



---

### **Ordering Information:**

<b>Item No.</b>	<b>Description</b>
YHY638AU	13.56MHz Multi Protocol Reader/Writer USB Interface
YHY638AR	13.56MHz Multi Protocol Reader/Writer RS232 Interface

---

### **1 Supported cards**

ISO 14443 TypeA: Mifare\_One(S50&S70)/UltraLight/Mifare\_ProX

### **2 Special Features**

- Read and write contactless smart cards
- Frequency: 13.56 MHz.
- Typical time to read and write cards: <100ms
- Communications Interface: USB or RS232, baud rate 9600 ~ 115200 bps
- Power supply : DC 5V
- Two LED indicators (software controlled)
- Buzzer alarm (software controlled)
- Mechanic and environmental characteristics:
  - Size: 110 ×81× 26 (mm)
  - Cable length: 1.5m
  - Color: Black

- Operating temperature: 0 ~60 °C
- Storage temperature: - 25 ~80°C
- Relative humidity: up to 90%
- Weight:**90g (YHY638AU)**  
**160g (YHY638AR)**

### 3 Connecting

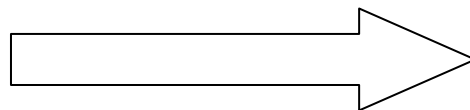
Connect the YHY638A to the USB or RS232 port of PC, after power on the RED led and GREEN led will flash one time, then the GREEN led will light on again, next the buzzer will beep twice, it means that the reader is ready now.

1) For YHY638AU item

Connect one of the A type USB connector to the rear side of the YHY638AU, another USB connector connect to the host USB port.



(Item No:YHY638AU)



