevel, VARIANT SwitchKey);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
DesLevel	BYTE	数字	要切换到的层级，最高3级 0x02-> 从当前层级切换到Level2 0x03-> 从当前层级切换到Level3 注意：只能从低层级切换到高层级，

			不可逆。
SwitchKey	VARIANT	byte数组, 字符串	切换AES密钥, 16 字节 若切换到Level2, 用Switch Key2 若切换到Level3, 用Switch Key3

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.6.4 Mifare Plus 卡 Level 3 级授权

函数原形: `Long YW_MFP_L3_Authorization(Long ReaderID, Long KeyMode, Long BlockID, VARIANT Key);`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000~0xFFFF, 如果未知, 则ReaderID=0
KeyMode	BYTE	数字	授权密钥类别: 0x60→A密钥 0x61→B密钥 Level3 A/B密钥定义见附录A
BlockID	Long	数字	块号(范围0x00~0xFF)
Key	VARIANT	byte数组, 字符串	密钥, 16 字节

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.6.5 Mifare Plus 卡 Level 3 级读块数据

函数原形: `Long YW_MFP_L3_Read(Long ReaderID, Long StartBlock, Long BlockNums, Long *DataLen, VARIANT pData);`

参数列表:

参数	类型	可接受类型	含义
----	----	-------	----

ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
StartBlock	Long	数字	开始块号(范围0x00~0xFF)
BlockNums	Long	数字	块数量(建议在一个扇区内)

输出数据列表：

属性	类型	含义
OutData1	VARIANT ，BYTE数组	读到的数据
OutData1Size	Long	OutData1的字节长度

返回值：大于0为命令发送成功，小于0为命令发送失败

9.6.6 Mifare Plus 卡 Level 3 级写块数据

函数原形：Long YW_MFP_L3_Write(Long ReaderID, Long StartBlock, Long BlockNums, VARIANT pData);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
StartBlock	Long	数字	开始块号(范围0x00~0xFF)
BlockNums	Long	数字	块数量(建议在一个扇区内)
pData	BYTE *	byte数组，字符串	要写入的数据，长度必须是 16 * BlockNums

返回值：大于0为命令发送成功，小于0为命令发送失败

9.6.7 Mifare Plus 卡 Level 3 级将某一块初始化为钱包

函数原形：Long YW_MFP_L3_Purse_Initial(Long ReaderID, Long BlockID, Long InitialValue);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
BlockID	Long	数字	绝对块号地址(范围0x00~0xFF)
InitialValue	Long	数字	初始化钱包时的初始值

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.6.8 Mifare Plus 卡 Level 3 级读取钱包值

函数原形: Long YW_MFP_L3_Purse_Read(Long ReaderID, Long BlockID);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
BlockID	Long	数字	绝对块号地址(范围0x00~0xFF)

输出数据列表:

属性	类型	含义
OutData1	VARIANT , Long类型数据	钱包的当前值
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.6.9 Mifare Plus 卡 Level 3 级钱包充值

函数原形：Long YW_MFP_L3_Purse_Charge(Long ReaderID, Long BlockID, Long Value);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
BlockID	Long	数字	绝对块号地址(范围0x00~0xFF)
Value	Long	数字	充值金额(范围 大于0)

返回值：大于0为命令发送成功, 小于0为命令发送失败

9.6.10 Mifare Plus 卡 Level 3 级钱包扣款

函数原形：Long YW_MFP_L3_Purse_Decrease(Long ReaderID, Long BlockID, Long Value);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
BlockID	Long	数字	绝对块号地址(范围0x00~0xFF)
Value	Long	数字	扣款金额(范围 大于0)

返回值：大于0为命令发送成功, 小于0为命令发送失败

9.6.11 Mifare Plus 卡 Level 3 级备份钱包

函数原形：Long YW_MFP_L3_Purse_Backup(Long ReaderID, Long

BlockID, Long DesBlockID);

参数列表:

参数	类型	可接受类型	方向	含义
ReaderID	Long	数字	IN	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
BlockID	Long	数字	IN	要备份的钱包块号(范围0x00~0xFF)
DesBlockID	Long	数字	IN	备份到的目标块号(范围0x00~0xFF)

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.6.12 Mifare Plus 卡 Level 3 级通用读写第一次授权

函数原形: Long YW_MFP_Authorization_First(Long ReaderID, Long AESKeyAddr, VARIANT AESKey);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
AESKeyAddr	Long	数字	AES 密钥地址(范围0x0000~0xffff) 密钥对应保护的区块见MifarePlus手册
AESKey	VARIANT	Byte数组, 字符串	AES密钥, 16字节

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.6.13 Mifare Plus 卡 Level 3 级通用读写第二次授权

函数原形：Long YW_MFP_Authorization_Follow(Long ReaderID, Long AESKeyAddr, VARIANT AESKey);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000~0xFFFF，如果未知，则ReaderID=0
AESKeyAddr	Long	数字	AES 密钥地址(范围0x0000~0xffff) 密钥对应保护的区块见MifarePlus手册
AESKey	VARIANT	Byte 数组， 字符串	AES密钥, 16字节

返回值：大于0为命令发送成功，小于0为命令发送失败

9.6.14 Mifare Plus 卡 Level 3 级通用读块

函数原形：Long YW_MFP_CommonRead(Long ReaderID, Long BlockID, Long BlockNums);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000~0xFFFF，如果未知，则ReaderID=0
BlockID	Long	数字	通用块地址(范围0x0000~0xffff)
BlockNums	Long	数字	块数量(建议在一个扇区内)

输出数据列表：

属性	类型	含义
----	----	----

OutData1	VARIANT , byte数组	数据(每个块数据长度16字节, 总字节= BlockNums*16)
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.6.15 Mifare Plus 卡 Level 3 级通用写块

函数原形: Long YW_MFP_CommonWrite(Long ReaderID, Long BlockID, Long BlockNums, VARIANT pData);

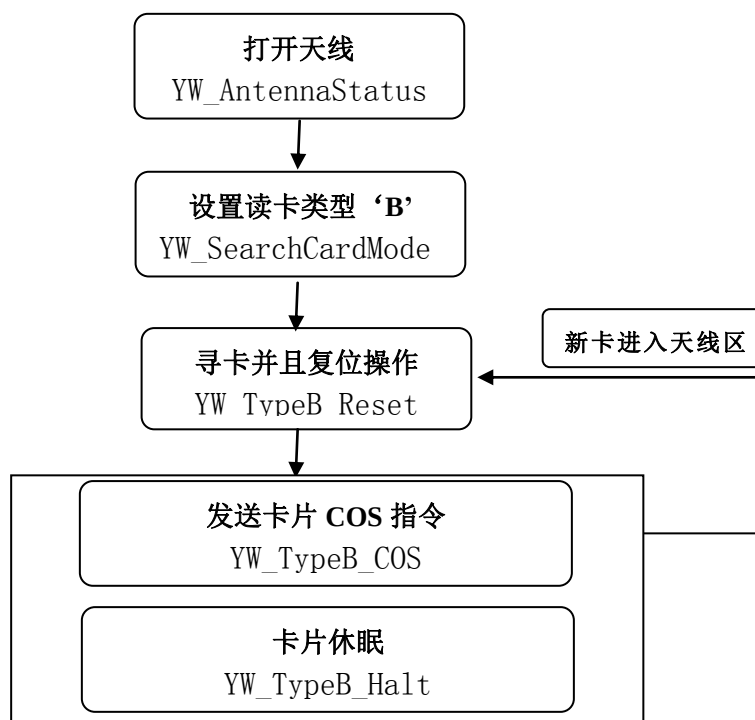
参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
BlockID	Long	数字	通用块地址(范围0x0000~0xffff)
BlockNums	Long	数字	块数量(建议在一个扇区内)
pData	VARIANT	Byte数组, 字符串	要写入的数据, 长度必须为16*BlockNums

