



Ordering Information:

Item No.	Description
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 $\,\sim\,$ 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.









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	Тур	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	Тур	Max	Unit
IDVDD	Supply	Reading started		90		mA
	Current	but no cards in				
		the reader range				
IDVDD		Reading started,		95		mA
		1 card in the				
		reader range				

Table 2 - Current Consumption

4.3 Operating Distance

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

Table 3 - Operating Distance

4.4 COM Interface Characteristics

Symbol	Parameter	Conditions	Min	Тур	Max	Unit
USB	baud rate			115200		
RS232	baud rate		9600	115200	115200	baud
				-		

 Table 4 - Serial Interface

5. DEMO

This software rfidxray.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.

RFIC_READER File(F) Choose C:	DEIO - ard(I) (Set(P)	Set Help(H)						
Port: COM3	Baud:	115200	Card Mifar	e_UltraLig Mem	o: <mark>OK</mark>			
	1							
		-Port Setting Port COM3 Baud 115200 Connect		Information Model THY632F DIL Ver 5.0	-0701	3	-Light Color green Send	
		-Icdeve		Beep	(Send)	X 10ms	-Antenna © On C Of Send	f

5.2 UltraLight

🚣 RFIC	READER DELC) - T ifare_Ultra	aLight
File(F)	Choose Card(L)	Set(P) Help(H)	
Port:	IS014443A 🕨	✔ Mifare_UltraLight	C
	IS014443B 🕨	Mifare_Std S50	
120.80	IS015693 🕨	Mifare_Std S70	
		Mifare_Pro(X)	SN
	2000		-

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



5NU			PAGEO	04C11E53	
04C11EC9850280		PAGE1	C9850280		
RegALL	ReqIDL	Halt	PAGE2	CE48	0000
			PAGE3	00000000	
Page			PAGE4	FFFFFFF	
5		*	PAGE5	12345678 >	write nage5
Read C	ard Wr	ita Paga	PAGE6	00000000	wiito pagoo
i neau c	<u>aru</u> "r.	ICC I AGO	PAGE7	00000000	
read all		PAGE8	00000000		
pages			PAGE9	00000000	
			PAGE10	00000000	
			PAGE11	00000000	
			PAGE12	00000000	
			PAGE13	00000000	
			PAGE14	00000000	
			PAGE15	00000000	

5.3 Mifare_1K (STD S50)

🚣 RFIC	_READER DEM) - H ifare_UltraL
File(F)	Choose Card(<u>I</u>)	Set(P) Help(H)
Port:	IS014443A 🕨	✔ Mifare UltraLight
1	IS014443B 🕨	(Mifare_Std S50)
	IS015693 🕨	Mifare_Std S70
		Mifare_Pro(X)

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



Port: COM3 Baud: 115200 Card Mifare_Std S50 Me	mo: <mark>OK</mark>
TAB98990 Request ReqIDL Halt	
Geeter 1 V Block 1 V	Sector 0 V Block 0 V Read Sector Write Block
• Keyk C KeyB Key (ffffffffff	TAB9B990EA88040047C12BA7B9001807 Cread only! this block
Value (Her) Value (Her)	000000000000000000000000000000000000000
(98) (6200000	- 0000000000000000000000000000000000000
	000000000000 FF078069 FFFFFFFFF
Initialize Increment Decrement Balance	© Keyλ ⊂ KeyB Key ffffffffff
Sector 2 Block Read Sector Frite Block	Sector 15 Block 0 Read Sector Write Block
123456789123456789abcdef12345678	000000000000000000000000000000000000000
000000000000000000000000000000000000000	000000000000000000000000000000000000000
000000000000000000000000000000000000000	000000000000000000000000000000000000000
Control bytes C KeyA C KeyB Key Effetteffeff	O0000000000 FF07808C FFFFFFFFF

5.4 Mifare_4K (STD S70)

👫 RFIC	_READER DE	[()) - L ifare_Std S50
File(F)	Choose Card(])	Set(P) Help(H)
Port:	IS014443A	۲	Mifare_UltraLight
	IS014443B	۲	✓ Mifare Std S50
7AB9B9	IS015693	۲	(Mifare_Std S70)
Purse	Function —	-	Mifare_Pro(X)

