

USB Reader Programmer Guide

1. imprint(Revision V1.00).

* All rights reserved, Specifications subject to change without notice.

2. explain of backtrack value.

2.1 the function return values

0x00	Command OK. (success)
0x01	Command FAILURE, the error code is:
0x02	checksum error.
0x03	Not selected COM port
0x04	time out reply
0x05	check sequence error
0x07	Receive error
0x0A	the parameter value out of range

2.2 System Error/Status Codes (0x80-0xFF)

0x80	SET OK. (success)
0x81	SET FAILURE
0x82	Reader reply time out error
0x83	the card do not exist
0x84	the data is error
0x85	the authentication failure
0x86	Unknown Internal Error
0x89	operation error
0x8f	Reader received unknown command
0x90	show the card could not support this command
0x91	show the command format have a mistake
0x92	show the command could not support OPTION form
0x93	show the inputed block is inexistence.
0x94	show the inputed block had been locked
0x95	show Locked the block is not successful
0x96	show the write card operation is not successful.

3、System Commands

3.5 int SetSerNum(unsigned char *newValue,
unsigned char *buffer);

Description:

set 8 byte serial number which be supplied by manufactory

Input Parameter Description:

*newValue 8 byte serial number

*buffer buffer send a pointer, it is used to return the received value.

Output Paramete:

*buffer Returns the STATUS of state

If set success *buffer=0x 80

If failure *buffer System Error/Status Codes(You can consult the 2.2)

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE

3.6 int GetSerNum(unsigned char *buffer);

Description:

read one byte reader address and 8 byte serial number which be supplied by manufactory

Input Parameter Description:

*buffer buffer send a pointer, it is used to return the received value.

Output Paramete:

*buffer buffer[0] reader address

buffer[1...8] 8 byte reader serial number

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE

3.7 int WriteUserInfo(int num_blk,
int num_length,
char *user_info);

Description:

the reader consist of 4 blocks(each block less than 120 byte),the user data space in all 480 byte. the user could base the requirement to deposited the relevant userinfo into the reader.

Input Parameter Description:

num_blk	the block number
num_length	data length
*user_info	user information

Return value:

0x00	Command OK. (success)
0x01	Command FAILURE

```
3.8  int  ReadUserInfo( int      num_blk,
                        int      num_length,
                        char      *user_info);
```

Description:

Read the date from the reader, the reader consist of 4 blocks
(each block less than 120 byte)

Input Parameter Description:

int num_blk	the block number
int num_length	data length
*user_info	wait for reading user date

Output Paramete:

*user_info	If Command Failure,then user_info[0] is error code.
	If Command success,then user_info[0..N] is read user information

Return value:

0x00	Command OK. (success)
0x01	Command FAILURE

```
3.9  int  GetVersionNum( char      *VersionNum);
```

Description:

read reader version number

Input Parameter Description:

*VersionNum wait for reading version number

Output Paramete:

*VersionNum If Command Failure,then VersionNum [0] is error code.
if Command success,then VersionNum [0..N] is reading version number

Return value:

0x00	Command OK. (success)
0x01	Command FAILURE

3.10 int ControlLED(unsigned char freq,
 unsigned char duration,
 unsigned char *buffer);

Description: the work state of set light,include,light Cyc and repeating times

Input Parameter Description:

freq periodicity
duration times
*buffer wait for return value parameter

Output Paramete:

*buffer If Command Failure,then buffer [0] 为 is error code.
If Command success ,then buffer [0] is 0x80

Return value:

0x00	Command OK. (success)
0x01	Command FAILURE

3.11 int ControlBuzzer(unsigned char freq,
 unsigned char duration,
 unsigned char *buffer);

Description: the work state of setting buzzer,include buzzer work Cyc and repeating times

Input Parameter Description:

freq	periodicity
duration	times
*buffer	wait for return value parameter

Output Paramete:

*buffer	If Command Failure,,then buffer [0] 为 is error code. If Command success,then buffer [0] is 0x80
---------	--

Return value:

0x00	Command OK. (success)
0x01	Command FAILURE

4 ISO14443 Type-A Commands

4.1 Type-A Commands

4.1.1 int MF_Request(unsigned char inf_mode,
unsigned char *buffer);

