FEITIAN

aR530 developer's Guide

For Android

V1.1

Revision History:

Date	Revision	Description
Jan, 2014	1.0	First version
13th, May, 2014	1.1	1. Removed MSR card support 2. Support find Specify card type 3. Add get reader hardware ID and firmware version
26th, May, 2014	1.2	1. Add error code 2. Modify API parameter
28th, May, 2014	1.3	Add GetCardInfoData() API
30th, May, 2014	1.4	Add comments in get card type API
11th, June, 2014	1.5	Change manual name to aR530

Copyright © Feitian Technologies Co., Ltd.
Website: www.FTsafe.com

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 **4**

Chapter 2. Features **5**

Chapter 3. Definitions **8**

3.1 Error codes 8
3.2 Card type 9

Chapter 4. API Reference **10**

4.1 Initial function 10
4.2 Release function 10
4.3 Get reader hardware serial number and firmware version 10

4.3.1 GetDevicID 10

4.3.1 GetDevicID 11

4.4 Contactless section 12

4.4.1 FTNFC_connect 12

4.4.2 FTNFC_transmitCmd 12

4.4.3 FTNFC_disconnect 13

4.4.4 FTNFCCardType 13

4.4.5 GetCardInfoData 13

4.5 Mifare card section 15

4.5.1 GeneralAuthenticate 15

4.5.2 ReadBinary 15

4.5.3 ClassicBlockInitial 15

4.5.4 ClassicReadValue 16

4.5.5 ClassicStoreBlock 16

4.5.6 ClassicIncrement 16

4.5.7 ClassicDecrement 17

4.6 Reader (plug-in/out) monitor function 18

4.6.1 OnInsertHeadSet 18

4.6.2 OnInsertHeadSet 18

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 NFC only 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.

Copyright © Feitian Technologies Co., Ltd.
Website: www.FTsafe.com

Chapter 2. Features

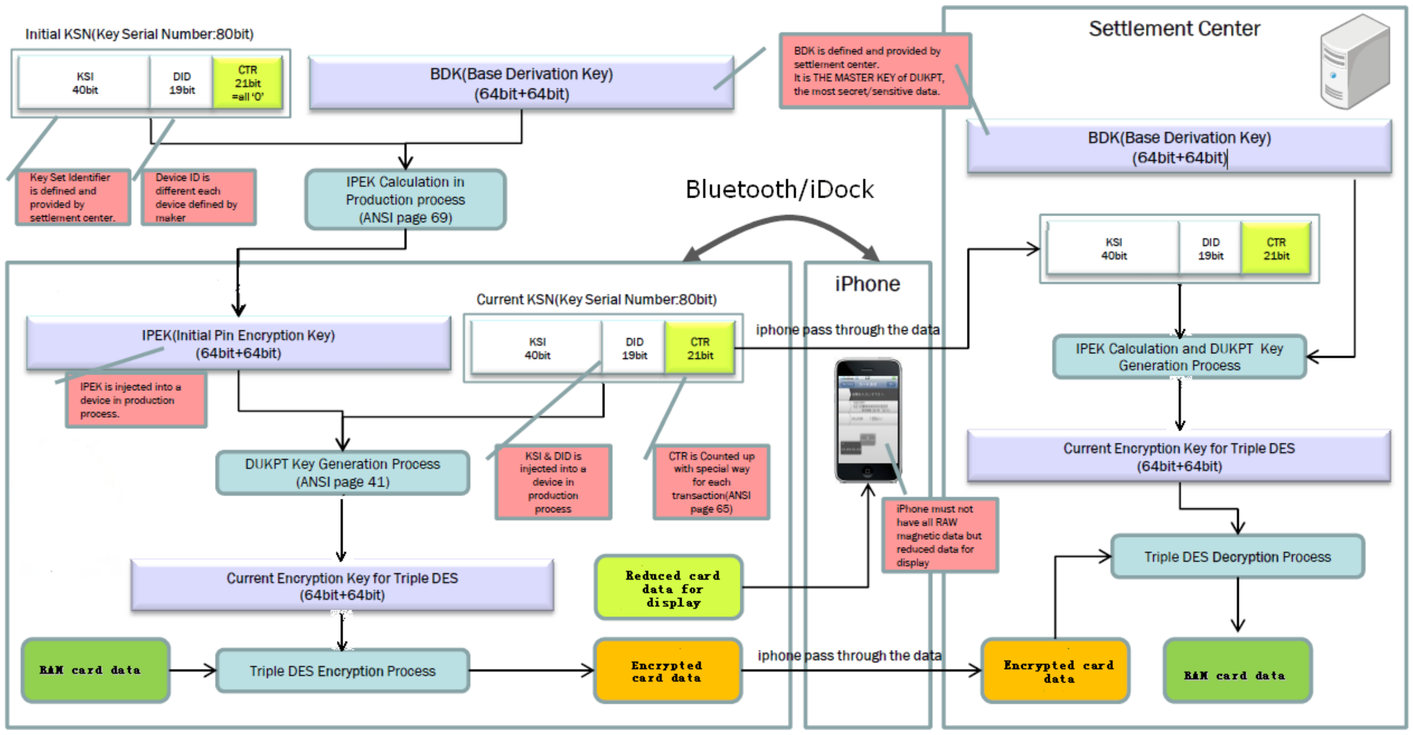
The new reader has been published, included key management and data space.

