DAILY RFID CF & SD & USB Interface Mini HF RFID Reader

CFDLL.DLL Dynamic Link Library

User's Manual V3.11_1

CFDLL.DLL is a dynamic link library designed to facilitate ISO/IEC 15693 protocol or ISO/IEC 14443 A/B HF tag application software development when using DAILY RFID CF & SD & USB Interface Mini HF RFID Reader.

CONTENT

1) Function Explanation

1.Get DLL Version.

Function antitype:

int LibVersion(unsigned int *pVer);

Function:

Obtaining dynamic edition No., the return data length is 2byte

E.g. the return data is “0x02, 0x00”， after the formatting showing “2.0”.

Input parameter: Null

Output parameter:

pVer – API edition No.

2.Set API Work Mode.

Function antitype:

int SetWorkMode (BYTE bType);

Function:

Set API WorkMode, ‘CF’ type or ‘SD’ type

Input parameter:

bType:

0 : CF Mode, API default type.

1 : SD mode

Output parameter:

NULL.

3.Initialize Port

Function antitype:

int CFInitCom(int nPort, long nBaud);

Function:

Open the communication port

Input parameter:

nPort - waiting the opening port No., like’1’,the max port No. is “9”

nBaud - Communication baud rate，baud rate is “2400、4800、9600、19200、38400、57600、115200”.

Output parameter:

Zero if successful; Nonzero if Requst fails,

4.Uninitialize Port

Function antitype:

int CFCloseCom();

Function:

Close the communication port

Input parameter:

Null

Output parameter:

Zero if successful; Nonzero if Requst fails,

5.Set Device Identity

Function antitype:

int CFSetDeviceNumber(WORD wDevID);

Function:

Setting reader's equipment ID

Input parameter:

wDevID - Equipment ID, like"0x0101"

Output parameter:

Zero if successful; Nonzero if Requst fails,

6.Get Device Identity

Function antitype:

int CFGetDeviceNumber(WORD *pDevID);

Function:

Setting reader's equipment ID

Input parameter:

pDevID - Equipment ID, E.g. when the got ID No."0x0101", formatted character string is "0101",

Output parameter:

Zero if successful; Nonzero if Requst fails,

7.Get Hardware Version.

Function antitype:

int CFGetHardwareVersion (WORD , WORD *pVersion);

Function:

Read the reader's hardware edition No.

Input parameter:

wDevID - equipment ID

Output parameter:

pVersion - hardware edition No., E.g. the edition No. is "0x0101",the formatted character string is "0101",

Zero if successful; Nonzero if Requst fails,

8.Get Serial No.

Function antitype:

int CFGetReaderNo(WORD wDevID, BYTE *pSnr);

Function:

Read the reader's Version.

Input parameter:

wDevID - Equipment ID

Output parameter:

pSnr - 8byte Serial No(hex)E.g."0x01、0x02…" Formatted character string is "0102".

Zero if successful; Nonzero if Requst fails,

9.Beep

Function antitype:

int CFBeep(WORD wDevID, BYTE bMs);

Function:

The CFBeep function generates simple tones.

Input parameter:

wDevID - Equipment ID

bMs - buzzer time limit, unit is 10MSEL

Output parameter:

Zero if successful; Nonzero if Requst fails,

due to there is no buzzer in this equipment. So when using this function there is no sound.

10.Set SAM Card baud rate

Function antitype:

int CFSetSamBaud(WORD wDevID, BYTE bBaud);

Function:

The function.is called to set SAM buadrate;

Input parameter:

wDevID - Equipment ID

bBaud - SAM card baud rate, numerical value 0x01：9600
0x02：38400

Output parameter:

Zero if successful; Nonzero if Requst fails,

11.Reset SAM Card

Function antitype:

int CFSamReset(WORD wDevID, BYTE *pData, BYTE* pLength);

Function:

The function.is called to reset SAM buadrate.

Input parameter:

wDevID - Equipment ID

Output parameter:

pData - Returned and reposition SAM data, E.g."0xA1,0xA2…", so the formatted data is "A1A2…"

pLength - the length of the returning data;

Zero if successful; Nonzero if Requst fails,

12.SAMCOS Instrucion

Function antitype:

int CFSamCos(WORD wDevID, BYTE *pCommand, BYTE bCmdLen, BYTE *pData, BYTE * pLength);

Function:

The function.is called to Send COS order to SAM card via reader

Input parameter:

wDevID - Equipment ID

pCommand - data of COS instruction

bCmdLen - length of COS instruction

Output parameter:

pData - Return the data of the SAM answering,E.g."0xA1,0xA2…",and the formatted data is "A1A2…"

pLength - Length of returning data

Zero if successful; Nonzero if Requst fails,

13.Set Reader's Mode

Function antitype:

int CFSetWorkMode(WORD wDevID, BYTE bType);

