

IC Card R&W Reader

IC cards Read&Write reader User Manual

Product Name:RFID reader/writer

Model No.(HVIN):RFCP-2



RW210Ax/RW210Cx

Version 1.0

Dec.2017

0.1 Disclaimers

Shanghai Electronics Co, Ltd reserves the right to make changes without further notice to the information published in this document. Shanghai Electronics Co, Ltd does not assume any liability arising out of the application or use of any product or circuits described herein.

0.2 Revision history

Revision	Description	Date
V1.0	Initial version	Dec. 2017

0.3 Abbreviations and Symbols

APDU	Application Protocol Data Unit
ChkS	Checksum
CMD	Command
COS	Card Operating System
Rd-addr	Reader-address
DF	Data Field
DLL	Dynamic Link Library
DII	Dynamic Link Library
ESAM	Embedded Secure Access Module
Fr-E	Frame-End
Fr-S	Frame-Start
ISAM	Increment Secure Access Module
Len	Length
LSB	Least Significant Byte
MSB	Most Significant Byte
RF	Radio Frequency
RFU	Reserved for Future Use
R&W	Read and Write
PPS	Protocol and Parameter Selection
PSAM	Purchase Secure Access Module
SAM	Secure Access Module
SDK	Software Development Kit
SIM	Subscriber Identity Module

CONTENTS

0.1 Disclaimers	2
0.2 Revision history	2
0.3 Abbreviations and Symbols	2
1 General Description.....	6
1.1 RW210Ax/RW210Cx series card readers	6
1.2 Ordering information	6
1.3 Main features	7
1.4 Technical characteristics	8
2 Hardware connection.....	8
2.1 Appearance.....	8
2.1.1 RS232 reader	8
2.1.2 USB reader.....	9
2.1.3 RS485 reader/TTL232 reader	9
2.2 Hardware connection	10
2.2.1 RS232 reader.....	10
2.2.2 RS485 reader.....	10
2.2.3 TTL232 reader.....	11
2.2.4 USB reader (USB virtual COM port).....	12
2.2.5 USB reader (Not need to install driver)	13
2.3 Power-on status.....	13
3. Description of card operation steps	14
3.1 Flowchart for MF1 Card:	14
3.2 Flowchart for Mifare Ultralight / Ntag Card:	14
3.3 Flowchart for ISO14443-4 TYPE A CPU card	15
3.4 Flowchart for Mifare Desfire EV0/EV1/EV2 card:	15
3.5 Flowchart for ISO17816-4 SAM/SIM/CPU Card:	16
3.6 Flowchart for ISO17816-4 SAM/SIM/CPU Card (Only support 1 pc card)	16
4. Description of DLL library functions	17
4.1 General description of DLL library	17
4.2 General system function.....	17
General system function: Get dll version	17
General system function: DES encrypt.....	18
General system function: DES Decrypt	18
General system function: Calculate MAC	18
General system function: Key scatter	19
General system function: Port connect.....	19

General system function: Close port.....	20
General system function: Set reader working mode.....	20
General system function: Reader antenna set.....	20
General system function: Reader antenna set(new).....	21
General system function: Buzzer control	21
General system function: Led control.....	22
General system function: Get reader version	22
General system function: Get reader serial number.....	22
General system function: Set reader address	22
General system function: Get reader address	23
4.3 Functions for MF1 card.....	23
Special for MF1 card: ISO14443A card request	23
Special for MF1 card: ISO14443A card anticollision.....	23
Special for MF1 card: ISO14443A card select.....	24
Special for MF1 card: Mifare One Authentication.....	24
Special for MF1 card: Mifare One read block	24
Special for MF1 card: Mifare One write block	25
Special for MF1 card: Mifare One initail purse value	25
Special for MF1 card: Mifare One read purse value	25
Special for MF1 card: Mifare One decrease purse value	26
Special for MF1 card: Mifare One increase purse value.....	26
Special for MF1 card: Mifare One restore purse value	26
Special for MF1 card: Mifare One transfer purse value.....	27
Special for MF1 card: Mifare One read sector.....	27
Special for MF1 card: Mifare One write sector.....	27
Special for MF1 card: Halt ISO14443A card	28
4.4 Functions for Ultraligh/Ntag card.....	28
Special for Ultralight/Ntag card: ISO14443A card request.....	28
Special for Ultralight/Ntag card: Ultralight/Ntag card select	28
Special for Ultralight/Ntag card: Ultralight/Ntag card read pages.....	28
Special for Ultralight/Ntag card: Ultralight/Ntag card write page.....	29
Special for Ntag21x card: Ntag21x card fast read	29
Special for Ntag21x card: Ntag21x card get version	29
Special for Ntag21x card: Ntag21x card key authentication.....	30
Special for Ntag21x card: Ntag21x card read CNT	30
Special for Ntag21x card: Ntag21x card read signature.....	30
Special for NFC type2 card: NFC type2 card select sector	31
Special for Ntag21x card: Halt ISO14443A card	31
4.5 ISO14443A-4 CPU card.....	31
Special for TYPE A CPU card: ISO14443A-4 CPU card reset.....	31

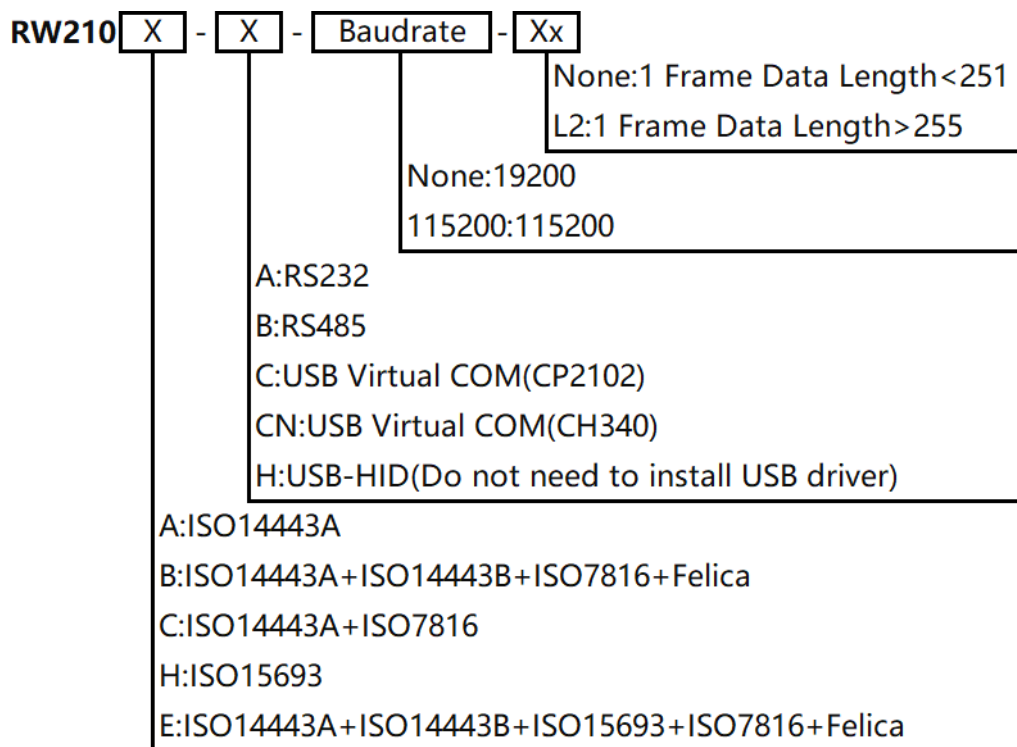
Special for TYPE A CPU card: Send APDU(COS) to ISO14443A-4 CPU card	32
Special for TYPE A CPU card: Send PPS to ISO14443A-4 CPU card	32
Special for TYPE A CPU card: Deselect ISO14443A-4 CPU card	32
4.6 ISO7816-4 SAM/SIM/CPU card(new).....	33
Special for ISO7816-4 SAM/SIM Card: No.x SAM/SIM Card reset	33
Special for ISO7816-4 SAM/SIM Card: Send APDU(COS) to No.x SAM/SIM Card	33
Special for ISO7816-4 SAM/SIM Card: Send PPS request to No.x SAM/SIM card.....	34
4.7 ISO7816-4 SAM/SIM/CPU card.....	35
Special for ISO7816-4 SAM/SIM Card : Initial baudrate.....	35
Special for ISO7816-4 SAM/SIM Card : SAM/SIM Card reset.....	35
Special for ISO7816-4 SAM/SIM Card : Send APDU(COS) to SAM/SIM Card.....	35