More security

DUKPT (Derived Unique Key per Transaction) is a key management scheme in which for every transaction, a unique key is used which is derived from a fixed key. Therefore, if a derived key is compromised, future and past transaction data are still protected since the next or prior keys cannot be determined easily. DUKPT is specified in ANSI X9.24 part 1.

We through below picture to give customer a clear concept of DUKPT:

http://en.wikipedia.org/wiki/DUKPT


Initial KSN(Key Serial Number:80bit)
KSI 40bit
DID 19bit
CTR 21bit =all '0'
BDK(Base Derivation Key) (64bit+64bit)
BDK is defined and provided by settlement center. It is THE MASTER KEY of DUKPT, the most secret/sensitive data.
Settlement Center
BDK(Base Derivation Key) (64bit+64bit)
Key Set Identifier is defined and provided by settlement center.
Device ID is different each device defined by maker
IPEK Calculation in Production process (ANSI page 69)
Bluetooth/iDock
iPhone
KSI 40bit
DID 19bit
CTR 21bit
Current KSN(Key Serial Number:80bit)
IPEK(Initial Pin Encryption Key) (64bit+64bit)
KSI 40bit
DID 19bit
CTR 21bit
iphone pass through the data
IPEK Calculation and DUKPT Key Generation Process
IPEK is injected into a device in production process.
DUKPT Key Generation Process (ANSI page 41)
KSI & DID is injected into a device in production process
CTR is Counted up with special way for each transaction(ANSI page 65)
Current Encryption Key for Triple DES (64bit+64bit)
iPhone must not have all RAW magnetic data but reduced data for display
Triple DES Decryption Process
Current Encryption Key for Triple DES (64bit+64bit)
Reduced card data for display
RAM card data
Triple DES Encryption Process
Encrypted card data
iphone pass through the data
Encrypted card data
RAM card data

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

Copyright © Feitian Technologies Co., Ltd.
Website: www.FTsafe.com

Chapter 3. Definitions

3.1 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.

```
//The firmware return status

public String errContent(int errCode) {

    switch (errCode) {

    case Card.CODE_FAIL:

        return "Fail";

    case Card.CODE_DEVICE_NOT_AVAILABLE:

        return "device is not available";

    case Card.CODE_CARD_NOT_CONNECTED:

        return "card is not connected";

    case Card.CODE_DEVICE_COMM_ERROR:

        return "communication error";

    case Card.CODE_PARAM_ERROR:

        return "illegal parameters";

    case Card.CODE_TIMEOUT:

        return "timeout";

    default:

        return "unkown error " + errCode;

    }

}
```

3.2 Card type

Can through card type to choose specify card

```
        //The firmware return status
    Class Card have below member objects
    Card.CARD_TYPE.A_CARD
    Card.CARD_TYPE.B_CARD
    Card.CARD_TYPE.Felica_CARD
    Card.CARD_TYPE.A_M1_CARD
    Card.CARD_TYPE.B_M1_CARD
    Card.CARD_TYPE.Topaz_CARD
```

Through call FTNFC_connect(Card.Type) to choose specify card, more information, please follow FEITIAN sample code

Chapter 4. API Reference

4.1 Initial function

Synopsis:

public native static int initial (Context con);

Parameters:

Context con	IN	the type must be 1 or 2, more information, please follow sample code

Description:

Initial context and environment before using

Example:

More information, please follow sample code.

Returns:

Reference errContent API

4.2 Release function

Synopsis:

public native static int release ();

Parameters:

NULL

Description:

Initial environment before use

Example:

More information, please follow sample code.

Returns:

Reference errContent API

4.3 Get reader hardware serial number and firmware version

4.3.1 GetDevicID

Synopsis:

public int GetDeviceID(byte[] deviceID, JKeyInt len);

Parameters:

DeviceID	out	using to saved reader hardware serial number

Len	out	Return length of hardware serial number

Description:
This function get device serial number from reader.

Example:
More information, please follow sample code.
Returns:
Reference error code section

4.3.1 GetDevicID

Synopsis:
public int GetFirmwareVersion(byte[] version, JKeyInt len);
Parameters:

version	out	using to saved reader firmware version
Len	out	Return length of firmware version

Description:
This function get device firmware version from reader.

Example:
More information, please follow sample code.
Returns:
Reference error code section

4.4 Contactless section

4.4.1 FTNFC_connect

Synopsis:

public int FTNFC_connect(Card.CARD_TYPE[] cardTypes, int timeout);

Parameters:

Card.card_type[]	in	input array of card type

Card type can be below:

Card.CARD_TYPE.A_CARD
Card.CARD_TYPE.B_CARD
Card.CARD_TYPE.Felica_CARD
Card.CARD_TYPE.A_M1_CARD
Card.CARD_TYPE.B_M1_CARD
Card.CARD_TYPE.Topaz_CARD

Timeout	in	timeout while in scan card (second) at list 1 second

Description:

This function using to connect specify card

Example:

More information, please follow sample code.