返回值: 大于0为命令发送成功, 小于0为命令发送失败

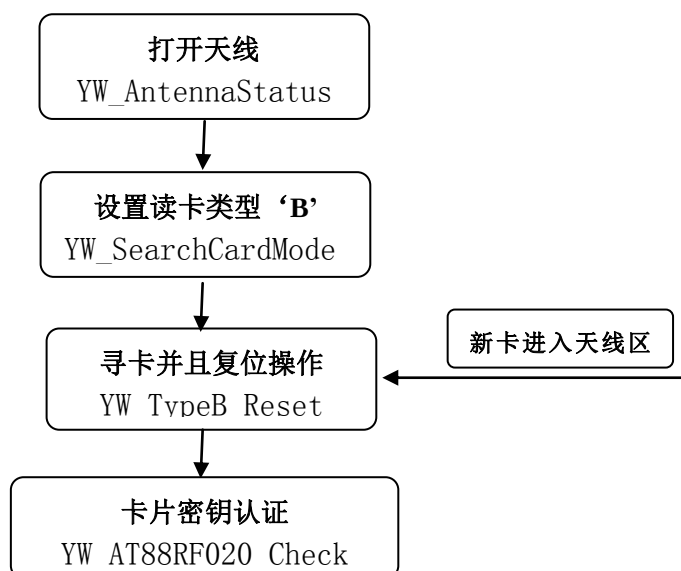
9.7 ISO14443 TypeB

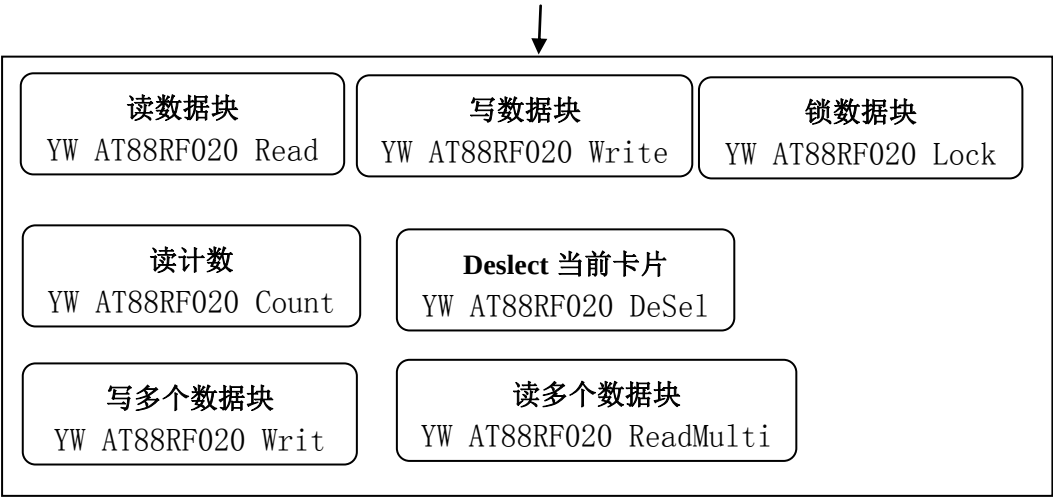
ISO14443-4 TypeB CPU卡操作流程:



(图-8 ISO14443-4 TypeB CPU卡 操作流程)

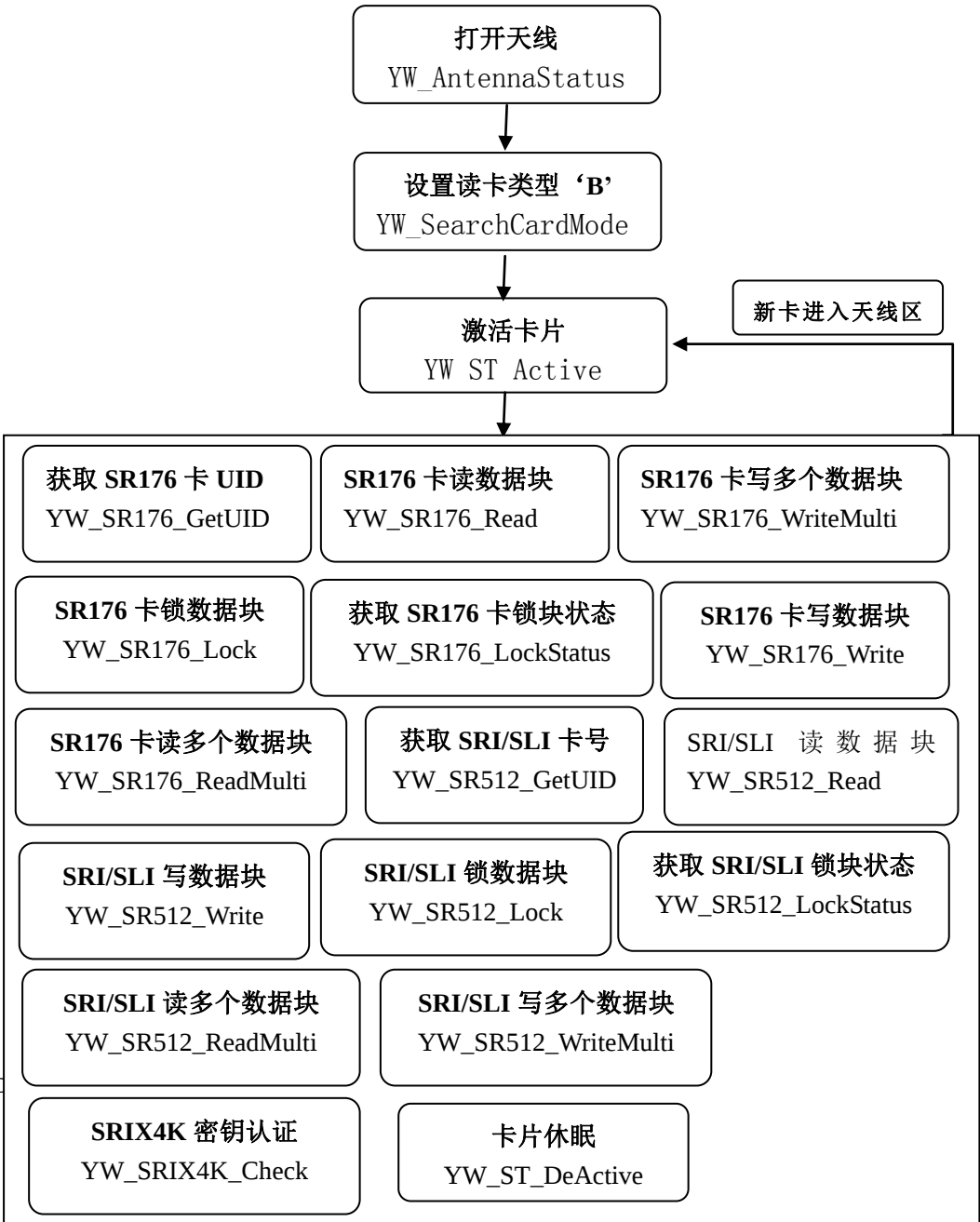
AT88RF020/AT88RF080 卡片操作函数





(图-9 AT88RF020/AT88RF080卡片 操作流程图)

SR176、SRI512/1k/2k/4k、SLIX4K操作流程



RFID

(图-10 SR176、SRI512/1k/2k/4k、SLIX4K卡 操作流程图)

9.7.1 ISO14443 Type B 卡复位

函数原形: `Long YW_TypeB_Reset(Long ReaderID, BYTE Mode);`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
Mode	BYTE	数字	寻卡的模式: 0x01->所有卡 0x00->未休眠卡

输出数据列表:

属性	类型	含义
OutData1	VARIANT , byte数组	Type B卡复位信息
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.7.2 ISO14443 Type B 卡复位 (扩展函数)

函数原形: `Long YW_TypeB_ResetEx(Long ReaderID, BYTE Mode, VARIANT Attrib);`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0

Mode	BYTE	数字	寻卡的模式： 0x01->所有卡 0x00->未休眠卡
Attrib	VARIANT	BYTE数组	复位时的Attrib参数，4个字节

输出数据列表：

属性	类型	含义
OutData1	VARIANT ，byte数组	Type B卡复位信息
OutData1Size	Long	OutData1的字节长度

返回值：大于0为命令发送成功，小于0为命令发送失败

9.7.3 ISO14443 Type B 卡 COS 命令

函数原形：Long YW_TypeB_COS(Long ReaderID, Long LenCOS, VARIANT Com_COS) ;

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
LenCOS	Long	数字	执行的COS命令的长度
Com_COS	VARIANT	Byte数组，字符串	要执行的COS命令

输出数据列表：

属性	类型	含义
OutData1	VARIANT ，byte数组	执行COS后返回的数据
OutData1Size	Long	OutData1的字节长度

返回值：大于0为命令发送成功，小于0为命令发送失败

9.7.4 ISO14443 Type B 卡 Halt

函数原形: Long YW_TypeB_Halt(ReaderID, VARIANT PUPI);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
PUPI	VARIANT	Byte数组	卡片PUPI

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.7.5 获取中国居民身份证卡号

函数原形: Long YW_ChinaIDV2_RequestCardNo(Long ReaderID);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0

输出数据列表:

属性	类型	含义
OutData1	VARIANT, byte数组	返回居民二代证卡号
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, 小于0为命令发送失败

注意: 仅仅是卡号, 不是身份证号。

9.7.6 AT88RF020/080 密钥认证

函数原形: Long YW_AT88RF020_Check(Long ReaderID, VARIANT Key);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
Key	VARIANT	Byte数组, 字符串	密钥, 8 字节, 出厂密钥是8个0x00

