IC卡读卡器开发指南

1 概述

随着社会的发展和科技的进步，IC卡应用越来越广泛。会员卡，学生卡，社保卡，公交卡，金融卡等已得到大量的应用，可以说IC卡在我们身边无处不在。IC卡的普及一方面取决于各个组织的大力推广，另外也得力于无数程序员的辛苦工作，将IC卡读卡器和IC卡融合到各个系统中。本文将着重介绍IC卡读卡器在实际应用中编程的流程和步骤，使得IC卡的开发简单明了，更缩短我们的开发周期。

2 非接触IC卡介绍

非接触 IC 卡是 IC 卡中的一种，由于非接触 IC 卡没有物理磨损，寿命更长，价格更便宜，使得非接触 IC 卡的使用越来越多，最具有代表性的非接触 IC 卡就是 M1 卡及其兼容卡。下面将以 M1 卡为例，来介绍 M1 的内部结构。
M1 卡有 1k 和 4k 之分，内部结构基本差不多，以 M1 卡 1K 为例：
该 IC 卡共有 16 个扇区，每个扇区 4 个块，每块 16 字节，所以总共有字节数 16*4*16=1024 个字节。

第 15 扇区	第 63 块	密钥块
	第 62 块	数据块
	第 61 块	数据块
	第 60 块	数据块
	. . .	
第 1 扇区	第 7 块	密钥块
	第 6 块	数据块
	第 5 块	数据块
	第 4 块	数据块
第 0 扇区	第 3 块	密钥块
	第 2 块	数据块
	第 1 块	数据块
	第 0 块	存卡号，只读

每个扇区最后一块用来保存密钥，故不能当作数据来使用。第0块由于是只读的，也不能用

来存取数据，所以此卡实际可用的内存为（16*3-1）*16= 752字节。

在读写某个数据块之前，必须首先进行密钥认证，如果密钥认证失败，则不能读写，只有认证成功，方可进行读写等操作。每个扇区共用一组密钥，所以一个扇区只要认证成功一次，就可以读写此扇区中的四个数据块。

每个可用的数据块可以初始化为整形值或者原始数据。原始数据可以当作普通内存使用，整形值可以当作钱包等具有加减功能数字使用。

3 IC卡读卡器介绍

IC 卡读卡器的种类繁多，这里以性价比较高的 YW-605 系列读卡器来作介绍。
YW-605 系列读卡器具有多种接口，外观简洁，美观大方，可以读写市面上大部分非接触 IC 卡。

4 IC卡读卡器API函数介绍

YW-605系列读卡器提供二次开发功能，用户可以在我们的DLL的基础上调用相应的函数开发应用程序，我们提供Delphi,C++Builder,VB,VC等的调用例程和相关函数声明单元，或者按照读卡器的通信协议直接开发应用程序。

库函数，C++语言版，其它语言见相应的函数声明文件。

动态库及读写器相关函数

1. 读取库函数内部版本号

函数原形：`int stdcall YW_GetDLLVersion(void);`

参数列表：无

返 回 值：大于0为版本号，小于0为错误

2. DES加解密函数

函数原形：
```
int stdcall DES(unsigned char cModel, unsigned char *pkey,
        unsigned char *in, unsigned char *out);
```

参数列表：

参数	类型	含义
cModel	unsigned char	加解密方向，0为加密，1为解密
pkey	unsigned char*	加解密秘钥，8个字节
in	unsigned char*	原始数据，8个字节
out	unsigned char*	加解密后的数据，8个字节

返 回 值：无意义

3. 3DES加解密函数

函数原形：int stdcall DES3(unsigned char cModel, unsigned char *pKey, unsigned char *In, unsigned char *Out);

参数列表：

参数	类型	含义
cModel	unsigned char	加解密方向，0为加密，1为解密
pkey	unsigned char*	加解密秘钥，16个字节
in	unsigned char*	原始数据，8个字节
out	unsigned char*	加解密后的数据，8个字节

返 回 值：无意义

4. 带向量的3DES加解密函数

函数原形：int stdcall DES3_CBC(unsigned char cModel, unsigned char *pKey,unsigned char *In, unsigned char *Out, unsigned char *pIV);

