WBE RF Card Reader/Writer

API

Ver 1.10

SHENZHEN WORLD BRIDGE ELECTRONICS CO., LTD.

web: www.wbe.com.cn

Table of Contents

1 SUMMARIZE

MF_API.DLL is 32 bit DLL document which base on Windows, our client can use it for second development; it can be used as standard .DLL of Windows.

1.1 API description

MF_API.h	API function prototype definition
MF_API.dll	API DLL of function
EasyD12.dll	USB Dynamic assistant link

1.2 Applicable operating system

Windows NT: NT 3.1 or above

Windows: Windows 98, Windows 2000 or above

1.3 Applicable language

Visual C++ 5.0 or above

Visual Basic 5.0 or above

Visual C++Builder 5.0 or above

DELPHI 3.0 or above

PowerBuilder 6.0 or above

Other Windows 32bit developing tool

2 API FUNCTION

2.1 Device universal function

2.1.1 int _stdcall MF_GetDLL_Ver(char *rVER)

Function: Get DLL version number

Parameter: rVER: return buffer of version number, end as “\0”, less or equal to 32 byte

Return: success returns 0, if not, please consult error code table

2.1.2 int _stdcall MF_InitComm(char *portname, unsigned long baud)

Function: Initialize communication port

Parameter: portname: serial port number, the name of communication port, COM1~COM8 or USB

Baud: Baud Rate, 9600~115200bps

Return: success returns 0, if not, please consult error code table

Example:

Status = MF_ InitComm(“COM”1, 115200); // Initialize COM1, Baud Rate is 115200bps

Status = MF_InitComm(“USB”, 0); // Initialize USB

2.1.3 int _stdcall MF_ExitComm()

Function: Close the communication port

Return: success returns 0, if not, please consult error code table

2.1.4 int _stdcall MF_GetDevice_Ver(int DeviceAddr, char *ver)

Function: Get device version

Parameter: DeviceAddr: device address, 0 is valid to all address

Ver: return buffer of device version, end as “\0”, less or equal to 32 byte

Return: success returns 0, if not, please consult error code table

2.1.5 int _stdcall MF_ControlLED(int DeviceAddr, unsigned char LED1, unsigned char LED2)

Function: Control LED

Parameter: DeviceAddr: device address, 0 is valid to all address

LED 1: control LED 1, 0 turn off, 1 turn on

LED 2: control LED 2, 0 turn off, 1 turn on

Return: success returns 0, if not, please consult error code table

Example:

Status = MF_Control_LED(0,0,1); // LED1 turn off, LED2 turn on

2.1.6 int _stdcall MF_ControlBuzzer(int DeviceAddr, unsigned char BeepTime)

Function: Control buzzer

Parameter: DeviceAddr: device address, 0 is valid to all address

BeepTime: the beeping time, unit is 10ms

Return: success returns 0, if not, please consult error code table

Example:

Status = IFD_Control_Buzzer (0,30); // buzzer beeping time is 30*10ms = 300ms

2.1.7 int _stdcall MF_SetDeviceBaud(int DeviceAddr, unsigned char baud)

Function: Aet device Baud Rate

Parameter: DeviceAddr: device address, 0 is valid to all address

Baud: Baud Rate; 0 -> 9600, 1 -> 19200, 2 -> 38400, 3 -> 57600, 4 -> 115200

Return: success returns 0, if not, please consult error code table

Note:

The setting will be only valid after reset or re-power on.

Data can be kept when power off

2.1.8 int _stdcall MF_SetDeviceAddr(int DeviceAddr, unsigned char addr)

Function: Set device address:

Parameter: DeviceAddr: device address, 0 is valid to all address

Addr: new device address, 0~254

Return: success returns 0, if not, please consult error code table

2.1.9 int _stdcall MF_GetDeviceAddr(int DeviceAddr, unsigned char *addr)

Function: get device address

Parameter: DeviceAddr: device address, 0 is valid to all address

Add: return buffer of device address, 1 byte

Return: success return 0, if not, please consult error code table

Example:

Unsigned char addr;

Status = MF_GetDeviceAddr(0,&addr); // get device version

2.1.10 int _stdcall MF_SetDeviceSNR(int DeviceAddr, unsigned char *SNR)

Function: Set device serial number

Parameter: DeviceAddr: device address, 0 is valid to all address

SNR: return buffer of device serial number, 8 byte

Return: success returns 0, if not, please consult error code table.