1 General Description

1.1 RW210Ax/RW210Cx series card readers

RW210Ax/RW210Cx series card readers are designed base on the NXP's contactless R&W IC, and adopt the 13.56M RF contactless technology. These readers support ISO/IEC 14443A /MIFARE mode. Users do not need to care about the complex control methods, and only need to send simple commands through RS232 or RS485 or USB or USB(HID) to implement all operations of the card. The reader supports ISO14443-A Mifare One S50, S70, FM11RF08, Ultralight, Ntag203, Ntag213, 215, 216, FM1208, Desfire EV1/EV2 D21, D41, D81,T=CL CPU card , ISO7816-4 SAM9600,SIM9600, SAM38400, SIM38400, SAM115200, SIM115200,and compatible card.

1.2 Ordering information



Type number	Description	Remarks
RW210AA	RS232, No SAM card connector socket embedded	
RW210AB	RS485, No SAM card connector socket embedded	
RW210AC	USB(CP2102), No SAM card connector socket embedded	
RW210ACN	USB(CH340), No SAM card connector socket embedded	
RW210AH	USB(HID), No SAM card connector socket embedded	
RW210AT	TTL232, No SAM card connector socket embedded	

RW210CA	RS232, 1 pc SAM card connector socket embedded	
RW210CB	RS485, 1 pc SAM card connector socket embedded	
RW210CC	USB(CP2102), 1 pc SAM card connector socket embedded	
RW210CCN	USB(CH340), 1 pc SAM card connector socket embedded	
RW210CH	USB(HID), 1 pc SAM card connector socket embedded	
RW210CT	TTL232, No SAM card connector socket embedded	
Remarks: (1) RW210AH and RW210CH do not need to install Windows driver. (2) Default factory setting baud rate is 19200, if need 115200, please add "-115200" after type number. E.g., RW210XX-115200. (3) RW210Cx default with 1 pc SAM card connector socket , if need 2 pcs , please add "-2SAM" after type number. E.g., RW210CA-2SAM (4) If you need to support the Key functions of Ultralight C card , you need to contact YuanZhi to get a new firmware version.		

1.3 Main features

- NXP 32-bit ARM processor
- NXP Contactless Read-Write ASIC
- RF frequency: 13.56MHz
- Support ISO14443A/Mifare/ISO7816-4 mode
- Support tags:
 - ISO14443A: Mifare 1K/4K , FM11RF08 , Ultralight, Ntag203, Ntag213, Ntag215, Ntag216, T=CL CPU card (ISO14443A) ,FM1208, Desfire EV1/ EV2 D21, D41, D81
 - Ultralight C key functions are not support by default. If you need to support, please contact YuanZhi for a new firmware version.
 - ISO17816-4: SAM/SIM9600(T=0 & T=1), SAM/SIM38400(T=0 & T=1), SAM/SIM115200(T=0 & T=1)
 - and compatible card
- Optional 1-2 pcs SAM card connector sockets
- Interface: RS232, RS485, USB or TTL232
- Simple instruction set
- Many SDKs for reference. E.g. java, JavaScript, VC, VB, C++ Builder, DELPHI, Power Builder, C#, CVI, Android, Linux, etc.
- Firmware can be customized according to user requirements without changing the hardware
- Widely used in the fields of finance, health, traffic, industrial, etc.

1.4 Technical characteristics

- RF frequency: 13.56MHz
- Support protocols: ISO14443A/Mifare/ISO7816
- Support Tags:
 - ISO14443A: Mifare 1K/4K , FM11RF08 , Ultralight, Ntag203, Ntag213, Ntag215, Ntag216, T=CL CPU card (ISO14443A) ,FM1208, Desfire EV1/ EV2 D21, D41, D81
 - ISO17816-4: SAM/SIM9600(T=0 & T=1), SAM/SIM38400(T=0 & T=1), SAM/SIM115200(T=0 & T=1)
and compatible card
- Power supply: DC5V $\pm$ 10%
- Max current : 150mA
- Interface: RS232/RS485/USB/TTL232
- Baud rate: Default is 19200 or 115200
- Tag operate distance : 40~60mm (Test result according to MF1 standard card)
- Ambient temperature: -20°C ~ +70°C
- Dimension : 125mm*85mm*19mm
- Weight: 130~160g (USB cable is about 45g, RS232 cable is about 75g)

2 Hardware connection

2.1 Appearance

2.1.1 RS232 reader

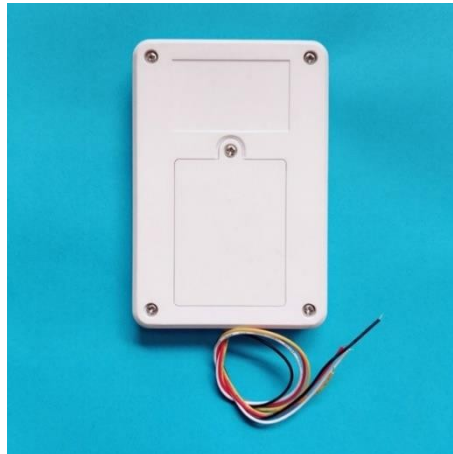


Remarks: USB connector is just for power supply, and female DB9 connector is for communication.

