

# RRMifare32.DLL Dynamic Link Library

## User's Manual

RRMifare32.DLL is a dynamic link library designed to facilitate Mifare I card application software development when using Mifare card RHMMF1RW reader/writer.

The following is the detail description of RRMifare32.DLL functions.

### 1) Operation System Requirement

WINDOWS 2000/XP

### 2) Function List

RRMifare32.DLL includes the following functions for the Mifare card operation.

#### 2.1) General Function

```
extern "C" __declspec(dllexport) long WINAPI AutoOpenComPort(long *Port);
extern "C" __declspec(dllexport) long WINAPI OpenComPort(long Port);
extern "C" __declspec(dllexport) long WINAPI CloseComPort(void);
extern "C" __declspec(dllexport) long WINAPI BUZZER(long Device , long OnTime , long BetweenTime , long Times);
extern "C" __declspec(dllexport) long WINAPI ControlLed(long Setlight);
extern "C" __declspec(dllexport) long WINAPI GetInfo(unsigned char *Info);
extern "C" __declspec(dllexport) long WINAPI OpenRf(void);
extern "C" __declspec(dllexport) long WINAPI CloseRf(void);
```

#### 2.2) Basic Card Operation Function

```
extern "C" __declspec(dllexport) long WINAPI RRMifare_LoadKey(long Mode , long KeySectorNo , unsigned char *Key);
extern "C" __declspec(dllexport) long WINAPI RRMifare_Request(long Mode , long *CardTypeNo);
extern "C" __declspec(dllexport) long WINAPI RRMifare_AntiColl(long *CardSn);
extern "C" __declspec(dllexport) long WINAPI RRMifare_Select(long CardSn, long *retsize);
extern "C" __declspec(dllexport) long WINAPI RRMifare_Authentication(long Mode , long Sector , long KeySectorNo);
extern "C" __declspec(dllexport) long WINAPI RRMifare_Read(long BlockNo , unsigned char *ReadBuff);
extern "C" __declspec(dllexport) long WINAPI RRMifare_Write(long BlockNo , unsigned char *WriteBuff);
extern "C" __declspec(dllexport) long WINAPI RRMifare_Increment(long BlockNo , long Value);
extern "C" __declspec(dllexport) long WINAPI RRMifare_Decrment(long BlockNo , long Value);
extern "C" __declspec(dllexport) long WINAPI RRMifare_Halt(void);
extern "C" __declspec(dllexport) long WINAPI RRMifare_DirectAuthentication(long Mode , long Sector , unsigned char *Keys);
extern "C" __declspec(dllexport) long WINAPI RRMifare_SetKey(long Sector , unsigned char *Keya , unsigned char *Keyb);
```

### 3) Function Explanation

#### 3.1) General Function

##### 3.1.1) AutoOpenComPort(long \*Port): Auto-detect and connect with the reader

**Function description:**

This function is used to detect the Mifare reader attached automatically and establish the connection with it through the correct communication port. The communication protocol is 9600bps, 8 data bits, 1 start bit, 1 stop bit, no parity bit.

**Usage:**

```
extern "C" __declspec(dllexport) long WINAPI AutoOpenComPort(long * Port);
```

**Parameter:**

Port: Pointed to the communication port number that the reader is detected and connected. Port number constants (COM1~COM10) are defined in the head file RRMifare32.h.

**Returns:**

Zero value when successful, non-zero value when error occurred.

**Examples:** retcode=AutoOpenComPort(port);

##### 3.1.2) OpenComPort(long Port): Connect reader with specified communication port

**Function description:**

This function is used to establish the connection with the Mifare reader through a specified communication port. The communication protocol is 9600bps, 8 data bits, 1 start bit, 1 stop bit, no parity bit.

**Usage:**

```
extern "C" __declspec(dllexport) long WINAPI OpenComPort(long Port);
```

**Parameter:** Port: Communication port number which is a constant from COM1 to COM10 defined in the head file RRMifare32.h

**Returns:**

Zero value when successful, non-zero value when error occurred.

