

RFID_R6 API SPECIFICATION with ISO14443A/Mifare Classic

This document is about how to use CLRC663 RFID reader to work with ISO14443A/Mifare Classic protocol. ISO15693 protocol and China's Second Generation UID please refer to other two files. This document describe how to use all native func in java source file which is locate in the path of /src/com/geomobile/rc663/Mifare_native.java. Customs can use these func to develop application layer program, but not all types of card support all instructions. All native func will print debug message, if func run failed, please use eclipse or adb to check debug message, this will help you to fix problems.

1 FUNC PROTOTYPE: **int init_dev();**

FUNC DESCRIPTION: Init reader device.

PARAM DESCRIPTION: none

RETURN VALUE: Return -1 means failed; return 0 means init device ok.

2 FUNC PROTOTYPE: **void release_dev();**

FUNC DESCRIPTION: Close reader device.

PARAM DESCRIPTION: none

RETURN VALUE: none

3 FUNC PROTOTYPE: **byte[] search_card();**

FUNC DESCRIPTION: Search card, deal with collision then select card, finally return the SN of card which is selected.

PARAM DESCRIPTION: none。

RETURN VALUE: If search failed, will return **null**; if success, will return a byte array. It's content is the selected card's SN.

4 FUNC PROTOTYPE: **int halt_card();**

FUNC DESCRIPTION: Make the current selected card to halt status. It will not be searched again.

PARAM DESCRIPTION: none

RETURN VALUE: Return -1 means failed; return 0 means halt card ok.

5 FUNC PROTOTYPE: **int active_card(byte[] cid);**

FUNC DESCRIPTION: If a card was halted, it will not be search and selected again. Use this func will active the halted card which has the SN store in param cid.

PARAM DESCRIPTION:

PARAM **cid**: The SN of the card want to active.

RETURN VALUE: Return -1 means failed; return 0 means active card ok.

6 FUNC PROTOTYPE: **int store_key_eeprom(int place, byte[] key);**

FUNC DESCRIPTION: CLRC663 can store many **key pairs** in EEPROM. This means when we

need to authenticate a block we don't need to send clear text key from cpu to reader chip, just needed send the place which key pair already store in, make security improved. This func is use to store a **key pair** to the right place of EEPROM in CLRC663. A **key pair** has the size of 12 bytes. First 6 bytes are **key of type A**, other 6 bytes are the **key of type B**. Related concepts please refer to ISO/IEC 14443 protocol.

PARAM DESCRIPTION:

PARAM **place**: Where the **key pair** should store in. For CLRC663, it can store 85 **key pairs**, so the value should be from 0 to 84.

PARAM **key**: The **key pair** want to store.

RETURN VALUE: Return -1 means failed; return 0 means active card ok.

7 FUNC PROTOTYPE: **int auth_card_eeeprom(int type, byte[] cid, int block, int place);**

FUNC DESCRIPTION: Use pre-stored key to authenticate a block on current selected card.

PARAM DESCRIPTION:

PARAM **type**: The type of authenticate, can only set to **AUTH_TYPEA** or **AUTH_TYPEB**, which means to authenticate **Key A** or **Key B**.

PARAM **cid**: The SN of current selected card.

PARAM **block**: The block number want to authenticate.

PARAM **place**: The place number where the key store in.

RETURN VALUE: Return -1 means failed; return 0 means authenticate card ok.

8 FUNC PROTOTYPE: **int auth_card_key(int type, byte[] cid, int block, byte[] key);**

FUNC DESCRIPTION: Use clear text key to authenticate a block on current selected card.

PARAM DESCRIPTION:

PARAM **type**: The type of authenticate, can only set to **AUTH_TYPEA** or **AUTH_TYPEB**, which means to authenticate **Key A** or **Key B**.

PARAM **cid**: The SN of current selected card.

PARAM **block**: The block number want to authenticate.

PARAM **key**: The key use to authenticate.

RETURN VALUE: Return -1 means failed; return 0 means authenticate card ok.

9 FUNC PROTOTYPE: **byte[] read_block(int block);**

FUNC DESCRIPTION: Read a block's raw data. Before use this, need to authenticate the block with key which offer the read permission success.

PARAM DESCRIPTION:

PARAM **block**: The block number want to read.

RETURN VALUE: When failed, will return **null**; when success, return a byte array which contain the raw data of block. For Mifare Classic card, a block has the size of 16 bytes.

10 FUNC PROTOTYPE: **int[] read_block_value(int block);**

FUNC DESCRIPTION: Read the value which store in block. Before use this, need to authenticate the block with key which offer the read permission success.

PARAM DESCRIPTION:

PARAM **block**: The block number want to read.

RETURN VALUE: When failed, will return **null**; when success, return a int array. It has only one element which store the value read from block. For example, can use the code below to get value. **`int[] res = read_block_value(3); if(res != null) res_value = res[0];`**.

11 FUNC PROTOTYPE: `int write_block(int block, byte[] data);`

FUNC DESCRIPTION: Write raw data to block. Before use this, need to authenticate the block with key which offer the write permission success.

PARAM DESCRIPTION:

PARAM **block**: The block number want to write.

PARAM **data**: The data want to write. For Mifare Classic card, the size of block is 16 bytes. So data param should has the size of 16 bytes.

RETURN VALUE: Return -1 means failed; return 0 means write card ok.

12 FUNC PROTOTYPE: `int write_block_value(int block, int value);`

FUNC DESCRIPTION: Write a value to block. Before use this, need to authenticate the block with key which offer the write permission success.

PARAM DESCRIPTION:

PARAM **block**: The block number want to write.

PARAM **value**: The value want to write.

RETURN VALUE: Return -1 means failed; return 0 means write card ok.

13 FUNC PROTOTYPE: `inc_value(int srcBlock, int destBlock, int value);`

FUNC DESCRIPTION: Read a block's value, increment it, then write to another block. The source block and the destination block can be one block. Before use this, need to authenticate the block with key which offer the right permission success.

PARAM DESCRIPTION:

PARAM **srcBlock**: The block number of source.

PARAM **destBlock**: The block number of destination.

PARAM **value**: the value to increment.

RETURN VALUE: Return -1 means failed; return 0 means write card ok.

14 FUNC PROTOTYPE: `dec_value(int srcBlock, int destBlock, int value);`

FUNC DESCRIPTION: Read a block's value, decrement it, then write to another block. The source block and the destination block can be one block. Before use this, need to authenticate the block with key which offer the right permission success.

PARAM DESCRIPTION:

PARAM **srcBlock**: The block number of source.

PARAM **destBlock**: The block number of destination.

PARAM **value**: the value to decrement.

RETURN VALUE: Return -1 means failed; return 0 means write card ok.