Description:: send ISO14443 A seeking card instruction

Input Parameter Description:

Inf_mode	seeking card mode
----------	-------------------

0x01 –Idle mode (operate one card once a time)

0x00 –All mode (operate a lot of card once a time)

*buffer	wait for return value parameter
---------	---------------------------------

Output Paramete:

*buffer	If Command FAILURE,then buffer [0] is error code. if Command OK,then buffer [0..1], return data bunch within 2 byte
---------	--

Return value:

0x00	Command OK. (success)
0x01	Command FAILURE

```
4.1.2 int MF_Anticoll( unsigned char *snr,
                      unsigned char &status);
```

Description:: test card quantity, single or much more, and then return the card number within 4 byte(if there are many cards, the only return one of card number

Input Parameter Description:

Status send a finger, return to the number of the card

*snr send a finger, return to 4 byte card number

Output Paramete:

If Command success

Status the checked card number(0x00 means have checked a single card,0x01 means have checked more cards

*snr 4 byte card number (snr[0..3])

If Command Failure

*snr System Error/Status Codes(You can consult the 2.2)

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE

```
4.1.3 int MF_Select( unsigned char *snr);
```

Description:

Select card, make the card as been select stat...

Input Parameter Description:

*snr send a finger,input 4 byte card number,and return to 4 byte card numbe

Output Paramete:

Status checked card numbe

*snr 4 byte card number4 (snr[0..3])

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE

4.1.4 int MF_Halt();

Description:

select card,make the card as been break off

Input Parameter Description:

none

Output Paramete:

None.

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE

4.2 Mifare Appilication Commands

4.2.1 int MF_Read(unsigned char mode,
unsigned char blk_add,
unsigned char num_blk,
unsigned char *snr,
unsigned char *buffer);

Description:

read the appointed length date at the appointed station

Input Parameter Description:

mode, read mode

(Request Idle + Key A mode=00 , Request Idle + Key B mode= 02,
Request All + Key A mode=01 , Request All + Key B mode=03)
(the up number is hex)

blk_add, read block address

num_blk, read block amount

*snr, a finger, transfer eight byte secret key

*buffer wait receive the variable of output finger

Output Paramete:

If Command success

*snr, 4 byte card number

*buffer, the read date (the fact number is: num_blk*16)

If Command Failure

buffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE

```
4.2.2 int MF_Write( unsigned char mode,
                    unsigned char blk_add,
                    unsigned char num_blk,
                    unsigned char *snr,
                    unsigned char *buffer);
```

Description:

Read-in date At appoint station

Input Parameter Description:

mode, needs write mode

(Request Idle + Key A mode=00 , Request Idle + Key B mode= 02,
 Request All + Key A mode=01 , Request All + Key B mode=03)

blk_add, needs fill in block address

num_blk, needs fill in block number

*snr, wait fill in date

*buffer, afferent pointer sign to output the date

Output Paramete:

If Command success

snr[0..3], four byte card number

If Command Failure

buffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE

```
4.2.3 int MF_InitVal( unsigned char mode,
                      unsigned char SectNum,
                      unsigned char *snr,
                      int value);
```

Description:

Initialize card

Input Parameter Description:

mode, initialize mode
(Request Idle + Key A mode=00 , Request Idle + Key B mode= 02,
Request All + Key A mode=01 , Request All + Key B mode=03)
SectNum, need initialize fan number
*snr, six byte secret key (introduction as pointer)
value 4 byte initialize the date

Output Paramete:

If Command success

snr[0..3] 4 byte card number

If Command Failure

snr[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE

```
4.2.4 int MF_Dec( unsigned char mode,  
                  unsigned char SectNum,  
                  unsigned char *snr,  
                  int *value);
```

Description:

Devalue work to the fan of the card

Input Parameter Description:

mode, work mode
(Request Idle + Key A mode=00 , Request Idle + Key B mode= 02,
Request All + Key A mode=01 , Request All + Key B mode=03)
SectNum, the fan number which need to write value 00-0F
*snr, 6 byte secret key (introduction as pointer)
value need decrease value,4 byte length

Output Paramete:

If Command success

snr[0..3], 4 byte card number

value[0..3] date bunch after 4 byte work

If Command Failure

snr[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE

```
4.2.5    int   MF_Inc (    unsigned char    mode,
                         unsigned char    SectNum,
                         unsigned char    *snr,
                         int               *value);
```

Description:

Add value work to appointed fan of the card.