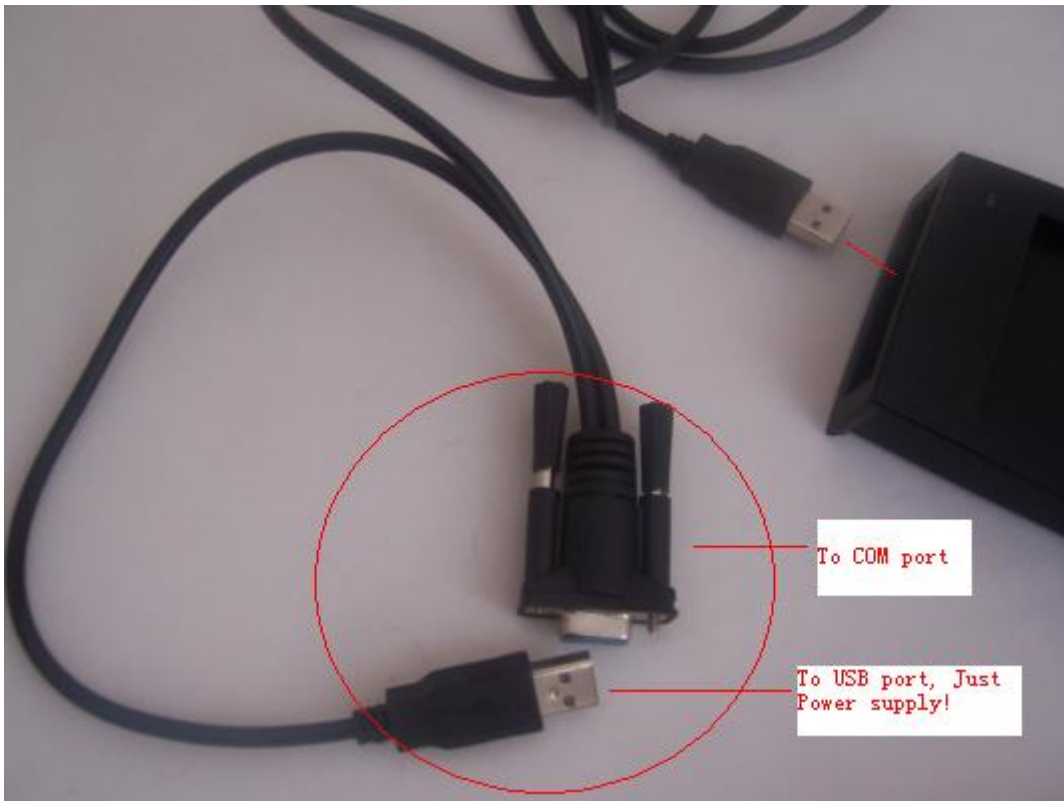
TO PC USB PORT



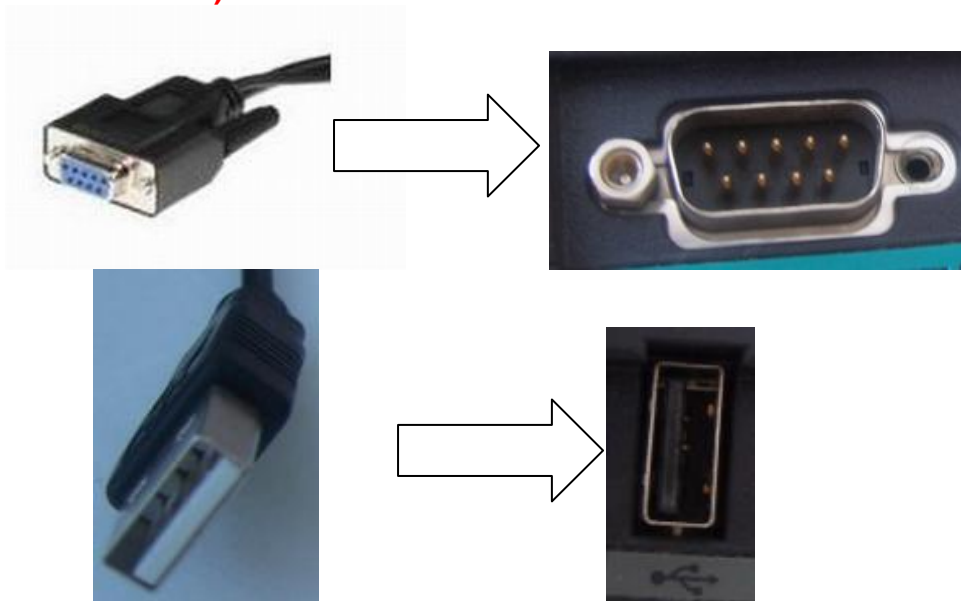
2) For YHY638AR item

Connect one of the A type USB connector to the rear side of the YHY638AR, another USB connector which near the serial

connector connect to the host USB port for power the device, and connect the 9pin COM port to HOST COM port.



(Item No:YHY638AR)



#### 4 Electrical Characteristics

## 4.1 Operating Condition Range

**Relative humidity: up to 90%**

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Tamb	Ambient Temperature		0	25	60	°C
VDD	DC Supply Voltage	DVSS = 0V	4.5	5	5.5	V

**Table 1 - Operating Condition Range**

## 4.2 Current Consumption

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
IDVDD	Supply Current	Reading started but no cards in the reader range		90		mA
IDVDD		Reading started, 1 card in the reader range		95		mA

**Table 2 - Current Consumption**

## 4.3 Operating Distance

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
OD	14443 Type A S50 tag Operating Distance,	Measured from the reader bottom		0-70		mm

**Table 3 - Operating Distance**

## 4.4 COM Interface Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
USB	baud rate			<b>115200</b>		
RS232	baud rate		9600	<b>115200</b>	115200	baud

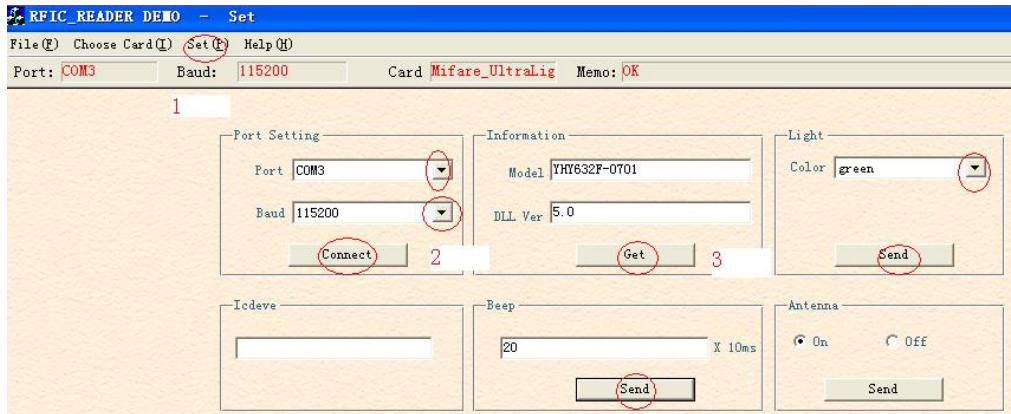
**Table 4 - Serial Interface**

## 5. DEMO

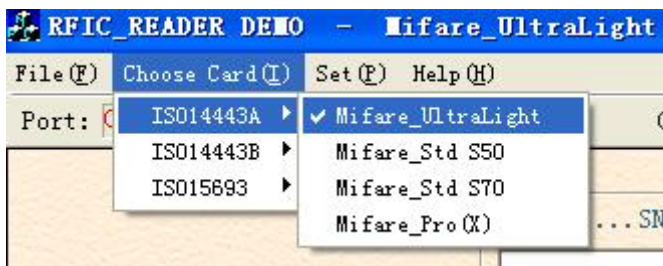
This software rfidrx.exe run on Win32 system.