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.7.7 AT88RF020/080 读数据块

函数原形: Long YW_AT88RF020_Read(Long ReaderID, Long BlockID);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
BlockID	Long	数字	要读取的数据块地址

输出数据列表:

属性	类型	含义
OutData1	VARIANT, byte数组	读取的数据(每个块8字节数据)
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.7.8 AT88RF020/080 写数据块

函数原形: Long YW_AT88RF020_Write(Long ReaderID, Long BlockID, VARIANT pData);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围

			0x0000-0xFFFF, 如果未知, 则 ReaderID=0
BlockID	Long	数字	要写入的数据块号
pData	VARIANT	Byte数组, 字符串	要写入的数据(每个块8字节数据)

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.7.9 AT88RF020/080 锁数据块

函数原形: Long YW_AT88RF020_Lock(Long ReaderID, VARIANT LockByte);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
LockByte	VARIANT	Byte数组	锁块数据(4字节, 对应32块锁存, 详情见手册)

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.7.10 AT88RF020/080 读取计数

函数原形: Long YW_AT88RF020_Count(Long ReaderID);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0

输出数据列表:

属性	类型	含义
----	----	----

OutData1	VARIANT , byte数组	签名, 6 字节
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.7.11 AT88RF020/080 Deslect 当前卡片

函数原形: Long YW_AT88RF020_DeSel (Long ReaderID);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.7.12 AT88RF020/080 读取多块

函数原形: Long YW_AT88RF020_ReadMulti (Long ReaderID, Long
BlockID, Long BlockNums);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
BlockID	Long	数字	要读取的起始块号
BlockNums	Long	数字	块数量

输出数据列表:

属性	类型	含义
OutData1	VARIANT , byte数组	读取的数据
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.7.13 AT88RF020/080 写多块

函数原形: Long YW_AT88RF020_WriteMulti(Long ReaderID, Long BlockID, Long BlockNums, Long LenData, VARIANT pData);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
BlockID	Long	数字	要写入数据的起始块号
BlockNums	Long	数字	块数量
LenData	Long	数字	要写入的数据长度
pData	VARIANT	byte数组, 字符串	要写入的数据

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.7.14 激活 S T 系列卡

函数原形: Long YW_ST_Active(Long ReaderID);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0

输出数据列表:

属性	类型	含义
OutData1	VARIANT, BYTE数据	卡的Chip ID, 一个字节
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, 小于0为命令发送失败

注意: 本功能适合SR176、SRI512/1K/2K/4K、SLIX4K卡。

9.7.15 S T 卡休眠

函数原形: `Long YW_ST_DeActive(Long ReaderID);`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则ReaderID=0

返回值: 大于0为命令发送成功, 小于0为命令发送失败

注意: 本功能适合SR176、SRI512/1K/2K/4K、SLIX4K卡。

9.7.16 SR176 卡获取 UID

函数原形: `Long YW_SR176_GetUID(Long ReaderID);`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0

输出数据列表:

属性	类型	含义
OutData1	VARIANT , BYTE数据	SR176卡的UID, 8个字节
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.7.17 SR176 卡读数据块

函数原形: `Long YW_SR176_Read(Long ReaderID, Long BlockID);`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围

			0x0000-0xFFFF, 如果未知, 则 ReaderID=0
BlockID	Long	数字	所读数据块号

输出数据列表:

属性	类型	含义
OutData1	VARIANT , BYTE数据	读取的数据, 2个字节
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.7.18 SR176 卡写数据块

函数原形: `Long YW_SR176_Write(Long ReaderID, Long BlockID, VARIANT pData);`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
BlockID	Long	数字	所写数据块块号
pData	VARIANT	Byte 数组, 字符串	要写入的数据, 2个字节

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.7.19 SR176 卡锁数据块

函数原形: `Long YW_SR176_Lock(Long ReaderID, BYTE LockByte);`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0

LockByte	BYTE	数字	锁数据，1个字节，详情见SR176手册
----------	------	----	---------------------

返回值：大于0为命令发送成功，小于0为命令发送失败

9.7.20 SR176 卡获取锁块状态

函数原形：Long YW_SR176_LockStatus(Long ReaderID);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0

输出数据列表：

属性	类型	含义
OutData1	VARIANT，BYTE数据	锁块状态数据，一个字节
OutData1Size	Long	OutData1的字节长度

返回值：大于0为命令发送成功，小于0为命令发送失败

9.7.21 SR176 卡读取多个数据块

函数原形：Long YW_SR176_ReadMulti(Long ReaderID, Long BlockID, Long BlockNums);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockID	Long	数字	读取的起始块号
BlockNums	Long	数字	块数量

输出数据列表：

属性	类型	含义
----	----	----

OutData1	VARIANT , BYTE数据	读取的块数据
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.7.22 SR176 卡写多个数据块

函数原形: Long YW_SR176_WriteMulti(Long ReaderID, Long BlockID, Long BlockNums, Long LenData, VARIANT pData);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
BlockID	Long	数字	写入块的起始块号
BlockNums	Long	数字	块数量
LenData	Long	数字	要写入的数据长度
pData	VARIANT	Byte 数组 , 字符串	要写入的数据

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.7.23 SRI/SLI 卡获取 UID

函数原形: Long YW_SR512_GetUID(Long ReaderID);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0

输出数据列表:

属性	类型	含义
----	----	----

OutData1	VARIANT , BYTE数据	卡片UID, 8个字节
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, 小于0为命令发送失败

注意: 本功能适合SRI512/1K/2K/4K、SLIX4K卡。

9.7.24 SRI/SLI 卡读数据块

函数原形: Long YW_SR512_Read(Long ReaderID, Long BlockID);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
BlockID	Long	数字	所读块的块号

输出数据列表:

属性	类型	含义
OutData1	VARIANT , BYTE数据	块数据, 4个字节
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, 小于0为命令发送失败

注意: 本功能适合SRI512/1K/2K/4K、SLIX4K卡。

9.7.25 SRI/SLI 卡写数据块

函数原形: Long YW_SR512_Write(Long ReaderID, Long BlockID, VARIANT pData);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0

BlockID	Long	数字	所写块的块号
pData	VARIANT	Byte数组, 字符串	要写入的数据, 4个字节

返回值: 大于0为命令发送成功, 小于0为命令发送失败

注意: 本功能适合SRI512/1K/2K/4K、SLIX4K卡。

9.7.26 SRI/SLI 卡锁数据块

函数原形: Long YW_SR512_Lock(Long ReaderID, SHORT LockByte);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
LockByte	SHORT	数字	锁数据 (0-0xFFFF) 详情见卡手册

返回值: 大于0为命令发送成功, 小于0为命令发送失败

注意: 本功能适合SRI512/1K/2K/4K、SLIX4K卡。

9.7.27 SRI/SLI 卡获取锁块状态

函数原形: Long YW_SR512_LockStatus(Long ReaderID)

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0

输出数据列表:

属性	类型	含义
OutData1	VARIANT , SHORT数	锁块数据

	据	
OutData1Size	Long	OutData1的字节长度

返回值：大于0为命令发送成功，小于0为命令发送失败

注意：本功能适合SRI512/1K/2K/4K、SLIX4K卡。

9.7.28 SRI/SLI 卡读多块

函数原形：Long YW_SR512_ReadMulti(Long ReaderID, Long BlockID, Long BlockNums)

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockID	Long	数字	所读数据的起始块号
BlockNums	Long	数字	块数量

输出数据列表：

属性	类型	含义
OutData1	VARIANT，byte数组	读取的数据
OutData1Size	Long	OutData1的字节长度

返回值：大于0为命令发送成功，小于0为命令发送失败

注意：本功能适合SRI512/1K/2K/4K、SLIX4K卡。

9.7.29 SRI/SLI 卡写多块

函数原形：Long YW_SR512_WriteMulti(Long ReaderID, Long BlockID, Long BlockNums, Long LenData, VARIANT pData)

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围

			0x0000-0xFFFF, 如果未知, 则 ReaderID=0
BlockID	Long	数字	所写数据的起始块号
BlockNums	Long	数字	块数量
LenData	Long	数字	写入数据长度
pData	VARIANT	Byte数组, 字符串	要写入的数据

返回值: 大于0为命令发送成功, 小于0为命令发送失败

注意: 本功能适合SRI512/1K/2K/4K、SLIX4K卡。

9.7.30 SRIX4K 卡密钥认证

函数原形: Long YW_SRIX4K_Check(Long ReaderID, VARIANT Key)

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
Key	VARIANT *	Byte数组, 字符串	密钥, 6 个字节