Input Parameter Description:

mode, work mode

(Request Idle + Key A mode=00 , Request Idle + Key B mode= 02,
 Request All + Key A mode=01 , Request All + Key B mode=03)

SectNum, need add value fan number 00-0F

*snr, 6 byte secret key(introduction as pointer)

value need add value, 4 byte length

Output Paramete:

 If Command success

 snr[0..3], 4 byte card number

 value[0..3] the date bunch after 4 byte work

 If Command Failure

 snr[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE

```
4.2.6    int   MF_Getsnr (
                         unsigned char    mode,
                         unsigned char    halt,
                         unsigned char    *snr
                         unsigned char    *value);
```

Description:

Output Paramete:

If Command success
 carddata[0..N], incept the return date
 If Command Failure
 carddata[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00 Command OK. (success)
 0x01 Command FAILURE

5. ISO14443 Type-B Commands

5.1 int RequestType_B(
 unsigned char *buffer)

Description: this order perform REQB order of ISO14443B, get PUPI code of the card

Input Parameter Description:

*buffer, the date bunch after card reposition (ATQB)

Output Paramete:

If Command success

 *buffer, the date bunch after card reposition (ATQB)
 buffer[0] card reposition date length
 buffer[0..N] the date after work (ATQB)

If Command Failure

 buffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00 Command OK. (success)
 0x01 Command FAILURE

5.2 int AntiType_B(
 unsigned char *buffer);

Description: this order perform Anticol1B of ISO14443B

Input Parameter Description:

*buffer, The date after return card (ATQB)

Output Paramete:

If Command success

Buffer[0..N], the date of card return (ATQB)

If Command Failure

buffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE

5.3 int SelectType_B (
unsigned char *SerialNum);

Description:

this order perform ATTRIB of ISO14443B, distribute a sign for CID to the know card

Input Parameter Description:

*SerialNum, the serial number of the card

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE

5.4 int Request_AB(
unsigned char * buffer);

Description:

This order performance integration REQUEST AND ATTRIB order of ISO14443B,use one order to make card reposition.

Input Parameter Description:

* buffer, return to 4 byte serial number of the worked card

Output Paramete:

If Command success

buffer[0..3], return to 4 byte serial number of the worked card

If Command Failure

buffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE

```
5.5 int  API_ISO14443TypeBTransCOSCmd(
                                     unsigned char  *cmd,
                                     int              cmdSize,
                                     unsigned char  *buffer);
```

Description:

ISO14443 remit order, any effect order and date can be transfer by this order

Input Parameter Description:

*cmd, date which wait to be send

cmdSize, date length

* buffer, return date

Output Paramete:

 If Command success

 buffer[0..N] the date which returned from the card

 If Command Failure

 buffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE

6 ISO15693 COMMANDS

```
6.1 int  ISO15693_Inventory(
                                     unsigned char  *cardnumber
                                     unsigned char  *pBuffer);
```

Description:

 This order is to get all the card's serial number which in the reading card district (the getable card number is relate to the output rate of the module antenna, commonly can read

2~6 card within anticollision)

Input Parameter Description:

*Cardnumber the return card number(length is 1 byte)
*pBuffer the return date (include LAG, DSFID and 8*n byte card number)

Output Paramete:

IF: Command success (return 0x00)

*nrOfCard return card number (one byte)
*pBuffer return date (include FLAG, DSFID and 8*n byte card number)

IF: Command Failure (return 0x01)

*nrOfCard System Error/Status Codes(You can consult the 2.2)

Return value:

0x00	Command OK. (success)
0x01	Command FAILURE, the error code is 2.2

6.2 int ISO15693_Read (

unsigned char	flags,
unsigned char	blk_add,
unsigned char	num_blk,
unsigned char	*uid,
unsigned char	*buffer);

Description:

It's for reading 1 and numerous fan value. If need to read the safe digit of every block, make Option flag of FLAGS as 1 ,means FLAG=0x42,every fan will return 5 byte, include one byte show safe state and 4 byte block content, **here the most can read 12 block**, IF FLAG=02,will only return 4 byte block content, **here the most can read 63 block**.

