FEITIAN

aR530 developer’s Guide For iOS

V1.3

Revision History:

Date	Revision	Description
Apr. 2013	V1.0	Release of the first version
Dec,2013	V1.1	Add new features in
Jan, 2014	V1.2	Support maximum 258 bytes data return Can through card type to choose specific card
June 11, 2014	V1.3	Removed MSR API support Add get lib/fw version API, and others

Software Developer’s Agreement

All Products of Feitian Technologies Co., Ltd. (Feitian) including, but not limited to, evaluation copies, diskettes, CD-ROMs, hardware and documentation, and all future orders, are subject to the terms of this Agreement. If you do not agree with the terms herein, please return the evaluation package to us, postage and insurance prepaid, within seven days of their receipt, and we will reimburse you the cost of the Product, less freight and reasonable handling charges.

1. Allowable Use – You may merge and link the Software with other programs for the sole purpose of protecting those programs in accordance with the usage described in the Developer’s Guide. You may make archival copies of the Software.

2. Prohibited Use – The Software or hardware or any other part of the Product may not be copied, reengineered, disassembled, decompiled, revised, enhanced or otherwise modified, except as specifically allowed in item 1. You may not reverse engineer the Software or any part of the product or attempt to discover the Software’s source code. You may not use the magnetic or optical media included with the Product for the purposes of transferring or storing data that was not either an original part of the Product, or a Feitian provided enhancement or upgrade to the Product.

3. Warranty – Feitian warrants that the hardware and Software storage media are substantially free from significant defects of workmanship or materials for a time period of twelve (12) months from the date of delivery of the Product to you.

4. Breach of Warranty – In the event of breach of this warranty, Feitian’s sole obligation is to replace or repair, at the discretion of Feitian, any Product free of charge. Any replaced Product becomes the property of Feitian.

Warranty claims must be made in writing to Feitian during the warranty period and within fourteen (14) days after the observation of the defect. All warranty claims must be accompanied by evidence of the defect that is deemed satisfactory by Feitian. Any Products that you return to Feitian, or a Feitian authorized distributor, must be sent with freight and insurance prepaid.

EXCEPT AS STATED ABOVE, THERE IS NO OTHER WARRANTY OR REPRESENTATION OF THE PRODUCT, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

5. Limitation of Feitian’s Liability – Feitian’s entire liability to you or any other party for any cause whatsoever, whether in contract or in tort, including negligence, shall not exceed the price you paid for the unit of the Product that caused the damages or are the subject of, or indirectly related to the cause of action. In no event shall Feitian be liable for any damages caused by your failure to meet your obligations, nor for any loss of data, profit or savings, or any other consequential and incidental damages, even if Feitian has been advised of the possibility of damages, or for any claim by you based on any third-party claim.

6. Termination – This Agreement shall terminate if you fail to comply with the terms herein. Items 2, 3, 4 and 5 shall survive any termination of this Agreement.

Contents

Chapter 1. Overview

This chapter describes how to develop applications through aR530 SDK, including the development interfaces supported by the product (aR530) and how to develop applications based on these interfaces.

FEITIAN aR530 is swipe card reader and NFC contactless reader specially engineered to accommodate a range of smart card applications. Developers use it as a platform to generate and deploy related products and services. Moreover, FEITIAN aR530 is a terminal unit which is seamlessly integrated to all major systems of operation. Additional features such as the built-in inclusive support for different smart card interfaces has facilitated the wide scale and cross industry adoption of aR530.

aR530 suits customers where security concerns are the most salient and satisfies the demand for a flexible solution for ID authentication, e-commerce, e-payment, information security and access control.

Chapter 2. Features

Features:

Contact part:

1. Support magnetic strip card
2. Audio jack compatible to different mobile OS
3. Support track 1,2,3 and key management with DUKPT(3DES&AES)
4. Low battery consumption
5. Micro-USB port for pass-through charging

Contactless part:

1. Firmware supports upgrading in encryption
2. Supports contactless smart cards compliant with ISO 14443 type A and type B, Mifare card,Felica.
3. Through beeper and light to informed card status

Battery usage cycle:

Status	Power consumption	Hours of use
Standby	23mA	20h
FEITIAN CPU Card	73 mA	4.5h
HongKong Octopus Card	66 mA	5h
Felica	81 mA	4h
Mifare card	72 mA	4.5h

We provide three lights which is red/blue/yellow, each means charge battery/low battery/card status.