参数列表：

参数	类型	含义
cModel	unsigned char	加解密方向，0为加密，1为解密
pkey	unsigned char*	加解密秘钥，16个字节
in	unsigned char*	原始数据，8个字节
out	unsigned char*	加解密后的数据，8个字节
pIV	unsigned char*	加解密向量，8个字节

返 回 值：无意义

5. 初始化串口

函数原形：int stdcall YW_ComInitial(int PortIndex, int Baud);

参数列表：

参数	类型	含义
PortIndex	int	串口号，1--255
Baud	int	通信波特率，2400—115200，默认为19200

返 回 值：1成功，0失败

6. 释放串口

函数原形：int stdcall YW_ComFree(void);

参数列表：无

返 回 值：1成功，0失败

7. USB无驱读写器，初始化USB

函数原形：`int stdcall YW_USBHIDInitial(void);`

参数列表：无

返 回 值：1成功，0失败

8. USB无驱读写器，释放USB

函数原形：`int stdcall YW_USBHIDFree(void);`

参数列表：无

返 回 值：1成功，0失败

9. 修改读写器串口波特率

函数原形：`int stdcall YW_ComNewBound(int ReaderID ,int NewBound);`

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围 0x0000-0xFFFF，如果未知，则 ReaderID=0
NewBound	int	新的波特率 0x01->9600bps 0x02->14400bps 0x03->19200bps 0x04->28800bps 0x05->38400bps 0x06->57600bps 0x07->115200bps

返 回 值：1成功，0失败

10. 设置设备标识

函数原形：`int stdcall YW_SetReaderID(int OldID, int NewID);`

参数列表：

参数	类型	含义
OldID	int	老的设备标示ID，范围 0x0000-0xFFFF
NewID	int	修改成新的设备标示ID，范围 0x0000-0xFFFF

返 回 值：1成功，0失败

11. 查询设备标识

函数原形：`int stdcall YW_GetReaderID(int ReaderID);`

参数列表：

参数	类型	含义
ReaderID	int	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

12. 读取读卡器内部版本号

函数原形：`int stdcall YW_GetReaderVersion(int ReaderID);`

参数列表：

参数	类型	含义
ReaderID	int	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，小于0为错误

13. 查询读写器产品序列号

函数原形：int stdcall YW_GetReaderSerial(int ReaderID, char *ReaderSerial);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
ReaderSerial	Char *	读取的产品序列号，长度为8个字节

返 回 值：大于0为成功，小于0为失败

14. 蜂鸣器控制函数

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

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
Time_ON	int	蜂鸣器鸣叫时间，单位：秒
Time_OFF	int	蜂鸣器静音时间，单位：秒
Cycle	int	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，小于0为命令发送失败

15. LED指示灯控制

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

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
LEDIndex	int	LED灯序号 01：红灯 02：绿灯 04：黄灯
Time_ON	int	LED灯亮时间，单位：秒
Time_OFF	int	LED灯灭时间，单位：秒
Cycle	int	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	int	最后要亮的灯： 00：全灭 01：红灯 02：绿灯 04：黄灯

返 回 值：大于0为命令发送成功，小于0为命令发送失败

16. 设置LED显示器显示的内容

函数原形：int stdcall YW_LEDDisplay(int ReaderID,int Alignment,char *LEDText);

参数列表：

参数	类型	含义

ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
Alignment	int	显示时的对齐方式： 1：左对齐 2：居中对齐 3：右对齐
LEDText	Char *	要显示的字符串。 可显示的字符如下： 0123456789AbCdEF.-

返 回 值：大于0为命令发送成功，小于0为命令发送失败

17. 设置天线的状态

函数原形：`int stdcall YW_AntennaStatus(int ReaderID,bool Status);`

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
Status	bool	True: 开天线 False:关天线

返 回 值：大于0为命令发送成功，小于0为命令发送失败

18. 设置寻卡模式

函数原形：`int stdcall YW_SearchCardMode(int ReaderID,int SearchMode);`

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围 0x0000-0xFFFF，如果未知，则 ReaderID=0
SearchMode	char	卡类型 0x41-----ISO14443A 0x42----- ISO14443B 0x31----- ISO15693 0x53------ST系列卡 0x52------AT88RF020等

