API Interface Specification

Programmer's Reference Manual (Android)

V- 3.0.2

2020-10-28

Shenzhen Xinguodu Technology Co., Ltd Published

www.xinguodu.com

This document may adjust the technical errors that may be included without notice. Modifications to this document will be released in the next release. This document prohibits any form of reproduction or propagation without permission.

Version	Date	Editor	Description
V0.1	2016-08-12	Liu Ting	
V0.2	2016-9-12	Liu Ting	Startprint () function to modify the number of Parameters
V0.3	2016-10-12	Zhou Xiaoxin	Add the appendix Return Value that is refined into the class Description, as well as some function changes
V 0.4	2016-10-31	Zhou Xiaoxin	Modify the buzzer interface Modify EMV card module power up and down from the application layer control Modify PIN pad work Cadogan a key type
V 0.5	2016-11-11	Zhou Xiao Xin	Increase silent uninstall Interface
V 0.6	2016-11-25	Zhou Xin	EMV interface modification, the amount of correction, whether the use of electronic cash callback
V 0.7	2016-12-05	Zhou Xin	Print Increment Print One-dimensional code Two-dimensional code interface
V0.8	2016-12-08	Zhou Xiaoxin	Increase the camera scan interface
V1.0. 4	2017-01-13	Wang Hongyang	According to the reference document, organize the structure of the document
V2.0.1	2017-05-04	QiangFang	Increase DUKTP
V2.0.6	2018-01-04	Hassan	Add onSetAfterFinalSelectedAppResponse, and onAfterFinalSelectedApp method.
V2.0.7	2018-05-28	Hassan	Sync new methods
V2.0.9	2018-10-29	Hassan	1. OnEmvProcessListener add callback: onPrompt, onRemoveCard 2. EmvHandler add method: onSetRemoveCardResponse, onSetPromptResponse 3. Remove armeabi(include libnexgo_emvjni.so, libnexgo_gencode.so, libnexgo_sdkemvjni.so). 4. Change SDK package format: jar-→aar 5. CardReader add Felica method: setSupportFelica: if support Felica card

			setFelicaRequestCode, setFelicaSystemCode 6. CardReader add Read Mag Stripe original Track data method: setMagReaderRawData 7. Delete method loadKeyByCom and cancelLoadKey 8. Change others
V2.1.1	2019-01-24	Hassan	1. Fixed mag-stripe reader issue 2. Fixed emv onRequestAmount method 3. Other fix
V2.1.1	2019-03-18	ShaoPu	Add Desfire card API
V2.2.1	2019-06-20	Hassan	1. EMV add new funcion for Pure kernel 2. add new API for set pure kernel Aid and CapkEntity 3. add new API for set pure enable aid select first 4. fix custom layout pinpad issue 5. paypass add RRP function 6. add new API for get sigature statue . 7. add new API for security
V2.2.1	2019-09-20	Randall	1. Add scanner 2 for customized UI scan
V2.3.1	2019-12-20	Hassan	1. Add Mifare Ultralight API 2. Add DRL for paywave and amex
V2.3.3	2020-05-11	Hassan	1. Optimize MAC algorithm 2. Fix paypass RRP issue 3. Optimize emv flow 4. other issue fixed
V3.0.1	2020-06-11	Hassan	1. add emvHandler2 API 2. Fix Pure card terminal capability issue. 3. Remove the install and uninstall API
V3.0.2	2020-10-21	Hassan	1. Emv add new MB kernel 2. Compatible with N86 3. Remove the HSM APIs 4. other issue fixed

SOFTWARE LICENSE AGREEMENT

IMPORTANT: YOU SHOULD CAREFULLY READ ALL THE TERMS, CONDITIONS AND RESTRICTIONS OF THIS LICENSE AGREEMENT BEFORE INSTALLING THE SOFTWARE PACKAGE. YOUR INSTALLATION OF THE SOFTWARE PACKAGE PRESUMES YOUR ACCEPTANCE OF THE TERMS, CONDITIONS, AND RESTRICTIONS CONTAINED IN THIS AGREEMENT. IF YOU DO NOT AGREE WITH THESE TERMS, CONDITIONS, AND RESTRICTIONS, PROMPTLY RETURN THE SOFTWARE PACKAGE AND ASSOCIATED DOCUMENTATION TO THE ADDRESS ON THE FRONT PAGE OF THIS DOCUMENT, ATTENTION: CUSTOMER SUPPORT.

TERMS, CONDITIONS, AND RESTRICTIONS

Shenzhen Xinguodu Technology Co., Ltd (the "Licensor") owns and has the right to distribute the described software and documentation, collectively referred to as the "Software".

LICENSE: Licensor grants you (the "Licensee") the right to use the Software in conjunction with Xinguodu products. LICENSEE MAY NOT COPY, MODIFY, OR TRANSFER THE SOFTWARE IN WHOLE OR IN PART EXCEPT AS EXPRESSLY PROVIDED IN THIS AGREEMENT. Licensee may not decompile, disassemble, or in any other manner attempt reverse reverse engineer the Software.
Licensee shall not tamper with, bypass, or alter any security features of the software or attempt to do so.

TRANSFER: Licensee may not transfer the Software or license to the Software to another party without the prior written authorization of the Licensor. If Licensee transfers the Software without authorization, all rights granted under this Agreement are automatically terminated.

COPYRIGHT: The Software is copyrighted**.** Licensee may not copy the Software except for archival purposes or to load for business operations. All other copies of the Software are in violation of this Agreement.

TERM: This Agreement is in effect as long as Licensee continues the use of the Software. The Licensor also reserves the right to terminate this Agreement if Licensee fails to comply with any of the terms, conditions, or restrictions contained herein. Should Licensor terminate this Agreement due to Licensee's failure to comply, Licensee agrees to return the Software to Licensor. Receipt of returned Software by the Licensor shall mark the termination.

LIMITED WARRANTY: Licensor warrants to the Licensee that the disk(s) or other media on which the Software is recorded are free from defects in material or workmanship under normal use.
THE SOFTWARE IS PROVIDED AS IS. LICENSOR MAKES NO OTHER WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
Because of the diversity of conditions and PC hardware under which the Software may be used, Licensor does not warrant that the software will meet Licensee specifications or that the operation of the Software will be uninterrupted or free of errors.
IN NO EVENT WILL LICENSOR BE LIABLE FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE, OR INABILITY TO USE, THE SOFTWARE. Licensee's sole remedy in the event of a defect in material or workmanship is expressly limited to replacement of the Software disk(s), if applicable.

GOVERNING LAW: If any provision of this Agreement is found to be unlawful, void, or unenforceable, that provision shall be removed from consideration under this Agreement and will not affect the enforceability of any of the remaining provisions. This Agreement shall be governed by the Laws of the State of California and shall inure to the benefit of Shenzhen Xinguodu Technology Co., Ltd, its successors or assigns.

ACKNOWLEDGMENT: LICENSEE ACKNOWLEDGES THAT HE HAS READ THIS AGREEMENT, UNDERSTANDS ALL OF ITS TERMS, CONDITIONS, AND RESTRICTIONS, AND AGREES TO BE BOUND BY THEM. LICENSEE ALSO AGREES THAT THIS AGREEMENT SUPERSEDES ANY AND ALL VERBAL AND WRITTEN COMMUNICATIONS BETWEEN LICENSOR AND LICENSEE OR THEIR ASSIGNS RELATING TO THE SUBJECT MATTER OF THIS AGREEMENT.

QUESTIONS REGARDING THIS AGREEMENT SHOULD BE ADDRESSED IN WRITING TO Shenzhen Xinguodu Technology Co., Ltd, ATTENTION: CUSTOMER SUPPORT, AT THE ADDRESS LISTED IN THIS DOCUMENT OR E-MAILED TO SUPPORT@XINGUODU.COM.

Table of Content

1 Introduction

This document helps developers develop third-party applications on N5 devices. The company provides API interface in the form of jar package, based on the API interface, the developer can conveniently and efficiently develop third-party applications to meet the personalized needs of developers.

1.1 Demo Description

The demo program demonstrates how to use the aar package to call API's various interfaces to meet the needs of the developer.

1.2 The Term

SdkResult returns the value class. The fields in the class define all the Return Values in the document. All fields are described in the Appendix.

1.3 SDK Content

File	Description
nexgo-smart-sdk-vx.x.x.aar	aar package API interface

1.4 System Requirements

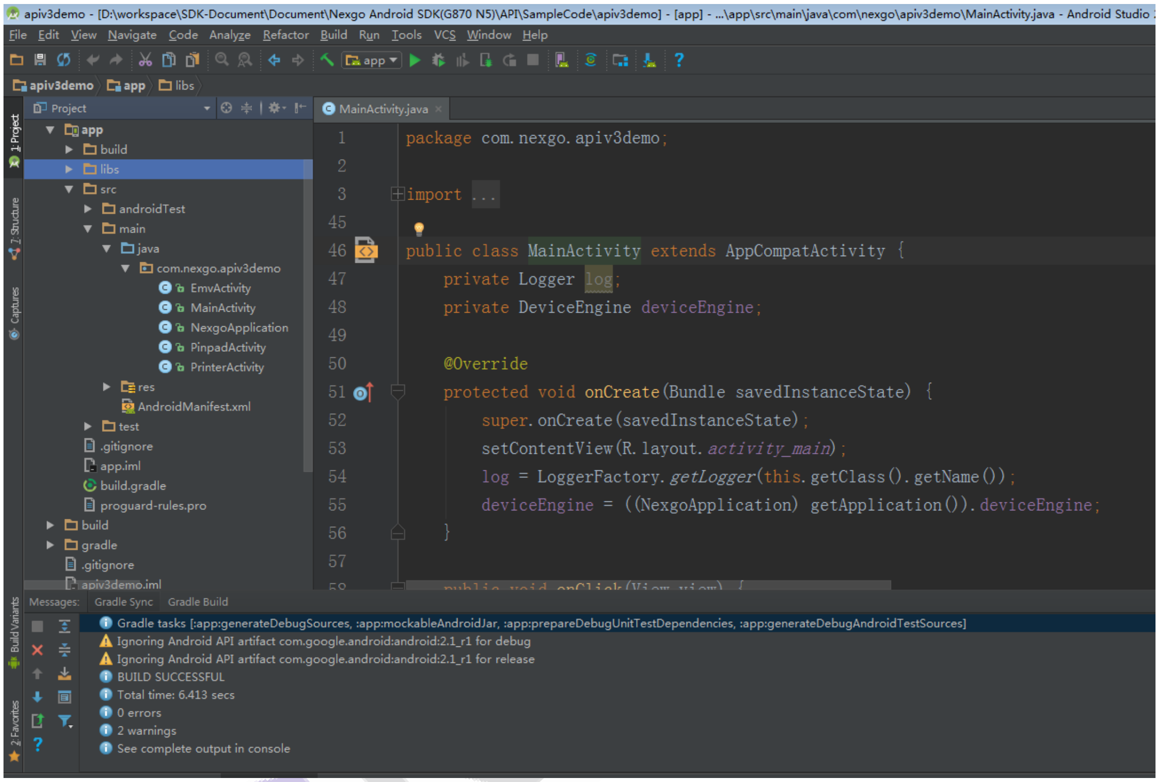
Development environment: Android Studio 2.0 or later

N5 Operating System Version: 5.1.1

2 How to Create a Project

To add the jar package to the developer project, follow these steps:

1) With Android studio to create or open the customer's project.

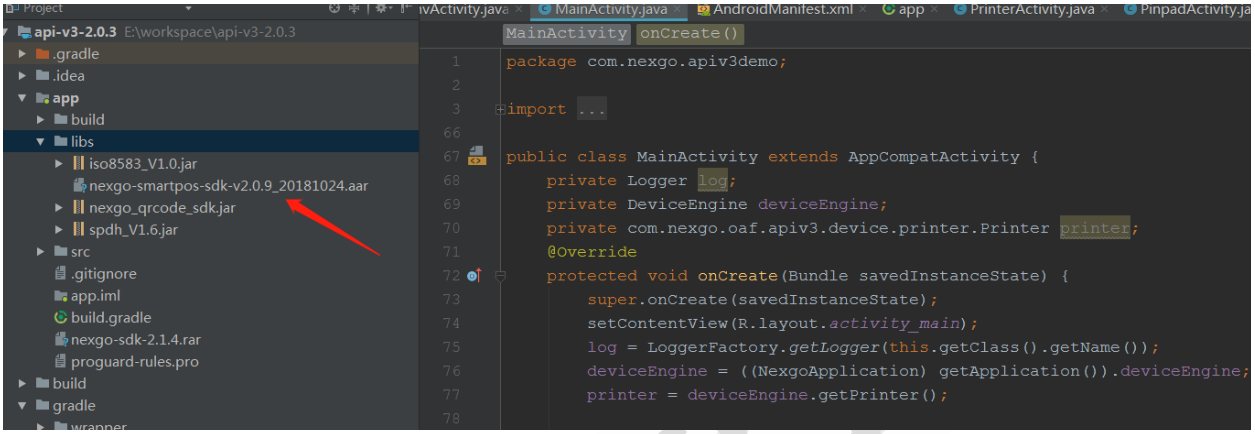


```
package com.nexgo.apiv3demo;

import ...

public class MainActivity extends AppCompatActivity {
    private Logger log;
    private DeviceEngine deviceEngine;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        log = LoggerFactory.getLogger(this.getClass().getName());
        deviceEngine = ((NexgoApplication) getApplication()).deviceEngine;
    }

    public void onClick(View view) {
```

```
Gradle tasks [:app:generateDebugSources, :app:mockableAndroidJar, :app:prepareDebugUnitTestDependencies, :app:generateDebugAndroidTestSources]
Ignoring Android API artifact com.google.android:android:2.1_r1 for debug
Ignoring Android API artifact com.google.android:android:2.1_r1 for release
BUILD SUCCESSFUL
Total time: 6.413 secs
0 errors
2 warnings
See complete output in console
```

2) Copy nexgo-smart-sdk-vx.x.x.aar to libs / directory, configure project build.gradle to load aar package.

```
repositories{
    flatDir{
        dirs 'libs'
    }
}
 dependencies{
complie(name: 'nexgo-smartpos-sdk-vx.x.x', ext: 'aar')
}
```



```
package com.nexgo.apiv3demo;

import ...

public class MainActivity extends AppCompatActivity {
    private Logger log;
    private DeviceEngine deviceEngine;
    private com.nexgo.oaf.apiv3.device.printer.Printer printer;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        log = LoggerFactory.getLogger(this.getClass().getName());
        deviceEngine = ((NexgoApplication) getApplication()).deviceEngine;
        printer = deviceEngine.getPrinter();

```

3) Get the global object of the device operation.