**Examples:** retcode=OpenComPort(COM1);

##### 3.1.3) CloseComPort(void): Disconnect and release communication port resources

**Function description:**

This function is used to disconnect the reader and release the corresponding communication port resources. In some development environment, the communication port resources must be released before exiting. Otherwise the operation system will become unstable.

**Usage:**

```
extern "C" __declspec(dllexport) long WINAPI CloseComPort(void);
```

**Parameter:** None.

**Returns:**

Zero value when successful, non-zero value when error occurred.

**Examples:** retcode=CloseComPort();

### **3.1.4) BUZZER(long Device, long OnTime, long BetweenTime, long Times): Control the buzzer**

#### **Function description:**

This function is used to control the status of the reader's buzzer.

#### **Usage:**

```
extern "C" __declspec(dllexport) long WINAPI BUZZER(long Device , long OnTime , long BetweenTime , long Times);
```

#### **Parameter:**

Device: Reserved and should be set to 0.

OnTime: The duration that the buzzer is active and beeping. The time unit is 15ms.

Example: if OnTime=15, the buzzer will beep for 225ms.

BetweenTime: The duration that the buzzer is inactive and silent. The time unit is 15ms.

Times: the counts of the buzzer's being active.

#### **Returns:**

Zero value when successful, non-zero value when error occurred.

**Examples:** retcode=BUZZER(0 ,OnTime , BetweenTime , Times);

### **3.1.5) ControlLed(long Setlight) : Control the LED.**

#### **Function description:**

This function is used to control the LED.

#### **Usage:**

```
extern "C" __declspec(dllexport) long WINAPI ControlLed(long Setlight);
```

#### **Parameter:**

Setlight: 0 to turn on the LED, 1 to turn off the LED.

#### **Returns:**

Zero value when successful, non-zero value when error occurred.

**Example:** retcode= ControlLed (Setlight).

### **3.1.6 ) GetInfo(unsigned char \*Info): Get the reader's information**

#### **Function description:**

This function is used to get the reader's information such as its firmware version.

#### **Usage:**

```
extern "C" __declspec(dllexport) long WINAPI GetInfo(unsigned char *Info);
```

#### **Parameter:**

Info: pointed to the reader's description string. The length of the description string is 13bytes.

#### **Returns:**

Zero value when successful, non-zero value when error occurred.

**Example:** retcode= GetInfo(Info);

### **3.1.7 ) OpenRf(void): Turn on the RF transmission**

#### **Function description:**

This function is used to turn on the RF transmission to activate the inductive area.

**Usage:**

extern "C" \_\_declspec(dllexport) long WINAPI OpenRf(void);

**Parameter:** none.

**Returns:**

Zero value when successful, non-zero value when error occurred.

**Example:** retcode= OpenRf();

### 3.1.8 )CloseRf(void): Turn off the RF transmission

**Function description:**

This function is used to turn off the RF transmission to deactivate the inductive area.

**Usage:**

extern "C" \_\_declspec(dllexport) long WINAPI CloseRf(void);

**Parameter:** none.

**Returns:**

Zero value when successful, non-zero value when error occurred.

**Example:** retcode= CloseRf();

## 3.2) Basic Card Operation Function

### 3.2.1) RRMifare\_LoadKey(long Mode , long KeySectorNo , unsigned char \*Key):

Save the card's sector authentication key in the reader's EEPROM area.

**Function description:**

This function is used to save a card's sector authentication key in the reader's EEPROM area.

**Usage:**

extern "C" \_\_declspec(dllexport) long WINAPI RRMifare\_LoadKey(long Mode , long KeySectorNo , unsigned char \*Key);

**Parameter:** Mode: 0 for KEY A and 4 for KEY B.

KeySectorNo: the save location in the reader's EEPROM area.

Key: pointed to the 6 bytes key data.

