

# 13.56M射频卡读卡器YW-610系列

## RS485

### 使用手册

(version 1.4)

ISO14443A

- 简体中文版



网址:	<a href="http://www.youwokeji.com.cn">http://www.youwokeji.com.cn</a>
EMAIL:	coodor@126.com
电话:	010-59395668
手机:	013671114914
联系人:	周先生

## 目录

1	概述.....	4
2	YW-610 系列型号.....	4
3	功能特点.....	5
4	技术指标.....	6
5	二次开发指南.....	6
	动态库及读写器相关函数.....	6
1.	读取库函数内部版本号.....	6
2.	DES加解密函数.....	7
3.	3DES加解密函数.....	7
4.	带向量的 3DES加解密函数.....	8
5.	初始化串口.....	8
6.	释放串口.....	9
7.	USB无驱读写器, 初始化USB.....	9
8.	USB无驱读写器, 释放USB.....	9
9.	修改读写器串口波特率.....	9
10.	设置设备标识.....	10
11.	查询设备标识.....	10
12.	读取读卡器内部版本号.....	10
13.	查询读写器产品序列号.....	11
14.	蜂鸣器控制函数.....	11
15.	LED指示灯控制.....	12
16.	LCD清屏控制.....	12
17.	设置LCD上要显示的内容.....	13
18.	设置读写器的时间.....	13
19.	获取读写器的时间.....	14
20.	设置天线的状态.....	14
21.	设置寻卡模式.....	15
	ISO14443A相关函数.....	15
22.	寻卡.....	15
23.	Type A卡访冲突.....	16
24.	Type A选卡.....	16
25.	访冲撞读卡序列号并且选定一张卡.....	17
26.	寻卡、访冲撞读卡序列号并且选定一张卡.....	18
27.	Type A卡n级访冲突.....	18
28.	Type A卡n级选卡.....	19
	S50/S70 卡相关操作.....	19
29.	下载密钥到只写区.....	19
30.	用只写区密钥验证扇区密钥.....	20
31.	验证某扇区密钥.....	20
32.	读取一块数据.....	21
33.	写入一块数据.....	21

34.	将某一扇区初始化为钱包.....	22
35.	读取钱包值.....	22
36.	钱包扣款.....	23
37.	钱包充值.....	23
38.	Restore 命令.....	23
39.	Transfer命令.....	24
40.	读取多块数据.....	24
41.	写多块数据.....	25
UltraLight卡操作函数.....		25
42.	读取UltraLight卡块数据.....	25
43.	写UltraLight卡块数据.....	26
44.	Type A CPU 卡复位.....	26
45.	Type A CPU 卡执行COS命令.....	27
Mifare Plus卡操作函数.....		28
46.	Mifare Plus卡Level 0 级写数据.....	28
47.	Mifare Plus卡Level 0 级向Level 1 或 3 级切换.....	28
48.	Mifare Plus卡从低级向高级切换.....	28
49.	Mifare Plus卡Level 3 级授权.....	29
50.	Mifare Plus卡Level 3 级读块数据.....	29
51.	Mifare Plus卡Level 3 级写块数据.....	30
52.	Mifare Plus卡Level 3 级将某一扇区初始化为钱包.....	31
53.	Mifare Plus卡Level 3 级读取钱包值.....	31
54.	Mifare Plus卡Level 3 级钱包扣款.....	31
55.	Mifare Plus卡Level 3 级钱包充值.....	32
56.	Mifare Plus卡Level 3 级备份钱包.....	32
57.	Mifare Plus卡Level 3 级通用读写第一次授权.....	33
58.	Mifare Plus卡Level 3 级通用读写第二次授权.....	33
59.	Mifare Plus卡Level 3 级通用读块.....	33
60.	Mifare Plus卡Level 3 级通用写块.....	34
SAM卡操作函数.....		34
61.	SAM卡波特率设置.....	35
62.	SAM卡复位.....	35
63.	SAM卡执行COS命令.....	36
64.	SAM卡PPS波特率设置.....	36
6	读卡操作流程.....	37
7	关于射频卡读写模块.....	39
8	订购方式.....	39