   DeviceEngine deviceEngine = APIProxy getDeviceEngine (this) ;

4) Get the object of the device sub-module, and operate the interface

   For example: get buzzer operation object, ring 500 milliseconds.

   final Beeper beeper = deviceEngine getBeeper () .;

   beeper.beep (500);

3 Class methods

The following is divided into 10 categories, 5 global methods. First, get the object of each class, then call the member method in the class.

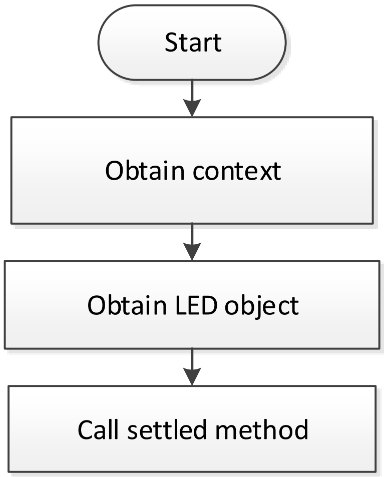
3.1 LED class

LED class is responsible for managing POS LED lights.

Get the object of the LED class:

LEDDriver ledDriver = deviceEngine.getLEDDriver ();

This module operates using the basic flow chart:



3.1.1 SetLed

Drive POS red, green, yellow, blue light switch.

Public void setLed (LightModeEnum light, boolean isOn);

Parameters:

Parameter	Description
light	Enumerated type red, green, yellow, blue LED lights
isOn	True: on, false: off

LightModeEnum

Enumeration Name	Description
RED	Red light
GREEN	Green light
YELLOW	Yellow light
BLUE	Blue light

Return Value: None

3.2 Printer class

The printer class is responsible for managing POS printers.

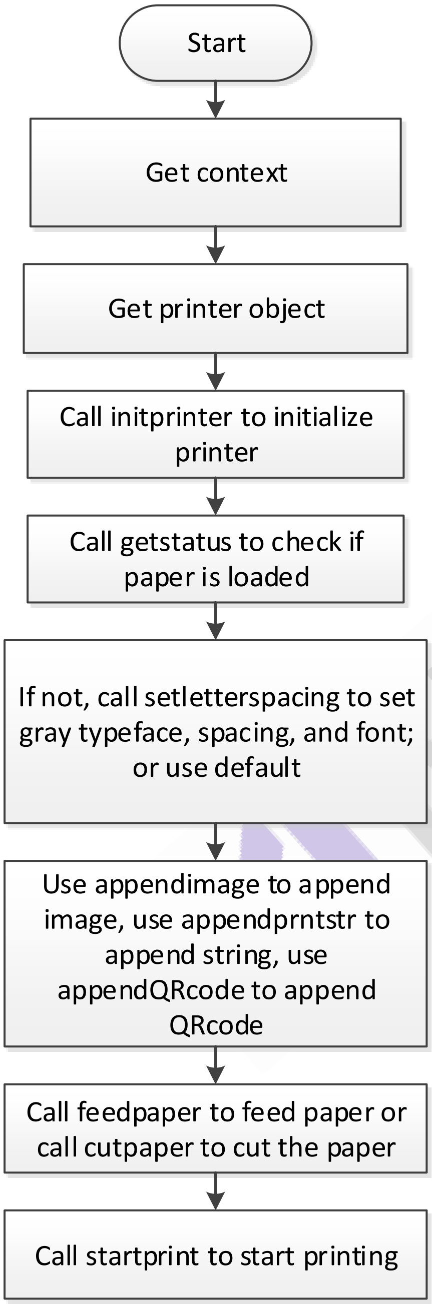
Get the object of the printer class:

Printer printer = deviceEngine getPrinter (). ;

The following table shows the Return Values supported by the method of the printer class:

Constant Name	Constant Value	Description
Printer_Base_Error	-1000	
Printer_Print_Fail	Printer_Base_Error – 1	Print failed
Printer_AddPrnStr_Fail	Printer_Base_Error – 2	Setting string buffer failed
Printer_AddImg_Fail	Printer_Base_Error – 3	Setting image buffer failed
Printer_Busy	Printer_Base_Error – 4	The printer is busy
Printer_PaperLack	Printer_Base_Error – 5	The printer is out of paper
Printer_Wrong_Package	Printer_Base_Error – 6	Print packet is wrong
Printer_Fault	Printer_Base_Error – 7	Printer failure
Printer_TooHot	Printer_Base_Error – 8	The printer is overheating
Printer_UnFinished	Printer_Base_Error – 9	The print is not complete
Printer_NoFontLib	Printer_Base_Error – 10	The printer does not have a font
Printer_OutOfMemory	Printer_Base_Error – 11	The packet is too long
Printer_Other_Error	Printer_Base_Error-999	Other exception error

This module operates using the basic flow chart:


Start
Get context
Get printer object
Call initprinter to initialize printer
Call getstatus to check if paper is loaded
If not, call setletterspacing to set gray typeface, spacing, and font; or use default
Use appendimage to append image, use appendprntstr to append string, use appendQRcode to append QRcode
Call feedpaper to feed paper or call cutpaper to cut the paper
Call startprint to start printing

3.2.1 initPrinter

Initialize the printer.

public int initPrinter ();

Parameters: None

Return Value:

SdkResult.Success success

3.2.2 getStatus

Get the printer status.

Public int getStatus ();

Parameters: None

Return Value:

SdkResult.Success	print successful
SdkResult. Printer_UnFinished	print is not complete
SdkResult. Printer_PaperLack	printer is out of paper
SdkResult. Printer_Too_Hot	printer is overheating
SdkResult. Printer_Fail	print failed
SdkResult.Fail	other errors

3.2.3 appendImage

Append bitmap.

Public int appendImage (Bitmap bitmap, AlignEnum align);

Parameters:

Parameter	Description
bitmap	Bitmap data
align	Enumerated type of alignment

AlignEnum

Enumeration Name	Description
LEFT	Left alignment

RIGHT	Right alignment
CENTER	Centered

Return Value:

SdkResult.Success success

SdkResult.Printer_AddImg_Fail additional picture failure

3.2.4 appendPrnStr

Add text.

```
Public int appendPrnStr (String text, int fontsize, AlignEnum align,
                                            Boolean isBoldFont);
```

Parameters:

Parameter	Description
text	The string data to be added
fontSize	Font Size small: 16; normal: 20; large: 24; x-large: 32
align	Enumerated type of alignment
isBoldFont	Whether bold, true: yes, false: no

AlignEnum

Enumeration Name	Description
LEFT	Left alignment
RIGHT	Right alignment
CENTER	Centered

Return Value:

SdkResult.Success success

SdkResult.Printer_Wrong_Package print packet format error

SdkResult.Printer_AddPrnStr_Fail string buffer is set to fail

3.2.5 appendPrnStr

Add text.

```
public int appendPrnStr(String text,int fontsize,AlignEnum align,
```

boolean isBoldFont, LineOptionEntity ops);

Parameters:

Parameter	Description
text	The string data to be added
fontSize	Font Size small: 16; normal: 20; large: 24; x-large: 32
align	Enumerated type of alignment
isBoldFont	Whether bold, true: yes, false: no
ops	LineOptionEntity :additional option

AlignEnum

Enumeration Name	Description
LEFT	Left alignment
RIGHT	Right alignment
CENTER	Centered

LineOptionEntity

attribute	Description
boolean isUnderline	Print underline: true: yes; false:no
int marginLeft	Left margin size

Return Value:

SdkResult.Success success

SdkResult.Printer_Wrong_Package print packet format error

SdkResult.Printer_AddPrnStr_Fail string buffer is set to fail

3.2.6 appendPrnStr

Append string both sides at the same time.

Public int appendPrnStr (String leftText, String rightText, int fontsize,
AlignEnum align, Boolean isBoldFont);

Parameters:

Parameter	Description
leftText	Left alignment data
rightText	Right alignment data
fontsize	Font Size small: 16; normal: 20; large: 24; x-large: 32

align	Enumerated type of alignment
isBoldFont	Whether bold, true: yes, false: no

AlignEnum

Enumeration Name	Description
LEFT	Left alignment
RIGHT	Right alignment
CENTER	Centered

Return Value:

SdkResult.Success success

SdkResult.Printer_Wrong_Package print packet format error

SdkResult.Printer_AddPrnStr_Fail string buffer is set to fail

3.2.7 appendPrnStr

Append string both sides at the same time.

```
public int appendPrnStr(String leftText,String rightText,int fontsize,
                                   Boolean isBoldFont, LineOptionEntity ops);
```

Parameters：

Parameter	Description
leftText	Left alignment data
rightText	Right alignment data
fontsize	Font Size small: 16; normal: 20; large: 24; x-large: 32
align	Enumerated type of alignment
ops	LineOptionEntity :additional option

LineOptionEntity

attribute	Description
boolean isUnderline	Print underline: true: yes; false:no
int marginLeft	Left margin size

Return Value:

SdkResult.Success success

SdkResult.Printer_Wrong_Package print packet format error

SdkResult.Printer_AddPrnStr_Fail string buffer is set to fail

3.2.8 appendBarcode

Append barcode.

public int appendBarcode (String content, BarcodeFormatEnum format, int width, int height,
AlignEnum align);

Parameters:

Parameter	Description
content	Generates Barcode data
format	Barcode format
width	Generate Barcode width for printing; range 1-384
height	Generates Barcode height for printing; ranges greater than 0
align	Alignment

BarcodeFormatEnum

Format Name	Description
AZTEC	
CODABAR	
CODE_39	
CODE_93	
CODE_128	
DATA_MATRIX	
EAN_8	
EAN_13	
ITF	
MAXICODE	
PDF_417	
QR_CODE	
RSS_14	
RSS_EXPANDED	
UPC_A	

UPC_E	
UPC_EAN_EXTENSION	

AlignEnum

Enumeration Name	Description
LEFT	Left alignment
RIGHT	Right alignment
CENTER	Centered

Return Value:

SdkResult.Success success

SdkResult.Printer_AddImg_Fail failure

3.2.9 appendQRCode

Append QR code.

public int appendQRcode (String content, int width, int height, AlignEnum align);

Parameters:

Parameter	Description
content	Generates QR code data
width	Generates QR code printing width; range 1-384
height	Generates QR code height for printing; ranges greater than 0
align	Alignment

AlignEnum

Enumeration Name	Description
LEFT	Left alignment
RIGHT	Right alignment
CENTER	Centered

Return Value:

SdkResult.Success success

SdkResult.Printer_AddImg_Fail failure

3.2.10 appendQRcode

Append QR code.

```
public int appendQRcode(String content,int height, int version,
                                                int level,AlignEnum align);
```

Parameters：

Parameter	Description
content	Generates QR code data
height	Generates QR code height for printing; ranges greater than 0
version	QR code version is 1-40
level	Error correction level, from low to high, 0-3
align	Alignment

AlignEnum

Enumeration Name	Description
LEFT	Left alignment
RIGHT	Right alignment
CENTER	Centered

Return Value:

SdkResult.Success success

SdkResult.Printer_AddImg_Fail failure

3.2.11 feedPaper

Feed paper.

```
Public void feedPaper (int value);
```

Parameters:

Parameter	Description
value	Paper length is in pixels; range of greater than or equal to 0; if the user has not set, the default value equals 0

Return Value: None

3.2.12 cutPaper

Feed paper, the default is to feed to the end of print section.

Public void cutPaper ();

Parameters: None

Return Value: None

3.2.13 startPrint

Start printing.

public int startPrint (boolean rollPaperEnd, OnPrintListener listener);

Parameters:

Parameter	Description
rollPaperEnd	Advance to the end of the paper automatically; true: yes, false: no
listener	The callback interface after printing is complete

Return Value:

SdkResult.Success operation is successful; listener can successfully callback

SdkResult.Printer_Busy printer is busy

SdkResult.Printer_Print_Fail print data is empty

SdkResult.Param_In_Invalid illegal Parameter

3.2.14 setLetterSpacing

Set the spacing between the print order lines.

public void setLetterSpacing (int value);

Parameters:

Parameter	Description
value	Line spacing is in pixels; the default value equals 4

Return Value: None

3.2.15 setGray

Set the grayscale.

public void setGray (GrayLevelEnum level);

Parameters:

Parameter	Description
level	Establish gray value; if the user has not set, the default value is LEVEL_0. The higher the grayscale, the darker the print font, the slower the print speed.