返 回 值：大于0为命令发送成功，小于0为命令发送失败

ISO14443A相关函数

19. 寻卡

函数原形：int stdcall YW_RequestCard(int ReaderID,char RequestMode,unsigned short *CardType);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围 0x0000-0xFFFF，如果未知，则 ReaderID=0
RequestMode	char	寻卡的模式 0x52----- 所有卡 0x26----- 激活卡
CardType	unsigned short	返回卡的类型 0x4400 = Ultralight/UltraLight C /MifarePlus(7Byte UID) 0x0400 = Mifare Mini/Mifare 1K (S50) /MifarePlus(4Byte UID)

	rt *	0x0200 = Mifare_4K(S70)/ MifarePlus(4Byte UID) 0x0800 = Mifare_Pro 0x0403 = Mifare_ProX 0x4403 ->Mifare_DESFire 0x4200 -> MifarePlus(7Byte UID)

返 回 值：大于0为命令发送成功，小于0为命令发送失败

20. Type A卡访冲突

函数原形：int stdcall YW_AntiCollide(int ReaderID,unsigned char *LenSNO, unsigned char *SNO)

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
LenSNO	unsigned char*	访冲突获得卡号的长度
SNO	unsigned char *	访冲突获得卡号

返 回 值：大于0为命令发送成功，小于0为命令发送失败

21. Type A选卡

函数原形：int stdcall YW_CardSelect(int ReaderID,char LenSNO, unsigned char *SNO)

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围

		0x0000-0xFFFF，如果未知，则ReaderID=0
LenSNO	unsigned char	选择卡的卡号长度
SNO	unsigned char *	要选择的卡号

返 回 值：大于0为命令发送成功，小于0为命令发送失败

22. 访冲撞读卡序列号并且选定一张卡

函数原形：int stdcall YW_AntiCollideAndSelect(int ReaderID, unsigned char MultiCardMode, unsigned char *CardMem, int *SNLen, unsigned char *SN);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
MultiCardMode	unsigned char	对多张卡的处理方式 0：多张卡返回错误 1：返回一张卡号
CardMem	unsigned char *	卡片容量代码
SNLen	int *	输出卡号的长度
SN	unsigned char *	输出卡的序列号

返 回 值：大于0为命令发送成功，小于0为命令发送失败

23. 寻卡、访冲撞读卡序列号并且选定一张卡

函数原形：int stdcall YW_RequestAntiandSelect(int ReaderID,int SearchMode,int MultiCardMode,unsigned short *ATQA,unsigned char *SAK,unsigned char *LenSNO,unsigned char *SNO);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
RequestMode	unsigned char	寻卡的模式 0x52 所有卡 0x26 激活卡
MultiCardMode	unsigned char	对多张卡的处理方式 0：多张卡返回错误 1：返回一张卡号
ATQA	unsigned short *	ATQA值
SAK	unsigned char *	SAK值
SNLen	int *	输出卡号的长度
SN	unsigned char *	输出卡的序列号

返 回 值：大于0为命令发送成功，小于0为命令发送失败

24. Type A卡n级访冲突

函数原形：int stdcall YW_AntiCollide_Level(int ReaderID,int Leveln,char *LenSNO, unsigned char *SNO);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
Leveln	int	访冲突级别，最高为3级
LenSNO	unsigned char*	访冲突获得卡号的长度

SNO	unsigned char *	访冲突获得卡号

返 回 值：大于0为命令发送成功，小于0为命令发送失败

25. Type A卡n级选卡

函数原形：int stdcall YW_SelectCard_Level(int ReaderID,int Leveln,unsigned char *SAK)

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
Leveln	int	访冲突级别，最高为3级
SAK	unsigned char*	SAK值

返 回 值：大于0为命令发送成功，小于0为命令发送失败

S50/S70卡相关操作

26. 下载秘钥到只写区

函数原形：int stdcall YW_ DownLoadKey(int ReaderID, int KeyIndex,char * Key);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
KeyIndex	int	只写区秘钥序号0~31，共可写32