Function:

Setting reader untouching working format:ISO14443-A or ISO14443-B。

Input parameter:

wDevID - Equipment ID

bType - 0x00:TYPE_A mode
0x01':TYPE_B mode
0x02:AT88RF020 mode
0x04:ISO15693 mode

Output parameter:

Zero if successful; Nonzero if Requst fails,

14.Setting the Antenna Status

Function antitype:

int CFSetAntennaStatus(WORD wDevID, BYTE bMode);

Function:

Setting the working format of the reader's antenna

Output parameter:

wDevID - Reader ID

bMode - 0x00:Close Antenna
0x01:Open Antenna

Output parameter:

Zero if successful; Nonzero if Requst fails,

15.Mifare Request Card

Function antitype:

int CFISO14443_3ARequest(WORD wDevID, BYTE bMode, BYTE *pTagType, BYTE *pLength);

Function:

Request the Mifare card in the reading range

Input parameter:

wDevID - Equipment ID

bMode - 0x26: Seek the card not in the dormancy state
0x52: Seek cards in all different states

Output parameter:

pTagType - return the type of sought card

pLength - Length of the returned data

Zero if successful; Nonzero if Requst fails,

16.Mifare Anti-collision

Function antitype:

int CFISO14443_3AAnticoll(WORD wDevID, BYTE *pSnr, BYTE* pLength);

Function:

Anti-collision

Input parameter:

wDevID - Equipment ID

Output parameter:

pSnr - Return the card batch No.

pLength - Return the length of the data

Zero if successful; Nonzero if Requst fails,

17.Mifare Select Card

Function antitype:

int CFISO14443_3ASelect(WORD wDevID, BYTE *pSnr, BYTE bLen, BYTE *pLength);

Function:

Select a Mifare card

Input parameter:

wDevID - Equipment ID

pSnr - card batch No.

bLen - the length of the card batch No.

Output parameter:

pLength - Return 1byte card capacity

Zero if successful; Nonzero if Requst fails,

18.Mifare Halt

Function antitype:

int CFISO14443_3AHalt(WORD wDevID);

Function:

Making the enabled Mifare card to be dormancy state

Input parameter:

wDevID - Equipment ID

Output parameter:

Zero if successful; Nonzero if Requst fails,

19.Mifare Authenticate

Function antitype:

int CFISO14443_3AAuthentication2(WORD wDevID, BYTE bMode, BYTE bBlock, BYTE *pKey);

Function:

Authenticate a certain block of the Mifare card

Input parameter:

wDevID - Equipment ID

bMode - Secret key attribute, 0x60 ='A', 0x61 = 'B'

bBlock - Exclusive block No.
pKey - 6 byte secret key

Output parameter:

Zero if successful; Nonzero if Requst fails,

20.Mifare Read Block

Function antitype:

int CFISO14443_3ARead(WORD wDevID, BYTE bBlock, BYTE *pData);

Function:

Read a certain block of the Mifare card

Input parameter:

wDevID - Equipment ID
bBlock - Exclusive block No.

Output parameter:

pData - Return 16 byte data

Zero if successful; Nonzero if Requst fails,

21.Mifare Write Block

Function antitype:

int CFISO14443_3AWrite(WORD wDevID, BYTE bBlock, BYTE *pData);

Function:

Write to a certain block of the Mifare card

Input parameter:

wDevID - Equipment ID
bBlock - Exclusive block No.
pData - 16 byte data

Output parameter

Zero if successful; Nonzero if Requst fails,

22.Mifare Initializate Purse

Function antitype:

int CFISO14443_3APurseInit(WORD wDevID, BYTE bBlock, long lValue);

Function:

Initialize a certain block of the Mifare card to purse structure

Input parameter:

wDevID - Equipment ID
bBlock - Exclusive block No.
lValue - Initialized the value of the purse

Output parameter:

Zero if successful; Nonzero if Requst fails,

23.Read Value

Function antitype:

int CFISO14443_3AReadVal(WORD wDevID, BYTE bBlock, long *plValue);

Function:

Read Mifare purse

Input parameter:

wDevID - Equipment ID
bBlock - Exclusive block No.