Card status light – blue color light

Number	Progress	Status
1	No card	Light OFF
2	Card detected	Light ON

Low battery light – yellow color light

Number	Progress	Status
1	Full battery	Light OFF
2	Low battery	Light ON

Charge battery light – red color light

Number	Progress	Status
1	Charging completed	Light OFF
2	Charging	Light ON

Chapter 3. Definitions

3.1 Card Type

Can through below card type to choose specific card

```
#define CARD_TYPE_A                  0x000A
#define CARD_TYPE_B                  0x000B
#define CARD_TYPE_C                  0x000C
#define CARD_INNOVISION_TOPAZ    0x10
#define CARD_NXP_MIFARE_UL        0x20
#define CARD_NXP_MIFARE_UL_C      0x21
#define CARD_NXP_MIFARE_1K        0x22
#define CARD_NXP_MIFARE_4K        0x23
#define CARD_NXP_DESFIRE_EV1      0x40
#define CARD_NXP_FELICA           0x50
#define CARD_NXP_TYPE_B            0x60
#define CARD_NXP_M_1_B              0x61
```

3.2 Error codes

The following is a list of commonly used errors. Since different cards produce different errors they must map over to these error messages.

```
#define NFC_CARD_ES_SUCCESS                          0x00000000
#define NFC_CARD_ES_GENERAL_ERROR                     0x00000001
#define NFC_CARD_ES_ARGUMENTS_BAD                      0x00000002
#define NFC_CARD_ES_INVALID_CARD_HANDLE              0x00000003
#define NFC_CARD_ES_RESPONSE_TOO_SHORT                0x00000004
#define NFC_CARD_ES_TIMEOUT                           0x00000005
#define NFC_CARD_ES_MEMORY_INSUFFICIENT              0x00000006
#define NFC_CARD_ES_BUFFER_TOO_SMALL                  0x00000007
#define NFC_CARD_ES_WAIT                             0x00000008
#define NFC_CARD_ES_KEY_LOCKED                        0x00000009
#define NFC_CARD_ES_DEVICE_BUSY                      0x0000000A
#define NFC_CARD_ES_NO_SMARTCARD                         0x0000000B
#define NFC_CARD_ES_FUNCTION_NOT_IMPLEMENTED        0x0000000C
#define NFC_CARD_ES_CC_FILE_IS_EMPTY               0x0000000D
#define NFC_CARD_ES_INVALID_LEN_IN_CC_FILE          0x0000000F
#define NFC_CARD_ES_INVALID_TLV_IN_CC_FILE          0x00000010
#define NFC_CARD_ES_NDEF_IS_NOT_READABLE              0x00000011
#define NFC_CARD_ES_FAILED_SELECT_NDEF_FILE         0x00000012
```

```
#define NFC_CARD_ES_FAILED_READ_NDEF_FILE          0x00000013
    #define NFC_CARD_ES_ACCEPTABLE_ERROR                  0x00000014
```

Chapter 4. API Reference

4.1 LibVersion

Synopsis:

#include “FT_aR530.h”

```
+ (NSString *)FTaR530_LibVersion;
```

Description:

None.

Example: please follow FEITIAN demo code

Returns:

The library version description string.

4.2 Get Device ID

Synopsis:

#include “FT_aR530.h”

```
/*@Name:        +(void)aR530_GetDeviceID:(id<FTaR530Delegate>)delegate;
 *@Function:    Get device ID
 *@Parameter:   IN:(id<FTaR530Delegate>)delegate:
 */
+(void)FTaR530_GetDeviceID:(id<FTaR530Delegate>)delegate;
```

Parameters:

The data through delegate to return

Description:

This function gets the device ID of the card reader.

Example: please follow FEITIAN demo code

Returns: please follow error code

4.3 Get Firmware Version

Synopsis:

#include “FT_aR530.h”

```
/*@Name:        +(void)aR530_GetFirmwareVersion:(id<FTaR530Delegate>)delegate;
 *@Function:    Get firmware version
 *@Parameter:   IN:(id<FTaR530Delegate>)delegate:
 */
```

+(void)FTaR530_GetFirmwareVersion:(id<FTaR530Delegate>)delegate;

Parameters:

The data through delegate to return

Description:

This function gets the firmware version of the card reader.

Example: please follow FEITIAN demo code

Returns: please follow error code

4.4 NFC Card Open

Synopsis:

#include “FT_aR530.h”

+(void)NFC_Card_Open:(id<FTaR530Delegate>)delegate;

Parameters:

The data through delegate to return

Description:

This function creates a communication between reader and cards.