		个秘钥
Key	char *	秘钥，每个秘钥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

27. 用只写区秘钥验证扇区秘钥

函数原形：int stdcall YW_KeyDown_Authorization (int ReaderID, char KeyMode , int BlockAddr,int KeyIndex);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
KeyMode	char	KeyMode=0x60为A密钥 KeyMode=0x61为B密钥
BlockAddr	int	要验证的绝对块号地址
KeyIndex	int	只写区秘钥序号0~31

返 回 值：大于0为命令发送成功，小于0为命令发送失败

28. 验证某扇区密钥

函数原形：int stdcall YW_KeyAuthorization (int ReaderID,char KeyMode,int BlockAddr, unsigned char *Key);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
KeyMode	char	KeyMode=0x60为A密钥 KeyMode=0x61为B密钥
BlockAddr	int	要验证的绝对块号地址

Key	unsigned char *	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

29. 读取一块数据

函数原形：int stdcall YW_ReadaBlock (int ReaderID,int BlockAddr,int LenData, unsigned char *Data);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockAddr	int	绝对地址块号
LenData	int	要读出的数据的字节数，Mifare One为16个字节
Data	unsigned char *	输出读到的块的数据

返 回 值：大于0为命令发送成功，小于0为命令发送失败

30. 写入一块数据

函数原形：int stdcall YW_WriteaBlock (int ReaderID,int BlockAddr,int LenData, unsigned char *Data);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockAddr	int	绝对块号地址
LenData	int	要写入的数据的字节数，

		Mifare One为16个字节
Data	unsigned char *	要写入的块的数据

返 回 值：大于0为命令发送成功，小于0为命令发送失败

31. 将某一扇区初始化为钱包

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

参数列表：

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

32. 读取钱包值

函数原形：int stdcall YW_Purse_Read (int ReaderID,int BlockAddr,int *Money);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockAddr	int	绝对块号地址
Money	Int *	读取的块号钱包的当前值

返 回 值：大于0为命令发送成功，小于0为命令发送失败

33. 钱包扣款

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

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockAddr	int	绝对块号地址
Decrement	Int	钱包中要扣掉的值

返 回 值：大于0为命令发送成功，小于0为命令发送失败

34. 钱包充值

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

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockAddr	int	绝对块号地址
Charge	Int	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

35. Restore 命令

函数原形：int stdcall YW_Restore (int ReaderID,int BlockAddr);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范

		围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockAddr	int	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

36. Transfer命令

函数原形：`int stdcall YW_Transfer (int ReaderID,int BlockAddr);`

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockAddr	int	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

37. 读取多块数据

函数原形：`int stdcall YW_ReadM1MultiBlock(int ReaderID, int StartBlock, int BlockNums, int *LenData, char *pData);`

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
StartBlock	int	绝对地址开始块号
BlockNums	int	块的数量
LenData	Int*	要读出的数据的字节数
Data	unsigned char *	输出读到的块的数据

返 回 值：大于0为命令发送成功，小于0为命令发送失败

38. 写多块数据

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

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
StartBlock	int	绝对地址开始块号
BlockNums	int	块的数量
LenData	int	要写入的数据的字节数，Mifare One为16* BlockNums个字节
Data	unsigned char *	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

UltraLight卡操作函数

39. 读取UltraLight卡块数据

函数原形：int stdcall YW_UltraLightRead(int ReaderID, int BlockID, unsigned char *pData);

参数列表：

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

BlockID	int	绝对地址块号
pData	unsigned char *	输出读到的块的数据，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

40. 写UltraLight卡块数据

函数原形：int stdcall YW_UltraLightWrite(int ReaderID, int BlockID, unsigned char *pData);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockID	int	绝对地址块号
pData	unsigned char *	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

Type A CPU卡操作函数

41. Type A C P U 卡复位

函数原形：int stdcall YW_TypeA_Reset(int ReaderID, unsigned char Mode, unsigned char MultiMode, int *rtLen, unsigned char *pData);

参数列表：

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

Mode	unsigned char	寻卡的模式 0x52 所有卡 0x26 激活卡
MultiMode	unsigned char	对多张卡的处理方式 0：多张卡返回错误 1：返回一张卡号
rtLen	int *	返回复位信息的长度
pData	unsigned char *	返回复位信息