GrayLevelEnum

Enumeration Name	Description
LEVEL_0	Primary grayscale
LEVEL_1	Secondary grayscale
LEVEL_2	Tertiary grayscale

Return Value: None

3.2.16 setTypeface

Set the font type.

public void setTypeface (Typeface typeface);

Parameters:

Parameter	Description
typeface	Android SDK Typeface font type; user can use the default value:Typeface. DEFAULT

Return Value: None

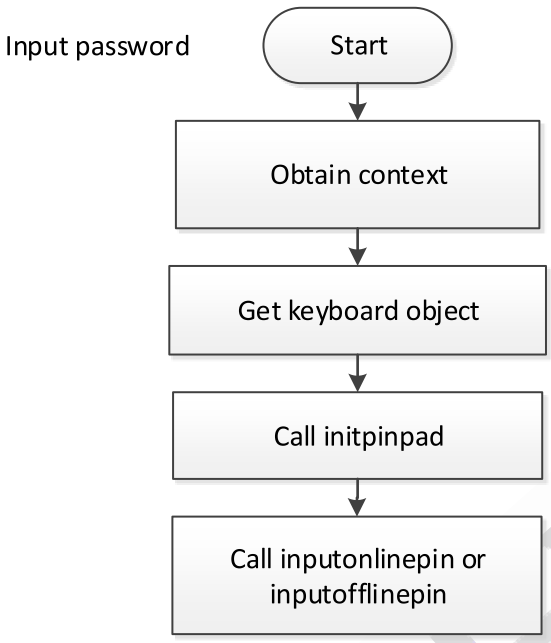
3.3 Pinpad Class

The password keyboard class is responsible for managing the POS password keyboard.

Get the object of the password keyboard class:

. PinPad pinpad = deviceEngine getPinPad () ;

This module operates using the basic flow chart:



3.3.1 initPinPad

Initialize the password keyboard.

Public int initPinPad (PinPadTypeEnum ppType);

Parameters:

Parameter	Description
ppType	PinPadTypeEnum Enum type Password keyboard type; currently only supports built-in password keyboard

PinPadTypeEnum

Enumeration Name	Description
INTERNAL	Built-in password keyboard
EXTERNAL	External password keyboard

Return Value:

SdkResult.Success success

SdkResult.Fail fail

3.3.2 setAlgorithmMode

Set PinPad work in DUKPT model or classical model. Default is DES model.

Public void setAlgorithmMode(AlgorithmModeEnum algMode);

AlgorithmModeEnum

Enumeration Name	Description
DES	DES model
SM4	PBOC use it
DUKPT	DUKPT model

3.3.3 setCipherMode

Set PinPad Cipher mode, default is ECB mode.

Public void setCipherMode (CipherModeEnum cipherMode);

CipherModeEnum

Enumeration Name	Description
ECB	
CBC	

3.3.4 setCipherInitializationVector

Set PinPad Cipher iv, it is used for CBC Cipher mode.

Public void setCipherInitializationVector (byte[] iv);

3.3.5 setPinKeyboardMode

Set password keyboard mode, default mode is random password keyboard.

public void setPinKeyboardMode(PinKeyboardModeEnum keyboardMode);
Parameters:

Parameter	Description

keyboardMode	Mode, default value RANDOM

PinKeyboardModeEnum

Enumeration Name	Description
RANDOM	Random key board
FIXED	Fixed key board

Return Value:

None

3.3.6 writeMKey

Inject the master key(plaintext key)

Public int writeMKey (int mKeyIdx, byte [] keyData, int keyDataLen);

Parameters:

Parameter	Description
mKeyIdx	Master Key Index 0-99
keyData	Plaintext master key data
keyDataLen	The length of the plain key range: 8,16,24

Return Value:

SdkResult.Success success

SdkResult.Param_In_InValid Parameter is not legitimate

SdkResult.PinPad_Dstkey_Idx_Error wrong key index

SdkResult.PinPad_Key_Len_Error wrong key length

SdkResult.Fail other errors

3.3.7 writeMKey

Inject the master key(ciper key)

Public int writeMKey (int mKeyIdx, byte [] keyData, int keyDataLen, int decMKeyIdx);

Parameters:

Parameter	Description
mKeyId	Master Key Index 0-99

keyData	Ciphertext master key data
keyDataLen	Ciphertext master key length should be not less than 8, and must be a multiple of 8 bytes
decMKeyIdx	Decrypt the master key index 0-99

Return Value:

SdkResult.Success success

SdkResult.Param_In_InValid Parameter is not legitimate

SdkResult.PinPad_Dstkey_Idx_Error wrong key index

SdkResult.PinPad_Key_Len_Error wrong key length

SdkResult.PinPad_No_Key_Error key does not exist

SdkResult.Fail other errors

3.3.8 isKeyExist

Whether the master key exists. It is only suitable for TMK/session key, do not suitable for DUKPT.

Public boolean isKeyExist (int mKeyIdx);
Parameters:

Parameter	Description
mKeyId	Master Key Index 0-99

Return Value:

True success, key exist

False failure, key not exist or error.

3.3.9 calcWKeyKCV

Calculate the work key KCV (check value).

Public byte [] calcWKeyKCV (int mKeyIdx, WorkKeyTypeEnum wKeyType);

Parameters:

Parameter	Description
mKeyIdx	Master Key Index 0-99

wKeyType	Working key type

WorkKeyTypeEnum

Enumeration Name	Description
PINKEY	PIN key
MACKEY	MAC key
TDKEY	Track key
ENCRYPTIONKEY	Data encryption key, providing encryption and decryption

Return Value:

Success, return an array of check values

Failure, returning null

3.3.10 writeWKey

Inject work key.

Public int writeWKey (int mKeyIdx, WorkKeyTypeEnum wKeyType, byte [] keyData, int keyDataLen);

Parameters:

Parameter	Description
mKeyIdx	Master key index number 0-99
wKeyType	Working key type
keyData	Working key cipher text data
keyDataLen	Working key cipher text length

WorkKeyTypeEnum

Enumeration Name	Description
PINKEY	PIN key
MACKEY	MAC key
TDKEY	Track key
ENCRYPTIONKEY	Data encryption key, providing encryption and decryption

Return Value:

SdkResult.Success success

SdkResult.Param_In_InValid Parameter is invalid

SdkResult.PinPad_Dstkey_Idx_Error wrong key index object; not within the scope

SdkResult.PinPad_Key_Len_Error wrong key length

SdkResult.Fail other errors

3.3.11 isKeyExist

Whether the work key exists. It is only suitable for TMK/session key, do not suitable for DUKPT.

Public boolean isKeyExist (int mKeyIdx, WorkKeyTypeEnum wKeyType);
Parameters:

Parameter	Description
mKeyId	Master Key Index 0-99
wKeyType	Working key type

WorkKeyTypeEnum

Enumeration Name	Description
PINKEY	PIN key
MACKEY	MAC key
TDKEY	Track key
ENCRYPTIONKEY	Data encryption key, providing encryption and decryption

Return Value:

True success ,work key exist

False failure, work key not exist, or error

3.3.12 calcByWkey

public byte[] calcByWKey(int mKeyIdx, WorkKeyTypeEnum wKeyType, byte[] data, int dataLen, CalcModeEnum calcMode);
Parameters:

Parameter	Description
mKeyIdx	Master Key Index 0-99
wKeyType	Working key type
data	Input data
dataLen	Data length

calcMode	encryption or decryption

CalcModeEnum

Enumeration Name	Description
ENCRYPT	ENCRYPT mode
DECRYPT	DECRYPT mode

3.3.13 desByWKey

Work key encryption and decryption.

Public byte [] desByWKey (int mKeyIdx, WorkKeyTypeEnum wKeyType, byte [] data, int dataLen, DesKeyModeEnum keyMode, CalcModeEnum calcMode);

Parameters:

Parameter	Description
mKeyIdx	Master Key Index 0-99
wKeyType	Working key type
data	Input data
dataLen	Data length
keyMode	Key Mode
calcMode	Calc Mode

WorkKeyTypeEnum

Enumeration Name	Description
PINKEY	PIN key
MACKEY	MAC key
TDKEY	Track key
ENCRYPTIONKEY	Data encryption key, providing encryption and decryption

DesKeyModeEnum

Enumeration Name	Description
KEY_ALL	
KEY_FIRST	Specify the type of algorithm, key double length is used to do the first 8 bytes DES operation
KEY_LAST	Specify the type of algorithm, key double length is used to do the last 8 bytes DES operation

CalcModeEnum

Enumeration Name	Description
ENCRYPT	Encrypt
DECRYPT	Decrypt

Return Value:

Success returns the computed array

Failure, returns null

3.3.14 encryptTrackData

Use TDK work key to Encrypt track data.

public byte[] encryptTrackData(int mKeyId, byte[] trackData, int trackDataLen);

Parameters:

Parameter	Description
mKeyId	Master key index 0-99
trackData	Track data
trackDataLen	Len of track data

Return Value:

Success returns the computed array

Failure, returns null

3.3.15 calcMac

Use TAK work key to Calculate MAC.

Public byte [] calcMac (int mKeyIdx, MacAlgorithmModeEnum macAlgMode, byte [] data);

Parameters:

Parameter	Description
mKeyIdx	Master Key Index 0-99
macAlgMode	MAC algorithm approach
data	Input data

MacAlgorithmModeEnum

Enumeration Name	Description
ECB	ECB Algorithm
X99	ANSI X9.9 Encryption Algorithm
X919	ANSI X9.19 Encryption Algorithm

Return Value:

Success returns the computed array

Failure, returning null

3.3.16 calcMac

Calculate MAC.

Public byte [] calcMac (int mKeyIdx, MacAlgorithmModeEnum macAlgMode, DesKeyModeEnum keyMode , byte [] data);

Parameters:

Parameter	Description
mKeyIdx	Master Key Index 0-99
macAlgMode	MAC algorithm approach
desAlgMode	Algorithm type
data	Input data

MacAlgorithmModeEnum

Enumeration Name	Description
ECB	ECB Algorithm
X99	ANSI X9.9 Encryption Algorithm
X919	ANSI X9.19 Encryption Algorithm

DesKeyModeEnum

Enumeration Name	Description
KEY_ALL	
KEY_FIRST	Do des with the first 8 bytes of the key
KEY_LAST	Do des with the last 8 bytes of the key

Return Value:

Success returns the computed array

Failure, returning null

3.3.17 calcMac(DUKPT)

byte[] calcMac(int mKeyIdx, MacAlgorithmModeEnum macAlgMode, DukptKeyModeEnum keyMode, byte[] data);

Parameters:

Parameter	Description
mKeyId	Key Index 0-19(DUKPT only support 0-19 key index)
macAlgMode	Mac Alg mode
keyMode	Key mode
data	Data, lack of an integer multiple of 8, after the meeting 0x00 or …the fill data is decided by application

MacAlgorithmModeEnum

Enumeration Name	Description
ECB	
CBC	
X919	
MAC9606	

DukptKeyModeEnum

Enumeration Name	Description
REQUEST	Request mode
RESPONSE	Response mode

3.3.18 encryptByMKey

Master key encryption.

Public byte [] encryptByMKey (int mKeyId, byte [] data, int dataLen);

Parameters:

Parameter	Description
mKeyId	Master Key Index 0-99
data	Data, lack of an integer multiple of 8, after the meeting 0x00
dataLen	Length, maximum 1024 bytes
desAlgMode	DES algorithm type

DesAlgorithmModeEnum

Return Value:

Success returns the computed array

Failure, returning null

3.3.19 setPinpadLayout

Set the password keyboard layout. After this method is called, when inputOnlinePin or inputOfflinePin is called , the layout of the password keyboard will be drawned by the app layer itself, without using the system default password keyboard interface.

public byte[] setPinpadLayout(PinpadLayoutEntity pinpadLayout);

Parameters:

Parameter	Description
pinpadLayout	Coordinates of 10 digital keys and 3 function keys

PinpadLayoutEntity

attribute	Description
Rect key1	The coordinate of the number key "1"
Rect key2	The coordinate of the number key "2"
Rect key3	The coordinate of the number key "3"
Rect key4	The coordinate of the number key "4"
Rect key5	The coordinate of the number key "5"
Rect key6	The coordinate of the number key "6"
Rect key7	The coordinate of the number key "7"
Rect key8	The coordinate of the number key "8"
Rect key9	The coordinate of the number key "9"
Rect key10	The coordinate of the number key "0"

Rect keyCancel	The coordinate of the key "cancel"
Rect keyConfirm	The coordinate of the key "confirm"
Rect keyClear	The coordinate of the key "clear"

Return Value:

Byte[]	Returns 0-9 digits for 10 key successfully
null	Failed

3.3.20 inputOnlinePin

Enter the online PIN.

Public int inputOnlinePin (int [] pinLen, int timeout, byte [] panBlock, int mKeyId,
PinAlgorithmModeEnum pinAlgMode, OnPinPadInputListener listener);

Parameters:

Parameter	Description
pinLen	The length of the support, such as{0x00,0x04,0x05,0x06,0x07,0x08,0x09,0x0a, 0x0b, 0x0c}
timeout	Enter a timeout in seconds; recommended value 60
panBlock	Card number, asc coding
mKeyId	Master Key Index 0-99(if DUKPT, is 0-19)
pinAlgMode	PIN encryption algorithm mode
listener	Monitor callback interface

PinAlgorithmModeEnum

Enumeration Name	Description
ISO9564FMT1	Currently only supports the standard encryption algorithm CUP pin
ISO9564FMT2	Currently only supports the standard encryption algorithm CUP pin
ISO9564FMT3	Currently only supports the standard encryption algorithm CUP pin

Return Value:

SdkResult.Success successful execution listener callback interface

SdkResult.Param_In_InValid illegal Parameter

SdkResult.Fail other errors

3.3.21 inputOfflinePin

Enter the offline PIN(offline plaintext pin, or offline cipher pin).

Public int inputOfflinePin (int [] pinLen, int timeout, OnPinPadInputListener listener);
Parameters:

Parameter	Description
pinLen	The length of the support, such as{0x00,0x04,0x05,0x06,0x07,0x08,0x09,0x0a, 0x0b, 0x0c}
timeout	Enter a timeout in seconds; recommended value 60
listener	Monitor callback interface

Return Value:

SdkResult.Success successful execution listener callback interface

SdkResult.Param_In_InValid illegal Parameter

SdkResult.Fail other errors

3.3.22 isInputting

Whether the keyboard is typing.

