/**

RFID DLL User Guide 1.1

First Written in : 2012-04-20

**/

Content:

1. Development Environment

2. Usage

3. Execution

4. API Reference

5. Program Flow Sheet

1. Development Environment

Visual Studio 2005 VC.

2. Usage

(1) Library header

: “RFID_DLL.h” is the library header file.

(2) Library file

: “RFID_DLL.dll” is the library

3. Execution

(1) Copy the provided RFID_DLL.dll and MF2351DLL.dll to the device. The location can be“ \ Windows” directory or the same directory with the application which uses the library .

(2) Copy the application that uses the RFID library in the device and execute it.

4. API Reference

[1] _DLL_EXPORT_ int WINAPI RFID_Init (void);

Description :

Initialization the RFID model.

Return Values :

None.r

Parameter :

None

Note: RFID_Init() should be executed before operating RFID model.

---

[2] _DLL_EXPORT_ int WINAPI RFID_Open (void);

Description :

Turn on RFID model, Reset RFID model .

Return Values :

RFID_OK : Success

Orther value : Failed (details see exception error)

Note: RFID_Open() should be executed once for every transaction

---

[3] _DLL_EXPORT_ int WINAPI RFID_Close ();

Description :

Close RFID model.

Return Values :

RFID_OK : Success

Orther value : Failed

---

[4] _DLL_EXPORT_ int RFID_Check(BYTE mode, BYTE *CardType, BYTE *SerialNo);

Description :

Searched the RFID card, and active the card.

Return Value :

RFID_OK(0) : Success

RFID_ParameterErr(-3503) : Parameter error(Invalid ‘mode’)

RFID_NotOpen(-3502) : RFID not opened

RFID_Collision(-3526) : More Cards

Orther value : Operating failed

Parameter :

BYTE mode: [in]:

0 : Search all cards in the effective range，First A cards.

More A cards->return error

No A card->Search B card->Just one B card->Active B card

Just one A card->Active A card

‘a’ or ‘A’ or hex 0x0a : Just search A card once and active A card (including M1 card)

‘b’ or ‘B’ or hex 0x0b : Just search B card once and active B card

‘m’ or ‘M’ or hex 0x0b : Just search M1 card once and active M1 card

BYTE *CardType [out]:

Buffer of card type returned, two bytes.

CardType[0]: ‘A’ A card founed

‘B’ B card founed

CardType[1]: ‘C’ CPU card founed

‘M’ M1 card founed

BYTE *SerialNo [out]:

Buffer of card ID and length

SerialNo[0]: length of card ID

SerialNo[1~10]: card ID (Aligh left)

Note: If CPU card is searched when parameter is set to be 0, RFID_Reset() should be execute before next operating.

---

[5]_DLL_EXPORT_ int WINAPI RFID_Command(APDU_SEND *ApduSend, APDU_RESP *ApduResp);

Description :

In the specified channel, to the card to send the data in APDU format, and receiving a response.

Return Value :

RFID_OK(0) : Success

RFID_ParameterErr(-3503) : Parameter error(Invalid ‘mode’)

RFID_NotOpen(-3502) : RFID not opened

RFID_ApduErr (-3524) : Data exchange error

Orther value : Operating failed

Parameters :

APDU_SEND *ApduSend : [in]

```
APDU_SEND：
struct{
        BYTE    Command[4];
        WORD    Lc;
        BYTE    DataIn[512];
        WORD    Le;
      };
```

Command[] = {CLA, INS, P1, P2}。

Lc = length of data to send。

DataIn = data to send to RFID model。

Le = data length expected return。

Case1：Lc=0; Le=0

Neither data nor return data

Case2: Lc=0; Le>0

No data transmission but expect return of data, if return data number unknown, should put the Le = 256; otherwise the returned data length must be determined by numerical.

Case3: Lc>0; Le=0

Send data is not empty,but no expected data returned

Case4: Lc>0; Le>0

Send data is not empty, and expected data returned. If return data number unknown, should put the Le = 256; otherwise the returned data length must be determined by numerical。

APDU_RESP *ApduResp :[out]

```
APDU_RESP：
        struct{
              WORD    LenOut;
              BYTE    DataOut[512];
              BYTE    SWA;
              BYTE    SWB;
              };
```

LenOut : The actual data length returned from IC card。

DataOut : The data buffer returned from IC card。

SWA : Status returned。

SWB : Status returned。

Note: API RFID_Check(BYTE mode, BYTE *CardType, BYTE *SerialNo) mustbe be called successfully before this API is called.

---

[6] _DLL_EXPORT_ int WINAPI RFID_M1Authority(BYTE Type,BYTE BlkNo,BYTE *Pwd,BYTE *SerialNoBuf);

Description :

Verify that the M1card access read / write the corresponding block to be submitted to the A B password.

Return Value :

RFID_OK(0) : Success

RFID_NotOpen(-3502) : RFID not opened

RFID_ParameterErr(-3503) : Parameter error

RFID_AuthErr(-3523) : Password authentication failed

Parameters :

BYTE Type : [in]

'a' or 'A' or hex 0x0a :Presented is the A password

'b' or 'B' or hex 0x0b : Presented is the B password

BYTE BlkNo : [in]

Specify access block number, range 0~63.

BYTE *Pwd :[in]

Password buffer.