Output parameter:

plValue - purse value

Zero if successful; Nonzero if Requst fails,

24.Mifare Decrease

Function antitype:

int CFISO14443_3ADecrement(WORD wDevID, BYTE bBlock, long lValue);

Function:

Input parameter:

wDevID - Equipment ID
bBlock - Exclusive block No.
lValue - The value the purse has

Output parameter:

Zero if successful; Nonzero if Requst fails,

25.Mifare Increment

Function antitype:

int CFISO14443_3AIncrement(WORD wDevID, BYTE bBlock, long lValue);

Function:

Operation to increment to the Mifare purse

Input parameter:

wDevID - Equipment ID
bBlock - Exclusive block No.
lValue - The amount wants increment

Output parameter:

Zero if successful; Nonzero if Requst fails,

26.Mifare Restore

Function antitype:

int CFISO14443_3ARestore(WORD wDevID, BYTE bBlock);

Function:

Zero if successful; Nonzero if Requst fails,

Input parameter:

wDevID - Equipment ID
bBlock - Exclusive block No.

Out put parameter:

Zero if successful; Nonzero if Requst fails,

27.Mifare Transfer

Function antitype:

int CFISO14443_3ATransfer(WORD wDevID, BYTE bBlock);

Function:

Transmit the data which is restored by CFISO14443_3ARestore fuction to the special block

Input parameter:

wDevID - Equipment ID

bBlock - Exclusive block No.

Output parameter:

Zero if successful; Nonzero if Requst fails,

28.UltraLight Select Card

Function antitype:

int CFULSelect(WORD wDevID, BYTE *pSnr);

Function:

Select certain Utri Light card

Input parameter:

wDevID - Equipment ID

Output parameter:

pSnr - Return 7 byte card Serial No.

Zero if successful; Nonzero if Requst fails,

29.UltraLight Write Card

Function antitype:

int CFULWrite(WORD wDevID, BYTE bPage, BYTE *pData, BYTE *pRev);

Function:

Write 16 bytes into the first blocks of certain section

Input parameter:

wDevID - Equipment ID

bPage - the starting section will be wrote

pData - the 16byte data

Output parameter:

pRev - Return 7byte card Serial No.

Zero if successful; Nonzero if Requst fails,

30.Reset CPU Card of the Mifare

Function parameter:

int CFTypeAReset(WORD wDevID, BYTE bMode, BYTE *pData, BYTE *pLength);

Function:

Reposition the CPU card which complies with ISO14443-A

Input parameter:

wDevID - Equipment ID

bMode - seek card method 0x52＝std,0x26=WUPa

Output parameter:

pData - Return the data after reposition

pLength - the length of the data after reposition

Zero if successful; Nonzero if Requst fails,

31.COS Instruction to CPU card of Mifare

Function antitype:

int CFCosCommand(WORD wDevID, BYTE*pCommand, BYTE bCLen,BYTE *pData, BYTE *pLength);

Function:

Send COS order to CPU card which complies with ISO14443-A

Input parameter:

wDevID - Equipment ID
pCommand - the order
bCLen - order length

Output parameter:

pData - returned data
pLength - length of returned data

Zero if successful; Nonzero if Requst fails,

32.TYPE-B Card Requst

Function antitype:

int CFSearchISO14443_3B(WORD wDevID, BYTE bMode, BYTE *pData);

Function:

Requst the IC card which complies with ISO14443-B

Input parameter:

wDevID - Equipment ID
bMode - Seek card method 0＝REQB,1=WUPB

Output parameter:

pData - Return 12byte data

Zero if successful; Nonzero if Requst fails,

33.15693_Inventory

Function parameter:

int CFISO15693_Inventory(WORD wDevID, BYTE *pData, BYTE *pLength);

Function:

Input parameter:

wDevID - Equipment ID

Output parameter:

pData - Return 9byte data, 1byte DSFID and 8byteUID
pLength - the length of the returned data

Zero if successful; Nonzero if Requst fails,

34.15693 Stay Quiet

Function parameter:

int CFISO15693_Stay_Quiet(WORD wDevID, BYTE *pUID);

Function:

Input parameter:

wDevID - Equipment ID
pUID - 8byte UID

Out put parameter:

Zero if successful; Nonzero if Requst fails,

35.15693 Select Card

Function parameter:

int CFISO15693_Select(WORD wDevID, BYTE *pUID);

Function:

Select a card complies with 15693

Input parameter:

wDevID - Equipment ID

pUID - 8byte UID

Output parameter:

Zero if successful; Nonzero if Requst fails,

36.15693 Reset Card

Function antitype:

int CFISO15693_ResetToReady(WORD wDevID, BYTE bMode, BYTE *pUID);

Function:

Input parameter:

wDevID - Equipment ID

bMode - bit0=Select_flags,bit1=Addres_flags

pUID - 8byte UID

Output parameter:

Zero if successful; Nonzero if Requst fails,

37.15693 Read Data

Function parameter:

int CFISO15693Read(WORD wDevID,BYTE bMode, BYTE *pUID,BYTE bBlock, BYTE bNumber, BYTE *pData, BYTE *pLength);

Function:

Read the pointed line of the pointed block in 15693 card

Input parameter:

wDevID - Equipment ID

bMode - bit0=Select_flags,bit1=Addres_flags

pUID - 8byte UID

bBlock - block No.