2.1.11 int _stdcall MF_GetDeviceSNR(int DeviceAddr, unsigned char *SNR)

Function: Get device serial number

Parameter: DeviceAddr: device address, 0 is valid to all address

SNR: return buffer of device serial number, 8 byte

Return: success returns 0, if not, please consult error code table

2.1.12 int _stdcall MF_SetRF_ON(int DeviceAddr)

Function: set RF on

Parameter: DeviceAddr: device address, 0 is available to all address

Return: success returns 0, if not, please consult error code table

Note: RF will be open automatically after power on

2.1.13 int _stdcall MF_SetRF_OFF(int DeviceAddr)

Function: Set RF off

Parameter: DeviceAddr: device address, 0 is available to all address

Return: success returns 0, if not, please consult error code table

2.1.14 int _stdcall MF_DeviceReset(int DeviceAddr)

Function: Device reset by software

Parameter: DeviceAddr: device address, 0 is available to all address

Return: success returns 0, if not, please consult error code table

2.1.15 int _stdcall MF_SetWiegandMode(int DeviceAddr, unsigned char mode, unsigned char alarm)

Function: Set wiegand mode

Parameter: DeviceAddr: device address, 0 is available to all address

Mode: 0: turn off wiegand mode, 1: turn on wiegand mode

Alarm: wiegand alarm control byte.

Bit1=0 turn on swipe card alarm, Bit1=1 turn off

Bit0=0 turn on external control alarm, Bit0=1 turn off

Return: success returns 0, if not, please consult error code table

2.2 Card operate function

2.2.1 int _stdcall MF_Request(int DeviceAddr, unsigned char mode, unsigned char *CardType)

Function: Request card

Parameter: DeviceAddr: device address, 0 is available to all address

Mode: 0 = free mode, 1 = active mode, request all card

CardType: return buffer of card type, 2 byte

Return: success returns 0, if not, please consult error code table

2.2.2 int _stdcall MF_Anticoll(int DeviceAddr, unsigned char *snr)

Function: Anticoll

Parameter: DeviceAddr: device address, 0 is valid to all address

snr: return buffer of serial number, 4 byte

Return: success returns 0, if not, please consult error code table

2.2.3 int _stdcall MF_Select(int DeviceAddr, unsigned char *snr)

Function: Select card

Parameter: DeviceAddr: device address, 0 is valid to all address

snr: return buffer of serial number, 4 byte

Return: success returns 0, if not, please consult error code table

2.2.4 int _stdcall MF_Halt(int DeviceAddr)

Function: Halt card

Parameter: DeviceAddr: device address, 0 is valid to all address

Return: success returns 0, if not, please consult error code table

2.2.5 int _stdcall MF_LoadKey(int DeviceAddr, unsigned char *key)

Function: Load key from host

Parameter: DeviceAddr: device address, 0 is valid to all address

Key: buffer of key storage, 6 byte

Return: success returns 0, if not, please consult error code table

2.2.6 int _stdcall MF_LoadKeyFromEE(int DeviceAddr, unsigned char KeyType, unsigned char KeyNum)

Function: Load key from EEPROM of reader

Parameter: DeviceAddr: device address, 0 is available to all address

KeyType: Key type. 0 = key A, 1 = key B

KeyNum: Key number

Return: success returns 0, if not, please consult error code table

2.2.7 int _stdcall MF_Authentication(int DeviceAddr, unsigned char AuthType, unsigned char block, unsigned char *snr)

Function: Authentication

Parameter: DeviceAddr: device address, 0 is available to all address

AuthType: Authenticate the key type. 0 = key A, 1 = key B

snr: buffer of serial number storage, 4 byte

Return: success returns 0, if not, please consult error code table

2.2.8 int _stdcall MF_Read(int DeviceAddr, unsigned char block, unsigned char numbers, unsigned char *databuff)