### 5.1 COM setup

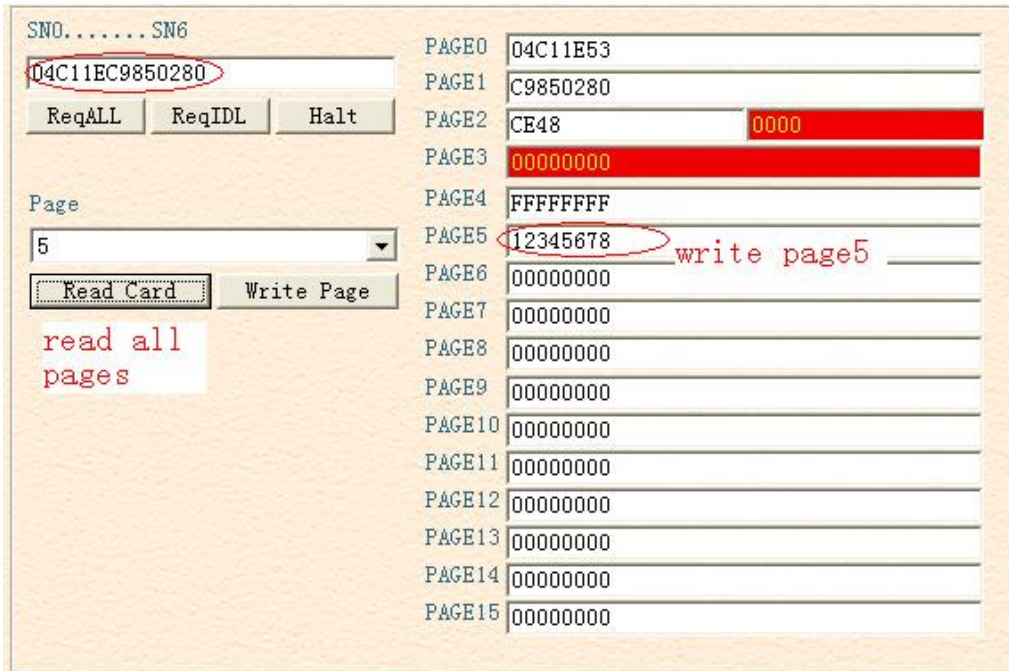
First run RFIDXray.exe, then connect the reader to PC COM port. Choose the correct COM number, click [Connect] button to connect the Reader to PC. Baud set to 115200. Click [Read] the product information button, you can check the specific type of the Reader and the supported cards.



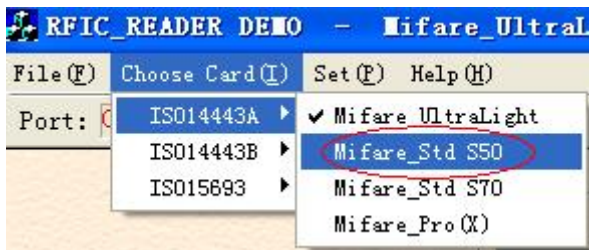
## 5.2 UltraLight



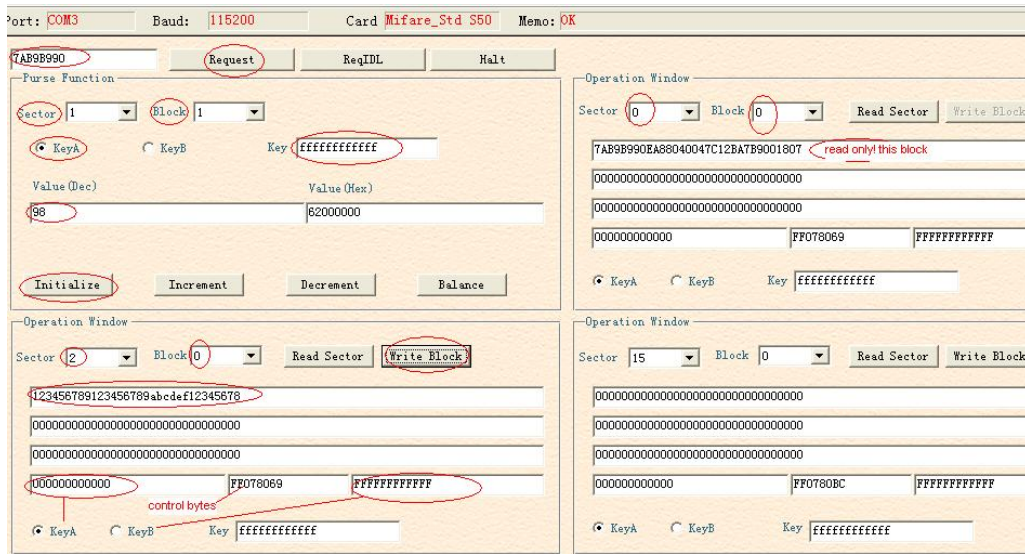
Click the [ReqALL] button to obtain the card Serial Number. Choose the corresponding address to read/write the card.