2.1.2 USB reader



2.1.3 RS485 reader/TTL232 reader



2.1.4 Dimension



2.2 Hardware connection

2.2.1 RS232 reader

To RW210AA, RW210CA



2.2.2 RS485 reader

To RW210AB, RW210CB



Cable color	Description
Red	DC5V
Black	Ground
Yellow	RS485-A
White	RS485-B

2.2.3 TTL232 reader

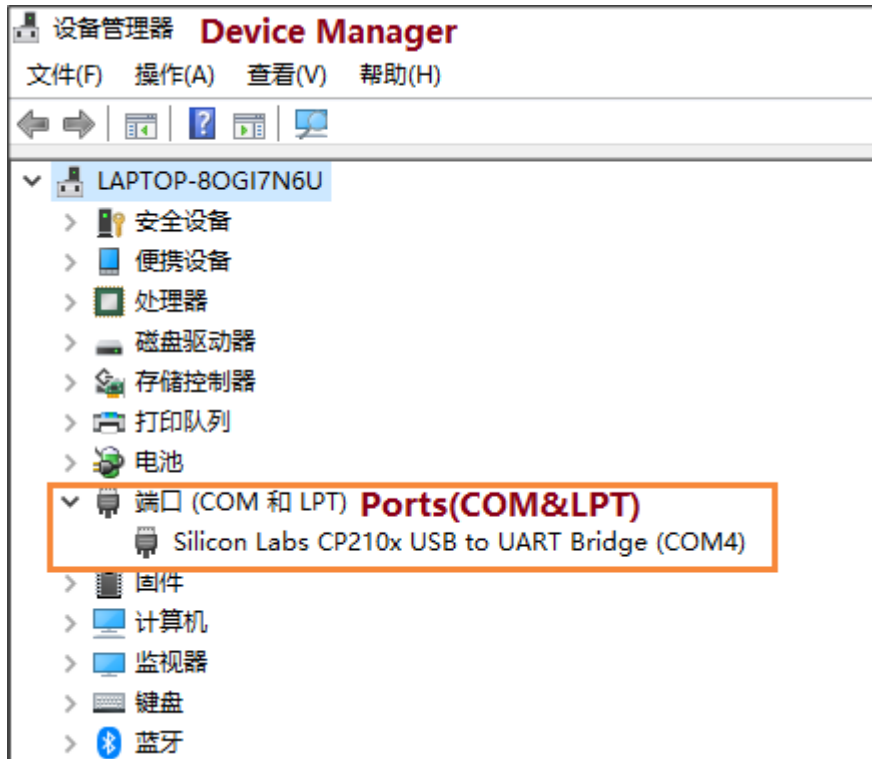
To RW210AT, RW210CT



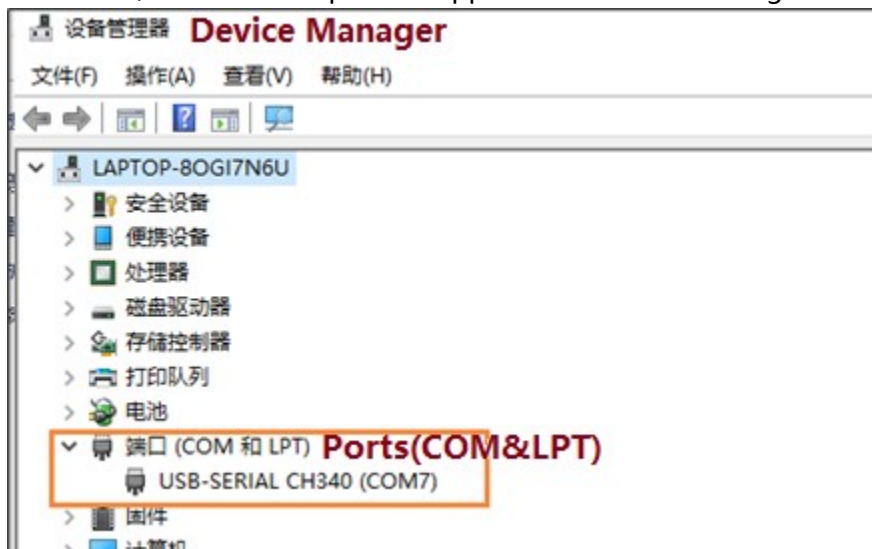
Cable color	Description
Red	DC5V
Black	Ground
Yellow	TTL232 : TXD
White	TTL232 : RXD

2.2.4 USB reader (USB virtual COM port)

(1) RW210AC and RW210CC use CP2102 as the USB-to-UART bridge controller. After install the reader driver, a virtual COM port will appear in the device manager. As shown below :



(2) RW210ACN/RW210CCN use CH340 as the USB-to-UART bridge controller. After install the reader driver, a virtual COM port will appear in the device manager. As shown below :



2.2.5 USB reader (Not need to install driver)

RW210AH or RW210CH is USB HID reader, and does not need to install driver. Once connect to PC USB port, a USB-HID device will appear in the device manager. As shown below(Different versions of Windows may display differently) :



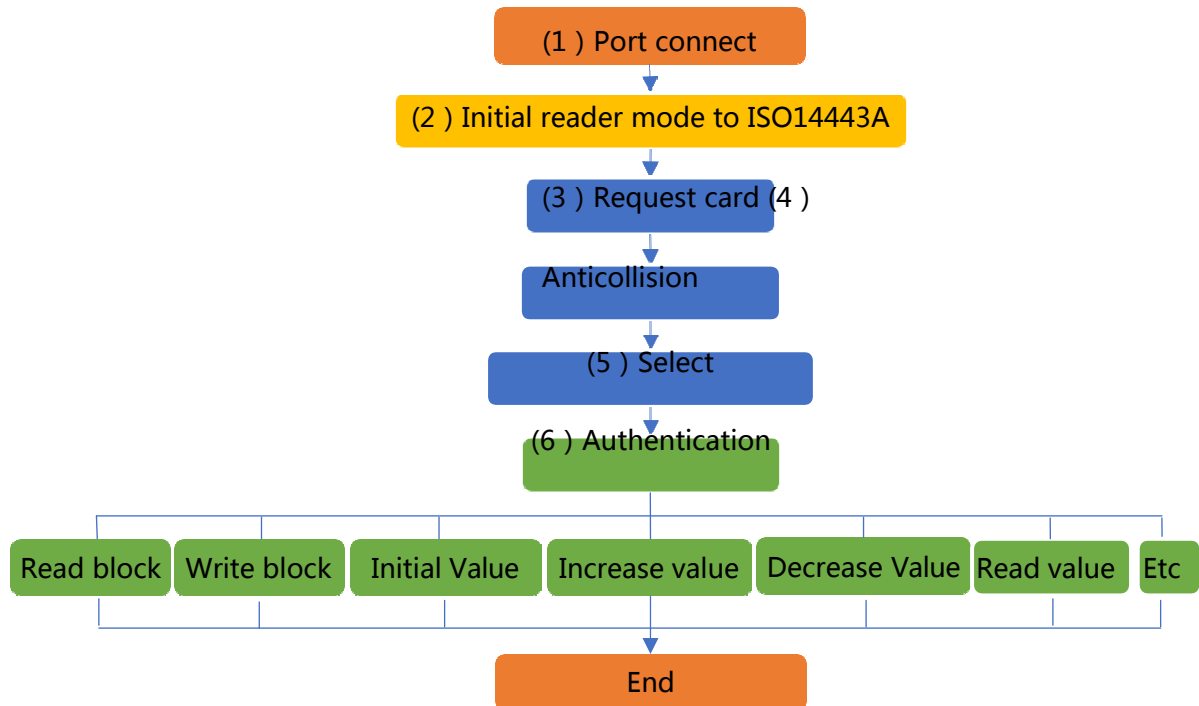
Remarks: After successful hardware connection, you can use PC_DEMO provided by YuanZhi for preliminary functional test.

2.3 Power-on status

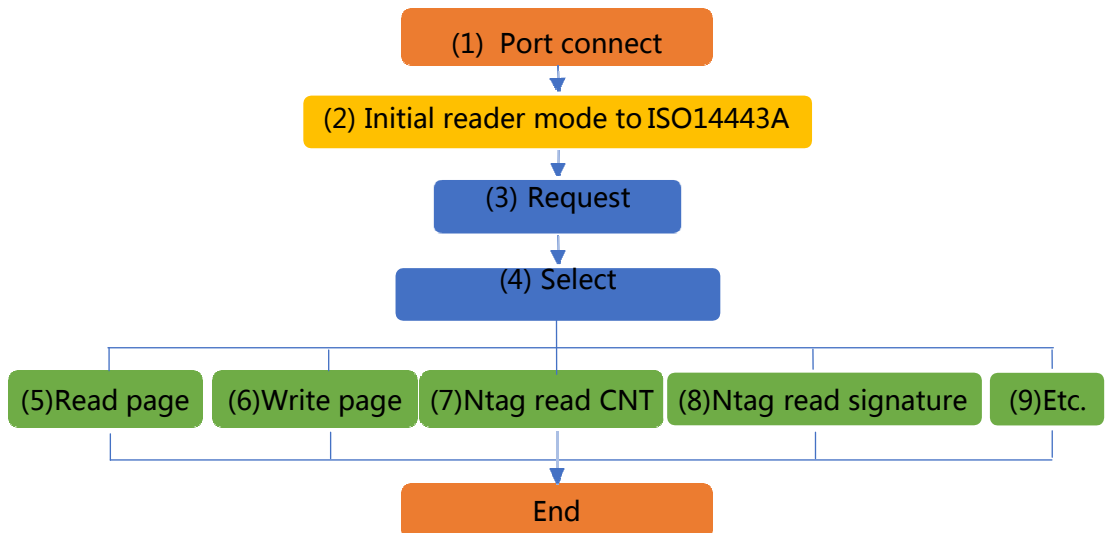
After the reader power on, the yellow and green LEDs flash several times and then turn off. The red LED is always on. The reader has the function of automatic detecting card after power on. When the supported card is in antenna, the green LED will be on, but no card UID is sent out. If you need to send card UID automatically, please contact YuanZhi to get a new firmware version. And this function will be closed automatically after receiving the first command from the host device.