返 回 值：大于0为命令发送成功，小于0为命令发送失败

42. Type A CPU 卡执行COS命令

函数原形：int stdcall YW_TypeA_COS(int ReaderID, int LenCOS, unsigned char *Com_COS, int *rtLen, unsigned char *pData);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
LenCOS	unsigned char*	输入的COS命令的长度
Com_COS	unsigned char*	COS命令
rtLen	int *	返回执行命令结果的长度
pData	unsigned char *	返回执行命令结果

返 回 值：大于0为命令发送成功，小于0为命令发送失败

Mifare Plus卡操作函数

43. Mifare Plus卡Level 0级写数据

函数原形：int stdcall YW_MFP_L0_WritePerso(int ReaderID, int Address, unsigned char *pData);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
Address	unsigned char	要写入数据的地址
Com_ pData	unsigned char*	要写入的数据， 1 6 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

44. Mifare Plus卡Level 0级向Level 1或3级切换

函数原形：int stdcall YW_MFP_L0_CommitPerso(int ReaderID);

参数列表：

参数	类型	含义
ReaderID	int	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

45. Mifare Plus卡从低级向高级切换

函数原形：int stdcall YW_MFP_SwitchToLevel(int ReaderID, int DesLevel,unsigned char *SwitchKey);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
DesLevel	unsigned char	要切换到的层级，最高 3 级
SwitchKey	unsigned char*	切换秘钥， 1 6 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

46. Mifare Plus卡Level 3级授权

函数原形：int stdcall YW_MFP_L3_Authorization(int ReaderID, int KeyMode,int BlockID,unsigned char *Key);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
KeyMode	unsigned char	KeyMode=0x60为A密钥 KeyMode=0x61为B密钥
BlockID	unsigned char	块号
Key	unsigned char*	秘钥， 1 6 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

47. Mifare Plus卡Level 3级读块数据

函数原形：int stdcall YW_MFP_L3_Read(int ReaderID, int StartBlock,int BlockNums,int *DataLen, unsigned char *pData);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
StartBlock	int	开始块号
BlockNums	int	块数量
DataLen	int*	读到的数据长度
pData	unsigned char*	读到的数据

返 回 值：大于0为命令发送成功，小于0为命令发送失败

48. Mifare Plus卡Level 3级写块数据

函数原形：int stdcall YW_MFP_L3_Write(int ReaderID, int StartBlock,int BlockNums, unsigned char *pData);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
StartBlock	int	开始块号
BlockNums	int	块数量
pData	unsigned char*	要写入的数据，长度必须是 1 6 * BlockNums

返 回 值：大于0为命令发送成功，小于0为命令发送失败

49. Mifare Plus卡Level 3级将某一扇区初始化为钱包

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

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockID	int	绝对块号地址
InitialValue	int	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

50. Mifare Plus卡Level 3级读取钱包值

函数原形：int stdcall YW_MFP_L3_Purse_Read(int ReaderID, int BlockID,int *Value);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockID	int	绝对块号地址
Value	Int *	读取的块号钱包的当前值

返 回 值：大于0为命令发送成功，小于0为命令发送失败

51. Mifare Plus卡Level 3级钱包扣款

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

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范

		围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockID	int	绝对块号地址
Value	Int	钱包中要扣掉的值

返 回 值：大于0为命令发送成功，小于0为命令发送失败

52. Mifare Plus卡Level 3级钱包充值

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

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockID	int	绝对块号地址
Value	Int	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

53. Mifare Plus卡Level 3级备份钱包

函数原形：int stdcall YW_MFP_L3_Purse_Backup(int ReaderID, int BlockID,int DesBlockID);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockID	int	要备份的钱包块号
DesBlockID	Int	目标块号

返 回 值：大于0为命令发送成功，小于0为命令发送失败

54. Mifare Plus卡Level 3级通用读写第一次授权

函数原形：int stdcall YW_MFP_Authorization_First(int ReaderID, int AESKeyAddr,unsigned char *AESKey);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
AESKeyAddr	int	要授权的地址
AESKey	unsigned char *	授权秘钥， 1 6 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