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

ort: COM3 Baud: 1152	00 Card Mifare_ Std S70 M	iemo: <mark>OK</mark>	
DC149433 Purse Function Sector 0 Block 1	st RegIDL Halt	Operation Window Sector 0 - Block 0	Bead Sector Write Block
• КеуА С КеуВ	Key (fffffffffff	QUC1494336F980200648E85906500280	Read only this block!
Value (Dec)	Value (Hex)	000000000000000000000000000000000000000	00
(98)	62000000		00
Initialize Increment -Operation Window Sector (32) Block(0)	Decrement Balance	6 blocks per sector	FFFFFFFFFFFFFFFF
000000000000000000000000000000000000000	11223344556677889900aabbccddeeff	000000000000000000000000000000000000000	000000000000000000000000000000000000000
000000000000000000000000000000000000000	000000000000000000000000000000000000000	000000000000000000000000000000000000000	000000000000000000000000000000000000000
000000000000000000000000000000000000000	000000000000000000000000000000000000000	000000000000000000000000000000000000000	000000000000000000000000000000000000000
000000000000000000000000000000000000000	000000000000000000000000000000000000000	000000000000000000000000000000000000000	000000000000 FF078069 FFFFFFFFF
• KeyA C KeyB Key	ffffffffff		

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.

	Sec. 1	0054000004						
		Sead	RetALL	RecITL.	Inislant			
	Talienitin							
		12779753						
	-Solbala	10011044010057				-	212	
	and the second	TIN 3. #12000H						
							THE .	
		-				_		
				Geat				
_cos_comma	nd:			Comm	Unication app	CHSSIU.		

5.6 HELP

File(F) Choose Card(I) Set(P)	Help(H)
Port: COM3	Baud:	Help Topic(<u>H</u>)
and the second second		About RFIDXray(A)

You can search DLL function defined in the help file.



· · · · · · · · · · · · · · · · · · ·	聞- 逢项 (Q) About
目录 CC 索引 (g) 搜索 (g) 书签 (L) General Information ● DLI INFORMATION ▲ Contact Information	The supported cards type contains: 1. ISO144438: Mifare_IK/Mifare_4K/UltraLight/Mifare_ProX 2. ISO144438: AT88KP020/SRI76/SRIX4K 3. ISO15693: Tag_it HF-1/I.CODE SLI The cards supported by the reader can be displayed after clicking [Read Product Informatic in [Port_Set] windows of DEMO software File Description RFIDXray.exe DEMO software RTHelp.html Help file MasterRd dll Application program interface library MasterCom.dll Serial port communication library, called by MasterRd SL_Paper.dll Demo program background picture library Example Sample source code of various language, VC/BC/VE/PB/DELPHI UsbDriver USB interface driver
	Device Identifier SL Series RF_READER supports networking function. The device identifier can be set by call int WIMAPI rf_init_device_number(unsigned short icdev), and the card reader only respond to command with matched identifier or with identifier equ If only one card reader is connected to COM port, just set icdeve = 0 for all commands. All USB reader device identifier is equal to 0x0000. (YHY632 reader device identifier is et All functions return 0x00 if executed successfully, non-zero is returned in error. Error Code Meanings 10 General error 11 Don't support this command 12 Command Parameter error

5.7 DISCONNECT READER



5.8 EXIT

ile(F)	Choose	Card(I)	Set (P)	Help(H)
Exit ()	0		Baud:	115200

6. DLL INFORMATION (icdev=0)

6.1 SYSTEM FUNCTION

6.1.1 INT WINAPI LIB_VER

Function:Get DLL VersionPrototype: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]

 Return:
 return 0 if successful

6.1.3 INT WINAPI RF_CLOSEPORT

Function:DisconnectPrototyp: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 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. Transfering 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 [IN] length of pSnr snrLen: pSize: [OUT] response data from card, capacity code return 0 if successful Return: 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 Authentify 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 [IN] block absolute address block: 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 [IN] block absolute address block: 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 bute in fr

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) [IN] Device ID Parameter: icdev: request mode model: [IN] pData: [OUT] response data from card pMsgLg: [OUT] length of response data return 0 if successful Return: 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

<u>Contact Information:</u> EHUOYAN Technology Co., Ltd. Tel: +86 -010-59870151 Fax: +86 -010-59754725 email: info@ehuoyan.com WebSite: <u>www.ehuoyan.com</u>

FILE END