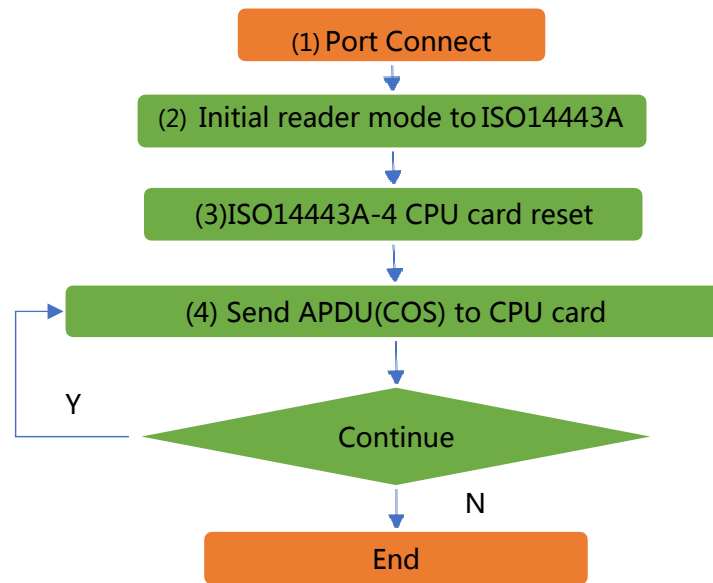
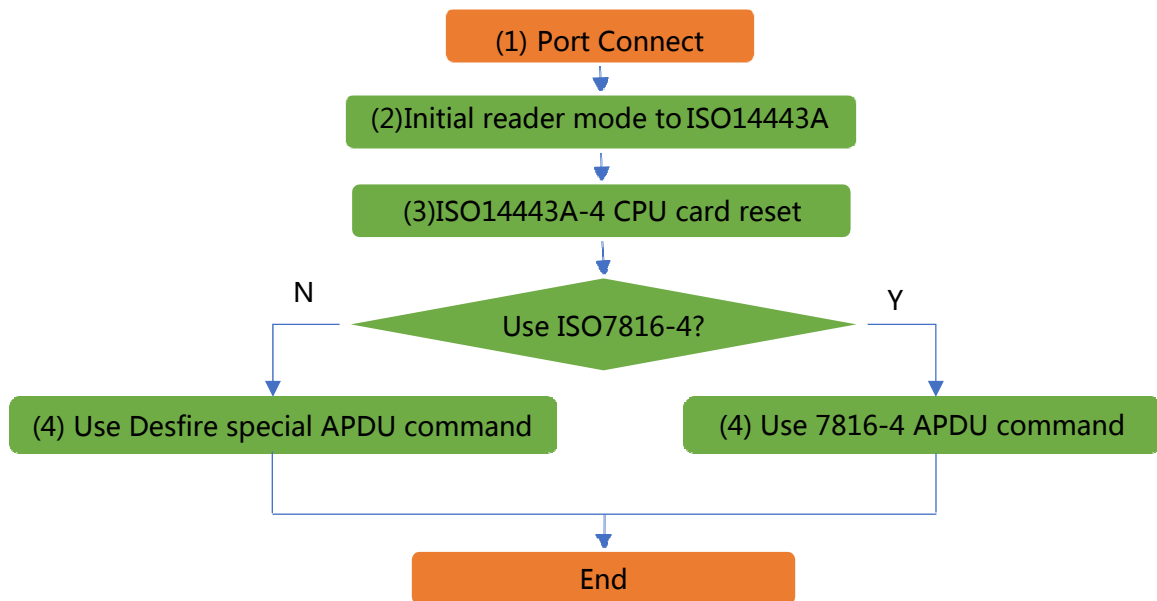
3. Description of card operation steps

3.1 Flowchart for MF1 Card:

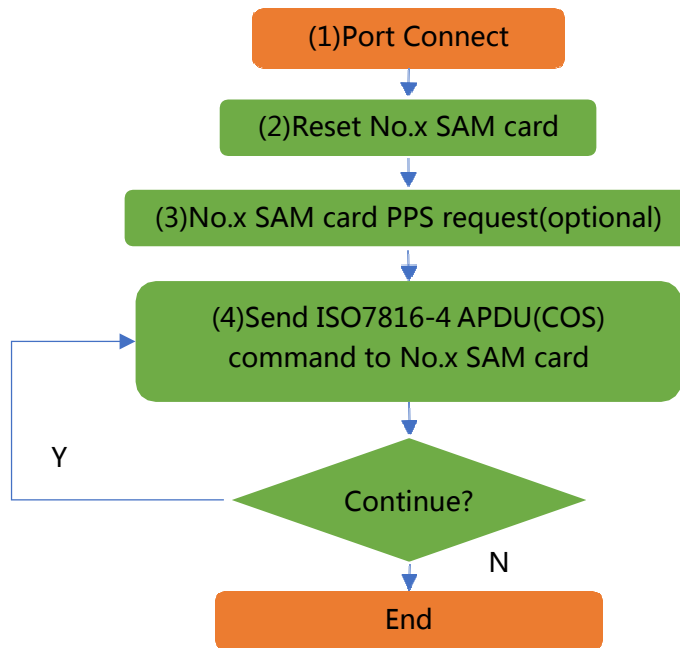


3.2 Flowchart for Mifare Ultralight / Ntag Card:



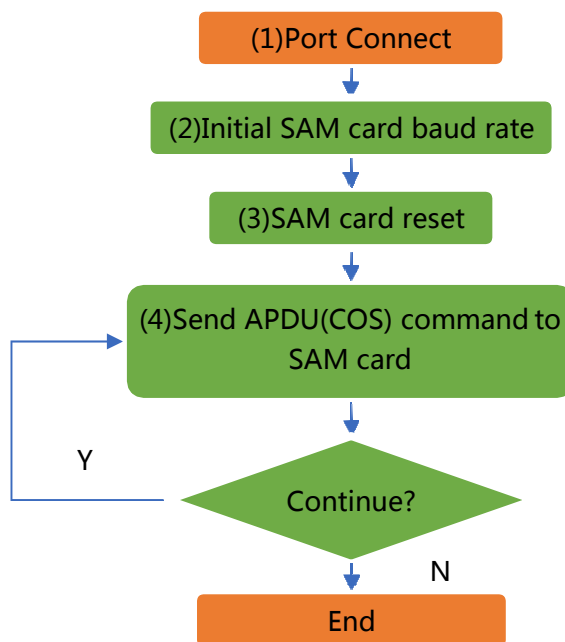
3.3 Flowchart for ISO14443-4 TYPE A CPU card**3.4 Flowchart for Mifare Desfire EV0/EV1/EV2 card:**

3.5 Flowchart for ISO17816-4 SAM/SIM/CPU Card:



Remarks: This flowchart is for new command list, supports multiple cards, and new users recommend to use this command list

3.6 Flowchart for ISO17816-4 SAM/SIM/CPU Card (Only support 1 pc card)



Remarks: This flowchart is for old command list, and new users recommend to use new command list

4. Description of DLL library functions

4.1 General description of DLL library

To RW210AA, RW210AB, RW210AC, RW210ACN, RW210AT:

RW210CA, RW210CB, RW210CC, RW210CCN, RW210CT:

- (1) Under 32-bit Windows System, please copy 32-bit DLLs of "MasterRDnew.dll" and "MasterCom.dll" to the folder of "C:\Windows\system32".
- (2) Under 64-bit Windows System, please copy 32-bit DLLs of "MasterRDnew.dll" and "MasterCom.dll" to the folder of " C:\Windows\SysWOW64".
- (3) DLL library provided by YuanZhi is 32-bit, and if you want to get 64-bit DLL library, please contact YuanZhi directly.
- (4) The length field of default DLL provided by YuanZhi is 1 byte, but to the reader of RW210XX-L2, it needs to support 2 bytes length, so if you wang to get the DLL which support 2 byte, please contact YuanZhi directly.