Returns:

Reference error code section

4.4.2 FTNFC_transmitCmd

Synopsis:

public native static int FTNFC_ transmitCmd (byte[] sendData, byte[] recvData);

Parameters:

sendData	IN	command which will send to card
recvData	OUT	return data from card

Description:

This function use to do transfer data between reader and card.

Example:

More information, please follow sample code.

Returns:

Please check error section

4.4.3 FTNFC_disconnect

Synopsis:

public native static int FTNFC_disconnect ();

Parameters:

NULL

Description:

This function use to disconnect reader.

Example:

More information, please follow sample code.

Returns:

SUCCESS	Successful

4.4.4 FTNFCCardType

Synopsis:

public Card.CARD_TYPE FTNFC_cardType();

Parameters:

NULL

Description:

Return current card type, after FTNFC_connect to call

Example:

More information, please follow sample code.

Returns:

SUCCESS	Successful

4.4.5 GetCardInfoData

Synopsis:

public byte[] GetCardInfoData();

Parameters:

NULL

Description:

Return connected card information

Example:

More information, please follow sample code.

Copyright © Feitian Technologies Co., Ltd.
Website: www.FTsafe.com

Returns:
Reference error code section

Notice:
A: return null if without any card connect
B: If the card connected, then return byte array which describe card information

Card type	**Return data**				
Type A	0x0A	Sak	Uid_len	UID	
	Type A	1 byte	Length of card UID	Card UID	
Type B	0x0B,ATQB,0x04,PUPI				
	0x0B	ATQB	0x04	PUPI	
	Type B	1 byte(the first four bits means maximum frame length, after four bits means protocol type)	Length of PUPI	4 bytes PUPI data	
Felica card	0x0C	0x00	0x10	felica_id	pad_id
	Felica	1 byte reserve	16 bytes data	8 bytes felica id	8bytes pad id
Topaz card	0x0D	ATQA	id		
	Topaz	1 byte	Topaz card ID		

4.5 Mifare card section

4.5.1 GeneralAuthenticate

Synopsis:

public int GeneralAuthenticate(int blockNum, int keyType, byte[] key)

Parameters:

blockNum	IN	block number which will do operation
keyType	IN	key's type
key	IN	Key's byte code

Description:

To do authenticate for operation block

Example:

More information, please follow sample code.

Returns:

SUCCESS	Successful
Others	fail

4.5.2 ReadBinary

Synopsis:

public int ReadBinary(int blockNum, byte[] data, int size)

Parameters:

blockNum	IN	block number which will do operation
data	OUT	return data which will be read
size	IN	size of how many data will be read

Description:

This function use to read block data

Example:

More information, please follow sample code.

Returns:

SUCCESS	Successful
Others	fail

4.5.3 ClassicBlockInitial

Synopsis:

public int ClassicBlockInitial(int blockNum)

Parameters:

blockNum	IN	block number which will do operation

Description:

To do initial of specify block

Example:

More information, please follow sample code.

Returns:

SUCCESS	Successful
Others	fail

4.5.4 ClassicReadValue

Synopsis:

public int ClassicReadValue(int blockNum, int[] valueAmount);

Parameters:

blockNum	IN	block number which will do operation
valueAmount	OUT	output block value into array

Description:

To read block value from card

Example:

More information, please follow sample code.

Returns:

For the error code, please follow error section

4.5.5 ClassicStoreBlock

Synopsis:

public int ClassicStoreBlock(int blockNum, int valueAmount);

Parameters:

blockNum	IN	block number which will do operation
valueAmount	IN	output block value into array

Description:

To write value into block

Example:

More information, please follow sample code.

Returns:

For the error code, please follow error section

4.5.6 ClassicIncrement

Synopsis:

public int ClassicIncrement(int blockNum, int valueAmount);

Parameters:

blockNum	IN	block number which will do operation
valueAmount	IN	Plus the value of the required

Description:

Plus the value opeartion

Example:

More information, please follow sample code.

Returns:

For the error code, please follow error section

4.5.7 ClassicDecrement

Synopsis:

public int ClassicDecrement(int blockNum, int valueAmount);

Parameters:

blockNum	IN	block number which will do operation
valueAmount	IN	Minus the value of the required

Description:

Minus the value opeartion

Example:

More information, please follow sample code.

Returns:

For the error code, please follow error section

4.6 Reader (plug-in/out) monitor function

4.6.1 OnInsertHeadSet

Synopsis:
public void OnInsertHeadSet ();
Parameters:
NULL
Description:
When audio jack insert to Phone then will execution this function
Example:
More information, please follow sample code.

4.6.2 OnInsertHeadSet

Synopsis:
public void OnPullHeadSet ();
Parameters:
NULL
Description:
When audio jack plug out from Phone then will execution this function
Example:
More information, please follow sample code.

Notice:

Add the uses-permission:

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" /> <!-- Get phones status -->
<uses-permission android:name="android.permission.RECORD_AUDIO" /> <!-- Play voice -->
```

In the AndroidManifest.xml More information, please follow sample code folder which name is "AndroidManifest.xml".