# 1 概述

YW-610系列射频卡读卡器是采用13.56M非接触射频技术设计而成的通用型读卡器，内嵌MF RC530、RC531和CL RC632等一些列原装芯片，读写性能稳定可靠，并且内嵌实时时钟。天线和主板一体化设计，使芯片的读写性能更出色的发挥出来。该系列读卡器采用RS485/RS232等数据传输方式与计算机相连接，使计算机二次开发变得更为简单。

该系列读卡器支持ISO14443-A Mifare One S50, S70, ISO14443B和ISO15693及其兼容卡片。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，门禁考勤读写卡器，汽车电子感应锁配套，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。各种型号支持的卡片请参见第二节。

此款读写器功能可以定制，欢迎来电咨询。



## 2 YW-610 系列型号

13.56M 射频卡读写模块 YW-610 系列有以下型号，请选择合适的射频卡读卡器。

型号	支持接口	读卡协议	支持的射频卡
YW-610SA	RS485	ISO14443-A	Mifare S50,S70

### 3 功能特点

型号.	YW-610
<b>特性:</b>	
读写射频卡	
频率	13.56MHz
协议	ISO 14443 TYPE A
支持卡:	Mifare_One (S50&S70)/ UltraLight / Mifare_ProX;
典型读写时间	<100ms
读些距离	up to 2 -- 5cm (for ISO card size)
电源	DC7-12V, 250mA
接口	RS485
波特率	4800 - 115200 bps(19200 default)
LED 灯	Two LED indicators (software controlled)
蜂鸣器	Built-in (software controlled)
存储器	1M bytes (to store information)
LCD 显示器	128*64 dots, 16 字符*4 行, 可以显示汉字
键盘	0-9 , *, # , and F1, F2, F3, F4 key. 共 16 个键
时钟 RTC	内嵌 RTC 实时时钟
接口	RS 485
输出线	4 Wire (Power, GND, D+, D-)
工作温度	-20-- +80
存储温度	-45 -- + 80
湿度	up to 90%

尺寸	130mm x 85mm x 23mm
重量	300g

## 4 技术指标

- ✎ 串口波特率：19200BPS
- ✎ 电源：DC5V ± 10%
- ✎ 最大功耗：1.5W
- ✎ 环境温度：-30℃ ~ +70℃
- ✎ 相对湿度：35% ~ 95%
- ✎ 外形尺寸：120 \* 84 \* 25 (mm)
- ✎ 重量：约100g

## 5 二次开发指南

YW-610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*	加解密密钥，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*	加解密密钥，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*	加解密密钥, 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 Bound);`

**参数列表:**

参数	类型	含义
PortIndex	int	串口号, 1--255
Bound	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	<code>int</code>	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0
NewBound	<code>int</code>	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	<code>int</code>	所要获取的设备标示ID，范围0x0000-0xFFFF，如果未知，则ReaderID=0
LEDIndex	<code>int</code>	LED灯序号 01：红灯 02：绿灯 04：黄灯
Time_ON	<code>int</code>	LED灯亮时间，单位：秒
Time_OFF	<code>int</code>	LED灯灭时间，单位：秒
Cycle	<code>int</code>	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	<code>int</code>	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. LCD清屏控制

**prototype:** `int stdcall YW_ClearLCD(int ReaderID);`

**Param:**

Param	Type	
ReaderID	<code>int</code>	所要获取的设备标示ID, 范围 0x0000-0xFFFF, 如果未知, 则 ReaderID=0