**Returns:**

Zero value when successful, non-zero value when error occurred.

**Example:** retcode= RRMifare\_LoadKey(Mode, KeySectorNo, Key).

### 3.2.2) RRMifare\_Request(long Mode, long \*CardType): Request for a card

**Function description:**

This function is used to request that if any card is in the inductive area and return the card type if a card respond to the request.

**Usage:**

extern "C" \_\_declspec(dllexport) long WINAPI RRMifare\_Request(long Mode , long \*CardTypeNo);

**Parameter:**

Mode: request mode.

Mode=01: request for all card.

Mode=00: request for all card except the card in HALT state.

CardType: pointed to a long integer describing the card type.

0x0010 for Mifare Light 10.

0x0004 for Mifare I.

**Returns:**

Zero value when successful, non-zero value when error occurred.

**Example:** retcode= RRMifare\_Request(Mode,CardType)

### 3.2.3) RRMifare\_AntiColl(long \*CardSn ):Anti-collision

**Function description:**

This function is used to fulfill the anti-collision procedure and return the serial number.(UID) of one card in the inductive area.

**Usage:**

extern "C" \_\_declspec(dllexport) long WINAPI RRMifare\_AntiColl(long \*CardSn);

**Parameter:**

CardSn: pointed to the UID of a card in the inductive area

**Returns:**

Zero value when successful, non-zero value when error occurred.

**Example:** retcode= RRMifare\_Anticoll(CardSn )

### 3.2.4) RRMifare\_Select(long CardSn, long \*retsize): Select the specified card

**Function description:**

This function is used to select a specified card in the inductive area by its UID. The card will be in selected status if the operation succeeds.

**Usage:**

extern "C" \_\_declspec(dllexport) long WINAPI RRMifare\_Select(long CardSn,long \*retsize);

**Parameter:**

CardSn: UID of the card to be selected. The UID is get by Anticoll procedure. Refer to function 3.2.3.

Retsize: pointed to a long integer describing the size of the selected card's capacity.

**Returns:**

Zero value when successful, non-zero value when error occurred.

**Example:** retcode= RRMifare\_Select(CardSn, retsize);

### 3.2.5) RRMifare\_Authentication(long Mode, long Sector, long KeySectorNo) :

**Authentication procedure**

**Function description:**

This function is used to fulfill the mutual authentication process between the card and the reader using the KEY stored in the reader's EEPROM. The card's sector data can only be read or written after successful authentication.

**Usage:**

extern "C" \_\_declspec(dllexport) long WINAPI RRMifare\_Authentication(long Mode, long Sector, long KeySectorNo);

**Parameter:**

Mode: Authenticate with KEY A or KEY B.

Mode=00: authenticate with KEY A.

Mode=01: authenticate with KEY B.

Sector: the number of the sector of the card to be authenticated.

KeySectorNo: the location of the KEY in the reader's EEPROM.

**Returns:**

Zero value when successful, non-zero value when error occurred.

**Example:** retcode= RRMifare\_Authentication(Mode, Sector, KeySectorNo);

**3.2.6) RRMifare\_Read(long BlockNo, unsigned char \*ReadBuff):Read one block data****Function description:**

This function is used to read out one block data of the card. The block size is 16 bytes.

**Usage:**

```
extern "C" __declspec(dllexport) long WINAPI RRMifare_Read(long BlockNo,  
unsigned char * ReadBuff);
```

**Parameter:**

BlockNo: the absolute block number of the card. The number is 0~63 for Mifare I card.

ReadBuff: pointed to the block data having been read out.

**Returns:**

Zero value when successful, non-zero value when error occurred.

**Example:** retcode= RRMifare\_Read(BlockNo, ReadBuff);

**3.2.7) RRMifare\_Write(long BlockNo , unsigned char \*WriteBuff):Write one block data****Function description:**

This function is used to write one block data into the card. The block size is 16 bytes.