Public boolean isInputting ();
Parameters: None

Return Value:

True success

False failure

3.3.23 cancelInput

Cancel the keyboard input.

Public void cancelInput ();
Parameters: None

Return Value: None

3.3.24 format

Format the key area.

Public boolean format ();
Parameters: None

Return Value:

True success

False failure

3.3.25 deleteMKey

Clear the master key.

Public boolean deleteMKey (int mKeyId);
Parameters:

Parameter	Description
mKeyId	Master Key Index 0-99

Return Value:

True success

False failure

3.3.26 dukptKeyInject

Inject BDK(or IPEK) and KSN for DUKPT.

Public int dukptKeyInject(int mKeyIdx, DukptKeyTypeEnum keyType, byte[] keyData, int keyDataLen, byte[] ksn);
Parameters:

Parameter	Description
mKeyIdx	Key Index 0-19
keyType	Key type
keyData	BDK
keyDataLen	BDK length
ksn	KSN

DukptKeyTypeEnum

Enumeration Name	Description

BDK	BDK
IPEK	IPEK

Return value:

SdkResult.Success,

SdkResult.Fail,

SdkResult.PinPad_KeyIdx_Error,

SdkResult.Param_In_Invalid

3.3.27 dukptKsnIncrease

Use it to increase ksn, otherwise the ksn will not change.

Public void dukptKsnIncrease(int mKeyIdx);
Parameters:

Parameter	Description
mKeyIdx	Key Index 0-19

3.3.28 dukptCurrentKsn

Get current Ksn value.

Public byte[] dukptCurrentKsn(int mKeyIdx);
Parameters:

Parameter	Description
mKeyIdx	Key Index 0-19

3.3.29 dukptEncrypt

Encrypt data in dukpt model.

Public byte[] dukptEncrypt(int mKeyIdx, DukptKeyModeEnum keyMode, byte[] data, int dataLen);
Parameters:

Parameter	Description
mKeyIdx	Key Index 0-19
keyMode	Encrypt model
data	Encrypt data

dataLen	Encrypt data's length

DukptKeyModeEnum

Enumeration Name	Description
REQUEST	
RESPONSE	

Return value:

Bytes Array,

Null

3.3.30 dukptEncrypt

Encrypt data in dukpt model.

Public byte[] dukptEncrypt(int mKeyIdx, DukptKeyModeEnum keyMode, byte[] data, int dataLen, DesAlgorithmModeEnum desMode, byte[] iv);

Parameters:

Parameter	Description
mKeyIdx	Key Index 0-19
keyMode	Encrypt model
data	Encrypt data
dataLen	Encrypt data's length
desMode	Use ECB or CBC
iv	Iv for CBC mode

DukptKeyModeEnum

Enumeration Name	Description
REQUEST	
RESPONSE	

DesAlgorithmModeEnum

Enumeration Name	Description
ECB	
CBC	

Return value:

Bytes Array,

Null

3.4 Scaner#1(default UI)

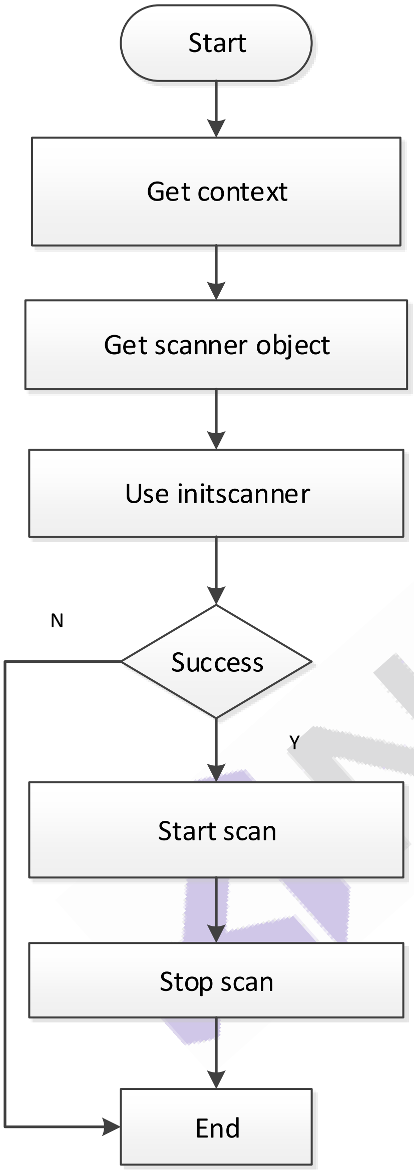
Camera scan code class is responsible for managing POS camera; must be initialized before use.

Get the object of the camera scan class:
Scanner scanner = deviceEngine getScanner();

The following table shows the Return Values supported by the method of the camera sweep class:

Constant Name	Constant Value	Description
Scanner_Base_Error	-2000	
Scanner_Customer_Exit	Scanner_Base_Error - 1	Active user exit
Scanner_Other_Error	Scanner_Base_Error - 2	Scan code fails

This module operates using the basic flow chart:



3.4.1 initScanner

Initialize the scan configuration.

Public int initScanner (ScannerCfgEntity cfgEntity, OnScannerListener listener);

Parameters:

Parameter	Description
cfgEntity	Initialize the configuration
listener	Callback interface

ScannerCfgEntity

Attributes	Description
boolean isUsedFrontCcd	Whether to use the front camera, if only back camera, then open the back camera by default
boolean isBulkMode	Whether continuous scan mode, open the scan after the success of the scan does not exit the interface
int interval	Continuous scan code interval, in milliseconds; default 1000
boolean isAutoFocus	Whether it is auto focus
boolean isNeedPreview	Whether it is need pre-view
Bundle mBundle	Use bundle to transfer parameter to customized the Scanner UI
Key	Description
boolean showBar	If show Bar
boolean showBack	Whether show the back button
boolean showTitle	Whether show the Title text
boolean showSwitch	Whether show the button for switching front and back camera
boolean showMenu	Whether show the Menu
String Title	Customized the title text
int TitleSize	The size of the Title text
string ScanTip	Customized the Scan tip text
int TipSize	The size of the tip text

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.4.2 startScan

start scan

Public int startScan (int timeout, OnScannerListener listener);

Parameters:

Parameter	Description
timeout	Scan code timeout in seconds; recommended value 60
listener	Callback interface

Return Value:

SdkResult.Sucess success

SdkResult.Fail failure

SdkResult.Param_In_Invalid illegal Parameter

3.4.3 stopScan

Stop scanning.

Public void stopScan();

Parameters: None

Return Value: None

3.4.4 decode

decode the image

public String decode(byte[] imageData, int imageWidth, int imageHeight);

Parameters：

Parameter	Description
imageData	Image, date type is YUV420SP
imageWidth	Image width
imageHeight	Image height

Return Value:

Failed : None

Success: decode result

3.5 Scanner#2(customizable UI)

The scanning UI can be customized. For details, see demo

Get the object of camera code scanning class:

Scanner scanner = deviceEngine.getScanner2() ;

3.5.1 initScanner

Initialize scan configuration

public void initScanner(ScannerCfgEntity cfgEntity, Set<SymbolEnum> enableSymbols);

Parameters:

Parameters:	Description
cfgEntity	Initialize the configuration
enableSymbols	Set supported code type

ScannerCfgEntity

Attributes	Description
boolean isUsedFrontCcd	Whether to use the front camera or not. If only the back camera is used, the back camera will be turned on by default

boolean isBulkMode	Continuous code scanning mode. If it is enabled, the code scanning interface will not exit after the code scanning succeeds
int interval	Continuous code scanning interval, unit: Ms default value: 1000
boolean isAutoFocus	Auto focus or not
boolean isNeedPreview	Preview required, default required
Bundle mBundle	User defined interface display settings, you can set the following table key values through bundle. Key Type description showBar boolean Show title bar or not BarColor int Title bar background color showBack boolean Show back button or not showTitle boolean Display title text or not showSwitch boolean is front/back camera switch button displayed showMenu boolean Show menu or not Title String Custom title text TitleSize int Title Text Size TitleColor int Title Text Color MaskColor int Preview mask color AngleColor int Color of four corners of code box FrameColor int Frame color SlideColor int Scanline color ScanTip int Custom prompt text

	TipColor int Prompt text color TipSize int Prompt text size Pendant String Image mount path

return value : none

3.5.2 getBestPreviewSize

Get the best preview resolution

public Size getBestPreviewSize();

parameter：none

return value：Size

3.5.3 setSurface

Set the preview surface. If not, there will be no preview scanning. Generally, this method is used to call getbestpreviewsize() to return the resolution supported by the camera, and then set it

public void setSurface(Surface surface, int width, int height);

parameter：

parameter	Description
surface	
width	width
height	height

return value : none

3.5.4 start

Start camera scanning

public void start(OnScannerListener listener);

parameter :

parameter	Description
listener	Decode listener

return value : none

3.5.5 stop

Stop scanning code and call when user initiatively exits.

public void stop();

parameter：none

return value : none

3.5.6 switchCamera

Before and after the switch, the camera is called after start. If you want to set up which camera to use from the beginning, please send it in initScanner configuration.

public void switchCamera(boolean usedFrontCcd);

parameter :

parameter	Description
usedFrontCcd	Front camera or not

return value : none

3.5.7 flashTrigger

Turn on the flash and call after start.

public void flashTrigger(boolean on);

parameter :

parameter	Description
on	Turn on flash or not

return value : none

3.5.8 focusTrigger

Open autofocus and call after start.

public void focusTrigger(boolean auto);

parameter :

parameter	Description
auto	Turn on auto connect focus

return value : none

3.5.9 setZoom

Set up an enlarged preview and call it after start.

public void setZoom(float scale);

Parameter term :

Parameter	Description
scale	0f~1.0f, restore default when 0

return value : none

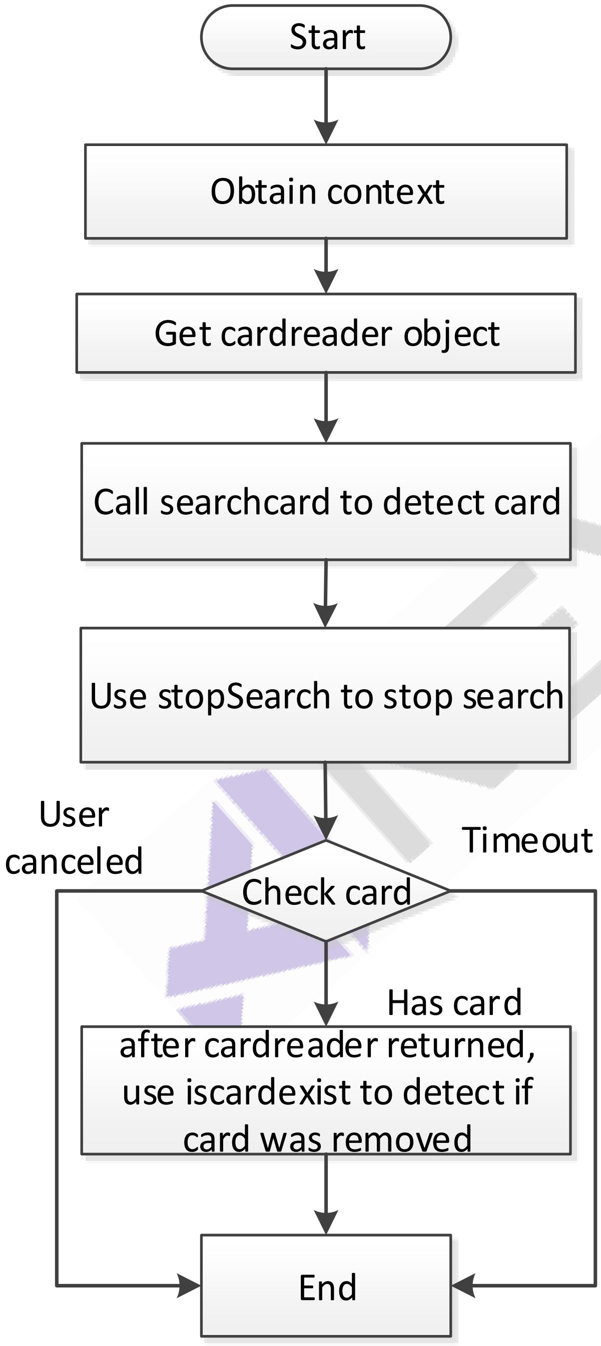
3.6 Card Reader Class

Card reader is responsible for managing the POS card reader (Note: 1, the user can take the initiative to stop the card operation; 2, find card operation automatically stop after the card is found).

Get the object of the reader class:

CardReader reader = deviceEngine getCardReader (). ;

This module operates using the basic flow chart:


Start
Obtain context
Get cardreader object
Call searchcard to detect card
Use stopSearch to stop search
User canceled
Timeout
Check card
Has card
after cardreader returned, use iscardexist to detect if card was removed
End

3.6.1 searchCard

Open the corresponding card reader, check the corresponding card slot has a card.

public int searchCard(HashSet<CardSlotTypeEnum> slotTypes,int timeout, OnCardInfoListener listener);
Parameters:

Parameter	Description
slotTypes	Slot enumerated type CardSlotTypeEnum; supports a variety of combinations of slots
timeout	Timeout in seconds; recommended value 60
listener	Callback interface OnCardInfoListener

CardSlotTypeEnum

Enumeration Name	Description
ICC1	Default IC card slot
ICC2	Unavailable
ICC3	Unavailable
PSAM1	PSAM slot 1
PSAM2	PSAM slot 2
PSAM3	Unavailable
RF	Non-access card slot
SWIPE	Magnetic stripe card slot

Return Value:

SdkResult.Success successful execution listener callback interface

SdkResult.Fail other errors

3.6.2 stopSearch

Turn off the corresponding card reader and stop detecting if there is a card in the card slot.