Function: Read card

Parameter: DeviceAddr: device address, 0 is available to all address

block: Block number

numbers: Quantity of blocks

snr: return buffer of card data, (16*numbers) byte

Return: success returns 0, if not, please consult error code table

2.2.9 int _stdcall MF_Write(int DeviceAddr, unsigned char block, unsigned char numbers, unsigned char *databuff)

Function: Write card

Parameter: DeviceAddr: device address, 0 is available to all address

block: Block number

numbers: Quantity of blocks

snr: return buffer of card data, (16*numbers) byte

Return: success returns 0, if not, please consult error code table

2.2.10 int _stdcall MF_Value(int DeviceAddr, unsigned char ValOption, unsigned char block, unsigned char *value)

Function: Value operation

Parameter: DeviceAddr: device address, 0 is available to all address

Valopton: Value operation type. 0 = Decrease value (Dec); 1 = Increase value (Inc); 2 = Restore (Restore)

block: Block number

value: buffer of storage data, 4 byte

Return: success returns 0, if not, please consult error code table

2.2.11 int _stdcall MF_Transfer(int DeviceAddr, unsigned char block)

Function: Write Value

Parameter: DeviceAddr: device address, 0 is available to all address

block: Block number

Return: success returns 0, if not, please consult error code table

2.2.12 int _stdcall MF_StoreKeyToEE(int DeviceAddr, unsigned char KeyAB, unsigned char KeyAddr, unsigned char *key)

Function: Store key to EEPROM of reader

Parameter: DeviceAddr: device address, 0 is available to all address

KeyAB: Key type, 0 = key A, 1 = key B

KeyAddr: Key address

Key: buffer of key storage

Return: success returns 0, if not, please consult error code table

2.3 ISO14443A/B & ISO15693 common function

2.3.1 int _stdcall MF_General_ISO14443A(int DeviceAddr, unsigned char enable_crc, unsigned char *cData, unsigned char cLen, unsigned char *rData, unsigned char *rLen)

Function: ISO14443A common command

Parameter: DeviceAddr: device address, 0 is valid to all address

enable_crc: permit or prohibit CRC. 0 = prohibit, 1= permit

cData: common command data (Reader to VICC)

cLen: number of cData

rData: data of card return(VICC to Reader)

rLen: length of rData

Return: success returns 0, if not, please consult error code table

2.3.2 int _stdcall MF_General_ISO14443B(int DeviceAddr, unsigned char *cData, unsigned char cLen, unsigned char *rData, unsigned char *rLen)

Function: ISO14443B common command

Parameter: DeviceAddr: device address, 0 is valid to all address

cData: common command data (Reader to VICC)

cLen: number of cData

rData: data of card return (VICC to Reader)

rLen: length of rData

Return: success returns 0, if not, please consult error code table

2.3.3 int _stdcall MF_General_ISO15693(int DeviceAddr, unsigned char *cData, unsigned char cLen, unsigned char *rData, unsigned char *rLen)

Function: ISO15693 common command

Parameter: DeviceAddr: device address, 0 is valid to all address

cData: common command data (Reader to VICC)

cLen: number of cData

rData: data of card return (VICC to Reader)

rLen: length of rData

Return: success returns 0, if not, please consult error code table

2.3.4 int __stdcall MF_ISO15693_Inventory(int DeviceAddr,unsigned char flags, unsigned char AFI, unsigned char masklengh ,unsigned char *maskvalue, unsigned char *databuffer)

Function: ISO15693 inventory command

Parameter: DeviceAddr: device address, 0 is valid to all address

Return: success returns 0, if not, please consult error code table

3 RETURN STATUS /ERROR CODE

```
#define MI_OK                   0x00    // normal

#define MI_NOTAGERR             0x01
#define MI_COLLERR              0x02
#define MI_BITERR               0x03
#define MI_SAKERR               0x04
#define MI_AUTHERR              0x05
#define MI_Value                0x0D
#define MI_ACCESSERR            0x0E
#define MI_ACCESSTIMEOUT        0x0F

#define MI_CommandErr           0x10
#define MI_OtherErr             0x11

#define Code_RetFrameErr        0x20
#define Code_TimeoutErr             0x21
#define Code_SetCommPortErr     0x22
```