55. Mifare Plus卡Level 3级通用读写第二次授权

函数原形：int stdcall YW_MFP_Authorization_Follow(int ReaderID, int AESKeyAddr,unsigned char *AESKey);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
AESKeyAddr	int	要授权的地址
AESKey	unsigned char *	授权秘钥， 1 6 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

56. Mifare Plus卡Level 3级通用读块

函数原形：int stdcall YW_MFP_CommonRead(int ReaderID, int BlockID,

int BlockNums,int *DataLen,unsigned char *pData);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockID	int	通用块地址
BlockNums	int	块数量
DataLen	Int*	返回的数据长度
pData	unsigned char *	返回的数据

返 回 值：大于0为命令发送成功，小于0为命令发送失败

57. Mifare Plus卡Level 3级通用写块

函数原形：int stdcall YW_MFP_CommonWrite(int ReaderID, int BlockID, int BlockNums, unsigned char *pData);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
BlockID	int	通用块地址
BlockNums	int	块数量
pData	unsigned char *	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

SAM卡操作函数

58. SAM卡波特率设置

函数原形：int __stdcall YW_SAM_ResetBaud(int ReaderID,int SAMIndex, int BaudIndex);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
SAMIndex	int	SAM卡序号
BaudIndex	int	0x00->9600（默认复位波特率） 0x01->19200 0x02->38400 0x03->55800 0x04->57600 0x05->115200

返 回 值：大于0为成功，小于0为失败

59. SAM卡复位

函数原形：int __stdcall YW_SAM_Reset(int ReaderID,int SAMIndex, int *rtLen, unsigned char *pData);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
SAMIndex	int	SAM卡序号
rtLen	int *	SAM卡复位返回的数据pData的长度
pData	unsigned char *	SAM卡复位返回的数据

返 回 值：大于0为成功，小于0为失败

60. SAM卡执行COS命令

函数原形：int __stdcall YW_SAM_COS(int ReaderID,int SAMIndex, int LenCOS, unsigned char *Com_COS, int *rtLen, unsigned char *pData);

参数列表：

参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
SAMIndex	int	SAM卡序号
LenCOS	int	向SAM卡要发送的COS命令的长度
Com_COS	unsigned char *	向SAM卡要发送的COS命令
rtLen	unsigned char *	SAM执行COS命令后返回的数据的长度
pData	unsigned char *	SAM执行COS命令后返回的数据