**Return:** 1大于0为命令发送成功, 小于0为命令发送失败

## 17. 设置LCD上要显示的内容

**prototype:** `int stdcall YW_LCDDisplay(int ReaderID, bool`

`ClearBeforeDisp, int StartPos, int txtLen, char *LCDText);`

**Param:**

Param	Type	
ReaderID	<code>Int</code>	Reader ID
ClearBeforeDisp	<code>Bool</code>	显示前是否清屏
StartPos	<code>Int</code>	开始位置
txtLen	<code>Int</code>	字符串长度
LCDText	<code>Char *</code>	要显示的字符串

**Return:** 1大于0为命令发送成功, 小于0为命令发送失败

## 18. 设置读写器的时间

**prototype:** `int stdcall YW_SetReaderDateTime (int ReaderID, int aYear, int aMonth, int aDay, int aHour, int aMin, int aSec);`

**Param:**

Param	Type	
ReaderID	<code>Int</code>	Reader ID
aYear	<code>Int</code>	年份
aMonth	<code>Int</code>	月份
aDay	<code>Int</code>	日期天

aHour	Int	小时
aMin	Int	分
aSec	Int	秒

**Return:** 1大于0为命令发送成功，小于0为命令发送失败

## 19. 获取读写器的时间

**prototype:** `int stdcall YW_GetReaderDateTime (int ReaderID, int *aYear, int *aMonth, int *aDay, int *aHour, int *aMin, int *aSec);`

**Param:**

Param	Type	
ReaderID	Int	Reader ID
aYear	Int*	年份
aMonth	Int*	月份
aDay	Int*	日期天
aHour	Int*	小时
aMin	Int*	分
aSec	Int*	秒

**Return:** 1大于0为命令发送成功，小于0为命令发送失败

## 20. 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

## 21. 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

### 22. 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) 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

### 23. 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

### 24. 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

## 25. 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

## 26. 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

## 27. 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

## 28. 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

### 29. 下载密钥到只写区

**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	只写区密钥序号0~31，共可写32个密钥
Key	char *	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

### 30. 用只写区密钥验证扇区密钥

**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	只写区密钥序号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

### 31. 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

### 32. 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

### 33. 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

### 34. 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

### 35. 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

### 36. 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

### 37. 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

### 38. 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

### 39. 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

### 40. 读取多块数据

**函数原形:** int stdcall YW\_ReadMIMultiBlock(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	输出读到的块的数据

	char *	
--	--------	--

**返回值:** 大于0为命令发送成功, 小于0为命令发送失败

#### 41. 写多块数据

**函数原形:** `int stdcall YW_WriteMIMultiBlock(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

#### 42. 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

### 43. 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

#### 44. Type A CPU 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

#### 45. 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

### 46. 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, 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

### 47. 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

### 48. 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

**49. 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*	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

**50. 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

### 51. 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 $16 * \text{BlockNums}$

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

## 53. 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	<code>int</code>	所要获取的设备标示ID, 范围0x0000-0xFFFF, 如果未知, 则ReaderID=0
BlockID	<code>int</code>	绝对块号地址
Value	<code>Int *</code>	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