### 5.3 Mifare\_1K (STD S50)



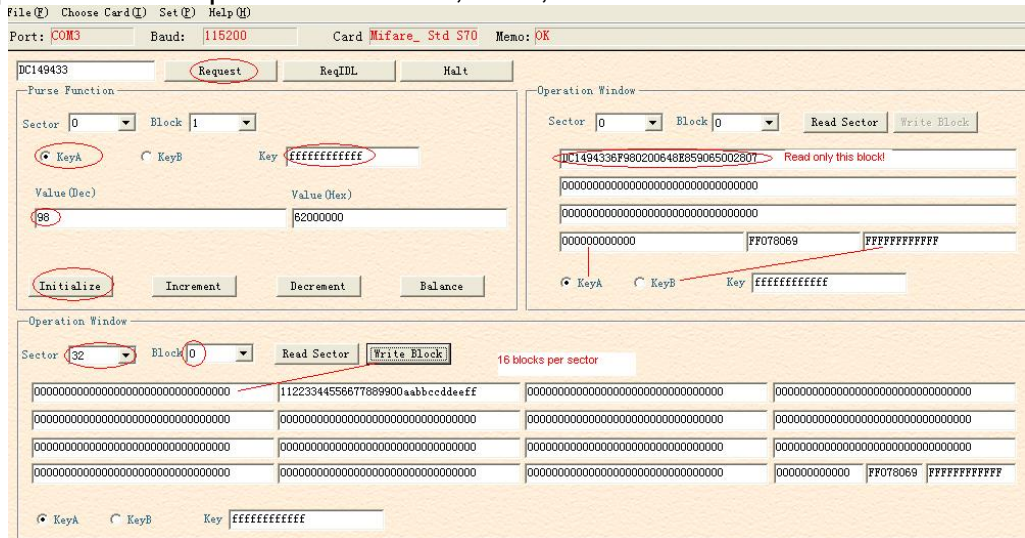
Click the [Request] button to obtain the card serial number.  
 Input the correct password to read, write, increase or decrease the card.



### 5.4 Mifare\_4K (STD S70)



Click the [Request] button to obtain the card serial number.  
 Input the correct password to read, write, increase or decrease the card.