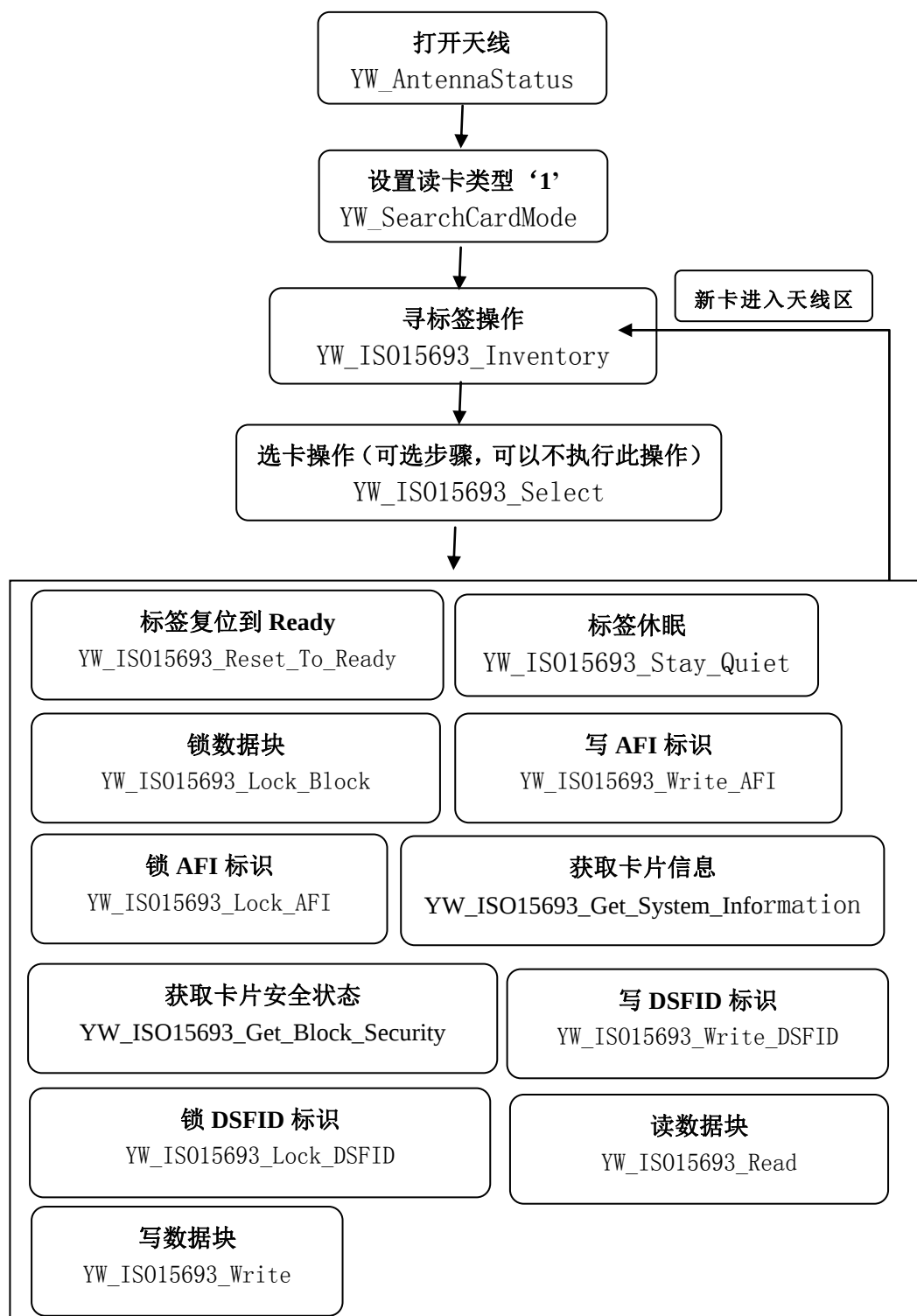
输出数据列表:

属性	类型	含义
OutData1	VARIANT, byte数组	输出签名, 3 个字节
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, 小于0为命令发送失败

9.8 ISO15693 标签

ISO15693 Tag操作流程



(图-11 ISO15693 标签操作流程)

9.8.1 ISO15693 标签 Inventory

函数原形: Long YW_ISO15693_Inventor(Long ReaderID);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0

输出数据列表:

属性	类型	含义
OutData1	VARIANT ,byte 数组	返回卡片UID(8Byte)
OutData1Size	Long	OutData1的字节长度
OutData2	VARIANT ,byte 数据	返回数据Flag(1byte)
OutData2Size	Long	OutData2的字节长度
OutData3	VARIANT ,byte 数据	返回数据DSFID(1Byte)
OutData3Size	Long	OutData3的字节长度

返回值: 大于0为成功, 小于0为失败

9.8.2 ISO15693 标签 Stay Quiet

函数原形: Long YW_ISO15693_Stay_Quiet(Long ReaderID, VARIANT PUID);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
PUID	VARIANT	byte数组	卡的序列号UID(8Byte)

返回值：大于0为成功，小于0为失败

9.8.3 IS015693 标签选卡

函数原形：Long YW_IS015693_Select(Long ReaderID, BYTE Model, VARIANT PUID);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
Model	BYTE	数字	默认：0x02
PUID	VARIANT	byte数组	卡的序列号UID(8Byte)

返回值：大于0为成功，小于0为失败

注意：

1. 执行该函数后，标签被选定，除了Inventory API外，所有其它API的Model可以选择0x00，表示读卡器一直都是对选定的卡片操作，此时附带的API中UID可以给任意值。若Model=0x02，则表示执行的是指定UID的卡片，此时API中的UID是Inventory得到的UID。

9.8.4 IS015693 标签复位

函数原形：Long YW_IS015693_Reset_To_Ready(Long ReaderID, BYTE Model, VARIANT PUID);

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
Model	BYTE	数字	Model标识（Flag）： 0x01->对已选择的标签进行操作，PUID可为任意值，

			Option_flag = 0; 0x02->对PUID指定的标签进行操作, PUID为当前操作标签的UID, Option_flag = 0; 0x04->对已选择的标签进行操作, PUID可为任意值, Option_flag = 1; 0x06->对PUID指定的标签进行操作, PUID为当前操作标签的UID, Option_flag = 1;
PUID	VARIANT	byte 数组	卡的序列号UID(8Byte)

返回值: 大于0为成功, 小于0为失败

注意:

1. IS015693 中Model, 其实为IS015693中的Flag标志位。

Model中位0 (bit0) : IS015693 Flag中 Select_flag位(选择标识位). 表示处于选择状态标签执行该指令, 后UID为任意值, 执行前必须先执行 YW_IS015693_Select API操作选定卡片。

Model中位1 (bit1) : IS015693 Flag中 Addres_flag位(地址标识位). 表示读卡器与本API给定UID一致的标签执行该指令, 执行前可以不必先执行YW_IS015693_Select API操作选定卡片。

Model中位2 (bit2) : IS015693中Option_flag位, 不同标签指令, 该值各不相同。

9.8.5 IS015693 标签读块

函数原形: Long YW_IS015693_Read(Long ReaderID, BYTE Model, VARIANT PUID, BYTE StartBlockID, BYTE BlockNums);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0

Model	BYTE	数字	Model标识 (Flag) : 0x01->对已选择的标签进行操作, PUID可为任意值, Option_flag = 0; 0x02->对PUID指定的标签进行操作, PUID为当前操作标签的UID, Option_flag = 0; 0x04->对已选择的标签进行操作, PUID可为 任意值, Option_flag = 1; 0x06->对PUID指定的标签进行操作, PUID为当前操作标签的UID, Option_flag = 1;
PUID	VARIANT	byte数组	卡的序列号UID(8Byte)
StartBlockID	BYTE	数字	开始块号
BlockNums	BYTE	数字	块数量

输出数据列表:

属性	类型	含义
OutData1	VARIANT ,byte 数组	读到的数据
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为成功, 小于0为失败

注意:

1. Model的解释见 YW_IS015693_Reset_To_Ready 中注释。
2. 若Option_Flag = 1, 返回的数据带安全状态字节 (Block Security Status), 格式为: 块安全状态字节 + 块数据; 反之不带安全状态字节。

9.8.6 IS015693 标签写块

函数原形: Long YW_IS015693_Write(Long ReaderID, BYTE Model, VARIANT PUID, BYTE BlockID, VARIANT PData);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
Model	BYTE	数字	Model标识 (Flag): 0x01->对已选择的标签进行操作, PUID可为任意值, Option_flag = 0; 0x02->对PUID指定的标签进行操作, PUID为当前操作标签的UID, Option_flag = 0; 0x04->对已选择的标签进行操作, PUID可为任意值, Option_flag = 1; 0x06->对PUID指定的标签进行操作, PUID为当前操作标签的UID, Option_flag = 1;
PUID	VARIANT	byte数组	卡的序列号UID(8Byte)
BlockID	BYTE	数字	操作的绝对块号块号
PData	VARIANT	Byte数组, 字符串	写到Tag中的块数据

返回值: 大于0为成功, 小于0为失败

注意:

1. Model的解释见 *YW_IS015693_Reset_To_Ready* 中注释。
2. Option_Flag 见IS015693 标准, 和返回数据字节无关。

9.8.7 IS015693 标签锁块

函数原形: Long YW_IS015693_Lock_Block(Long ReaderID, BYTE Model, VARIANT PUID, BYTE BlockID);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围

			0x0000-0xFFFF，如果未知，则 ReaderID=0
Model	BYTE	数字	Model标识（Flag）： 0x01->对已选择的标签进行操作，PUID可为任意值，Option_flag = 0； 0x02->对PUID指定的标签进行操作，PUID为当前操作标签的UID，Option_flag = 0； 0x04->对已选择的标签进行操作，PUID可为任意值，Option_flag = 1； 0x06->对PUID指定的标签进行操作，PUID为当前操作标签的UID，Option_flag = 1；
PUID	VARIANT	Byte数组	卡的序列号UID(8Byte)
BlockID	BYTE	数字	绝对块号

返回值：大于0为成功，小于0为失败

注意：

- 1. Model的解释见 YW_IS015693_Reset_To_Ready 中注释。
- 2. Option_Flag 见IS015693 标准，和返回数据字节无关。

9.8.8 IS015693 标签写 AFI

函数原形：Long YW_IS015693_Write_AFI(Long ReaderID, BYTE Model, VARIANT PUID, BYTE AFI)；

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围 0x0000-0xFFFF，如果未知，则 ReaderID=0
Model	BYTE	数字	Model标识（Flag）： 0x01->对已选择的标签进行操作，PUID可为任意值，Option_flag = 0；

			0x02->对PUID指定的标签进行操作, PUID为当前操作标签的UID, Option_flag = 0; 0x04->对已选择的标签进行操作, PUID可为任意值, Option_flag = 1; 0x06->对PUID指定的标签进行操作, PUID为当前操作标签的UID, Option_flag = 1;
PUID	VARIANT	Byte数组	卡的序列号UID(8Byte)
AFI	BYTE	数字	AFI值

返回值: 大于0为成功, 小于0为失败

注意:

1. Model的解释见 YW_IS015693_Reset_To_Ready 中注释。

2. Option_Flag 见IS015693 标准, 和返回数据字节无关。

9.8.9 IS015693 标签锁 AFI

函数原形: Long YW_IS015693_Lock_AFI(Long ReaderID, BYTE Model, VARIANT PUID);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
Model	BYTE	数字	Model标识 (Flag) : 0x01->对已选择的标签进行操作, PUID可为任意值, Option_flag = 0; 0x02->对PUID指定的标签进行操作, PUID为当前操作标签的UID, Option_flag = 0; 0x04->对已选择的标签进行操作, PUID可为任意值, Option_flag = 1; 0x06->对PUID指定的标签进行操作, PUID为

			当前操作标签的UID, Option_flag = 1;
PUID	VARIANT	Byte数组	卡的序列号UID(8Byte)

返回值: 大于0为成功, 小于0为失败

注意:

1. Model的解释见 *YW_IS015693_Reset_To_Ready* 中注释。

2. Option_Flag 见IS015693 标准, 和返回数据字节无关。

9.8.10 IS015693 标签写 DSFID

函数原形: Long YW_IS015693_Write_DSFIID(Long ReaderID, BYTE Model, VARIANT PUID, BYTE DSFID);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
Model	BYTE	数字	Model标识 (Flag) : 0x01->对已选择的标签进行操作, PUID可为任意值, Option_flag = 0; 0x02->对PUID指定的标签进行操作, PUID为当前操作标签的UID, Option_flag = 0; 0x04->对已选择的标签进行操作, PUID可为任意值, Option_flag = 1; 0x06->对PUID指定的标签进行操作, PUID为当前操作标签的UID, Option_flag = 1;
PUID	VARIANT	Byte数组	卡的序列号UID(8Byte)
DSFID	BYTE	数字	DSFID值

返回值: 大于0为成功, 小于0为失败

注意:

1. Model的解释见 *YW_IS015693_Reset_To_Ready* 中注释。

2. Option_Flag 见IS015693 标准，和返回数据字节无关。

9.8.11 IS015693 标签锁 DSFID

函数原形: Long YW_IS015693_Lock_DSFID(Long ReaderID, BYTE Model, VARIANT PUID);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
Model	BYTE	数字	Model标识（Flag）： 0x01->对已选择的标签进行操作，PUID可为任意值，Option_flag = 0; 0x02->对PUID指定的标签进行操作，PUID为当前操作标签的UID，Option_flag = 0; 0x04->对已选择的标签进行操作，PUID可为任意值，Option_flag = 1; 0x06->对PUID指定的标签进行操作，PUID为当前操作标签的UID，Option_flag = 1;
PUID	VARIANT	Byte数组	卡的序列号UID(8Byte)

返回值: 大于0为成功，小于0为失败

注意:

1. Model的解释见 YW_IS015693_Reset_To_Ready 中注释。

2. Option_Flag 见IS015693 标准，和返回数据字节无关。

9.8.12 获取 IS015693 标签系统信息

函数原形: Long YW_IS015693_Get_System_Information(Long ReaderID, BYTE Model, VARIANT PUID);

参数列表:

参数	类型	可接受类型	含义
----	----	-------	----

ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
Model	BYTE	数字	Model标识 (Flag) : 0x01->对已选择的标签进行操作, PUID可为 任意值, Option_flag = 0; 0x02->对PUID指定的标签进行操作, PUID为 当前操作标签的UID, Option_flag = 0; 0x04->对已选择的标签进行操作, PUID可为 任意值, Option_flag = 1; 0x06->对PUID指定的标签进行操作, PUID为 当前操作标签的UID, Option_flag = 1;
PUID	VARIANT	Byte数组	卡的序列号UID (8Byte)

输出数据列表:

属性	类型	含义
OutData1	VARIANT , byte 数组	读到的数据
OutData1Size	Long	OutData1的字节长度

返回 值: 大于0为成功, 小于0为失败

注意:

1. Model的解释见 YW_IS015693_Reset_To_Ready 中注释。

2. Option_Flag 见IS015693 标准, 和返回数据字节无关。

9.8.13 获取 IS015693 标签块安全信息

函数原形: Long YW_IS015693_Get_Block_Security(Long ReaderID, BYTE
Model, VARIANT PUID, BYTE StartBlockID, BYTE
BlockNums);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围

			0x0000-0xFFFF, 如果未知, 则 ReaderID=0
Model	BYTE	数字	Model标识 (Flag) : 0x01->对已选择的标签进行操作, PUID可为任意值, Option_flag = 0; 0x02->对PUID指定的标签进行操作, PUID为当前操作标签的UID, Option_flag = 0; 0x04->对已选择的标签进行操作, PUID可为任意值, Option_flag = 1; 0x06->对PUID指定的标签进行操作, PUID为当前操作标签的UID, Option_flag = 1;
PUID	VARIANT	Byte数组	卡的序列号UID(8Byte)
StartBlockID	BYTE	数字	开始块号
BlockNums	BYTE	数字	块数量

输出数据列表:

属性	类型	含义
OutData1	VARIANT ,byte 数组	读到的数据
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为成功, 小于0为失败

注意:

1. Model的解释见 YW_IS015693_Reset_To_Ready 中注释。

2. Option_Flag 见IS015693 标准, 和返回数据字节无关。

9.8.14 IS015693 标签防冲突寻多张卡

函数原形: Long YW_IS015693_Multi_Inventory(Long ReaderID, BYTE

AFIEnable, BYTE AFI);

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
AFIEnable	BYTE	数字	
AFI	BYTE	数字	