Public void stopSearch();

Parameters: None

Return Value: None

3.6.3 isCardExist

When the card reader operation is finished, call the card to check whether the contact IC card is pulled out or if there is a non-card access card in the slot.

Public boolean isCardExist (CardSlotTypeEnum slotTypes);

Parameters:

Parameter	Description
slotTypes	Slot enumerated type CardSlotTypeEnum

CardSlotTypeEnum

Enumeration Name	Description
ICC1	Default IC card slot
ICC2	Unavailable
ICC3	Unavailable
PSAM1	PSAM slot 1
PSAM2	PSAM slot 2
PSAM3	Unavailable
RF	Non-access card slot
SWIPE	Magnetic stripe card slot

Return Value:

True exists

False does not exist

3.6.4 open

Open the specified slot, and if you have already called searchCard to find the card, you do not need to call open again

public void open(CardSlotTypeEnum cardSlotType);

Parameters:

Parameter	Description
cardSlotType	CardSlotTypeEnum

CardSlotTypeEnum

Enumeration Name	Description
ICC1	Default IC card slot
ICC2	Unavailable

ICC3	Unavailable
PSAM1	PSAM slot 1
PSAM2	PSAM slot 2
PSAM3	Unavailable
RF	Non-access card slot
SWIPE	Magnetic stripe card slot

Return Value: None

3.6.5 close

Close the specified slot.

public void close(CardSlotTypeEnum cardSlotType);

Parameters:

Parameter	Description
cardSlotType	CardSlotTypeEnum

CardSlotTypeEnum

Enumeration Name	Description
ICC1	Default IC card slot
ICC2	Unavailable
ICC3	Unavailable
PSAM1	PSAM slot 1
PSAM2	PSAM slot 2
PSAM3	Unavailable
RF	Non-access card slot
SWIPE	Magnetic stripe card slot

Return Value: None

3.6.6 getRfCardType

Get contactless card type

public RfCardTypeEnum getRfCardType(CardSlotTypeEnum cardSlotType);

Parameters:

Parameter	Description

cardSlotType	CardSlotTypeEnum

CardSlotTypeEnum

Enumeration Name	Description
ICC1	Default IC card slot
ICC2	Unavailable
ICC3	Unavailable
PSAM1	PSAM slot 1
PSAM2	PSAM slot 2
PSAM3	Unavailable
RF	Non-access card slot
SWIPE	Magnetic stripe card slot

RfCardTypeEnum

Enumeration Name	Description
TYPE_A_CPU	
TYPE_B_CPU	
S50	
FELICA	
S70	
ULTRALIGHT	
MEMORY_OTHER	
S50_PRO	
S70_PRO	

Return Value:

Success return RfCardTypeEnum

Fail null

3.6.7 setETU

reset ETU。

public void setETU(CardSlotTypeEnum cardSlotType, int val);

Parameters:

Parameter	Description
cardSlotType	CardSlotTypeEnum
val	Value 0:372(standard card, default support adaptive 4-fold)

CardSlotTypeEnum

Enumeration Name	Description
ICC1	Default IC card slot
ICC2	Unavailable
ICC3	Unavailable
PSAM1	PSAM slot 1
PSAM2	PSAM slot 2
PSAM3	Unavailable
RF	Non-access card slot
SWIPE	Magnetic stripe card slot

Return Value:

None

3.6.8 setSupportFelica

set if support Felica Card .

public void setSupportFelica(boolean var1);

Return Value:

None

3.6.9 setFelicaSystemCode

set Felica Card system code

void setFelicaSystemCode(byte[] code);

Return Value:

None

3.6.10 setFelicaRequestCode

set Felica Request Code.

void setFelicaRequestCode(byte code);

Return Value:

None

3.7 CPU Cards

The CPU card class is responsible for managing the CPU card.

Get the object of the CPU card class:

CPUCardHandler cpucard = deviceEngine getCPUCardHandler (CardSlotTypeEnum slotType).;

Parameters:

Parameter	Description
slotType	Card slot type

CardSlotTypeEnum

Enumeration Name	Description
ICC1	Default IC card slot
ICC2	Unavailable
ICC3	Unavailable
PSAM1	PSAM slot 1
PSAM2	PSAM slot 2
PSAM3	Unavailable
RF	Non-access card slot

This module operates using the basic flow chart:

3.7.1 readUid

Read Uid of the card

Public String readUid ();

Parameters: None

Return Value:

Success Uid

Note : if the card is Felica , the UID = IDm + PMm

Failure Null

3.7.2 powerOn

Power-on reset, only for ICC1, PSAM1, PSAM2.

Public boolean powerOn (byte [] atr);

Parameters:

Parameter	Description
atr	Power returns atr, the first length byte hexadecimal representation, followed by the standard atr data

Return Value:

True success

False failure

3.7.3 active

Activated, only for contactless card(RF).

Public boolean active ();

Parameters: None

Return Value:

True success

False failure

3.7.4 exchangeAPDUCmd

Interactive APDU command.

Public int exchangeAPDUCmd (APDUEntity cmd);

Parameters:

Parameter	Description
cmd	APDUEntity Command data

APDUEntity

Attributes	Description
byte p1	Instruction to attach a specific Parameter
byte p2	Instruction to attach a specific Parameter
int lc	The number of bytes to transfer data
int le	Expect the maximum number of bytes to return
byte ins	Instruction code
byte cla	Command category
byte swa	Back swa
byte swb	Back swb
int dataOutLen	Returns the length of the data
byte [] dataIn	Sent data
byte [] dataOut	Return data
int overtime	Timeout in milliseconds; recommended value 1000

Return Value:

SdkResult.Success success

SdkResult.Fail failure

3.7.5 exchangeAPDUCmd

Interactive APDU command.

public byte[] exchangeAPDUCmd(byte[] cmd);

Parameters:

Parameter	Description
cmd	Apdu command data

Return Value:

Success return response data

Fail null

3.7.6 powerOff

Power down.

Public void powerOff();

Parameters: None

Return Value: None

3.7.7 remove

remove contactless card

public boolean remove();

Return Value:

SdkResult.Success success

SdkResult.Fail failure

3.8 EMV class (Emvhandler2)

The EMV class is responsible for managing the EMV operation of the POS.

Get the object of the EMV class:

EmvHandler2 EmvHandler = deviceEngine getEmvHandler2 (String appId);

Parameters:

Parameter	Description
appId	Application ID is mainly used to distinguish between aid and capk storage paths

3.8.1 delAllAid

Remove all AIDs.

Public void delAllAid ();

Parameters: None

Return Value: None

3.8.2 delOneAid

Delete an AID.

Public boolean delOneAid (byte [] aid);

Parameters:

Parameter	Description
aid	Enter aid

Return Value:

True success

False failure

3.8.3 delAllCapk

Remove all CAPK.

Public void delAllCapk ();

Parameters: None

Return Value: None

3.8.4 delOneCapk

Delete a CAPK.

Public boolean delOneCapk (byte [] rid, int capkIdx);

Parameters:

Parameter	Description
rid	Enter rid
capkIdx	capk Index

Return Value:

True success

False failure

3.8.5 setAidParaList

Set the AID.

public int setAidParaList(List<AidEntity> aidParaTlvList);

Parameters:

Parameter	**Description**
aidParaTlvList	Aid list

AidEntity

attribute	**Description**
String aid	Application ID
int asi	Application selection indicator 0- needn't match exactly(partial match up to the length); 1- match exactly
String tacDefault	Terminal Action Code – Default
String tacOnline	Terminal Action Code – Online
String tacDenial	Terminal Action Code – Denial
String appVerNum	Application Version Number
String ddol	DDOL
long threshold	Threshold value for biased random selection
int maxTargetPercent	The maximum target percentage to be used for biased random selection
int targetPercent	The target percentage to be used for random selection
int onlinePinCap	Terminal online Pin capability
long floorLimit	Contact floor limit

long transLimit	Electronic cash limit(union pay used in china market)
long contactlessCvmLimit	Contactless cvm limit
long contactlessTransLimit	Contactless transaction limit
long contactlessFloorLimit	Contactless floor limit
String transType	Transaction type, EMV tag 9c, “00”-sale, “20”-refund.. Default value is “FF”,it means adapt to all transaction type
AidEntryModeEnum aidEntryModeEnum	AID_ENTRY_CONTACT_CONTACTLESS: default value, means this aid can used for both contact and contactless AID_ENTRY_CONTACT: This aid is only used for cotact AID_ENTRY_CONTACTLESS: This aid is only used for contactless So, the same aid can config 2 aid with aidEntryModeEnum different, one is only for contact, and one is only for contactless

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.8.6 setAidParaList

Set the AID.

Public int setAidParaList (List <byte []> aidParaTlvList);

Parameters:

Parameter	Description
aidParaTlvList	Enter the number of aid data list, such as: aidParaTlvList.add(ByteUtils.hexString2ByteArray("9F0607A0000000043060DF0101009F08020002DF1105FC5058A000DF1205F85058F800DF130504000000009F1B0400000000DF150400000000DF160199DF170199DF14039F3704DF180101DF2006000999999999"));

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.8.7 setAidParaList

Set the AID.

public int setAidParaList(List<String> aidParaTlvList);

Parameters:

Parameter	Description
aidParaTlvList	he number of aid data list, such as: aidParaTlvList.add("9F0607A0000000043060DF0101009F08020002DF1105FC5058A000DF1205F85058F800DF130504000000009F1B0400000000DF150400000000DF160199DF170199DF14039F3704DF180101DF2006000999999999");

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.8.8 setCAPKList

Set CAPK.

public int setCAPKList(List<CapkEntity> capkTlvList);
参数项：

Parameter	Description
capkTlvList	Capk list

CapkEntity

attribute	Description
String rid	Registered Application Identifier
int capkIdx	Unique CA public key index number
int hashInd	Cryptographic algorithm ID used to generate the CAPK
String modulus	CA Public Key modulus
String exponent	CA Public Key exponent
String checkSum	CA Public Key checkSum
String expireDate	CA Public Key expireDate(YYYYMMDD)

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.8.9 setCAPKList

Set CAPK.

Public int setCAPKList (List <byte []> capkTlvList);

Parameters:

Parameter	Description
capkTlvList	Enter multiple capk data list, such as: capkTlvList.add (ByteUtils.hexString2ByteArray ("9F0605A0000000659F220109DF05083230303931323331DF060101DF070101DF028180B72A8FEF5B27F2B550398FDCC256F714BAD497FF56094B7408328CB626AA6F0E6A9DF8388EB9887BC930170BCC1213E90FC070D52C8DCD0FF9E10FAD36801FE93FC998A721705091F18BC7C98241CADC15A2B9DA7FB963142C0AB640D5D0135E77EBAE95AF1B4FEFADCF9C012366BDDA0455C1564A68810D7127676D493890BDDF040103DF03144410C6D51C2F83ADFD92528FA6E38A32DF048D0A"));

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.8.10 setCAPKList

Set CAPK.

public int setCAPKList(List<String> capkTlvList);

Parameters:

Parameter	**Description**
capkTlvList	Enter multiple capk data list, such as: capkTlvList.add("9F0605A0000000659F220109DF05083230303931323331DF060101DF070101DF028180B72A8FEF5B27F2B550398FDCC256F714BAD497FF56094B7408328CB626AA6F0E6A9DF8388EB9887BC930170BCC1213E90FC070D52C8DCD0FF9E10FAD36801FE93FC998A721705091F18BC7C98241CADC15A2B9DA7FB963142C0AB640D5D0135E77EBAE95AF1B4FEFADCF9C012366BDDA0455C1564A68810D7127676D493890BDDF040103DF03144410C6D51C2F83ADFD92528FA6E38A32DF048D0A");

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.8.11 getAidListNum

get aid list number

public int getAidListNum();

Return Value:

Number of aid list

3.8.12 getAidList

get aid list

public List<AidEntity> getAidList();

AidEntity

Attributes	Description
String aid	Application ID
int asi	Application selection indicator 0- needn't match exactly(partial match up to the length); 1- match exactly
String tacDefault	Terminal Action Code – Default
String tacOnline	Terminal Action Code – Online
String tacDenial	Terminal Action Code – Denial
String appVerNum	Application Version Number
String DDOL	DDOL
long threshold	Threshold value for biased random selection
int maxTargetPercent	The maximum target percentage to be used for biased random selection
int targetPercent	The target percentage to be used for random selection
int onlinePinCap	Terminal online Pin capability
long floorLimit	
long transLimit	
long contactlessCvmLimit	
long contactlessTransLimit	
long contactlessFloorLimit	

Return Value:

Success	return aid list
Fail	return null

3.8.13 getCapkListNum

get capk list number

public int getCapkListNum();

Return Value:

Number of capk list

3.8.14 getCapkList

get capk list

public List<CapkEntity> getCapkList();

CapkEntity

Attributes	Description
String rid	Registered Application Identifier
int capkIdx	Unique CA public key index number
int hashInd	Cryptographic algorithm ID used to generate the CAPK
String modulus	CA Public Key modulus
String exponent	CA Public Key exponent
String checkSum	CA Public Key checkSum
String expireDate	CA Public Key expireDate(MMYY)

Return Value:

Success	return capk list
Fail	return null

3.8.15 emvDebugLog

enable EMV log for checking emv issues, default false

public void emvDebugLog(boolean isEnable);

Parameters:

Parameter	Description
isEnable	True , false

Return Value: None

3.8.16 setDynamicReaderLimitListForPaywave

Set DRL for paywave

public int setDynamicReaderLimitListForPaywave(List<DynamicReaderLimitEntity> drlEntityList)

Parameters:

Parameter	Description
drlEntityList	DRL list

DynamicReaderLimitEntity

attribute	Description
byte[] appProgID	
boolean statusCheck	
boolean authOfZeroCheck	
byte authOfZeroCheckOption;	
boolean readerContactlessTransLimitCheck;	
boolean readerCVMReqLimitCheck;	
boolean readerContactlessFloorLimitCheck;	
private boolean drlSupport;	
byte[] readerContactlessTransLimit;	
byte[] readerCVMReqLimit;	
byte[] readerContactlessFloorLimit;	

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.8.17 setDynamicReaderLimitListForExpressPay

Set DRL for Amex Expresspay

public int setDynamicReaderLimitListForExpressPay (List<DynamicReaderLimitEntity> drlEntityList)

Parameters:

Parameter	Description
drlEntityList	DRL list

DynamicReaderLimitEntity

attribute	Description
byte[] appProgID	Application Prog ID
boolean statusCheck	statusCheck
boolean authOfZeroCheck	
byte authOfZeroCheckOption;	
boolean readerContactlessTransLimitCheck;	
boolean readerCVMReqLimitCheck;	
boolean readerContactlessFloorLimitCheck;	
private boolean drlSupport;	
byte[] readerContactlessTransLimit;	
byte[] readerCVMReqLimit;	
byte[] readerContactlessFloorLimit;	

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.8.18 getTlv

Get tag.