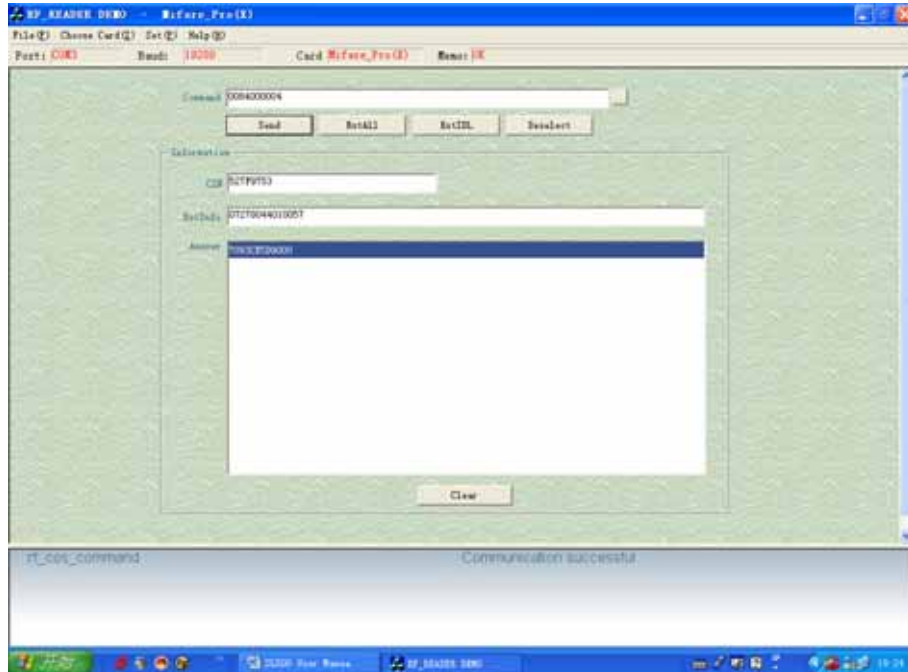


### 5.5 Mifare\_ProX

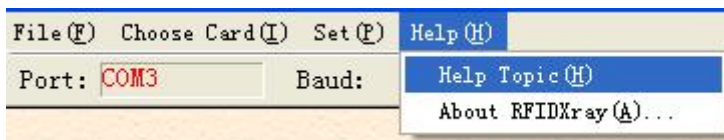
Click [Reset] button to obtain the serial number and the reset information of

the card according to ISO14443-4 protocol.

Input the COS command, click [Send] button to commute data to card.

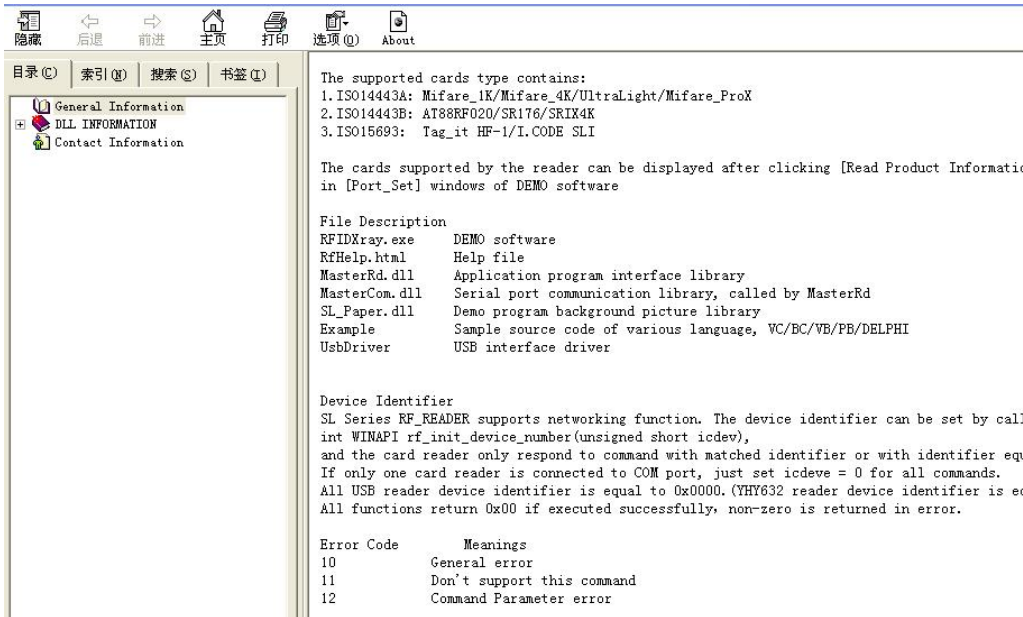


## 5.6 HELP



You can search DLL function defined in the help file.

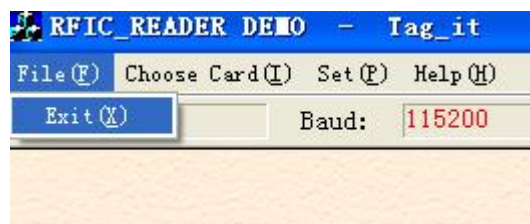




## 5.7 DISCONNECT READER



## 5.8 EXIT



## 6. DLL INFORMATION ( icdev=0 )

### 6.1 SYSTEM FUNCTION

#### 6.1.1 INT WINAPI LIB\_VER

Function: Get DLL Version  
 Prototype: int WINAPI lib\_ver (unsigned int \*pVer)  
 Parameter: pVer: [OUT] DLL version

Return: return 0 if successful

### **6.1.2 INT WINAPI RF\_INIT\_COM**

Function: Connect

Prototype: int WINAPI rf\_init\_com (int port, long baud)

Parameter: port: [IN] serial port number

baud: [IN] communication baud rate, 19200 ~ 115200 bps

Return: return 0 if successful

### **6.1.3 INT WINAPI RF\_CLOSEPORT**

Function: Disconnect

Prototyp: int WINAPI rf\_ClosePort(void)

Return: return 0 if successful

### **6.1.4 INT WINAPI RF\_GET\_MODEL**

Function: Get Device Type

Prototype: int WINAPI rf\_get\_model (unsigned short icdev,  
unsigned char \*pVersion,  
unsigned char \*pLen)