**Usage:**

```
extern "C" __declspec(dllexport) long WINAPI RRMifare_Write(long BlockNo,  
unsigned char *WriteBuff);
```

**Parameter:**

BlockNo: the absolute block number of the card. The number is 0~63 for Mifare I card.

WriteBuff: pointed to the block data to be written into the card

**Returns:**

Zero value when successful, non-zero value when error occurred.

**Example:** retcode= RRMifare\_Write(BlockNo, WriteBuff);

**3.2.8) RRMifare\_Increment(long BlockNo , long Value):Value-block increment****Function description:**

This function is used to increase the value of a value-block of the card

**Usage:**

```
extern "C" __declspec(dllexport) long WINAPI RRMifare_Increment(long BlockNo,  
long Value);
```

**Parameter:**

BlockNo: the absolute block number of the card. The number is 0~63 for Mifare I card.

This operation can only be applied to a value-block. Please refer to the datasheet of the Mifare card for the definition and format of the value-block.

Value: the 4 bytes value to be increased.

**Returns:**

Zero value when successful, non-zero value when error occurred.

**Example:** retcode= RRMifare\_Increment(BlockNo, Value);

### 3.2.9) RRMifare\_Decrment(long BlockNo, long Value):Value-block decrement

**Function description:**

This function is used to decrease the value of a value-block of the card

**Usage:**

```
extern "C" __declspec(dllexport) long WINAPI RRMifare_Decrment(long BlockNo,  
long Value);
```

**Parameter:**

BlockNo: the absolute block number of the card. The number is 0~63 for Mifare 1 card.

This operation can only be applied to a value-block. Please refer to the datasheet of the Mifare card for the definition and format of the value-block.

Value: the 4 bytes value to be decreased.

**Returns:**

Zero value when successful, non-zero value when error occurred.

**Example:** retcode= RRMifare\_Decrment(BlockNo, Value);

### 3.2.10) RRMifare\_Halt(void) : Halt the card.

**Function description:**

This function is used to set the current active card into halt status.

**Usage:**

```
extern "C" __declspec(dllexport) long WINAPI RRMifare_Halt(void);
```

**Parameter:** none.

**Returns:**

Zero value when successful, non-zero value when error occurred.

**Example:** retcode= RRMifare\_Halt();

### 3.2.11) RRMifare\_DirectAuthentication(long Mode, long Sector, unsigned char \*Keys):

**Direct authentication procedure**

**Function description:**

This function is used to fulfill the mutual authentication process between the card and the reader using the KEY directly input. The card's sector data can only be read or written after successful authentication.

**Usage:**

```
extern "C" __declspec(dllexport) long WINAPI RRMifare_Direct Authentication(long  
Mode, long Sector, unsigned char *Keys);
```

**Parameter:**

Mode: Authenticate with KEY A or KEY B.

Mode=00: -authenticate with KEY A.

Mode=01: authenticate with KEY B.

Sector: the number of the sector of the card to be authenticated.

Keys: pointed to the 6 bytes KEY.

**Returns:**

Zero value when successful, non-zero value when error occurred.

**Example:** retcode= RRMifare\_DirectAuthentication(Mode, Sector, Keys);

**3.2.12) RRMifare\_SetKey(long Sector, unsigned char \*Keya, unsigned char \*Keyb):Set the KEY of a sector of the card**

**Function description:**

This function is used to set the KEY of a sector of the card. The KEY A and KEY B of the sector are set at the same time.

**Usage:**

```
extern "C" __declspec(dllexport) long WINAPI RRMifare_SetKey(long Sector ,  
unsigned char *Keya, unsigned char *Keyb);
```

**Parameter:**

Sector: the sector number of which the KEY A and KEY B will be set.

Keya: pointed to the 6 bytes KEY A.

Keyb: pointed to the 6 bytes KEY B.

**Returns:**

Zero value when successful, non-zero value when error occurred.

**Example:** retcode= RRMifare\_SetKey(Sector, Keya, Keyb);