To RW210AH:

RW210CH:

- (1) Under 32-bit Windows System, please copy 32-bit DLLs of "MasterRDHID.dll" and "MasterHID.dll" to the folder of "C:\Windows\system32".
- (2) Under 64-bit Windows System, please copy 32-bit DLLs of "MasterRDHID.dll" and "MasterHID.dll" to the folder of " C:\Windows\SysWOW64".
- (3) DLL library provided by YuanZhi is 32-bit, and if you want to get 64-bit DLL library, please contact YuanZhi directly.
- (4) The length field of default DLL provided by YuanZhi is 1 byte, but to the reader of RW210XX-L2, it needs to support 2 bytes length, so if you wang to get the DLL which support 2 byte, please contact YuanZhi directly.

4.2 General system function

General system function: Get dll version

Functions: Gets dll version.

Prototype: int WINAPI lib_ver(unsigned int *nVer)

Parameters : nVer : [OUT] Dll version Return :

Sucess returns 0

Failure returns Not 0

General system function: DES encrypt

Functions: DES encrypt.

[illegible]

Parameters : szOut : [OUT] Ciphertext, and output of length is equal to input
 szIn : [IN] Plaintext, which is a multiple of 8 bytes in length
 nInLg : [IN] Length of input plaintext
 pKey : [IN] DES or TDES Key
 nKeyLg : [IN] Length of key=8(Single DES)
 =16(Triple DES)

Return : Success returns 0

Failure returns Not 0

General system function: DES Decrypt

Functions: DES decrypt.

[illegible]

Parameters :

- pszOut : [OUT] Ciphertext, and output of length is equal to input
- pszIn : [IN] Plaintext, which is a multiple of 8 bytes in length
- nInLg : [IN] Length of input plaintext
- pKey : [IN] DES or TDES Key
- nKeyLg : [IN] Length of key=8(Single DES)
=16(Triple DES)

Return : Success returns 0

Failure returns Not 0

General system function: Calculate MAC

Functions: Calculates MAC.

```

Prototype: int WINAPI Cal_MAC (unsigned char *pszOut,
                               unsigned char *pszIn, unsigned int nInLg,
                               unsigned char *pKey, unsigned int nKeyLg,
                               unsigned char *pInitVal, unsigned char byInitLg)

```

Parameters :

- pszOut : [OUT] Return 4 bytes MAC
- pszIn : [IN] Input data , <256
- nInLg : [IN] Length of input data
- pKey : [IN] DES or TDES Key
- nKeyLg : [IN] Length of key=8(Single DES)

=16(Triple DES)

pInitVal : [IN] Initial value for MAC calculation
 byInitLg : [IN] Length of initial value

Return : Success returns 0

Failure returns Not 0

General system function: Key scatter

Functions: DES decrypt.

Prototype: int WINAPI Key_Scatter (unsigned char *pszOut,
 unsigned char *pszIn, unsigned int nInLg,
 unsigned char *pKey, unsigned int nKeyLg)

Parameters : pszOut : [OUT] 16 bytes scatter key
 pszIn : [IN] 8 bytes input data, e.g., Card UID
 nInLg : [IN] Length of input data
 pKey : [IN] DES or TDES Key
 nKeyLg : [IN] Length of key=8(Single DES)
 =16(Triple DES)

Return : Success returns 0

Failure returns Not 0

General system function: Port connect

Functions: Host connects reader.

To RW210AA, RW210AB, RW210AC, RW210ACN, RW210AT:

RW210CA, RW210CB, RW210CC, RW210CCN, RW210CT:

Prototype: int WINAPI rf_init_com (unsigned short wIcdev, int nPort, long lBaud) Parameters :
 wIcdev : [IN] Reader address

=0 or same with the reader embedded address nPort :

[IN] Com port

lBaud : [IN] Communication baudrate (9600 , 14400 , 19200 , 28800 , 38400 ,
 57600 , 115200)

Return : Success returns 0

Failure returns Not 0

Remarks: (1)Factory default baudrate is 19200

(2)Dll library is "MasterRDnew.dll" and "MasterCom.dll"

To RW210AH:

RW210CH:

Prototype: int WINAPI rf_init_com (unsigned short wIcdev, int nVID, int nPID)

Parameters : wIcdev : [IN] Reader address
=0 or same with the reader embedded address nVID :
[IN] Value=6790 (0x1A86)
nPID : [IN] Value=57360 (0xE010)

Return : Success returns 0

Failure returns Not 0

Remarks: Dll library is "MasterRDHID.dll" and "MasterHID.dll"

General system function: Close port

Functions: Close port and free up Windows memory resource.

Prototype: int WINAPI rf_ClosePort ()

Parameters : Null Return :

Success returns 0

Failure returns Not 0

General system function: Set reader working mode

Functions: Sets reader working mode.

Prototype: int WINAPI rf_init_type (unsigned short wIcdev, unsigned char byType) Parameters :

wIcdev : [IN] Reader address

=0 or same with the reader embedded address

byType : [IN] Reader working mode

=0x41 ('A') : Set to ISO14443A mode (MF1, Ultralight, Ntag, TypeA CPU, etc)

=0x42 ('B') : Set to ISO14443B mode (Type B CPU card, Name Card, etc)

=0x43 ('C') : Set to ISO14443C mode (Felica, etc)

=0x73 ('s') : Set to ISO14443B-ST mode (SRI512, SRI4K, etc)

=0x72 ('r') : Set to ISO14443B-Atmel mode (AT88RF020, etc)

=0x31 ('1') : Set to ISO15693 mode (I CODE SL2, Tag it, etc)

Return : Success returns 0

Failure returns Not 0

General system function: Reader antenna set

Functions: Sets reader antenna.

Prototype: int WINAPI rf_antenna_sta (unsigned short wIcdev, unsigned char byMode) Parameters :

wIcdev : [IN] Reader address

=0 or same with the reader embedded address

byMode : [IN] Antenna status:

bit0=0 Close antenna ; =1 Open antenna
 bit1=0 Close auto polling; =1 Open auto polling
 bit2=0 Noting to do; =1 Reader IC hardware reset

Remarks: Bit1 and bit2 functions require reader support.

Bit3-bit7 : fixed to 00000b

Return : Success returns 0

Failure returns Not 0

General system function: Reader antenna set(new)

Functions: sets reader antenna. Dll version should be greater than or equal to V28.

Prototype: int WINAPI rf_antenna_sta_new (unsigned short wIcdev,
 unsigned char *pSet, unsigned char bySetLg)

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address

When bySetLg=1:

pSet[0] : [IN] Antenna status:

bit0=0 Close antenna ; =1 Open antenna
 bit1=0 Close auto polling; =1 Open auto polling
 bit2=0 Noting to do; =1 Reader IC hardware reset
 Bit3-bit7 : fixed to 00000b

When bySetLg=3:

pSet[0] : [IN] Antenna status:

bit0=0 Close antenna ; =1 Open antenna
 bit1=0 Close auto polling; =1 Open auto polling
 bit2=0 Noting to do; =1 Reader IC hardware reset
 Bit3-bit7 : fixed to 00000b

pSet[1] : [IN] The most significant byte of timeout setting pSet[2] :

[IN] The least significant byte of timeout setting

Total time(ms)= pSet [1] *256+ pSet [2]

Return : Success returns 0

Failure returns Not 0

Remarks: Bit1, bit2 and bySetLg=3 functions require reader support.

General system function: Buzzer control

Functions: Controls buzzer time.

Prototype: int WINAPI rf_beep (unsigned short wIcdev, unsigned char byMsec)

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address

byMsec : [IN] Buzzer time = msec * 10ms

Return : Success returns 0
Failure returns Not 0

General system function: Led control

Functions: Controls led on or off.

Prototype: int WINAPI rf_light (unsigned short wIcdev, unsigned char byColor)

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address

byColor : [IN] LED status

=0x00: Yellow LED turns off (GPIO pin outputs high level)

=0x01: Green LED turns off (GPIO pin outputs high level)

=0x02: Green LED turns on (GPIO pin outputs low level)

=0x03: Yellow LED turns on (GPIO pin outputs low level)

Return : Success returns 0
Failure returns Not 0

General system function: Get reader version

Functions: Gets reader SW/HW version.