返 回 值：大于0为成功，小于0为失败

61. SAM卡PPS波特率设置

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

参数列表：

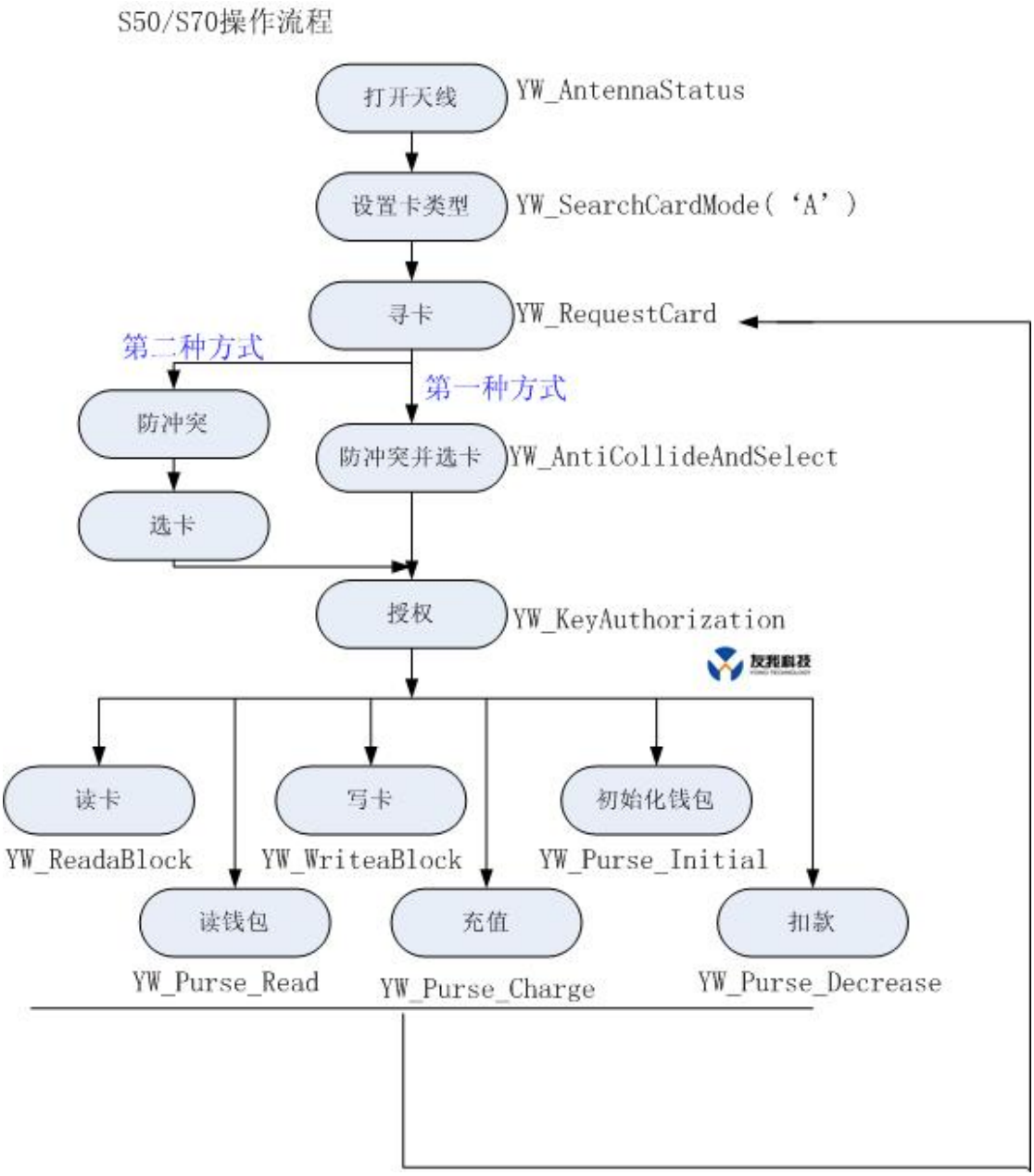
参数	类型	含义
ReaderID	int	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0