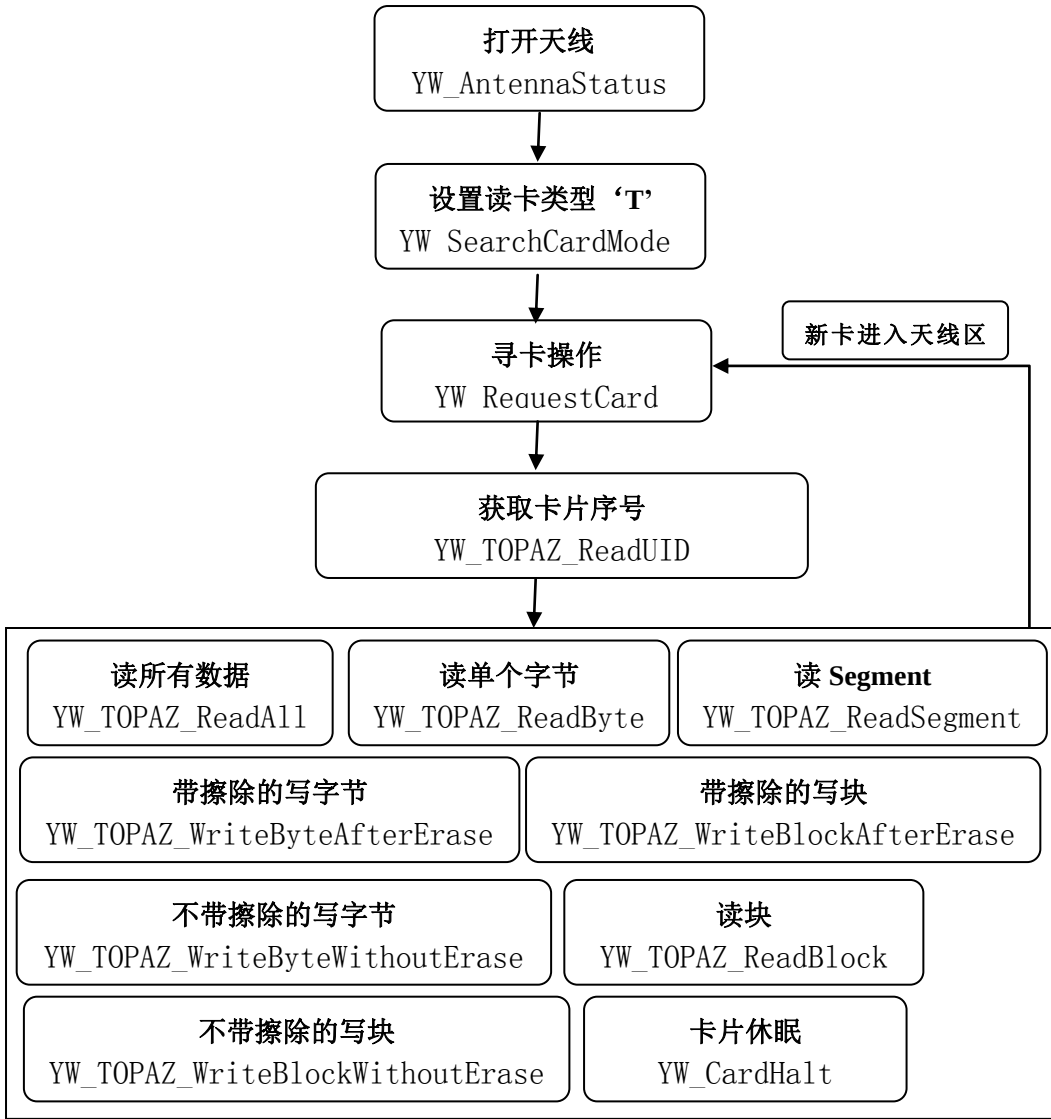
输出数据列表:

属性	类型	含义
OutData1	VARIANT ,byte 数组	多个卡的序列号UID
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为成功, 小于0为失败

9.9 NFC-Type1 TOPAZ 标签

TOPAZ 卡片操作流程:



(图-12 TOPAZ卡 操作流程)

9.9.1 TOPAZ 卡片获取卡号

函数原形: `Long YW_TOPAZ_ReadUID(Long ReaderID)`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围

			0x0000-0xFFFF, 如果未知, 则 ReaderID=0
--	--	--	--------------------------------------

输出数据列表:

属性	类型	含义
OutData1	VARIANT ,byte 数组	TOPAZ卡的卡号
OutData1Size	Long	OutData1的字节长度
OutData2	VARIANT ,byte 数据	TOPAZ卡的HR0
OutData2Size	Long	OutData1的字节长度
OutData3	VARIANT ,byte 数据	TOPAZ卡的HR1
OutData3Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, <=0为命令发送失败

9.9.2 TOPAZ 卡片读所有数据

函数原形: `Long YW_TOPAZ_ReadAll(Long ReaderID, VARIANT UID)`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
UID	VARIANT	byte数组	TOPAZ卡的UID, 目前4字节

输出数据列表:

属性	类型	含义
OutData1	VARIANT ,byte 数组	读到的数据
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, <=0为命令发送失败

9.9.3 TOPAZ 卡片读单个字节

函数原形: `Long YW_TOPAZ_ReadByte(Long ReaderID, VARIANT UID, BYTE BlockID, BYTE ByteAddr)`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
UID	BYTE *	byte数组	TOPAZ卡的UID, 目前4字节
BlockID	BYTE	数字	要读取的块号
ByteAddr	BYTE	数字	要读取字节所在的地址

输出数据列表:

属性	类型	含义
OutData1	VARIANT ,byte 数据	读到的数据, 一个字节
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, <=0为命令发送失败

9.9.4 TOPAZ 卡片先擦除再写单个字节

函数原形: `Long YW_TOPAZ_WriteByteAfterErase(Long ReaderID, VARIANT UID, BYTE BlockID, BYTE ByteAddr, BYTE DataWrite)`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0

UID	VARIANT	byte数组	TOPAZ卡的UID, 目前4字节
BlockID	BYTE	数字	要写入的块号
ByteAddr	BYTE	数字	要写入字节所在的地址
DataWrite	BYTE	数字	要写入的数据内容, 一个字节

输出数据列表:

属性	类型	含义
OutData1	VARIANT ,byte 数据	写入后的数据内容, 一个字节
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, <=0为命令发送失败

9.9.5 TOPAZ 卡片不带擦除写单个字节

函数原形: Long YW_TOPAZ_WriteByteWithoutErase(Long

ReaderID, VARIANT UID, BYTE BlockID, BYTE ByteAddr, BYTE

DataWrite)

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
UID	VARIANT	byte数组	TOPAZ卡的UID, 目前4字节
BlockID	BYTE	数字	要写入的块号
ByteAddr	BYTE	数字	要写入字节所在的地址
DataWrite	BYTE	数字	要写入的数据内容, 一个字节

输出数据列表:

属性	类型	含义
OutData1	VARIANT ,byte 数据	写入后的数据内容, 一个字节

OutData1Size	Long	OutData1的字节长度
--------------	------	---------------

返回值：大于0为命令发送成功，≤0为命令发送失败

9.9.6 TOPAZ 卡片读 Segment

函数原形：Long YW_TOPAZ_ReadSegment (Long ReaderID, VARIANT UID, BYTE Segment)

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
UID	VARIANT	byte数组	TOPAZ卡的UID，目前4字节
Segment	BYTE	数字	要读取的Segment，一个字节，0x00到0x03

输出数据列表：

属性	类型	含义
OutData1	VARIANT ,byte 数据	读到的数据
OutData1Size	Long	OutData1的字节长度

返回值：大于0为命令发送成功，≤0为命令发送失败

9.9.7 TOPAZ 卡片读块

函数原形：Long YW_TOPAZ_ReadBlock (Long ReaderID, VARIANT UID, BYTE BlockID)

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则

			ReaderID=0
UID	VARIANT	byte数组	TOPAZ卡的UID, 目前4字节
BlockID	BYTE	数字	要读取的块号

输出数据列表:

属性	类型	含义
OutData1	VARIANT ,byte 数据	读到的数据, 8个字节
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, <=0为命令发送失败

9.9.8 TOPAZ 卡片先擦除再写块

函数原形: `Long YW_TOPAZ_WriteBlockAfterErase(Long ReaderID, VARIANT UID, BYTE BlockID, VARIANT DataWrite)`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
UID	VARIANT	byte数组	TOPAZ卡的UID, 目前4字节
BlockID	BYTE	数字	要写入的块号
DataWrite	VARIANT	byte数组	要写入的数据, 8个字节

输出数据列表:

属性	类型	含义
OutData1	VARIANT ,byte 数据	写入后的数据, 8个字节
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, <=0为命令发送失败

9.9.9 TOPAZ 卡片不带擦除写块

函数原形: `Long YW_TOPAZ_WriteBlockWithoutErase(Long ReaderID, VARIANT UID, BYTE BlockID, VARIANT DataWrite)`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
UID	VARIANT	byte数组	TOPAZ卡的UID, 目前4字节
BlockID	BYTE	数字	要写入的块号
DataWrite	VARIANT	byte数组	要写入的数据, 8个字节