Parameter: icdev: [IN] Device ID

pVersion: [OUT] response information

pLen: [OUT] length of response information

Return: return 0 if successful

### **6.1.5 INT WINAPI RF\_INIT\_TYPE (RFU)**

Function: Set Reader contactless working mode

Prototype: int WINAPI rf\_init\_type(unsigned short icdev, unsigned char type)

Parameter: icdev: [IN] Device ID

type: [IN] reader working mode

Return: return 0 if successful

Explanation: this function is not effective to the readers only support single protocol.

type = 'A': set YHY638 into ISO14443A mode

type = 'B': set ISO14443B mode

type = 'r': set AT88RF020 card mode

type = '1': set ISO15693 mode

### **6.1.6 INT WINAPI RF\_ANTENNA\_STA**

Function: Manage RF Transmittal

Prototype: int WINAPI rf\_antenna\_sta (unsigned short icdev, unsigned char model)

Parameter: icdev: [IN] Device ID

model: [IN] transmittal state

Return: return 0 if successful

Explanation: model = 0: turn off RF transmittal

model = 1: turn on RF transmittal

### **6.1.7 INT WINAPI RF\_LIGHT**

Function: Manage LED  
Prototype: int WINAPI rf\_light (unsigned short icdev, unsigned char color)  
Parameter: icdev: [IN] Device ID  
          color: [IN] 0 = off  
                  1 = red  
                  2 = green  
                  3 = yellow (RFU)  
Return: return 0 if successful

### **6.1.8 INT WINAPI RF\_BEEP**

Function: beep  
Prototype: int WINAPI rf\_beep (unsigned short icdev, unsigned char msec)  
Parameter: icdev: [IN] Device ID  
          msec: [IN] beep time, unit 10 Msec  
Return: return 0 if successful

## **6.2 ISO14443A FUNCTION**

### **6.2.1 UltraLight**

#### **6.2.1.1 INT WINAPI RF\_REQUEST**

Function: ReqA  
Prototype: int WINAPI rf\_request ( unsigned short icdev,  
                                  unsigned char model,  
                                  unsigned short \*pTagType)  
Parameter: icdev: [IN] Device ID  
          model: [IN] REQ MODE  
          pTagType: [OUT] response data, chip type code  
Return: return 0 if successful  
Explanation: mode = 0x26: REQ\_STD  
              mode = 0x52: REQ\_ALL

#### **6.2.1.2 INT WINAPI INT RF\_UL\_SELECT**

Function: Select UltraLight  
Prototype: int WINAPI int\_rf\_ul\_select (unsigned short icdev,  
                                      unsigned char \*pSnr,  
                                      unsigned char \*pLen)  
Parameter: icdev: [IN] Device ID  
          pSnr: [OUT] response data, card unique serial number  
          pLen: [OUT] length of response data  
Return: return 0 if successful

#### **6.2.1.3 INT WINAPI RF\_M1\_READ**

Function: MifareOne read  
Prototype: int WINAPI rf\_M1\_read ( unsigned short icdev,

unsigned char block,  
unsigned char \*pData,  
unsigned char \*pLen)

Parameter: icdev: [IN] Device ID  
block: [IN] block absolute address  
pData: [OUT] response data from card  
pLen: [OUT] length of response data

Return: return 0 if successful

Explanation: this function is also applicable for UltraLight card. Every page of UltraLight card has 4 bytes. Transferring this function every time, return data of 4 consecutive pages.

#### **6.2.1.4 INT WINAPI INT RF\_UL\_WRITE**

Function: UltraLight Write

Prototype: int WINAPI int rf\_ul\_write ( unsigned short icdev,  
unsigned char page,  
unsigned char \*pData)

Parameter: icdev: [IN] Device ID  
page: [IN] UltraLight card page address , 0 ~ 0x0F  
pData: [IN] written data, 4 bytes

Return: return 0 if successful

#### **6.2.1.5 INT WINAPI RF\_HALT**

Function: TYPE\_A card HALT

Prototype: int WINAPI rf\_halt (unsigned short icdev)

Parameter: icdev: [IN] Device ID

Return: return 0 if successful

### **6.2.2 Mifare\_Std**

#### **6.2.2.1 INT WINAPI RF\_REQUEST**

Function: ReqA

Prototype: int WINAPI rf\_request ( unsigned short icdev,  
unsigned char model,  
unsigned short \*pTagType)

Parameter: icdev: [IN] Device ID  
model: [IN] REQ MODE  
pTagType: [OUT] response data, chip type code