Prototype: int WINAPI rf_get_model (unsigned short wIcdev, unsigned char *pVer) Parameters :

icdev : [IN] Reader address

=0 or same with the reader embedded address pVer :

[OUT] return SW/HW version

Return : Success returns 0
Failure returns Not 0

General system function: Get reader serial number

Functions: Gets reader serial number.

Prototype: int WINAPI rf_get_snr (unsigned short wIcdev, unsigned char *pSnr, int *pSnrLg)

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address

pSnr : [OUT] Return reader serial number

pSnrLg : [OUT] Length of return data

Return : Success returns 0
Failure returns Not 0

General system function: Set reader address

Functions: Sets new address, **and this function requires reader support.**

Prototype: int WINAPI rf_init_device_number (unsigned short wIcdev,

~~unsigned short wIcdevNew)~~

Parameters : wIcdev : [IN] Reader address
=0 or same with the reader embedded address
wIcdevNew : [IN] New reader address, and value = 0x0000~0xFFFF

Return : Sucess returns 0
Failure returns Not 0

General system function: Get reader address

Functions: Gets reader address.

Prototype: int WINAPI rf_get_device_number (unsigned short wIcdev,
unsigned short *IcdevCur)

Parameters : wIcdev : [IN] Reader address
=0 or same with the reader embedded address wIcdevCur :
[OUT] Return reader address

Return : Sucess returns 0
Failure returns Not 0

4.3 Functions for MF1 card

Special for MF1 card: ISO14443A card request

Functions: Requests an ISO14443A card.

Prototype: int WINAPI rf_request (unsigned short wIcdev, unsigned char byMode, unsigned short *pTagType) Parameters :

wlcdev : [IN] Reader address
=0 or same with the reader embedded address
byMode : [IN] Request mode
=0x26 Requests card in dile state
=0x52 Requests card in halt state pTagType :
[OUT] ATQA information

Return : Success returns 0
Failure returns Not 0

Special for MF1 card: ISO14443A card anticollision

Functions: Anticollision for ISO14443A card.

Prototype: int WINAPI rf_antcoll (unsigned short wIcdev, unsigned char byCnt,
unsigned char *pSnr, unsigned char *pSnrLq) Parameters :

wIcdev : [IN] Reader address
=0 or same with the reader embedded address
byCnt : [IN] =4

pSnr : [OUT] 4/7/10 bytes UID

pSnrLg : [OUT] Length of UID

Return : Success returns 0

Failure returns Not 0

Special for MF1 card: ISO14443A card select

Functions : Selects an IOS14443A card.

Prototype : int WINAPI rf_select (unsigned short wIcdev, unsigned char *pSnr,
unsigned char bySnrLg, unsigned char *pSize)

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address pSnr :

[IN] UID

bySnrLg : [IN] Length of UID

pSize : [OUT] SAK information

Return : Success returns 0

Failure returns Not 0

Special for MF1 card: Mifare One Authentication

Functions : Authenticates a block of MifareOne card.

```

Prototype : int WINAPI rf_M1_authentication2(unsigned short wIcdev, unsigned char byMode,
                                             unsigned char byBlockNo, unsigned char *pKey)

```

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address

byMode : [IN] Key mode

=0x60 : Authentication key A

=0x61 : Authentication key B

byBlockNo : [IN] BlockNo

=0x00~0x3F : Mifare S50

=0x00~0xFF : Mifare S70

pKey : [IN] 6 bytes key (Default key is 6 bytes "0xFF") Return :

Success returns 0

Failure returns Not 0

Special for MF1 card: Mifare One read block

Functions : Reads a block of MifareOne card.

Prototype : int WINAPI rf_M1_read (unsigned short wIcdev, unsigned char byBlockNo,
unsigned char *pData, unsigned char *pDataLq)

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address byBlockNo :

[IN] MF1 card block No.

pData : [OUT] 16 bytes block data

pDataLg: [OUT] Length of response data

Return : Success returns 0

Failure returns Not 0

Special for MF1 card: Mifare One write block

Functions : Writes a block of MifareOne card. Prototype :

```
int WINAPI rf_M1_write (unsigned short wIcdev,
                        unsigned char byBlockNo,
                        unsigned char *pData)
```

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address byBlockNo :

[IN] MF1 card block No.

pData : [IN] 16 bytes block data

Return : Success returns 0

Failure returns Not 0

Special for MF1 card: Mifare One initail purse value

Functions : Initials a block into purse format of Mifare One card. Prototype :

```
int WINAPI rf_M1_initval (unsigned short wIcdev,
                          unsigned char byBlockNo,
                          long lValue)
```

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address byBlockNo :

[IN] MF1 card block No.

lValue : [IN] Purse value (Decimal value)

Return : Success returns 0

Failure returns Not 0

Special for MF1 card: Mifare One read purse value

Functions : Reads purse value of Mifare One card. Prototype :

```
int WINAPI rf_M1_readval (unsigned short wIcdev,
                          unsigned char byBlockNo,
                          long *pValue)
```

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address

byBlockNo : [IN] MF1 card block No. pValue :
[OUT] Purse value (Decimal value)

Return : Success returns 0

Failure returns Not 0

Special for MF1 card: Mifare One decrease purse value

Functions : Decreases purse value of Mifare One card. Prototype :

```
int WINAPI rf_M1_decrement (unsigned short wIcdev,  
                             unsigned char byBlockNo,  
                             long lValue)
```

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address byBlockNo :

[IN] MF1 card block No.

lValue : [IN] Purse value (Decimal value)

Return : Success returns 0

Failure returns Not 0

Special for MF1 card: Mifare One increase purse value

Functions : Increases purse value of Mifare One card. Prototype :

```
int WINAPI rf_M1_increment (unsigned short wIcdev,  
                             unsigned char byBlockNo,  
                             long lValue)
```

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address byBlockNo :

[IN] MF1 card block No.

lValue : [IN] Purse value (Decimal value)

Return : Success returns 0

Failure returns Not 0

Special for MF1 card: Mifare One restore purse value

Functions : Copies Mifare One card purse value into card internal volatile memory.

Prototype : int WINAPI rf_M1_restore (unsigned short wIcdev, unsigned char bySrcBlockNo)

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address bySrcBlockNo :

[IN] Source block No.

Return : Success returns 0

Failure returns Not 0

Special for MF1 card: Mifare One transfer purse value

Functions : Writes the value stored in the internal volatile memory into the destination block.

Prototype : int WINAPI rf_M1_transfer (unsigned short wIcdev, unsigned char byDestBlockNo)

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address

byDestBlockNo : [IN] Destination block No.

Return : Success returns 0

Failure returns Not0

Remarks : (1) Destination block No, should be in same sector with the source block No. in Restore command.

(2) This function is called only after restore command.

Special for MF1 card: Mifare One read sector

Functions : Reads 3 blocks in the same sector of Mifare One card. Prototype :

```
int WINAPI rf_M1_read_sector (unsigned short wIcdev,
```

unsigned char byStartBlockNo,

```
unsigned char *pData, unsigned char *pDataLg)
```

Parameters : wlcdev : [IN] Reader address

=0 or same with the reader embedded address byStartBlockNo :

[IN] Start block No.

pData : [OUT] 48 bytes data of 3 blocks

pDataLq: [OUT] Length of response data

Return : Success returns 0

Failure returns Not 0

Remarks : (1) This function requires reader support.

(2) "Start block No.+2" should be in same sector

Special for MF1 card: Mifare One write sector

Functions : Writes 3 blocks in the same sector of Mifare One card. Prototype :

```
int WINAPI rf_M1_write_sector (unsigned short wIcdev,
```

unsigned char byStartBlockNo,

```
unsigned char *pData)
```

Parameters : wlcdev : [IN] Reader address

=0 or same with the reader embedded address byStartBlockNo :

[IN] Start block No.