Public byte [] getTlv (byte [] tag, EmvDataSourceEnum pathId);

Parameters:

Parameter	Description

tag	tag value
pathId	tag source

EmvDataSourceEnum

Enumeration Name	Description
FROM_KERNEL	Data sources kernel
FORM_CARD	Data sources cards

Return Value:

Tlv successful Return Value

Else return null

3.8.19 getTlvByTags

public String getTlvByTags(String[] tags);

Parameters:

Parameter	Description
tags	Tag such as: String[] TAGS = {"9f26", "9f27", "9f10", "9f37", "9f36", "95", "9a", "9c", "9f02", "5f2a", "82", "9f1a", "9f03","9f33", "9f34", "9f35", "9f1e", "9f09", "84", "9f41"}

Return Value:

Tlv successful Return string Value

Else return null

3.8.20 setTlv

Settings tag for EMV processing

public int setTlv (byte [] tag, byte [] value);

Parameters:

Parameter	Description

tag	tag value
value	data

Return Value:

SdkResult.Success success

SdkResult.Fail failure

SdkResult.Param_In_Invalid Parameter error

3.8.21 initTermConfig

Allows the user to set the terminal personalization attribute, initialize the EMV kernel, and use the EMV kernel default attribute if the user does not call it. (Not recommended, please use method setTlv instead of this method)

Public int initTermConfig (byte [] cfgTlv);

Parameters:

Parameter	Description
cfgTlv	Standard tlv data stream

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.8.22 emvProcess

Start emv process

Public int emvProcess (EmvTransConfigurationEntity transData, OnEmvProcessListener2 listener);

Parameters:

Parameter	Description
transData	EMV transactions Entity Info
listener	EMV flow monitor interfaces

EmvTransDataEntity

Attributes	Description

String traceNo	trace number, length 8
String transAmt	Amount, length 12, for example “000000010000” = 100.00
String cashbackAmt	Cash back amount, length 12
String transDate	Transaction date MMDD, length 4
String transTime	Transaction date HHMMSS, length 6
byte[] merName	merchant name
String merId	merchant ID, length 15
String termId	terminal ID, length 8
byte emvTransType	EMV Transaction Type, sale-0x00, refund-0x20…
String countryCode	Country code , emv tage 9f1a
String currencyCode	Currency code, emv tag 5f2a
EmvEntryModeEnum entryModeEnum	Entry mode: contact or contactless
EmvProcessFlowEnum processFlowEnum	Standard flow(full flow) Read app data flow
boolean isContactForceOnline	True: Contact transaction force online False: standard process
MasterCardTransDataEntity entity	Master card parameter
VisaTransDataEntity entity	Visa parameter
AmexTransDataEntity entity	Amex parameters
UnionPayTransDataEntity entity	UnionPay parameter

EmvProcessFlowEnum

Enumeration Name	Description
EMV_PROCESS_FLOW_STANDARD	Standard emv flow
EMV_PROCESS_FLOW_READ_APPDATA	Read application data, card number ..etc(it is only suitable for contact)

EmvEntryModeEnum

Enumeration Name	Description
EMV_ENTRY_MODE_CONTACT	Contact
EMV_ENTRY_MODE_CONTACTLESS	Contactless

MasterCardTransDataEntity

Enumeration Name	Description
Boolean isSupportContactQps	True: support conatact QPS False: do not support
String contactNoCvmLimit	Contact QPS limit, 12 bytes. If transaction amount < contactNoCvmLimit, for master credit card, No cvm replace signature.

VisaTransDataEntity

Enumeration Name	Description
Boolean isSupportContactQps	True : support conatact QPS False: do not support
String contactNoCvmLimit	Contact QPS limit, 12 bytes. If transaction amount < contactNoCvmLimit, for master credit card, No cvm replace signature.

AmexTransDataEntity

Enumeration Name	Description
Boolean isExpressPaySeePhoneTapCardAgain	express pay see phone test cases, the second tap should set the value true

UnionPayTransDataEntity

Enumeration Name	Description
Boolean isForceOnline	Force online
Boolean isSupportCDCVM	Support CDCVM, default value is true
Boolean isQpbocForGlobal	if use China market, please set false, others please set true. Default value is true
Boolean isSupportContactlessQps	Support QPS
String contactlessQpsLimit	QPS limit

Return Value:

SdkResult.Success success execution listener callback

SdkResult.Param_In_Invalid illegal Parameter

3.8.23 onSetSelAppResponse

After executing the OnEMVProcessListener2. OnSelApp method, call it to notify the EMV kernel to continue the process.

Public void onSetSelAppResponse (int selResult);

Parameters:

Parameter	Description
selResult	After selecting the AID index number, the index starts at 1; the method is performed by onSelApp after obtained.

Return Value: None

3.8.24 onSetTransInitBeforeGPOResponse

After executing the OnEMVProcessListener2. onTransInitBeforeGPO method, call it to notify the EMV kernel to continue the process.

Public void onSetTransInitBeforeGPOResponse (boolean isSuccess);

Parameters:

Parameter	Description
isSuccess	Default value: true. The result of final select application.

Return Value: None

3.8.25 onSetConfirmCardNoResponse

After executing the OnEmvProcessListener2. OnConfirmCardNo method, call it to notify the EMV kernel to continue the process.

Public void onSetConfirmCardNoResponse (boolean isConfirm);

Parameters:

Parameter	Description
isConfirm	Are you sure, true: yes, false: no

Return Value: None

3.8.26 onSetPinInputResponse

After executing the OnEMVProcessListener2. OnCardHolderInputPin method, call it to notify the EMV kernel to continue the process.

Public void onSetPinInputResponse (boolean isConfirm, boolean isBypass);

Parameters:

Parameter	Description
isConfirm	Whether the Enter key is pressed
isBypass	If no password is entered, press the Enter key

Return Value: None

3.8.27 onSetContactlessTapCardResponse

After executing OnEmvProcessListener2. onContactlessTapCardAgain method， call it to notify the EMV kernel to continue the process.

public void onSetContactlessTapCardResponse (boolean isSuccess);

Parameters:

Parameter	Description
isSuccess	isSuccess，true:yes，false:no

Return Value: None

3.8.28 onSetOnlineProcResponse

After executing the OnEmvProcessListener. OnOnlineProc method, call it to notify the EMV kernel to take the secondary authorization.

public void onSetOnlineProcResponse (int retCode, EmvOnlineResultEntity result);

Parameters:

Parameter	Description
retCode	Are you sure, true: yes, false: no
result	EmvOnlineResultEntity, EMV online results

EmvOnlineResultEntity

Attributes	Description
String rejCode	Host respond with transaction response codes
String authCode	Host respond with Transaction Authorization Code
Byte [] recvField55	Host respond 55 field data

Return Value: None

3.8.29 onSetPromptResponse

After executing OnEmvProcessListener. onPrompt method，call it to notify the EMV kernel to continue the process.

public void onSetPromptResponse (boolean isSuccess);

Parameters:

Parameter	Description
isSuccess	isSuccess，true:yes，false:no

Return Value: None

3.8.30 onSetRemoveCardResponse

After executing OnEmvProcessListener. onRemoveCard method，call it to notify the EMV kernel to continue the process.

public void onSetRemoveCardResponse (boolean isSuccess);

Parameters:

Parameter	Description
isSuccess	isSuccess，true:yes，false:no

Return Value: None

3.8.31 EMVProcessCancel

Cancel EMV process.

public void emvProcessCancel ();

Parameters: None

Return Value: None

3.8.32 EMVProcessAbort

Force quite EMV process.

public void emvProcessAbort ();

Parameters: None

Return Value: None

3.8.33 getEmvContactlessMode

get EMV contactless flow mode, EMV mode or MSD mode, should be called in method onOnlineProc or onFinish method

public EmvModeEnum getEmvContactlessMode();

Return Value:

EmvModeEnum

Enumeration Name	Description
EMV	EMV mode
MSD	MSD mode
UNDEF	UNDEF mode
LEGACY	LEGACY mode

3.8.34 contactlessSetAidFirstSelect

set which AID first select for contactless transaction. It should be called before emvProcess.

public int contactlessSetAidFirstSelect (byte aidLen, byte[] aid);

parameter：

Attributes	Description
aidLen	AID length
aid	AID

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.8.35 setPureKernelCapab

set pure kernel capability. It should be called in method "onTransInitBeforeGPO"

int setPureKernelCapab(byte[] capab);

parameter：

Attributes	Description
capab	Capability, 5 bytes

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.8.36 setJcbContactlessTIP

set JCB contactless TIP. It should be called in method "onTransInitBeforeGPO"

void setJcbContactlessTIP(byte[] terminalInterchangeProfile);

parameter：

Attributes	Description
terminalInterchangeProfile	TIP

Return Value:

None

3.8.37 setRupayTransType

set Rupay contactless transaction type. It should be called in method "onTransInitBeforeGPO"

void setRupayTransType (RupayTransType transType);

parameter：

Attributes	Description
transType	Transaction Type

RupayTransType

Attributes	Description
RUPAY_TRANSTYPE_GOODS	GOODS, 0x00
RUPAY_TRANSTYPE_CASH	CASH, 0x01
RUPAY_TRANSTYPE_CASHBACK	CASHBACK, 0x19
RUPAY_TRANSTYPE_MONEYADD	MONEYADD, 0x28
RUPAY_TRANSTYPE_BALANCEENQUIRY	BALANCEENQUIRY,0x31
RUPAY_TRANSTYPE_VOID	VOID,0x34
RUPAY_TRANSTYPE_SERVICECREATION	SERVICE CREATION,0x83
RUPAY_TRANSTYPE_OTHER	DEFAULT, 0xff

Return Value:

None

3.8.38 getJcbContactlessTIP

get JCB contactless TIP.

byte[] getJcbContactlessTIP();

Return Value:

JCB contactless TIP

3.8.39 getSignNeed

get cvm result if need signature

public boolean getSignNeed();

Return Value:

ture	need signature
false	not need signature

3.8.40 getEmvCvmResult

get EMV CVM result

EmvCvmResultEnum getEmvCvmResult();

Return Value:

EmvCvmResultEnum

Enumeration Name	Description
EMV_CVMR_NA	CVM result is not specified, or the result is null
EMV_CVMR_NOCVM	No cvm required
EMV_CVMR_SIGNATURE	Signature

EMV_CVMR_ONLINEPIN	Online pin
EMV_CVMR_CONFVERIFIED	ID verify (not used)
EMV_CVMR_CDCVM	CDCVM
EMV_CVMR_OFFLINEPIN_PLAINTEXT	Offline plaintext pin
EMV_CVMR_OFFLINEPIN_ENCIPHER	Offline encipher pin
EMV_CVMR_OFFLINEPIN_PLAINTEXT_SIGNATURE	Offline plaintext pin & signature
EMV_CVMR_OFFLINEPIN_ENCIPHER_SIGNATURE	Offline encipher pin & signature
EMV_CVMR_SKIP_CVM	Skip cvm, used for MIR

3.8.41 getEmvCardDataInfo

get EMV card data, such as pan, track2 data

CardInfoEntity getEmvCardDataInfo();

Return Value:

CardInfoEntity

Attributes	Description
String cardNo	Card number
CardSlotTypeEnum cardExistslot	CardSlotType
RfCardTypeEnum rfCardType	RfCardTyp
String tk1	track 1
String tk2	tracks 2
String tk3	tracks 3
String expiredDate	Card is valid
String serviceCode	Service Code
boolean isTk1Valid	A track LRC is correct
boolean isTk2Valid	Two tracks LRC is correct
boolean isTk3Valid	Three tracks LRC is correct
boolean isICC	If mag card has chip flag
String csn	Card serial number, only returnd in OnEmvProcessListener.onConfirmCardNo

CardSlotTypeEnum

Enumeration Name	Description
ICC1	The I C slot 1
ICC2	The I C slot 2
ICC3	The I C slot 3
PSAM1	PSAM slot 1
PSAM2	PSAM slot 1
PSAM3	PSAM slot 1
RF	Contactless card slot
SWIPE	Magnetic stripe card slot

RfCardTypeEnum

Enumeration Name	Description
TYPE_A_CPU	
TYPE_B_CPU	
S50	
FELICA	
S70	
ULTRALIGHT	
MEMORY_OTHER	
S50_PRO	
S70_PRO	

3.9 EMV class(Emvhandler) Deprecated

Please note: All the Emvhandler method, do not recommend use it anymore.

The EMV class is responsible for managing the EMV operation of the POS.