Return: return 0 if successful

Explanation: mode = 0x26: REQ\_STD  
mode = 0x52: REQ\_ALL

#### **6.2.2.2 INT WINAPI RF\_ANTICOLL**

Function: Mifare card Anticollision

Prototype: int WINAPI rf\_anticoll ( unsigned short icdev,  
unsigned char bcnt,  
unsigned char \*pSnr,  
unsigned char \*pLen)

Parameter: icdev: [IN] Device ID  
bcnt: [IN] must be 4  
pSnr: [OUT] response data from card, unique serial number  
pLen: [OUT] length of response data  
Return: return 0 if successful

#### **6.2.2.3 INT WINAPI RF\_SELECT**

Function: Mifare card Selectting  
Prototype: int WINAPI rf\_select (unsigned short icdev,  
                                  unsigned char \*pSnr,  
                                  unsigned char snrLen,  
                                  unsigned char \*pSize)  
Parameter: icdev: [IN] Device ID  
pSnr: [IN] card unique serial number  
snrLen: [IN] length of pSnr  
pSize: [OUT] response data from card, capacity code  
Return: return 0 if successful  
Explanation: card will be on active estate after received this command, only one TYPE\_A card  
on active estate at the same influence range at same time.

#### **6.2.2.4 INT WINAPI RF\_M1\_AUTHENTICATION2**

Function: Mifare\_Std Authenticate  
Prototype: int WINAPI rf\_M1\_authentication2 ( unsigned short icdev,  
  unsigned char model,  
  unsigned char block,  
  unsigned char \*pKey)  
Parameter: icdev: [IN] Device ID  
model: [IN] key validate mode  
block: [IN] block absolute address  
pKey: [IN] 6 bytes password  
Return: return 0 if successful  
Explanation: model = 0x60: via KeyA  
              model = 0x61: via KeyB

#### **6.2.2.5 INT WINAPI RF\_M1\_READ**

Function: MifareOne Read  
Prototype: int WINAPI rf\_M1\_read ( unsigned short icdev,  
                                      unsigned char block,  
                                      unsigned char \*pData,  
                                      unsigned char \*pLen)  
Parameter: icdev: [IN] Device ID  
block: [IN] block absolute address  
pData: [OUT] response data from card  
pLen: [OUT] length of response data  
Return: return 0 if successful

#### **6.2.2.6 INT WINAPI RF\_M1\_WRITE**

Function: Mifare\_Std Write

Prototype: int WINAPI rf\_M1\_write (unsigned short icdev,  
unsigned char block,  
unsigned char \*pData)

Parameter: icdev: [IN] Device ID  
block: [IN] block absolute address  
pData: [IN] written data, 16 bytes

Return: return 0 if successful

### **6.2.2.7 INT WINAPI RF\_M1\_INITVAL**

Function: Mifare\_Std card Initialize Value

Prototype: int WINAPI rf\_M1\_initval ( unsigned short icdev,  
unsigned char block,  
long value)

Parameter: icdev: [IN] Device ID  
block: [IN] block absolute address  
pValue: [IN] initialize purse value at HEX format, low byte in former

Return: return 0 if successful

### **6.2.2.8 INT WINAPI RF\_M1\_READVAL**

Function: Mifare\_Std Read Value

Prototype: int WINAPI rf\_M1\_readval ( unsigned short icdev,  
unsigned char block,  
long \*pValue)

Parameter: icdev: [IN] Device ID  
block: [IN] block absolute address  
pValue: [OUT] response value at HEX format, low byte in former

Return: return 0 if successful

### **6.2.2.9 INT WINAPI RF\_M1\_INCREMENT**

Function: Mifare purse increment

Prototype: int WINAPI rf\_M1\_increment (unsigned short icdev,  
unsigned char block,  
long value)

Parameter: icdev: [IN] Device ID  
block: [IN] block absolute address  
value: [IN] increase value at HEX format, low byte in former

Return: return 0 if successful

### **6.2.2.10 INT WINAPI RF\_M1\_DECREMENT**

Function: Mifare purse decrement

Prototype: int WINAPI rf\_M1\_decrement (unsigned short icdev,  
unsigned char block,  
long value)

Parameter: icdev: [IN] Device ID  
block: [IN] block absolute address  
value: [IN] decrease value at HEX format, low byte in former

Return: return 0 if successful

#### **6.2.2.11 INT WINAPI RF\_M1\_RESTORE**

Function: Mifare\_Std Restore  
Prototype: int WINAPI rf\_M1\_restore (unsigned short icdev, unsigned char block)  
Parameter: icdev: [IN] Device ID  
            block: [IN] block absolute address  
Return: return 0 if successful