Open the Card Reader and Connect to Smart Card

Example: please follow FEITIAN demo code

Returns: please follow error code

4.5 NFC Card Close

Synopsis:

#include “FT_aR530.h”

```
/*@Name:          +(void)NFC_Card_Close:(nfc_card_t)card delegate:(id<FTaR530Delegate>)delegate;
  *@Function:     Close the CardReader and Disconnect with SmartCard
  *@Parameter:    IN: (1).(nfc_card_t)card: SmartCard's handle has been Opened
  *                   (2).(id<FTaR530Delegate>)delegate
  */
+(void)NFC_Card_Close:(nfc_card_t)card delegate:(id<FTaR530Delegate>)delegate;
```

Parameters:

Use delegate to return data

Description:

This function disconnects a communication between reader and cards.

Example: please follow FEITIAN demo

Returns: Please follow error code

4.6 NFC Card Recognize

Synopsis:

```
#include "FT_aR530.h"
/*@Name:          +(void)NFC_Card_Recognize:(nfc_card_t)card delegate:(id<FTaR530Delegate>)delegate;
  *@Function:     Recognize the smartcard's type
  *@Parameter:    IN: (1).nfc_card_t card: the pointer SmartCard Handle
  *                   (2).(id<FTaR530Delegate>)delegate
  */
+(void)NFC_Card_Recognize:(nfc_card_t)card delegate:(id<FTaR530Delegate>)delegate;
```

Parameters:

Use delegate to get return data

Description:

Get card type

Example: please follow FEITIAN demo

Returns: Please follow error code

4.7 NFC Card Recognize

Synopsis:

```
#include "FT_aR530.h"
/*@Name:          +(void)NFC_CArd_Transmit:(nfc_card_t)card sendBuf:(unsigned char *)sendBuf
sendLen:(unsigned int)sendLen delegate:(id<FTaR530Delegate>)delegate;
  *@Function:     Transmit APDU to Smart Card
  *@Parameter:    IN: (1).nfc_card_t card: The pointer of smart card handle
  *                   (2).(unsigned char *)sendBuf: Send buffer
  *                   (3).(unsigned int)sendLen: Send buffer length
  *                   (4).(id<FTaR530Delegate>)delegate
  */

+(void)NFC_Card_Transmit:(nfc_card_t)card sendBuf:(unsigned char *)sendBuf sendLen:(unsigned int)sendLen
delegate:(id<FTaR530Delegate>)delegate;
```

Parameters:

Use delegate to get return data

Description:

Do transfer between reader and card

Example: please follow FEITIAN demo

Returns: Please follow error code

4.8 FTaR530Delegate Methods

```
//--------------------------FTaR530Delegate Methods------------------------------
@protocol FTaR530Delegate <NSObject>

@optional

/*@Name:              -(void)FTaR530GetInfoDIdComplete:(unsigned char *)retData retDataLen:(unsigned
int)retDataLen    functionNum:(unsigned int)functionNum errCode:(unsigned int)errCode;
  *@Function:       This function will be callback when Get Firmware Version or Get Device ID function completed.
  *@Parameter:      OUT:(1).(unsigned char *)retData:      return Data
  *                         (2).(unsigned int)retDataLen:      return Data length
  *                         (3).(unsigned int)functionNum:    The function number
  *                         (4).(unsigned int)errCode:           The error code(0-sucess,other value-error code)
  *
  */
-(void)FTaR530GetInfoDidComplete:(unsigned char *)retData retDataLen:(unsigned int)retDataLen
functionNum:(unsigned int)functionNum errCode:(unsigned int)errCode;
@end

/*@Name:              -(void)FTNFCDidComplete:(nfc_card_t)cardHandle retData:(unsigned char *)retData
retDataLen:(unsigned int)retDataLen functionNum:(unsigned int)funcNum errCode:(unsigned int)errCode;
  *@Function:       This function will be callback when NFC function completed.
  *@Parameter:      OUT:(1).(nfc_card_t)cardHandle:         the card's handle
  *                         (2).(unsigned char *)retData:      return data
  *                         (3).(unsigned int)retDataLen:      return data length
  *                         (4).(unsigned int)funcNum:           The function number
  *                         (5).unsigned int)errCode:            The error code(0-success, other value-error code)
  */
-(void)FTNFCDidComplete:(nfc_card_t)cardHandle retData:(unsigned char *)retData retDataLen:(unsigned
int)retDataLen functionNum:(unsigned int)funcNum errCode:(unsigned int)errCode;

@end
```