RFID_R6 API SPECIFICATION with ISO15693

This document is about how to use CLRC663 RFID reader to work with ISO15693 protocol. ISO14443A/Mifare Classic 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/iso15693_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(int type, int datarate, int setAFI, byte AFI, int oneslots, byte[] mask, int masklength);**

FUNC DESCRIPTION：Search card, if success, will make the card into **Quite** or **Selected** status which is decided by the value of type.

PARAM DESCRIPTION：

PARAM **type**： Can only be set to the value of ***ISO15693_ACTIVATE_ADDRESSED*** or ***ISO15693_ACTIVATE_SELECTED***. First make the card into **Quite** status, second make card into **Selected** status. In **Quite** status the communication is in **addressed mode**, in **Selected** statsu the communication is in **selected mode**. Please refer to the *ISO/IEC15693* protocol.

PARAM **datarate**：Can only be set to the value of ***ISO15693_FLAG_UPLINK_RATE_LOW*** or ***ISO15693_FLAG_UPLINK_RATE_HIGH***. First means use low speed, second means use high speed. Please refer to the *ISO/IEC15693* protocol.

PARAM **setAFI**：Can only be set to the value of ***ISO15693_FLAG_NO_USE_AFI*** or ***ISO15693_FLAG_USE_AFI***. First means don't use AFI to filter the card in search; second means use AFI to filter cards. Plesae refer to the *ISO/IEC15693* protocol.

PARAM **AFI**：The value of AFI to filter. This param is valid only if param **setAFI** is set to the value of ***ISO15693_FLAG_USE_AFI.*** Please refer to the *ISO/IEC15693* protocol.

PARAM **oneslots**：Can only be set to the value of ***ISO15693_FLAG_SIXTEEN_SLOTS*** or ***ISO15693_FLAG_ONE_SLOTS***. First means search cards in 16 slots; second means only use one. Plesae refer to the *ISO/IEC15693* protocol.

PARAM **mask**：Use this value to filter card's ID. If the mask bits you want to use is not an integer multiple of 8, leave the higher bit to zero of the byte. For example, if you want the mask bits to 11, the length of byte array it to be 2 bytes(16 bits), low 11 bits is the mask and higther five should be zero. Please refer to the *ISO/IEC15693* protocol.

PARAM **masklength**：The length of mask bits in the param **mask**. Plesae refer to the *ISO/IEC15693* protocol.

RETURN VALUE：If no card searched, **nul**l is returned. If success, a byte array is returned which has the card info. The length of the byte array is 10. First 8 bytes contain the card's SN. The SN in these 8 bytes is stored in **reverse order**. Then the place of ***ISO15693_UID_AT_DSFID*** in 10 bytes stored the value of **DSFID**. The place of ***ISO15693_UID_AT_MORECARD*** in 10 bytes stored the value which indicate whether there is more than one card. If it's value is 0, means only one card searched, if is 1, means more than one card can be searched.

4 FUNC PROTOTYPE：**byte[] read_card_info();**

FUNC DESCRIPTION：Execuate the Protocol's **Get System Information** cmd to get card's information. Some types of card may not support this cmd.

PARAM DESCRIPTION：none.

RETURN VALUE：If failed will return **null**; If success, a byte array is returned which has the card info. The place of ***ISO15693_INFO_AT_DSFID*** in the byte array stores the value of **DSFID**. The place of ***ISO15693_INFO_AT_AFI*** in the byte array stores the value of **AFI**. The place of ***ISO15693_INFO_AT_BLOCK_NR*** in the byte array stores the number of card's blocks. The place of ***ISO15693_INFO_AT_BLOCK_SIZE*** in the byte array stores the size of card's block. The place of ***ISO15693_INFO_AT_IC*** in the byte array stores the value of **IC**. Please refer to the *ISO/IEC15693* protocol.

5 FUNC PROTOTYPE：**int stay_quiet();**

FUNC DESCRIPTION：Make the current selected card which is in **Selected** status into **Quite** status. Please refer to the *ISO/IEC15693* protocol.

PARAM DESCRIPTION：none.

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

6 FUNC PROTOTYPE：**int select_card();**

FUNC DESCRIPTION：Make the current selected card which is in **Quite** status into **Selected** status. Please refer to the *ISO/IEC15693* protocol.

PARAM DESCRIPTION：none.

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

7 FUNC PROTOTYPE：**int reset_to_ready();**

FUNC DESCRIPTION：Make the current selected card into Ready status. Please refer to the *ISO/IEC15693* protocol.

PARAM DESCRIPTION：none.

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

8 FUNC PROTOTYPE：**byte[] read_block(int option, int block);**

FUNC DESCRIPTION：Read a block.

PARAM DESCRIPTION：

PARAM **option**： Can only be set to the value of ***ISO15693_OPTION_OFF*** or ***ISO15693_OPTION_ON***, this is depend on card type.