## 54. 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

### 55. 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

### 56. 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

### 57. 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

### 58. 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

### 59. 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

## 60. 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

## 61. 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

## 62. 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	SAM卡复位返回的数据

	char *	
--	--------	--

**返回值:** 大于0为成功, 小于0为失败

### 63. 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

### 64. 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

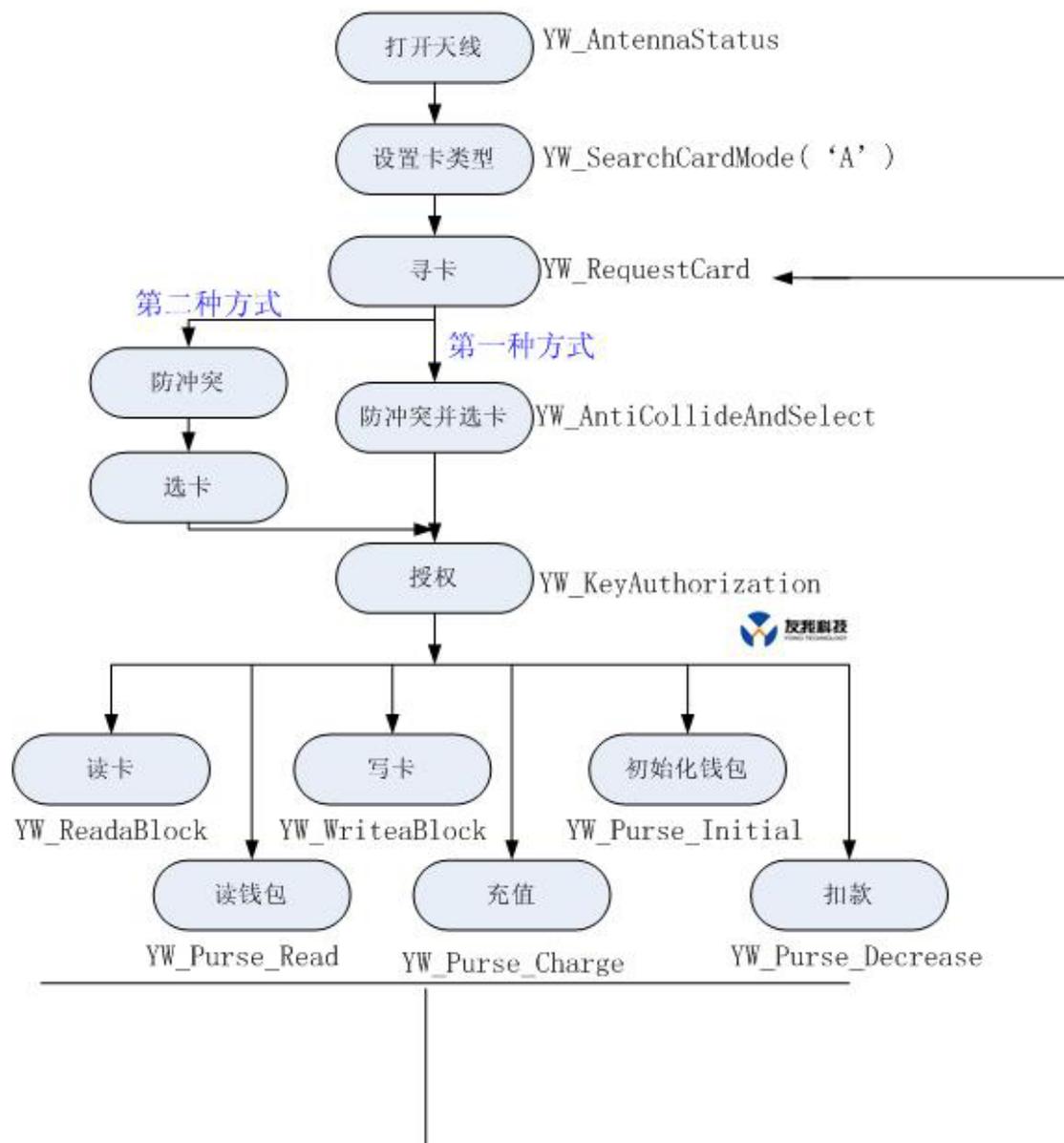
		知, 则ReaderID=0
SAMIndex	int	SAM卡序号
BaudIndex	int	0x00->9600 (默认复位波特率) 0x01->19200 0x02->38400 0x03->55800 0x04->57600 0x05->115200

**返回值:** 大于0为成功, 小于0为失败

## 6 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



## 7 关于射频卡读写模块

如果不在计算机下使用，如果想将读写卡的功能嵌入到你的系统中，可以选择射频卡读写模块YW201, YW202, YW203, YW204等。关于YW-200系列射频卡读写模块请登录网站<http://www.youwokeji.com.cn>查看。

## 8 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

24小时手机: 13671114914, 13691531038

Email: [coodor@126.com](mailto:coodor@126.com)