Remarks: Start block No. should be the first block of one sector pData :

[IN] 48 bytes data of 3 blocks

Return : Success returns 0

Failure returns Not 0

Remarks : this function requires reader support.

Special for MF1 card: Halt ISO14443A card

Functions : Halts ISO14443A card.

Prototype : int WINAPI rf_halt (unsigned short wIcdev)

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address

Return : Success returns 0

Failure returns Not 0

4.4 Functions for Ultraligh/Ntag card

Special for Ultralight/Ntag card: ISO14443A card request

Functions: Requests an ISO14443A card.

Prototype: int WINAPI rf_request (unsigned short wIcdev, unsigned char byMode,
unsigned short *pTagType) Parameters : wIcdev :

[IN] Reader address

=0 or same with the reader embedded address

byMode : [IN] Request mode

=0x26 Requests card in dile state

=0x52 Requests card in halt state pTagType :

[OUT] ATQA information

Return : Success returns 0

Failure returns Not 0

Special for Ultralight/Ntag card: Ultralight/Ntag card select

Functions: Selects a card of Ultralight/Ntag.

Prototype: int WINAPI rf_ul_select (unsigned short wIcdev,
unsigned char *pSnr, unsigned char *pSnrLg) Parameters :

wIcdev : [IN] Reader address

=0 or same with the reader embedded address pSnr :

[OUT] 7/10 bytes UID

pSnrLg : [OUT] Length of UID Return :

Success returns 0

Failure returns Not 0

Special for Ultralight/Ntag card: Ultralight/Ntag card read pages

Functions : Reads 4 pages of Ultralight/Ntag card.

Prototype : int WINAPI rf_M1_read (unsigned short wIcdev, unsigned char byBlockNo,
 unsigned char *pData, unsigned char *pDataLg)

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address byBlockNo :

[IN] Ultralight/Ntag card first page No.

pData : [OUT] 4 pages data

pDataLg: [OUT] Length of response data Return :

Sucess returns 0

Failure returns Not 0

Special for Ultralight/Ntag card: Ultralight/Ntag card write page

Functions: Writes a page of Ultralight/Ntag card.

Prototype: int WINAPI int rf_ul_write (unsigned short wIcdev, unsigned char byPage,
 unsigned char *pData)

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address

byPage : [IN] Ultralight/Ntag card page No.

pData : [IN] 4 bytes data

Return : Sucess returns 0

Failure returns Not 0

Special for Ntag21x card: Ntag21x card fast read

Functions: Reads multiple pages of Ntag21x card.

Prototype: int WINAPI rf_Ntag_Fast_Read (unsigned short wIcdev,
 unsigned char byStartAdr, unsigned char byEndAdr,
 unsigned char *pData, unsigned char *pDataLg)

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address byStartAdr :

[IN] Ntag21x card first page No.

byEndAdr : [IN] Ntag21x card end page No.

pData : [OUT] ((cEndAdr – cStartAdr +1) * 4) byte data

pDataLg: [OUT] Length of response data

Return : Sucess returns 0

Failure returns Not 0

Remarks: (cEndAdr – cStartAdr +1) < 0x0F

Special for Ntag21x card: Ntag21x card get version

Functions: Gets version of Ntag21x card.

Prototype: ~~int WINAPI rf_Ntag_Get_Version (unsigned short wIcdev,
unsigned char *pData, unsigned char *pDataLg);~~

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address

pData : [OUT] 8 bytes data

pDataLg: [OUT] Length of response data

Return : Success returns 0

Failure returns Not 0

Special for Ntag21x card: Ntag21x card key authentication

Functions: Ntag21x card key authentication, and before using this function, you need to enable the area protection.

Prototype: int WINAPI rf_Ntag_Password_Auth (unsigned short wIcdev,
unsigned char *pPWD,
unsigned char *pData, unsigned char *pDataLg)

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address

pPWD : [IN] 4 bytes key.

pData : [OUT] 2 bytes PACK

pDataLg: [OUT] Length of response data

Return : Success returns 0

Failure returns Not 0

Special for Ntag21x card: Ntag21x card read CNT

Functions: Reads CNT of Ntag21x card.

Prototype: int WINAPI rf_Ntag_Read_CNT (unsigned short wIcdev,
unsigned char *pData, unsigned char *pDataLg);

Parameters : wIcdev : [IN] Reader address

=0 or same with the reader embedded address

pData : [OUT] 3 bytes CNT

pDataLg: [OUT] Length of response data

Return : Success returns 0

Failure returns Not 0

Special for Ntag21x card: Ntag21x card read signature

Functions: Reads signature of Ntag21x card.

Prototype: int WINAPI rf_Ntag_Read_Signature (unsigned short wIcdev,
unsigned char *pData, unsigned char *pDataLg);

Parameters : wIcdev : [IN] Reader address
 =0 or same with the reader embedded address
 pData : [OUT] 32 bytes signature
 pDataLg: [OUT] Length of response data
 Return : Success returns 0
 Failure returns Not 0

Special for NFC type2 card: NFC type2 card select sector

Functions: Selects sector of NFC type2 card.

Prototype: int WINAPI rf_NFC_Select_Sector (unsigned short wIcdev,
 unsigned char bySec);

Parameters : wIcdev : [IN] Reader address
 =0 or same with the reader embedded address
 bySec : [IN] 1 byte sector No, and the value=0x00~0xFE.
 Return : Success returns 0
 Failure returns Not 0

Special for Ntag21x card: Halt ISO14443A card

Functions : Halts ISO14443A card.

Prototype : int WINAPI rf_halt (unsigned short wIcdev)

Parameters : icdev : [IN] Reader address
 =0 or same with the reader embedded address
 Return : Success returns 0
 Failure returns Not 0

4.5 ISO14443A-4 CPU card

Special for TYPE A CPU card: ISO14443A-4 CPU card reset

Functions : ISO14443A-4 CPU card reset

Prototype : int WINAPI rf_typea_rst (unsigned short wIcdev, unsigned char byMode,
 unsigned char *pData, unsigned char *pDataLg)

Parameters : wIcdev : [IN] Reader address
 =0 or same with the reader embedded address
 byMode : [IN]
 =0x26 Requests card in dile state
 =0x52 Requests card in halt state
 pData : [OUT] 4/7/10 bytes UID + CPU card reset information
 pDataLg : [OUT] Length of response data
 Return : Success returns 0

Failure returns Not 0

Special for TYPE A CPU card: Send APDU(COS) to ISO14443A-4 CPU card

Functions : Send APDU(COS) to ISO14443A-4 CPU card

Prototype: int WINAPI rf_cos_command (unsigned short wIcdev,
 unsigned char *pCmd, unsigned int nCmdLg,
 unsigned char *pData, unsigned int *pDataLg)

Parameters : wIcdev: [IN] Reader address
 =0 or same with the reader embedded address
 pCmd : [IN] APDU(COS) command
 nCmdLg: [IN] Length of APDU(COS) command
 pData: [OUT] Response data, and the data contains SW1, SW2
 pDataLg : [OUT] Length of response data

Return : Success returns 0

Failure returns Not 0

Special for TYPE A CPU card: Send PPS to ISO14443A-4 CPU card

Functions: Send PPS to ISO14443A-4 CPU card **Remarks :**

this function requires reader support.

Prototype: int WINAPI rf_ISO14443_PPS (unsigned short wIcdev,
 unsigned char byCID,
 unsigned char byMode)

Parameters: wIcdev: [IN] Reader address
 =0 or same with the reader embedded address
 byCID : [IN] byCID=0
 byMode: cMode=0x00: 106K
 cMode=0x01: 212K
 cMode=0x02: 424K
 cMode=0x03: 848K

Return : Success returns 0

Failure returns Not 0

Special for TYPE A CPU card: Deselect ISO14443A-4 CPU card

Functions: Deselect ISO14443A-4 CPU card

Prototype: int WINAPI rf_cl_deselect (unsigned short wIcdev)

Parameters: wIcdev: [IN] Reader address
 =0 or same with the reader embedded address

Return : Success returns 0

Failure returns Not 0

4.6 ISO7816-4 SAM/SIM/CPU card(new)

Special for ISO7816-4 SAM/SIM Card: No.x SAM/SIM Card reset

Functions : No.x ISO7816-4 CPU/SAM/SIM card reset.