Input Parameter Description:

flags	0x02 without uid
	0x22 with uid
	0x42 without uid but need to read the safe digit
blk_add,	needed reading origin block number
num_blk,	block quantity
*uid	UID message
*buffer	return value

Output Paramete:

If : Command success (return 0x00)

*buffer	return date
buffer[0]	return flag
buffer[1..N]	Data

If : Command Failure (return 0x01)

buffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00	Command OK. (success)
0x01	Command FAILURE, the error code is 2.2

6.3 int ISO15693_Write(

unsigned char	flags,
unsigned char	blk_add,
unsigned char	num_blk,
unsigned char	*uid,
unsigned char	*data);

Description: make writting work to every block(every time only can write one block)

Input Parameter Description:

flags	0x02 without uid
	0x22 with uid
	0x42 without uid but need to read the safe digit
blk_add,	needed writting origin block number
num_blk,	writing block quantity
*uid	UID message
*buffer	return value

Output Paramete:

If : Command success (return 0x00)

*buffer	return date
buffer[0]	return flag
buffer[1..N]	Data

If : Command Failure (return 0x01)

buffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00	Command OK. (success)
0x01	Command FAILURE, the error code is:2.2


```

6.4  int  ISO15693_Lock(
                                unsigned char  flags,
                                unsigned char  num_blk,
                                unsigned char  *uid
                                unsigned char  *buffer);

```

Description: for lock block content, caution : this process can not in reverse(can not unlock) the content can not revise when the block be locked.

Input Parameter Description:

flags	0x02 without uid
	0x42 without uid but need to read the safe digit
	0x22 with uid
num_blk,	locked block number
*uid	UID message
*buffer	return value

Output Paramete:

If : Command success (return 0x00),

buffer[0] return 0x80 , means work ok.,

If : Command Failure (return 0x01)

buffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00	Command OK. (success)
0x01	Command FAILURE, the error code is:2.2

```

6.5  int  ISO15693_StayQuiet(
                                unsigned char  flags,
                                unsigned char  *uid,
                                unsigned char  *buffer );

```

Description:

This order is to make the card to sleep sate, must use the address mode if the sent date is same to the card serial number, after this work finish, the card will be sleep, otherwise the state will not change.

Input Parameter Description:

flags	sign byte 1byte
*uid	UID message
*buffer	return value

Output Paramete:

If : Command success (return 0x00)

buffer[0] return 0x80, means work ok.

If : Command Failure (return 0x01)

buffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE, the error code is:2.2

6.6 int ISO15693_Select(

unsigned char flags,
unsigned char *uid,
unsigned char *buffer);

Description:

This order must use address mode, if the sent date is same to the card serial number, after the work ok, the card will be selected, otherwise the state will not change

Input Parameter Description:

flags sign byte 1 byte

*uid UID message

*buffer return value

Output Paramete:

If : Command success (return 0x00)

buffer[0] return to 0x80, means work ok

If : Command Failure (return 0x01)

buffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE, the error code is 2.2

6.7 int ISO15693_ResetToReady(

unsigned char flags,
unsigned char *uid,
unsigned char *buffer);

Description:

After the work ok, the card will return to Ready state .

Input Parameter Description:

flags	sign byte 1 byte
	0x02 without uid
	0x42 without uid but need to read safe digit
	0x22 with uid
*uid	UID message
*buffer	return value

Output Paramete:

If : Command success (return 0x00)

buffer[0] return 0x80, means work ok

If : Command Failure (return 0x01)

buffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00	Command OK. (success)
0x01	Command FAILURE, the error code is:

```
6.8 int ISO15693_WriteAFI(  
                                unsigned char flags,  
                                unsigned char afi,  
                                unsigned char *uid,  
                                unsigned char *buffer );
```

Description:

Write AFI to the card .

