SHENZHEN EXCELSOO TECHNOLOGY CO., LTD

EX-RW10X Mifare 1 Reader

SDK Manual

V2.1

Excelsoo

2/3/2021

Note:

1. EX-RW10X series readers module follow the ISO/IEC14443 standard and based on 13.56MHz frequency Mifare;
2. IC Card in this manual means 13.56MHz Mifare Card.

This SDK Manual is used for EX-RW10X series readers. Commonly used functions will be list in the manual.

EX-RW10X SDK Manual V2.1

Functions List

/**********

Function 1：OpenCom

Note: *You must call this function first to open the port of the card reader before operating it.*
Parameter: com_num- port number (1-255), Incoming Parameter
Return: Call result, 0-open successfully, others-open failed
**********/

```
int __stdcall OpenCom(int com_num);
```

/**********

Function 2: CloseCom

Note: *You can call this function to release the occupied port when you no longer need to use the dynamic library*
Parameter: com_num- port number (1-255), Incoming Parameter
Return: Call result, 0-Close successfully, others-Close failed
**********/

```
int __stdcall CloseCom(int com_num);
```

/**********

Function 3: IsComValid

Note: *Used at the application layer to test whether the port can be used normally*
Parameter: com_num- port number (1-255), Incoming Parameter
Return: Call result, 0-unavailable or port does not exist, 1-port can be used normally
**********/

```
int __stdcall IsComValid(int com_num);
```

/**

Function 4: BuzCtrl

Note: *Used to control the buzzer on the reader*

Parameter: com_num- port number (1-255), Incoming Parameter

act- buzzer time (0-turn off the buzzer, other values (1-255)-buzzer time, in units of 10 milliseconds), Incoming Parameter

Return: Call result, 0-operation succeeded, others-operation failed

**/

```
int __stdcall BuzCtrl(int com_num, int act);
```

/**

Function 5: LedCtrl

Note: *Used to control the LED lights on the reader*

P*arameter:* com_num-port number (1-255), Incoming Parameter

act- Command (0-turn off the LED, 1-turn on the LED), Incoming Parameter

Return: Call result, 0-operation succeeded, others-operation failed

**/

```
int __stdcall LedCtrl(int com_num, int act);
```

/**

Function 6: ReadBlock

Note: *Used to read the data of one block (16 bytes) of the IC card*

Parameter: com_num-port number (1-255), Incoming Parameter

block- Block number (0-63), Incoming Parameter

*key- Sector key, 6-byte BYTE array (the first address or pointer of the pass in array), Incoming Parameter, cannot be empty

*card_id- Scanned card ID (4 bytes, Little-endian format), Outgoing Parameter, cannot be empty, the calling program must pass the address of an array with sufficient storage space

*pdat- The read block data (16 bytes), Outgoing Parameter, cannot be empty, the calling program must pass the address of the array with sufficient storage space

Return: Call result, 0-operation succeeded, others-operation failed

**/

```
int __stdcall ReadBlock(int com_num, BYTE block, BYTE *key, DWORD *card_id, BYTE *pdat);
```

/**

Function 7: WriteBlock

Note: *Used to write data of one block (16 bytes) of IC card*

Parameter: com_num-port number (1-255), Incoming Parameter

block- Block number (0-63), Incoming Parameter

*key- Sector key, 6-byte BYTE array (the first address or pointer of the pass in array), Incoming Parameter, cannot be empty

card_id- ID of the card to write data (4 bytes, Little-endian format), Incoming Parameter

*pdat- The block data to be written (16 bytes), Incoming Parameter, cannot be empty, the calling program must pass the address of the array with sufficient storage space

Return: Call result, 0-operation succeeded, others-operation failed

**********/

```
int __stdcall WriteBlock(int com_num, BYTE block, BYTE *key, DWORD card_id, BYTE *pdat);
```

/**********

Function 8: ReadSector

Note: *Used to read the data of one sector of the IC card (the first 3 blocks, a total of 48 bytes)*

Parameter: com_num-port number (1-255), Incoming Parameter

sector- Block number (0-15), Incoming Parameter

*key- Sector key, 6-byte BYTE array (the first address or pointer of the pass in array), Incoming Parameter, cannot be empty

*card_id- Scanned card ID (4 bytes, Little-endian format), Outgoing Parameter, cannot be empty, the calling program must pass the address of an array with sufficient storage space

*pdat-The read block data (48 bytes), Outgoing Parameter, cannot be empty, the calling program must pass the address of the array with sufficient storage space

Return: Call result, 0-operation succeeded, others-operation failed

**********/

```
int __stdcall ReadSector(int com_num, BYTE sector, BYTE *key, DWORD *card_id, BYTE *pdat);
```

/**********

Function 9: WriteSector

Note: *Used to write data into one sector of the IC card (the first 3 blocks, a total of 48 bytes)*

Parameter: com_num-port number (1-255), Incoming Parameter

sector- Block number (0-15), Incoming Parameter

*key- Sector key, 6-byte BYTE array (pass in the first address or pointer of the array), Incoming Parameter, cannot be empty

card_id- ID of the card to write data (4 bytes, little endian format)

*pdat- The block data to be written (16 bytes), Incoming Parameter, cannot be empty, the calling program must pass the address of the array with sufficient storage space

Return: Call result, 0-operation succeeded, others-operation failed

**********/

```
int __stdcall WriteSector(int com_num, BYTE sector, BYTE *key, DWORD card_id, BYTE *pdat);
```