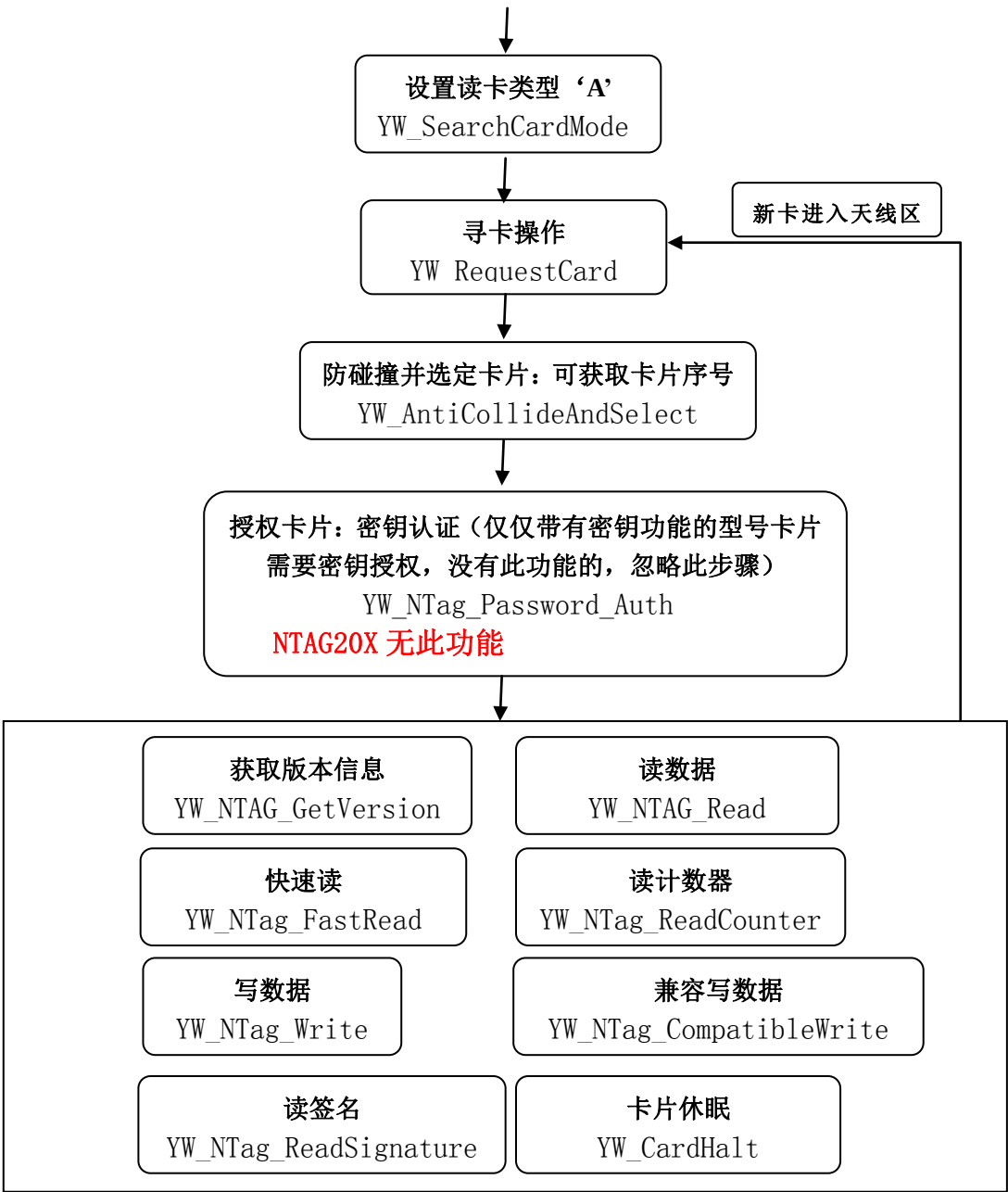
输出数据列表:

属性	类型	含义
OutData1	VARIANT ,byte 数据	写入后的数据, 8个字节
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, <=0为命令发送失败

9.10 NFC-Type2 NTAG 系列标签

NTAG 标签包括 NTAG203、NTAG210/212/213/215/216 等卡片类型，不同的卡片可能支持的 API 可能不尽相同，详细信息请参考各个型号的数据手册。NTAG 操作流程如下：



(图-13 NTAG 标签操作流程图)

9.10.1 NTAG 卡片获取版本信息

函数原形: `Long YW_NTAG_GetVersion(Long ReaderID)`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则ReaderID=0

输出数据列表:

属性	类型	含义
OutData1	VARIANT ,byte 数据	NTAG卡片的版本信息, 8个字节, NTAG20X 无此指令。 NTAG21X执行此指令的返回结果中第6字节含义: 0x0F为NTAG213卡 0x11为NTAG215卡 0x13为NTAG216卡
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, <=0为命令发送失败

9.10.2 NTAG 卡片读数据

函数原形: `Long YW_NTAG_Read(Long ReaderID, Long StartPage)`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
StartPage	Long	数字	要读取的开始页

输出数据列表:

属性	类型	含义
OutData1	VARIANT ,byte 数据	读取的数据,此处一般是读取4个数据页,即从开始页开始的连续的4个页,共16字节。
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, <=0为命令发送失败

9.10.3 NTAG 卡片快速读数据

函数原形: Long YW_NTag_FastRead(Long ReaderID, Long StartPage, Long EndPage)

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
StartPage	Long	数字	要读取的开始页
EndPage	Long	数字	要读取的结束页, 由于各种限制, 请保证读取页的跨度不要超过50页

输出数据列表:

属性	类型	含义
OutData1	VARIANT ,byte 数据	读取的数据
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, <=0为命令发送失败

9.10.4 NTAG 卡片写数据

函数原形: Long YW_NTag_Write(Long ReaderID, Long PageIndex, VARIANT pData)

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
PageIndex	Long	IN	要写入的页序号
pData	VARIANT	Byte数组, 字符串	写入的数据, 4个字节

返回值: 大于0为命令发送成功, <=0为命令发送失败

9.10.5 NTAG 卡片兼容写数据

函数原形: Long YW_NTag-CompatibleWrite(Long ReaderID, Long PageIndex, VARIANT pData)

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
PageIndex	Long	IN	要写入的页序号
pData	VARIANT	Byte数组, 字符串	写入的数据, 16个字节, 实际写入前 4个字节

返回值: 大于0为命令发送成功, <=0为命令发送失败

9.10.6 NTAG 卡片读计数器

函数原形: Long YW_NTag_ReadCounter(Long ReaderID)

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围

			0x0000-0xFFFF, 如果未知, 则ReaderID=0
--	--	--	----------------------------------

输出数据列表:

属性	类型	含义
OutData1	VARIANT , Long 数据	读到的计数器内容
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, <=0为命令发送失败

9.10.7 NTAG 卡片密码授权

函数原形: Long YW_NTag_Password_Auth(Long ReaderID, VARIANT PWD)

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
PWD	VARIANT	Byte数组, 字符串	密码, 4个字节

输出数据列表:

属性	类型	含义
OutData1	VARIANT , Byte 数组	如果授权失败, 此处显示失败的代码, 2字节
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功, <=0为命令发送失败

9.10.8 NTAG 卡片读签名

函数原形: Long YW_NTag_ReadSignature(Long ReaderID)

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围 0x0000-0xFFFF，如果未知，则 ReaderID=0

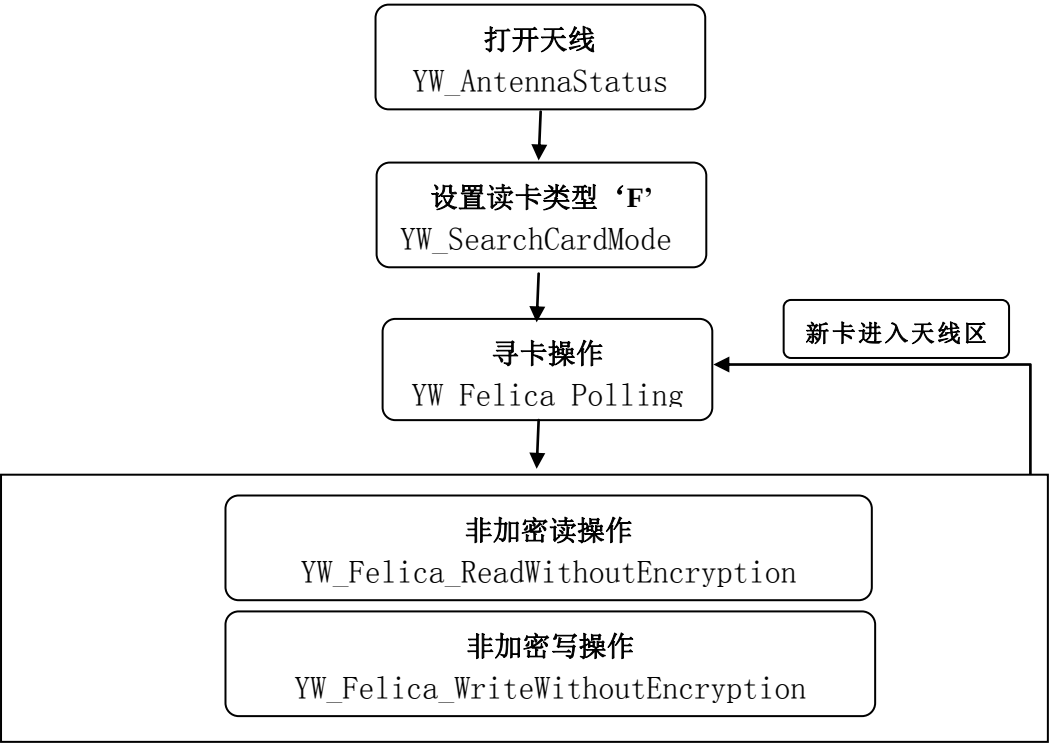
输出数据列表:

属性	类型	含义
OutData1	VARIANT ,Byte 数组	NTAG卡片的签名，32字节
OutData1Size	Long	OutData1的字节长度

返回值: 大于0为命令发送成功，≤0为命令发送失败

9.11 NFC-Type3 Felica 卡

Felica 卡片包括加密的和非加密的卡片操作，目前提供非加密操作 API，后续加密 Felica 操作根据用户需求而增加。非加密 Felica 操作流程如下：



(图-14 TOPAZ卡 操作流程图)

9.11.1 Felica 卡片寻卡

函数原形：`Long YW_Felica_Polling(Long ReaderID, unsigned short SystemCode, BYTE RequestCode, BYTE Slot)`

参数列表：

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID，范围 0x0000-0xFFFF，如果未知，则 ReaderID=0
SystemCode	short	数字	Felica参数，System Code
RequestCode	BYTE	数字	Felica参数，Request Code 0x00: No Request

			0x01: System Code Request 0x02: Communication Performance Request 其他: RFU
Slot	BYTE	数字	最大响应时间, 可以设置的值: 0x00, 0x01, 0x03, 0x07, 0x0F

输出数据列表:

属性	类型	含义
OutData1	VARIANT ,Byte 数组	IDM, 卡号, 8字节
OutData1Size	Long	OutData1的字节长度
OutData2	VARIANT ,Byte 数组	PMm, 8字节
OutData2Size	Long	OutData2的字节长度
OutData3	VARIANT ,Byte 数组	只有当RequestCode=0x01和0x02时, 才会返回Request data, 2字节
OutData3Size	Long	OutData3的字节长度

返回值: 大于0为命令发送成功, <=0为命令发送失败

9.11.2 Felica 卡片不加密读操作

函数原形: `Long YW_Felica_ReadWithoutEncryption(Long ReaderID, VARIANT IDM, BYTE NumberofService, VARIANT ServiceCodeList, BYTE NumberofBlock, BYTE LengthofBlockList, VARIANT BlockList)`

参数列表:

参数	类型	可接受类型	含义
ReaderID	Long	数字	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
IDM	VARIANT	Byte数组	Felica卡卡号, 8字节

NumberOfService	BYTE	数字	Service数量
ServiceCodeList	VARIANT	Byte数组	Service内容，字节数为2* NumberOfService
NumberOfBlock	BYTE	数字	Block块的数量
LengthofBlockList	BYTE	数字	Block列表的字节长度
BlockList	VARIANT	Byte数组	Block列表的内容

输出数据列表：

属性	类型	含义
OutData1	VARIANT ,Byte 数组	读到的块数据，只有当函数返回>0时，此处才有返回，字节数为 NumberOfBlock*16，如果失败，请查看状态Status1，Status2
OutData1Size	Long	OutData1的字节长度
OutData2	VARIANT ,Byte 数据	读取后的状态字节Status1，1字节
OutData2Size	Long	OutData2的字节长度
OutData3	VARIANT ,Byte 数据	读取后的状态字节Status2，1字节
OutData3Size	Long	OutData3的字节长度

返回值：大于0为命令发送成功，≤0为命令发送失败

9.11.3 Felica 卡片不加密写操作

函数原形：Long YW_Felica_WriteWithoutEncryption(Long ReaderID, VARIANT IDM, BYTE NumberOfService, VARIANT ServiceCodeList, BYTE NumberOfBlock, BYTE LengthofBlockList, VARIANT BlockList, VARIANT pBlockData)

参数列表：

参数	类型	可接受类型	含义
----	----	-------	----

ReaderID	Long	数字	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
IDM	VARIANT	Byte数组	Felica卡卡号，8字节
NumberOfService	BYTE	数字	Service数量
ServiceCodeList	VARIANT	Byte数组	Service内容，字节数为2*NumberOfService
NumberOfBlock	BYTE	数字	Block块的数量
LengthofBlockList	BYTE	数字	Block列表的字节长度
BlockList	VARIANT	Byte数组	Block列表的内容
pBlockData	VARIANT	Byte数组	要写入的块数据，字节数为NumberOfBlock*16

输出数据列表：

属性	类型	含义
OutData1	VARIANT , Byte 数据	如果写入失败，状态字节Status1, 1字节
OutData1Size	Long	OutData1的字节长度
OutData2	VARIANT , Byte 数据	如果写入失败，状态字节Status2, 1字节
OutData2Size	Long	OutData2的字节长度

返回值：大于0为命令发送成功，≤0为命令发送失败

10 订购方式

可以通过我们的网站或电话订购。或者联系当地的经销商。

北京友我科技有限公司

网站: <http://www.youwokeji.com.cn>

电话: 010-59395668 (传真)

手机: 18910685939 、 13671114914 、 13691531038

Email: coodor@126.com

附录 A

MifarePlus Level3 的数据及密钥存储结构和 Mifareone 有所区别，结构如下：

块相对地址		块地址	AES 对应密钥块地址
Sector0			
Block0	数据块	0x0000	A 密钥：0x4000 B 密钥：0x4001
Block1	数据块	0x0001	
Block2	数据块	0x0002	
Block3	Level1 :KeyA+KeyB, Level3: data+config	0x0003	
Sector1			
Block0	数据块	0x0004	A 密钥：0x4002 B 密钥：0x4003
Block1	数据块	0x0005	
Block2	数据块	0x0006	
Block3	Level1 :KeyA+KeyB, Level3: data+config	0x0007	
....			
Sector31			
Block0	数据块	0x007C	A 密钥：0x403E B 密钥：0x403F
Block1	数据块	0x007D	
Block2	数据块	0x007E	
Block3	Level1 :KeyA+KeyB, Level3: data+config	0x007F	
Sector32			
Block0	数据块	0x80	A 密钥：0x4040 B 密钥：0x4041
Block1	数据块	0x81	
...	数据块	...	
Block15	Level1 :KeyA+KeyB, Level3: data+config	0x8F	
...			
Sector39			
Block0	数据块	0xF0	A 密钥：0x404E B 密钥：0x404F
Block1	数据块	0xF1	
...	数据块	...	
Block15	Level1 :KeyA+KeyB, Level3: data+config	0xFF	
配置块			
	MFP Configuration Block	0xB000	
	Installation Identifier	0xB001	

	ATS Information	0xB002	
	Field Configuration Block	0xB003	
Key 块			
	AES Sector Keys	0x4000~0x403F	
	AES Sector Keys	0x4040~0x404F	
	Originality Key	0x8000	
	Card Master Key	0x9000	
	Card Configuration Key	0x9001	
	Level2 switch Key	0x9002	
	Level3 switch Key	0x9003	
	SL1 Card Authentication Key	0x9004	
	Select VC Key	0xA000	
	Proximity Check Key	0xA001	
	VC Polling ENC Key	0xA080	
	VC Polling MAC Key	0xA081	

注意：

- 1、蓝色和黄色部分是关联部分。即数据区和密钥区对应部分(仅仅是在 Level 2/3 才对应，因只有级别 2/3 才使用到 AES 密钥认证)。
- 2、在安全级别 Level 1，是和 Mifare classic 兼容的，每个扇区最后一块为密钥和配置块。
- 3、AES 密钥分为 A/B 密钥是人为划分，是为了同 Mifare classic 概念相同。在 PLUS 内部一个扇区是对应地址连续的 AES 密钥块。
- 4、主要掌握如下 key：

AES Sector Keys：

在 Level2/3 中对数据的授权采用 AES Key 授权。该密钥可以在 Level0 写入，或者通过 AES Sector Keys 对卡片授权而修改 AES Key。

CardMasterKey：

通过该 Key 的授权，可以改变 **Card Configuration Key** 和 **Level2/3 switch Key**

Card Configuration Key：

通过该 key 的授权，可以改变 MFP Configuration Block 配置块内容。

Level2 switch Key：

通过该 key 的授权，可以从 Level1 切换 Level2。

Level3 switch Key：

通过对该 key 的授权，可以从 Level2 切换 Level3，或从 Level1 切换到 Level3。

- 5、在 Level0，除了出厂写入的用户不能修改的密钥外，都可以以明文方式写入，一般在 Level0 做初始化操作。注意，必须在该安全级别写入 0x9000~0x9003 块。
- 6、Level3 级别支持明文、AES 加密、加密且带 MAC 方式读写方式。本读卡器采用的是最保密的方式读写 MifarePlus 块：*加密且带 MAC 方式*。