BYTE *SerialNobuf : [in]

Buffer to the storage card serial number. The buffer should be the ID part from API RFID_Check(BYTE mode, BYTE *CardType, BYTE *SerialNo).

SerialNobuf = SerialNo+1 (Serial1No[1~10]);

Note: API RFID_Check(BYTE mode, BYTE *CardType, BYTE *SerialNo) must be called successfully before this API is called.

---

[7] _DLL_EXPORT_ int WINAPI RFID_M1ReadBlock(BYTE BlkNo,BYTE *BlkValue);

Description :

Read the specified block data from M1 card.

Return Value :

RFID_OK(0) : Success

RFID_NotOpen(-3502) : RFID not opened

RFID_ReadBlockErr (-3522) : Read data failed

Parameters :

BYTE BlkNo : [in]

Specify access block number, range 0~63.

BYTE *BlkValue: [out]

Buffer to the block data, not less than 16 byte

Note: API RFID_M1Authority(BYTE Type,BYTE BlkNo,BYTE *Pwd,BYTE *SerialNoBuf) must be called successfully before this API is called.

---

[8] _DLL_EXPORT_ int WINAPI RFID_M1WriteBlock(BYTE BlkNo,BYTE *BlkValue);

Description :

Write data to the specified block of M1 card.

Return Value :

RFID_OK(0) : Success
RFID_NotOpen(-3502) : RFID not opened
RFID_ReadBlockErr (-3522) : Write data failed

Parameters :

BYTE BlkNo : [in]
Specify access block number, range 0~63.

BYTE *BlkValue: [in]
Buffer to the block data to write

Note: API RFID_M1Authority(BYTE Type,BYTE BlkNo,BYTE *Pwd,BYTE *SerialNoBuf) must be called successfully before this API is called.

---

[9]_DLL_EXPORT_ int WINAPI RFID_M1Operate(BYTE Type,
BYTE BlkNo,BYTE *Value,BYTE UpdateBlkNo);

Description :

Increase or reduce operation to the specified E-purse block

Return Value :

RFID_OK(0) : Success
RFID_NotOpen(-3502) : RFID not opened
RFID_ ParameterErr (-3503) : Write data failed
RFID_OperateErr(-3520) : Operation failed

Parameters :

BYTE Type : [in]
'+' : Increase
'-' : Reduce

BYTE BlkNo: [in]
Specify access block number, range 0~63

BYTE *Value :[in]
Buffer to the amount of cash to be increased or reduced, 4 bytes,Low byte left, Hight byte right. Such as [0x01, 0x00, 0x00,0x00] means 1.

BYTE UpdateBlkNo: [in]
Block number to storage the data of this opreation.

Note: BlkNo is the block number to be operated, UpdateBlkNo is the block number to save the result data of this operation. BlkNo and UpdateBlkNo mus be in same sector!

Note: API RFID_M1Authority(BYTE Type,BYTE BlkNo,BYTE *Pwd,BYTE *SerialNoBuf) must be called successfully before this API is called.

---

[10] _DLL_EXPORT_ int WINAPI RFID_Reset (void);

Description :

Send RATS command to CPU card of type A only. Orther cards should not use this function

Return Value :

RFID_OK(0) : Success

RFID_CardTyepErr (-3527) : Not A card
RFID_RatsErr (-3517) : RATS command error

Parameters :

None

Note: Just for CPU card of type A when searched cards with function RFID_Check()(only for first parameter is 0).

---

[11] _DLL_EXPORT_ int WINAPI RFID_Remove (void);

Description :

Determine whether the card has been removed sensing area

Return Value :

RFID_OK(0) : Success removed
RFID_NotOpen(-3502) : RFID not opened
RFID_HaveCard(-3525) : Card not removed

Parameters :

None

---

[12] _DLL_EXPORT_ int WINAPIRFID_GetFirmwareVersion(BYTE *pVersion);

Description :

Get firmware version

Return Value :

RFID_OK(0) : Success removed
Orhter value : Failed

Parameters :

BYTE *pVersion :[out]
Buffer of firmware version

---

5. Application Flow Sheet：

(1)Read data (Mifare card).

step1: Open the RFID model

RFID_Init();
RFID_Open();

Step2: Search the card and active card

RFID_Check(BYTE mode,BYTE *CardType,BYTE *SerialNo);

Step3: Verify the password

RFID_M1Authority(BYTE Type,BYTE BlkNo,BYTE *Pwd,BYTE *SerialNo);

Step4 : Read data

RFID_M1ReadBlock(BYTE BlkNo,BYTE *BlkValue);

(2)Write data (Mifare card)..

Step1: Search the card and active card

RFID_Check(BYTE mode,BYTE *CardType,BYTE *SerialNo);

Step2: Verify the password

RFID_M1Authority(BYTE Type,BYTE BlkNo,BYTE *Pwd,BYTE *SerialNo);

Step3 : Write data

RFID_M1WriteBlock(BYTE BlkNo,BYTE *BlkValue);

(3)APDU command (A card or B card)

Step1: Search the card and active card

RFID_Open();

RFID_Check(BYTE mode,BYTE *CardType,BYTE *SerialNo);

Step2 : APDU command

RFID_Command(APDU_SEND *ApduSend, APDU_RESP *ApduResp);