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

RETURN VALUE：If failed return null; if success, will return a byte array which contain the block's content. It's size is block size. For some type of card, if set ***ISO15693_OPTION_OFF***, may return block size + 1 bytes. The first byte is some status. Other block size bytes is the content of block. Please refer to the *ISO/IEC15693* protocol.

9 FUNC PROTOTYPE：**byte[] read_multi_block(int option, int firstblock, int block_nr);**

FUNC DESCRIPTION：Read from multi consecutive blocks. Some types of card may don't support this cmd.

PARAM DESCRIPTION：

PARAM **option**： Can only be set to the value of ***ISO15693_OPTION_OFF*** or ***ISO15693_OPTION_ON***, this is depend on card type.

PARAM **firstblock**：The first block number you want to read.

PARAM **block_nr**：The number of blocks want to read.

RETURN VALUE：If failed return null; if success, will return a byte array, which have the size 其大 of **block size * block_nr**, store the content of blocks. Like func **read_block**, if set ***ISO15693_OPTION_ON***, may return more content.

10 FUNC PROTOTYPE：**int write_block(int option, int block, byte[] data);**

FUNC DESCRIPTION：Write value to one block.

PARAM DESCRIPTION：

PARAM **option**： Can only be set to the value of ***ISO15693_OPTION_OFF*** or ***ISO15693_OPTION_ON***, this is depend on card type.

PARAM **block**：The number of block.

PARAM **data**：The data want write to block. Size should be block size which can call func **get_card_info**.

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

10 FUNC PROTOTYPE：**int write_multi_block(int option, int firstblock, int block_nr, byte[] data);**

FUNC DESCRIPTION：Write values to multi consecutive blocks. Some types of card may don't support this cmd.

PARAM DESCRIPTION：

PARAM **option**： Can only be set to the value of ***ISO15693_OPTION_OFF*** or ***ISO15693_OPTION_ON***, this is depend on card type.

PARAM **firstblock**：The first block number you want to write.

PARAM **block_nr**：The number of blocks want to write.

PARAM **data**：The content want to write. The size should be **block size * block_nr**.

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

11 FUNC PROTOTYPE：**int write_AFI(int option, byte AFI);**

FUNC DESCRIPTION：Write the value of **AFI** to card.

PARAM DESCRIPTION：

PARAM **option**：Can only be set to the value of ***ISO15693_OPTION_OFF*** or ***ISO15693_OPTION_ON***, this is depend on card type.

PARAM **AFI**：The value of **AFI**.

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

12 FUNC PROTOTYPE：**int write_DSFID(int option, byte DSFID);**

FUNC DESCRIPTION：Write the value of **DSFID** to card.

PARAM DESCRIPTION：

PARAM **option**：Can only be set to the value of ***ISO15693_OPTION_OFF*** or ***ISO15693_OPTION_ON***, this is depend on card type.

PARAM **DSFID**：The value of **DSFID**.

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

13 FUNC PROTOTYPE：**int lock_block(int option, int block);**

FUNC DESCRIPTION：Permanent lock the value of block. If this operation success, the value of the block will never be changed or unlock. If lock again, will failed.

PARAM DESCRIPTION：

PARAM **option**：Can only be set to the value of ***ISO15693_OPTION_OFF*** or ***ISO15693_OPTION_ON***, this is depend on card type.

PARAM **block**：The number of the block want to lock.

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

14 FUNC PROTOTYPE：**int lock_AFI(int option);**

FUNC DESCRIPTION：Permanent lock the value of **AFI**. If this operation success, the value of **AFI** will never be changed or unlock. If lock again, will failed.

PARAM DESCRIPTION：

PARAM **option**：Can only be set to the value of ***ISO15693_OPTION_OFF*** or ***ISO15693_OPTION_ON***, this is depend on card type.

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

15 FUNC PROTOTYPE：**int lock_DSFID(int option);**

FUNC DESCRIPTION：Permanent lock the value of **DSFID**. If this operation success, the value of **DSFID** will never be changed or unlock. If lock again, will failed.

PARAM DESCRIPTION：

PARAM **option**： Can only be set to the value of ***ISO15693_OPTION_OFF*** or ***ISO15693_OPTION_ON***, this is depend on card type.

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

16 FUNC PROTOTYPE：**byte[] get_multi_block_security_status(int firstblock, int nr_block);**

FUNC DESCRIPTION：Get the security status of multi consecutive blocks.

PARAM DESCRIPTION：

PARAM **firstblock**：The first block number you want to get.

PARAM **nr_block**：The number of blocks want to read.

RETURN VALUE：If failed, will return null. If success, return a byte array. The size of the byte array is **nr_block**, the value of each byte indicate the block security status. If the value is 0, means the corresponding block is not locked, if value is 1, means the corresponding block is locked.