Prototype: int WINAPI rf_sam_rst_new (unsigned short wIcdev,
 unsigned char *pSet, unsigned char bySetLg,
 unsigned char *pData, unsigned char *pDataLg)

Parameters: wIcdev : [IN] Reader address
 =0 or same with the reader embedded address
 pSet : [IN] bit1-bit0 SAM cards communication baudrate :
 =00: Set to 9600 ;
 =01: Set to 38400 ;
 =10: Set to 115200 ;
 bit3-bit2 :
 =00: Reset operation ;
 bit7-bit6-bit5-bit4 SAM card No :
 =0000 : No.1 ;
 =0001 : No.2 ;
 =0010 : No.3 ;
 =0011 : No.4 ;
 =0100 : No.5 ;
 bySetLg : [IN] value=1
 pDate : [OUT] Response data from Card
 pDataLg : [OUT] Length of response data

Return : Success returns 0

Failure returns Not 0

Special for ISO7816-4 SAM/SIM Card: Send APDU(COS) to No.x SAM/SIM Card

Functions: Send APDU(COS) to ISO7816-4 CPU/SAM/SIM card

Prototype: int WINAPI rf_sam_cos_new (unsigned short wIcdev, unsigned char bySAMNo,
 unsigned char *pCmd, unsigned int nCmdLg,
 unsigned char *pData, unsigned int *pDataLg)

Parameters: wIcdev : [IN] Reader address
 =0 or same with the reader embedded address
 bySAMNo : [IN] SAM card No
 pCmd : [IN] APDU(COS) command nCmdLg :
 [IN] Length of APDU(COS) command

pDate : [OUT] Response data from card, and the data contains SW1/SW2
 pDataLg : [OUT] Length of response data

Return : Success returns 0

Failure returns Not 0

Special for ISO7816-4 SAM/SIM Card: Send PPS request to No.x SAM/SIM card

Functions: Send PPS to ISO7816-4 CPU/SAM/SIM card

Prototype: int WINAPI rf_sam_rst_new (unsigned short wIcdev,
 unsigned char *pSet, unsigned char bySetLg,
 unsigned char *pData, unsigned char *pDataLg)

Parameters: wIcdev : [IN] Reader address
 =0 or same with the reader embedded address

When bySetLg =1:

pSet [0]: [IN] bit1-bit0 :
 =XX (Not use) ;
 bit3-bit2 :
 =10 (PPS to 38400 (cSetLen=1))
 bit7-bit6-bit5-bit4 SAM card No :
 =0000 : No.1 ;
 =0001 : No.2 ;
 =0010 : No.3 ;
 =0011 : No.4 ;
 =0100 : No.5 ;

When bySetLg =3:

pSet [0]: [IN] bit1-bit0 :
 =XX (Not use) ;
 bit3-bit2 :
 =11 (PPS to 115200 (cSetLen=3))
 bit7-bit6-bit5-bit4 SAM card No :
 =0000 : No.1 ;
 =0001 : No.2 ;
 =0010 : No.3 ;
 =0011 : No.4 ;
 =0100 : No.5 ;
 pSet [1]: [IN] =0x10 (cSetLen=3);
 pSet [2]: [IN] =0x13 (PPS to 38400 (cSetLen=3));
 =0x18 (PPS to 115200 (cSetLen=3));

bySetLg: [IN] =1 or =3;

pDate : [OUT] Response data from Card
 pDataLg : [OUT] Length of response data
 Return : Success returns 0
 Failure returns Not 0

4.7 ISO7816-4 SAM/SIM/CPU card

Special for ISO7816-4 SAM/SIM Card : Initial baudrate

Functions: Set ISO7816-4 CPU/SAM/SIM card communication baudrate.
 Prototype: int WINAPI rf_init_sam (unsigned short wIcdev, unsigned char byBaud)
 Parameters: wIcdev: [IN] Reader address
 =0 or same with the reader embedded address
 byBaud : [IN] SAM cards communication baudrate
 =0x00 : Set to 9600
 =0x01 : Set to 38400
 =0x02 : Set to 115200
 Return : Success returns 0
 Failure returns Not 0

Special for ISO7816-4 SAM/SIM Card : SAM/SIM Card reset

Functions : ISO7816-4 CPU/SAM/SIM card reset. Prototype :
 int WINAPI rf_sam_rst (unsigned short wIcdev,
 unsigned char *pData, unsigned char *pDataLg)
 Parameters : wIcdev : [IN] Reader address
 =0 or same with the reader embedded address
 pData: [OUT] Response data from Card
 pDataLg: [OUT] Length of response data
 Return : Success returns 0
 Failure returns Not 0

Special for ISO7816-4 SAM/SIM Card : Send APDU(COS) to SAM/SIM Card

Functions : ISO7816-4 CPU/SAM/SIM card send APDU(COS) command.
 Prototype : int WINAPI rf_sam_cos (unsigned short wIcdev,
 unsigned char *pCmd, unsigned char byCmdLg,
 unsigned char *pData, unsigned char *pDataLg)
 Parameters : wIcdev : [IN] Reader address
 =0 or same with the reader embedded address
 pCmd : [IN] APDU(COS) command

byCmdLg : [IN] Length of APDU(COS) command
 pData : [OUT] Response data from card, and the data contains SW1/SW2
 pDataLg : [OUT] Length of response data

Return : Success returns 0

Failure returns Not 0

FCC Statement

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

RF Exposure Information

Caution: Any changes or modifications to this device not explicitly approved by manufacturer could void your authority to operate this equipment.

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
 - Increase the separation between the equipment and receiver.
 - Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
 - Consult the dealer or an experienced radio/TV technician for help.
- The device has been evaluated to meet general RF exposure requirement. This equipment should be installed and operated with minimum distance 0mm between the radiator & your body.

ISED Statement

English: This device contains licence-exempt transmitter(s)/receiver(s) that comply with Innovation, Science and Economic Development Canada's licence-exempt RSS(s). Operation is subject to the following two conditions:

- (1) This device may not cause interference.
- (2) This device must accept any interference, including interference that may cause undesired operation of the device.

The digital apparatus complies with Canadian CAN ICES-3 (B)/NMB-3(B).

French: Cet appareil contient des émetteurs/récepteurs exempts de licence qui sont conformes aux RSS exemptés de licence d'Innovation, Sciences et Développement économique Canada.

L'exploitation est soumise aux deux conditions suivantes :

- (1) Cet appareil ne doit pas provoquer d'interférences.
- (2) Cet appareil doit accepter toute interférence, y compris les interférences susceptibles de provoquer un fonctionnement indésirable de l'appareil.

l'appareil numérique du ciem conforme canadien peut - 3 (b) / nmb - 3 (b).

This device meets the exemption from the routine evaluation limits in section 2.5 of RSS 102 and compliance with RSS 102 RF exposure, users can obtain Canadian information on RF exposure and compliance. cet appareil est conforme à l'exemption des limites d'évaluation courante dans la section 2.5 du cnr - 102 et conformité avec rss 102 de l'exposition aux rf, les utilisateurs peuvent obtenir des données canadiennes sur l'exposition aux champs rf et la conformité.

This equipment complies with Canada radiation exposure limits set forth for an uncontrolled environment. Cet équipement est conforme aux limites d'exposition aux rayonnements du Canada établies pour un environnement non contrôlé.

RF Exposure Statement

The device has been evaluated to meet general RF exposure requirement. This equipment should be installed and operated with minimum distance 0mm between the radiator & your body.

L'appareil a été évalué pour répondre aux exigences générales d'exposition aux RF. Cet équipement doit être installé et utilisé avec une distance minimale de 0 mm entre le radiateur et votre corps.