#### **6.2.2.12 INT WINAPI RF\_M1\_TRANSFER**

Function: Mifare\_Std Transfer  
Prototype: int WINAPI rf\_M1\_transfer (unsigned short icdev, unsigned char block)  
Parameter: icdev: [IN] Device ID  
            block: [IN] block absolute address  
Return: return 0 if successful  
Explanation: this function only be transferred after increment, decrement and restore command

#### **6.2.2.13 INT WINAPI RF\_HALT**

Function: Mifare Halt  
Prototype: int WINAPI rf\_halt (unsigned short icdev)  
Parameter: icdev: [IN] Device ID  
Return: return 0 if successful  
Explanation: card exit active estate after received this command

### **6.2.3 Mifare\_ProX**

#### **6.2.3.1 INT WINAPI RF\_TYPE\_RST**

Function: Request ISO14443A-4 card and reset  
Prototype: int WINAPI rf\_typea\_rst ( unsigned short icdev,  
  unsigned char model,  
  unsigned char \*pData,  
  unsigned char \*pMsgLg)  
Parameter: icdev: [IN] Device ID  
            model: [IN] request mode  
            pData: [OUT] response data from card  
            pMsgLg: [OUT] length of response data  
Return: return 0 if successful  
Explanation: mode = 0x26: REQ\_STD  
            mode = 0x52: REQ\_ALL  
            pData: 4bytes CSN + RATS according to ISO14443A

#### **6.2.3.2 INT WINAPI RF\_COS\_COMMAND**

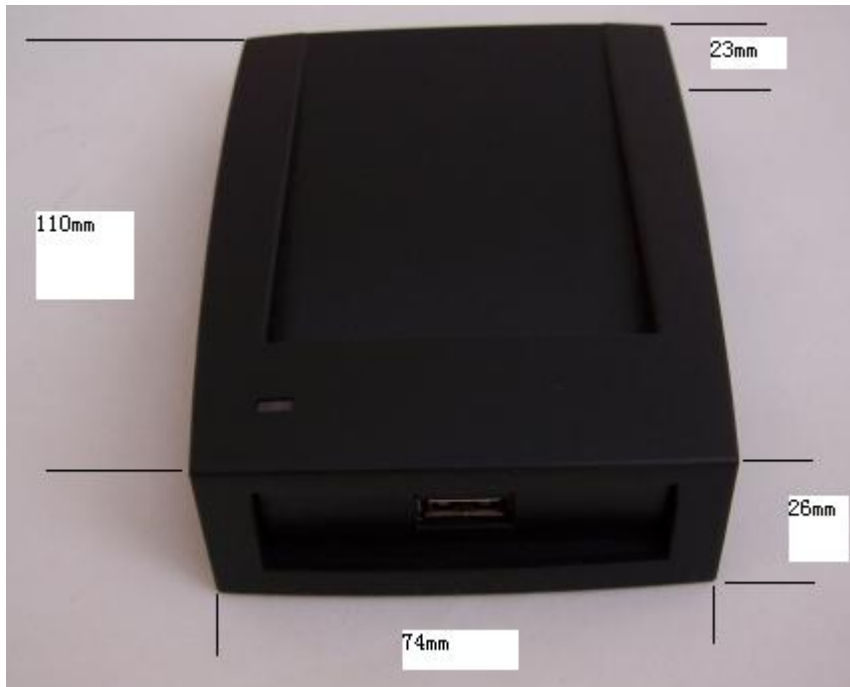
Prototype: int WINAPI rf\_cos\_command ( unsigned short icdev,  
  unsigned char \*pCommand,  
  unsigned char cmdLen,  
  unsigned char \*pData,  
  unsigned char \*pMsgLg)  
Parameter: icdev: [IN] Device ID  
            pCommand: [IN] COS command

cmdLen: [IN] length of COS command  
pData: [OUT] response data from card, including SW1& SW2  
pMsgLg: [OUT] length of response data  
Return: return 0 if successful

### **6.2.3.3 INT WINAPI RF\_CL\_DESELECT**

Prototype: int WINAPI rf\_cl\_deselect (unsigned short icdev)  
Parameter: icdev: [IN] Device ID  
Return: return 0 if successful

## **7 Pictures**



*YHY638 size picture*





*YHY638 Top View*



*YHY638 Side View*



*YHY638 Back Side View*



*YHY638 Rear Side View*

Contact Information:  
EHUOYAN Technology Co., Ltd.  
Tel: +86 -010-59870151  
Fax: +86 -010-59754725  
email: [info@ehuoyan.com](mailto:info@ehuoyan.com)  
WebSite: [www.ehuoyan.com](http://www.ehuoyan.com)

FILE END

---