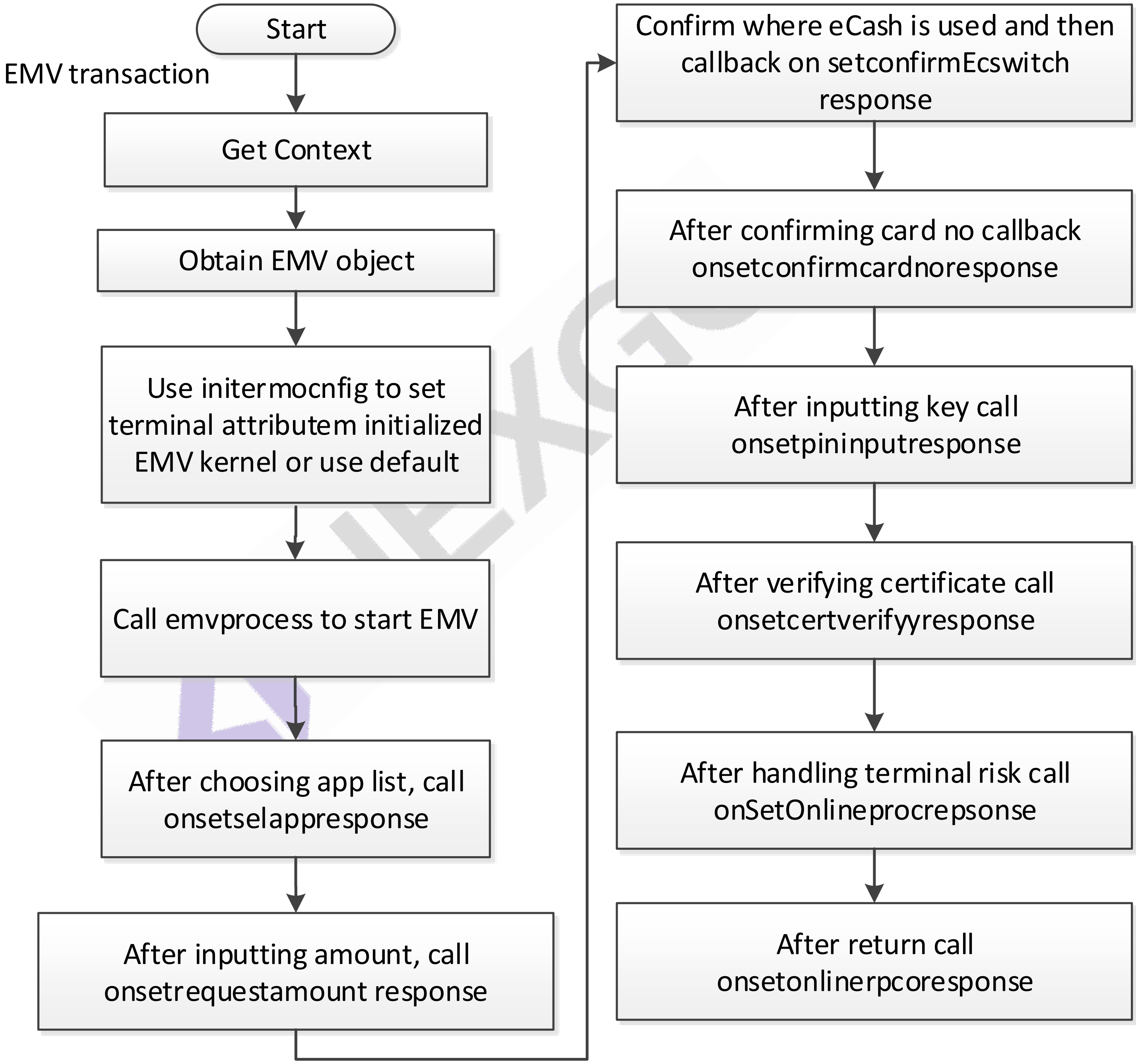
Get the object of the EMV class:

EmvHandler EmvHandler = deviceEngine getEmvHandler (String appId).;

Parameters:

Parameter	Description
appId	Application ID is mainly used to distinguish between aid and capk storage paths

This module operates using the basic flow chart:


Start
EMV transaction
Get Context
Obtain EMV object
Use intermocnfig to set terminal attributem initialized EMV kernel or use default
Call emvprocess to start EMV
After choosing app list, call onsetselappresponse
After inputting amount, call onsetrequestamount response
Confirm where eCash is used and then callback on setconfirmEcswitch response
After confirming card no callback onsetconfirmcardnoresponse
After inputting key call onsetpininputresponse
After verifying certificate call onsetcertverifyyresponse
After handling terminal risk call onSetOnlineprocrepsonse
After return call onsetonlinerpcoresponse

3.9.1 delAllAid

Remove all AIDs.

Public void delAllAid ();

Parameters: None

Return Value: None

3.9.2 delOneAid

Delete an AID.

Public boolean delOneAid (byte [] aid);

Parameters:

Parameter	Description
aid	Enter aid

Return Value:

True success

False failure

3.9.3 delAllCapk

Remove all CAPK.

Public void delAllCapk ();

Parameters: None

Return Value: None

3.9.4 delOneCapk

Delete a CAPK.

Public boolean delOneCapk (byte [] rid, int capkIdx);

Parameters:

Parameter	Description
rid	Enter rid

capkIdx	capk Index

Return Value:

True success

False failure

3.9.5 setAidParaList

Set the AID.

public int setAidParaList(List<AidEntity> aidParaTlvList);

Parameters:

Parameter	Description
aidParaTlvList	Aid list

AidEntity

attribute	Description
String aid	Application ID
int asi	Application selection indicator 2- needn't match exactly(partial match up to the length); 3- match exactly
String tacDefault	Terminal Action Code – Default
String tacOnline	Terminal Action Code – Online
String tacDenial	Terminal Action Code – Denial
String appVerNum	Application Version Number
String DDOL	DDOL
long threshold	Threshold value for biased random selection
int maxTargetPercent	The maximum target percentage to be used for biased random selection

int targetPercent	The target percentage to be used for random selection
int onlinePinCap	Terminal online Pin capability
long floorLimit	
long transLimit	
long contactlessCvmLimit	
long contactlessTransLimit	
long contactlessFloorLimit	

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.9.6 setAidParaList

Set the AID.

Public int setAidParaList (List <byte []> aidParaTlvList);

Parameters:

Parameter	Description
aidParaTlvList	Enter the number of aid data list, such as: aidParaTlvList.add(ByteUtils.hexString2ByteArray("9F0607A00000000043060DF0101009F08020002DF1105FC5058A000DF1205F85058F800DF130504000000009F1B0400000000DF150400000000DF160199DF170199DF14039F3704DF180101DF2006000999999999"));

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.9.7 setAidParaList

Set the AID.

public int setAidParaList(List<String> aidParaTlvList);

Parameters:

Parameter	Description
aidParaTlvList	he number of aid data list, such as: aidParaTlvList.add("9F0607A0000000043060DF0101009F08020002DF1105FC5058A000DF1205F85058F800DF130504000000009F1B0400000000DF150400000000DF160199DF170199DF14039F3704DF180101DF2006000999999999");

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.9.8 setCAPKList

Set CAPK.

public int setCAPKList(List<CapkEntity> capkTlvList);
参数项：

Parameter	Description
capkTlvList	Capk list

CapkEntity

attribute	Description
String rid	Registered Application Identifier
int capkIdx	Unique CA public key index number
int hashInd	Cryptographic algorithm ID used to generate the CAPK
String modulus	CA Public Key modulus
String exponent	CA Public Key exponent
String checkSum	CA Public Key checkSum
String expireDate	CA Public Key expireDate(YYYYMMDD)

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.9.9 setCAPKList

Set CAPK.

Public int setCAPKList (List <byte []> capkTlvList);

Parameters:

Parameter	Description
capkTlvList	Enter multiple capk data list, such as: capkTlvList.add (ByteUtils.hexString2ByteArray ("9F0605A0000000659F220109DF05083230303931323331DF060101DF070101DF028180B72A8FEF5B27F2B550398FDCC256F714BAD497FF56094B7408328CB626AA6F0E6A9DF8388EB9887BC930170BCC1213E90FC070D52C8DCD0FF9E10FAD36801FE93FC998A721705091F18BC7C98241CADC15A2B9DA7FB963142C0AB640D5D0135E77EBAE95AF1B4FEFADCF9C012366BDDA0455C1564A68810D7127676D493890BDDF040103DF03144410C6D51C2F83ADFD92528FA6E38A32DF048D0A"));

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.9.10 setCAPKList

Set CAPK.

public int setCAPKList(List<String> capkTlvList);

Parameters:

Parameter	Description
capkTlvList	Enter multiple capk data list, such as: capkTlvList.add("9F0605A0000000659F220109DF05083230303931323331DF060101DF070101DF028180B

	72A8FEF5B27F2B550398FDCC256F714BAD497FF56094B7408328CB626AA6F0E6A9DF8388EB9887BC930170BCC1213E90FC070D52C8DCD0FF9E10FAD36801FE93FC998A721705091F18BC7C98241CADC15A2B9DA7FB963142C0AB640D5D0135E77EBAE95AF1B4FEFADCF9C012366BDDA0455C1564A68810D7127676D493890BDDF040103DF03144410C6D51C2F83ADFD92528FA6E38A32DF048D0A");

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.9.11 setDynamicReaderLimitList

Set DRL for paywave

public int setDynamicReaderLimitList(List<DynamicReaderLimitEntity> drlEntityList)

Parameters:

Parameter	Description
drlEntityList	DRL list

DynamicReaderLimitEntity

attribute	Description
byte[] appProgID	
boolean statusCheck	
boolean authOfZeroCheck	
byte authOfZeroCheckOption;	
boolean readerContactlessTransLimitCheck;	
boolean readerCVMReqLimitCheck;	
boolean readerContactlessFloorLimitCheck;	
private boolean drlSupport;	

byte[] readerContactlessTransLimit;	
byte[] readerCVMReqLimit;	
byte[] readerContactlessFloorLimit;	

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.9.12 setDynamicReaderLimitListForExpressPay

Set DRL for Amex Expresspay

public int setDynamicReaderLimitListForExpressPay (List<DynamicReaderLimitEntity> drlEntityList)

Parameters:

Parameter	**Description**
drlEntityList	DRL list

DynamicReaderLimitEntity

attribute	**Description**
byte[] appProgID	Application Prog ID
boolean statusCheck	statusCheck
boolean authOfZeroCheck	
byte authOfZeroCheckOption;	
boolean readerContactlessTransLimitCheck;	
boolean readerCVMReqLimitCheck;	
boolean readerContactlessFloorLimitCheck;	
private boolean drlSupport;	
byte[] readerContactlessTransLimit;	
byte[] readerCVMReqLimit;	
byte[] readerContactlessFloorLimit;	

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.9.13 initTermConfig

Allows the user to set the terminal personalization attribute, initialize the EMV kernel, and use the EMV kernel default attribute if the user does not call it. (Not recommended, please use method setTlv instead of this method)

Public int initTermConfig (byte [] cfgTlv);

Parameters:

Parameter	Description
cfgTlv	Standard tlv data stream

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.9.14 emvProcess

Start emv process

Public int emvProcess (emvTransDataEntity transData, OnEmvProcessListener listener);

Parameters:

Parameter	Description
transData	EMV transactions Entity Info
listener	EMV flow monitor interfaces

EmvTransDataEntity

Attributes	Description
EmvAlgorithmTypeEnum algType	Type RSA algorithm or SM2, the default RSA

EmvTransFlowEnum procType	EMV Process Type
String traceNo	Serial number, length 8
String transAmt	Amount, length 12
String cashbackAmt	Cash back amount, length 12
String transDate	Transaction date MMDD, length 4
String transTime	Transaction date HHMMSS, length 6
byte[] merName	Business name
String merId	Business number, length 15
String termId	Terminal number, length 8
byte B9C	Transaction Type, sale-0x00, refund-0x20…
boolean isSupportEC	Whether support e-cash(only sued for union pay in china market)
EMVChannelTypeEnum channelType	Card approach, contact or contactless
boolean isQpbocForceLine	union pay whether to force go online
boolean isDefaultEC	When set to support EC is true, the default is e-cash; when false, then callback the method to let user to select whether to use electronic cash(only sued for union pay in china market)
isSupportCDCVM	union pay support CDCVM
isQpbocForGlobal	Union pay contactless check CVM limit for excute CVM method.(not force prompt online pin).

EMVAlgorithmTypeEnum

Enumeration Name	Description
RSA	RSA
SM2	Country code

EMVTransFlowEnum

Enumeration Name	Description
FULL	Standard full process
SIMPLE	Simple processonly confirms the callback number, then directlyOnFinish ends EMV process

QPASS	qpboc flow only confirms the callback number, then directly OnFinish ends EMV process

EMVChannelTypeEnum

Enumeration Name	Description
FROM_ICC	Contact
FROM_PICC	Contactless

Return Value:

SdkResult.Success success execution listener callback

SdkResult.Param_In_Invalid illegal Parameter

3.9.15 onSetSelAppResponse

After executing the OnEMVProcessListener. OnSelApp method, call the EMV kernel to continue the process.

Public void onSetSelAppResponse (int selResult);

Parameters:

Parameter	Description
selResult	After selecting the AID index number, the index starts at 1; the method is performed by onSelApp after obtained.

Return Value: None

3.9.16 onSetAfterFinalSelectedAppResponse

After executing the OnEMVProcessListener. onAfterFinalSelectedApp method, call the EMV kernel to continue the process.

Public void onSetAfterFinalSelectedAppResponse (boolean isSuccess);

Parameters:

Parameter	Description
isSuccess	Default value: true. The result of final select application.

Return Value: None

3.9.17 onSetRequestAmountResponse

After executing the OnEmvProcessListener. OnRequestAmount method, call the EMV kernel to continue the process.