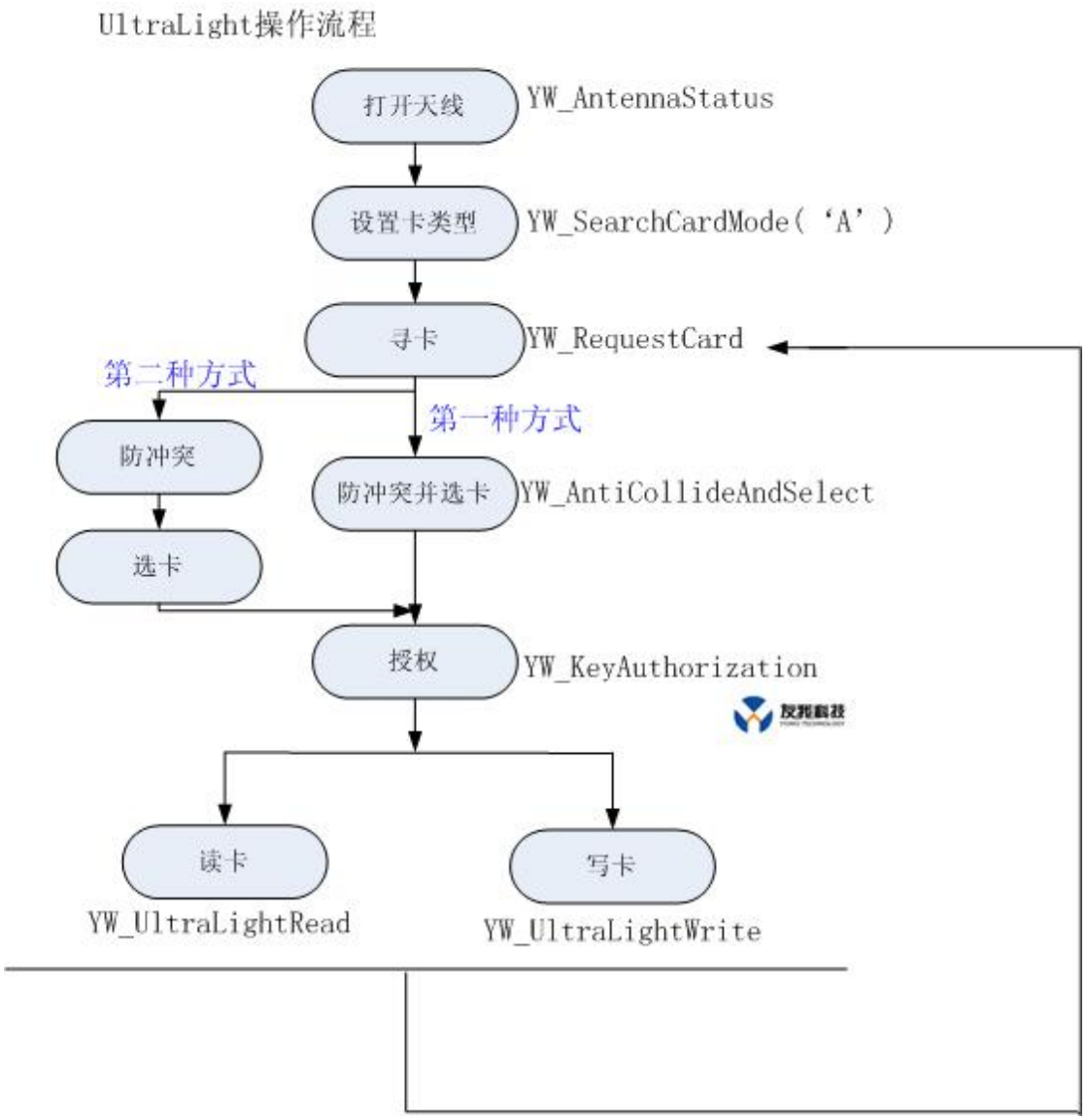
SAMIndex	int	SAM卡序号
BaudIndex	int	0x00->9600（默认复位波特率) 0x01->19200 0x02->38400 0x03->55800 0x04->57600 0x05->115200

返 回 值：大于0为成功，小于0为失败

5 读卡操作流程

YW605 在所有卡操作之前必须打开天线，读完卡后可关闭天线，也可以不关闭天线。对卡的操作流程如下图所示：


S50/S70操作流程
打开天线
YW_AntennaStatus
设置卡类型
YW_SearchCardMode（‘A’）
寻卡
YW_RequestCard
第二种方式
第一种方式
防冲突
防冲突并选卡
YW_AntiCollideAndSelect
选卡
授权
YW_KeyAuthorization
读卡
写卡
初始化钱包
YW_ReadaBlock
YW_WriteaBlock
YW_Purse_Initial
读钱包
充值
扣款
YW_Purse_Read
YW_Purse_Charge
YW_Purse_Decrease


UltraLight操作流程
打开天线
YW_AntennaStatus
设置卡类型
YW_SearchCardMode('A')
寻卡
YW_RequestCard
第二种方式
第一种方式
防冲突
防冲突并选卡
YW_AntiCollideAndSelect
选卡
授权
YW_KeyAuthorization
读卡
写卡
YW_UltraLightRead
YW_UltraLightWrite

6 程序开发注意事项

6.1 YW-605 系列读卡器具有多种接口，不同的接口，SDK中端口初始化函数有所区别。

- 串口，RS485，USB虚拟的串口的读卡器端口操作函数为：

打开端口：int stdcall YW_ComInitial(int PortIndex, int Baud);

释放端口：int stdcall YW_ComFree(void);

- USB HID的读卡器端口操作函数为：

打开端口：int stdcall YW_USBHIDInitial(void);

释放端口：int stdcall YW_USBHIDFree(void);

7 更多帮助

更多帮助请联系友我科技技术支持，或者QQ:896163157.