Output parameter:

pData - returned data after read

pLength - return the length of the read data

Zero if successful; Nonzero if Requst fails,

38.15693 Write Data

Function antitype:

int CFISO15693Write(WORD wDevID, BYTE bMode, BYTE *pUID, BYTE bBlock, BYTE *pData);

Function:

Write 4byte data into the 15693 card

Input parameter:

wDevID - Equipment ID
bMode - bit0=Select_flags,bit1=Addres_flags
pUID - 8byte UID
bBlock - Block No.
pData - 4byte data waiting be wrote

Output parameter:

Zero if successful; Nonzero if Requst fails,

39.15693 Lock certain Block

Function antitype:

int CFISO15693LockBlock(WORD wDevID, BYTE bMode, BYTE *pUID, BYTE bBlock);

Function:

Lock certain block data in 15693 card

Input parameter:

wDevID - Equipment ID
bMode - bit0=Select_flags,bit1=Addres_flags
pUID - 8byte UID
bBlock - Block No.

Output parameter:

Zero if successful; Nonzero if Requst fails,

40.15693 Write AFI

Function parameter:

int CFISO15693WriteAFI(WORD wDevID, BYTE bMode, BYTE *pUID, BYTE bAFI);

Function:

Read-in AFI to 15693 card

Input parameter:

wDevID - Equipment ID
bMode - bit0=Select_flags,bit1=Addres_flags
pUID - 8byte UID
bAFI - AFI data

Output parameter:

Zero if successful; Nonzero if Requst fails,

41.15693 Lock AFI

Function antitype:

int CFISO15693LockAFI(WORD wDevID, BYTE bMode, BYTE *pUID);

Function:

Lock AFI in 15693 card

Input parameter:

wDevID - Equipment ID
bMode - bit0=Select_flags,bit1=Addres_flags
pUID - 8byte UID

Output parameter:

Zero if successful; Nonzero if Requst fails,

42.15693 Write DSFID

Function antitype:

int CFISO15693WriteDSFID(WORD wDevID, BYTE bMode, BYTE *pUID, BYTE bDSFID);

Function:

Read-in DSFID of 15693

Input parameter:

wDevID - Equipment ID
bMode - bit0=Select_flags,bit1=Addres_flags
pUID - 8byte UID
bDSFID - 1byte DSFID

Output parameter:

Zero if successful; Nonzero if Requst fails,

43.15693 Lock DSFID

Function antitype:

int CFISO15693LockDSFID(WORD wDevID, BYTE bMode, BYTE *pUID);

Function:

Lock up the DSFID in 15693 card

Input parameter:

wDevID - Equipment ID
bMode - bit0=Select_flags,bit1=Addres_flags
pUID - 8byte UID

Output parameter:

Zero if successful; Nonzero if Requst fails,

44.15693 Get System Information

Function antitype:

int CFISO15693GetSystemInformation(WORD wDevID, BYTE bMode,BYTE *pUID, BYTE *pDSFID, BYTE *pAFI);

Function:

Read 15693 card

Input parameter:

wDevID - Equipment ID
bMode - bit0=Select_flags,bit1=Addres_flags

Output parameter:

pUID - 8byte UID
pDSFID - 1byte
pAFI - 1byte

Zero if successful; Nonzero if Requst fails,

45.15693 Read certain Block

Function antitype:

int CFISO15693GetBlockSecurity(WORD wDevID, BYTE bMode, BYTE *pUID, BYTE bBlock, BYTE bNumber, BYTE *pData, BYTE *pLength);

Function:

Read the data in certain block of 15693 card

Input parameter:

wDevID - Equipment ID
bMode - bit0=Select_flags,bit1=Addres_flags
pUID - 8byte UID
bBlock - pointed starting block
bNumber - the quantity of pointed blocks want to read

Output parameter:

pData - read data
pLength - length of the data

Zero if successful; Nonzero if Requst fails,

2) Error Code

Macroinstruction definition	Error code	Meaning
ERROR_WRITE_COM	1	Write failure
ERROR_WRITE_COM_FAILED	2	Write error
ERROR_READ_COM	3	Read failure
ERROR_CHECK	4	Return data inspected failure
ERROR_PARAM	5	Introduction interface parameter error