Input Parameter Description:

flags	sign byt 1 byte
	0x02 without uid
	0x42 without uid but need to read the safe digit
	0x22 with uid
afi	wait write AFI
*uid	UID message
*buffer	return value

Output Paramete:

If : Command success (return 0x00)

buffer[0] return 0x80, means work ok

If : Command Failure (return 0x01)

buffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE, the error code is:2.2

```
6.9 int ISO15693_LockAFI(  
                                unsigned char flags,  
                                unsigned char *uid,  
                                unsigned char *buffer );
```

Description:

For lock AFI of the card, after lock AFI can not change

Input Parameter Description:

flags sign byte 1 byte
0x02 without uid
0x42 without uid but need to read the safe digit
0x22 with uid
*uid UID message
*buffer return value

Output Paramete:

If : Command success (return 0x00),

buffer[0] return 0x80, means work ok

If : Command Failure (return 0x01)

buffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE, the error code is:2.2

```
6.10 int ISO15693_WriteDSFID(  
                                unsigned char flags,  
                                unsigned char DSFID,  
                                unsigned char *uid,
```

unsigned char *buffer);

Description:

Write DSFID to the card

Input Parameter Description:

flags	sign byte 1 byte
0x02	without uid
0x42	without uid but need to read the safe digit
0x22	with uid

DSFID the writted DSFID byte, the length is 1 byte

*uid UID message

*buffer return value

Output Paramete:

If : Command success (return 0x00),

buffer[0] return 0x80, means work ok

If : Command Failure (return 0x01)

buffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE, the error code is:

6.11 int ISO15693_LockDSFID(

unsigned char flags,
unsigned char *uid,
unsigned char *buffer);

Description:

For lock DSFID of the card, after lock, DSFID can not change

Input Parameter Description:

flags	sign byte (length is 1 byte)
0x02	without uid
0x42	without uid
0x22	with uid

*uid UID message
*buffer return value

Output Paramete:

If : Command success (return 0x00),

buffer[0] return 0x80, means work ok

If : Command Failure (return 0x01)

buffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00 Command OK. (success)

0x01 Command FAILURE, the error code is:2.2

6.12 int ISO15693_GetSysInfo(
 unsigned char flag,
 unsigned char *uid,
 unsigned char *Buffer);

Description:

For get the particular message of the card, the fact content please refer to ISO15693 agreement date.

Input Parameter Description:

flags sign byte 1 byte
 0x02 without uid
 0x42 withou uid but need to read the safe digit
 0x22 with uid

*uid UID message
*buffer return value

Output Paramete:

If : Command success (return 0x00),

Then Buffer [0]: Flags

Buffer[1]: INFO Flags

Buffer[2..9]: UID

Buffer[10]: DSFID

Buffer[11]: AFI

Buffer[12..N]: Other fields

If : Command Failure (return 0x01)

Buffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00	Command OK. (success)
------	------------------------

0x01	Command FAILURE, the error code is:2.2
------	--

6.13 int ISO15693_GetMulSecurity()

```

unsigned char    flag,
unsigned char    blkAddr,
unsigned char    blkNum,
const unsigned char *uid,
unsigned char    *pBuffer);

```

Description: for get the date of every safe state block of the card

Input Parameter Description:

flag	0x02	without uid
	0x22	with uid
	0x42	without uid but need to read the safe digit
blkAddr,		the read origin block number
blkNum,		the read block quantity
*uid		UID message
*pBuffer		return value

Output Paramete:

If : Command success (return 0x00),

*pBuffer	return value	
pBuffer [0]	return flags	pBuffer [1..N] Block security status (the safe state of block)

If : Command Failure (return 0x01)

pBuffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00	Command OK. (success)
------	------------------------

0x01	Command FAILURE, the error code is:2.2
------	--

```

6.14  int  ISO15693_TransCOSCmd(
                                     unsigned char  *cmd,
                                     int               cmdSize,
                                     unsigned char  *buffer);

```

Description: a currency order, the user can use this order to make kinds of handle to the card

Input Parameter Description:

*cmd, the date need to be send.
cmdSize, the date length
*buffer return value

Output Paramete:

If : Command success (return 0x00),

 *buffer return data
 buffer [0..N] the date return from the card

If : Command Failure (return 0x01)

 buffer[0] System Error/Status Codes(You can consult the 2.2)

Return value:

0x00	Command OK. (success)
0x01	Command FAILURE, the error code is:2.2