Public void onSetRequestAmountResponse (String amount);

Parameters:

Parameter	Description
amount	Amount length 12, prepend with 0s to make it 12 digits long.

Return Value: None

3.9.18 onSetConfirmEcSwitchResponse

After executing the OnEMVProcessListener. OnConfirmEcSwitch method, call the EMV kernel to continue the process.

Public void onSetConfirmEcSwitchResponse (boolean isConfirm);

Parameters:

Parameter	Description
isConfirm	Whether to use electronic cash, true: yes, false: no

Return Value: None

3.9.19 onSetConfirmCardNoResponse

After executing the OnEmvProcessListener. OnConfirmCardNo method, call the EMV kernel to continue the process.

Public void onSetConfirmCardNoResponse (boolean isConfirm);

Parameters:

Parameter	Description
isConfirm	Are you sure, true: yes, false: no

Return Value: None

3.9.20 onSetPinInputResponse

After executing the OnEMVProcessListener. OnCardHolderInputPin method, call the EMV kernel to continue the process.

Public void onSetPinInputResponse (boolean isConfirm, boolean isBypass);

Parameters:

Parameter	Description
isConfirm	Whether the Enter key is pressed
isBypass	If no password is entered, press the Enter key

Return Value: None

3.9.21 onsetCertVerifyResponse

After executing the OnEMVProcessListener. OnCertVerify method, call the EMV kernel to continue the process.

Public void onSetCertVerifyResponse (boolean isVerify);

Parameters:

Parameter	Description
isVerify	Are you sure, true: yes, false: no

Return Value: None

3.9.22 onSetReadCardAgainResponse

After executing OnEmvProcessListener. onReadCardAgain method，call the EMV kernel to continue the process.

public void onSetReadCardAgainResponse(boolean isSuccess);

Parameters:

Parameter	Description
isSuccess	isSuccess，true:yes，false:no

Return Value: None

3.9.23 onSetOnlineProcResponse

After executing the OnEmvProcessListener. OnOnlineProc method, call the EMV kernel to take the secondary authorization.

public void onSetOnlineProcResponse (int retCode, EmvOnlineResultEntity result);

Parameters:

Parameter	Description
retCode	Are you sure, true: yes, false: no
result	EmvOnlineResultEntity, EMV online results

EmvOnlineResultEntity

Attributes	Description
String rejCode	Host respond with transaction response codes
String authCode	Host respond with Transaction Authorization Code
Byte [] recvField55	Host respond 55 field data

Return Value: None

3.9.24 onSetPromptResponse

After executing OnEmvProcessListener. onPrompt method , call the EMV kernel to continue the process.

public void onSetPromptResponse (boolean isSuccess);

Parameters:

Parameter	Description
isSuccess	isSuccess，true:yes，false:no

Return Value: None

3.9.25 onSetRemoveCardResponse

After executing OnEmvProcessListener. onRemoveCard method，call the EMV kernel to continue the process.

public void onSetRemoveCardResponse (boolean isSuccess);

Parameters:

Parameter	Description
isSuccess	isSuccess，true:yes，false:no

Return Value: None

3.9.26 getTlv

Get tag.

Public byte [] getTlv (byte [] tag, EmvDataSourceEnum pathId);

Parameters:

Parameter	Description
tag	tag value
pathId	tag source

EmvDataSourceEnum

Enumeration Name	Description
FROM_KERNEL	Data sources kernel
FORM_CARD	Data sources cards

Return Value:

Tlv successful Return Value

Else return null

3.9.27 getTlvByTags

public String getTlvByTags(String[] tags);

Parameters:

Parameter	Description
tags	Tag such as: String[] TAGS = {"9f26", "9f27", "9f10", "9f37", "9f36", "95", "9a", "9c", "9f02", "5f2a", "82", "9f1a", "9f03","9f33", "9f34", "9f35", "9f1e", "9f09", "84", "9f41"}

Return Value:

Tlv successful Return string Value

Else return null

3.9.28 setTlv

Settings tag.

public int setTlv (byte [] tag, byte [] value);

Parameters:

Parameter	Description
tag	tag value
value	data

Return Value:

SdkResult.Success success

SdkResult.Fail failure

SdkResult.Param_In_Invalid Parameter error

3.9.29 getEMVCardLog

Read the log, this method is finished after the callback to onFinish method.

public int getEmvCardLog (EmvChannelTypeEnum channelType, OnEmvProcessListener listener);

Parameters:

Parameter	Description
channelType	Channel Type
listener	Callback

EmvChannelTypeEnum

Enumeration Name	Description
FROM_ICC	Contact
FROM_PICC	Contactless

Return Value:

SdkResult.Success successful execution of listener callback interface

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.9.30 Clear the Log

Clear the log.

public int clearLog();

Parameters: None

Return Value:

SdkResult.Success success

SdkResult.Fail failure

3.9.31 EMVGetEcBalance

Read electronic cash balance.

public int emvGetEcBalance (EmvChannelTypeEnum channelType, OnEmvProcessListener listener);

Parameters:

Parameter	Description
channelType	Channel Type
listener	Callback

EmvChannelTypeEnum

Enumeration Name	Description
FROM _ICC	Contact
FROM_PICC	Contactless

Return Value:

SdkResult.Success successful execution of listener callback interface

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.9.32 EMVProcessCancel

Cancel EMV process.

public void emvProcessCancel ();

Parameters: None

Return Value: None

3.9.33   emvDebugLog

enable EMV log, default false

public void emvDebugLog(boolean isEnable);

Parameters:

Parameter	Description
isEnable	True ，false

Return Value: None

3.9.34   getEmvContactlessMode

get EMV contactless flow mode, EMV mode or MSD mode, should be called in method onOnlineProc or onFinish method

public EmvModeEnum getEmvContactlessMode();

Return Value:

EmvModeEnum

Enumeration Name	Description
EMV	EMV mode
MSD	MSD mode

3.9.35   getAidListNum

get aid list number

public int getAidListNum();

Return Value:

Number of aid list

3.9.36 getAidList

get aid list

public List<AidEntity> getAidList();

AidEntity

Attributes	Description
String aid	Application ID
int asi	Application selection indicator 4- needn't match exactly(partial match up to the length); 5- match exactly
String tacDefault	Terminal Action Code – Default
String tacOnline	Terminal Action Code – Online
String tacDenial	Terminal Action Code – Denial
String appVerNum	Application Version Number
String DDOL	DDOL
long threshold	Threshold value for biased random selection
int maxTargetPercent	The maximum target percentage to be used for biased random selection
int targetPercent	The target percentage to be used for random selection
int onlinePinCap	Terminal online Pin capability
long floorLimit	
long transLimit	
long contactlessCvmLimit	
long contactlessTransLimit	
long contactlessFloorLimit	

Return Value:

Success	return aid list
Fail	return null

3.9.37 getCapkListNum

get capk list number

public int getCapkListNum();

Return Value:

Number of capk list

3.9.38 getCapkList

get capk list

public List<CapkEntity> getCapkList();

CapkEntity

Attributes	Description
String rid	Registered Application Identifier
int capkIdx	Unique CA public key index number
int hashInd	Cryptographic algorithm ID used to generate the CAPK
String modulus	CA Public Key modulus
String exponent	CA Public Key exponent
String checkSum	CA Public Key checkSum
String expireDate	CA Public Key expireDate(MMYY)

Return Value:

Success	return capk list
Fail	return null

3.9.39 newDelAllAid

Pure , MIR kernel API, delete all the AID

public void newDelAllAid();

Parameters: None

Return Value: None

3.9.40 newDelOneAid

Pure , MIR kernel API, delete one AID

public boolean newDelOneAid(byte[] aid);

parameter :

Attributes	Description
byte[] aid	aid

Return Value:

ture	delete success
false	delete failed

3.9.41 newDelAllCapk

Pure , MIR kernel API, delete all CAPK

public void newDelAllCapk();

Parameters: None

Return Value: None

3.9.42 newDelOneCapk

Pure , MIR kernel API, delete one CAPK

public boolean newDelOneCapk(byte[] rid,int capkIdx);

parameter :

Attributes	Description
rid	rid
capkIdx	Capk index

Return Value:

ture	delete success
false	delete failed

3.9.43 newSetAidParaList

Pure , MIR kernel API, set AID list

public int newSetAidParaList(List<byte[]> aidParaTlvList);

parameter：

Attributes	Description
aidParaTlvList	Aid list： aidParaTlvList.add(ByteUtils.hexString2ByteArray("9F0607A0000000043060DF0101009F08020002DF1105FC5058A000DF1205F85058F800DF13050400000000009F1B0400000000DF150400000000DF160199DF170199DF14039F3704DF180101DF2006000999999999"));

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.9.44 newSetAidParaList

Pure , MIR kernel API, set AID list

public int newSetAidParaList(List<String> aidParaTlvList);

parameter：

Attributes	Description
aidParaTlvList	Aid list： aidParaTlvList.add("9F0607A0000000043060DF0101009F08020002DF1105FC5058A000DF1205F85058F800DF13050400000000009F1B0400000000DF150400000000DF160199DF170199DF14039F3704DF180101DF2006000999999999");

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.9.45 newSetAidParaList

Pure , MIR kernel API, set AID list

public int newSetAidParaList(List<AidEntity> aidParaTlvList);

parameter：

Attributes	Description
aidParaTlvList	Aid list

AidEntity

Attributes	**Description**
String aid	Application ID
int asi	Application selection indicator 6- needn't match exactly(partial match up to the length); 7- match exactly
String tacDefault	Terminal Action Code – Default
String tacOnline	Terminal Action Code – Online
String tacDenial	Terminal Action Code – Denial
String appVerNum	Application Version Number
String DDOL	DDOL
long threshold	Threshold value for biased random selection
int maxTargetPercent	The maximum target percentage to be used for biased random selection
int targetPercent	The target percentage to be used for random selection
int onlinePinCap	Terminal online Pin capability
long floorLimit	
long transLimit	

long contactlessCvmLimit	
long contactlessTransLimit	
long contactlessFloorLimit	

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.9.46 newSetCAPKList

Pure , MIR kernel API, set CAPK list

public int newSetCAPKList(List<byte[]> capkTlvList);

parameter：

Attributes	Description
capkTlvList	Capk list： capkTlvList.add(ByteUtils.hexString2ByteArray("9F0605A0000000659F220109DF050832303931323331DF060101DF070101DF028180B72A8FEF5B27F2B550398FDCC256F714BAD497FF56094B7408328CB626AA6F0E6A9DF8388EB9887BC930170BCC1213E90FC070D52C8DCD0FF9E10FAD36801FE93FC998A721705091F18BC7C98241CADC15A2B9DA7FB963142C0AB640D5D0135E77EBAE95AF1B4FEFADCF9C012366BDDA0455C1564A68810D7127676D493890BDDF040103DF03144410C6D51C2F83ADFD92528FA6E38A32DF048D0A"));

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.9.47 newSetCAPKList

Pure , MIR kernel API, set CAPK list

public int newSetCAPKList(List<String> capkTlvList);

parameter :

Attributes	Description
capkTlvList	Capk list： capkTlvList.add("9F0605A0000000659F220109DF05083230303931323331DF060101DF070101DF028180B72A8FEF5B27F2B550398FDCC256F714BAD497FF56094B7408328CB626AA6F0E6A9DF8388EB9887BC930170BCC1213E90FC070D52C8DCD0FF9E10FAD36801FE93FC998A721705091F18BC7C98241CADC15A2B9DA7FB963142C0AB640D5D0135E77EBAE95AF1B4FEFADCF9C012366BDDA0455C1564A68810D7127676D493890BDDF040103DF03144410C6D51C2F83ADFD92528FA6E38A32DF048D0A");

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.9.48 newSetCAPKList

Pure , MIR kernel API, set CAPK list

public int newSetCAPKList(List<CapkEntity> capkTlvList);

parameter :

Attributes	Description
capkTlvList	Capk list

CapkEntity

Attributes	Description
String rid	Registered Application Identifier
int capkIdx	Unique CA public key index number

int hashInd	Cryptographic algorithm ID used to generate the CAPK
String modulus	CA Public Key modulus
String exponent	CA Public Key exponent
String checkSum	CA Public Key checkSum
String expireDate	CA Public Key expireDate(MMYY)

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.9.49 newGetAidListNum

Pure , MIR kernel API, get AID list number

public int getAidListNum();

Return Value:

Number of aid list

3.9.50 newGetAidList

Pure , MIR kernel API, get AID list number

public List<AidEntity> newGetAidList();

AidEntity

Attributes	**Description**
String aid	Application ID
int asi	Application selection indicator 8- needn't match exactly(partial match up to the length);

	9- match exactly
String tacDefault	Terminal Action Code – Default
String tacOnline	Terminal Action Code – Online
String tacDenial	Terminal Action Code – Denial
String appVerNum	Application Version Number
String DDOL	DDOL
long threshold	Threshold value for biased random selection
int maxTargetPercent	The maximum target percentage to be used for biased random selection
int targetPercent	The target percentage to be used for random selection
int onlinePinCap	Terminal online Pin capability
long floorLimit	
long transLimit	
long contactlessCvmLimit	
long contactlessTransLimit	
long contactlessFloorLimit	

Return Value:

Success	return aid list
Fail	return null

3.9.51 newGetCapkListNum

Pure , MIR kernel API, get CAPK list number

public int newGetCapkListNum();

Return Value:

Number of capk list

3.9.52 newGetCapkList

Pure , MIR kernel API, get CAPK list

public List<CapkEntity> newGetCapkList();

CapkEntity

Attributes	Description
String rid	Registered Application Identifier
int capkIdx	Unique CA public key index number
int hashInd	Cryptographic algorithm ID used to generate the CAPK
String modulus	CA Public Key modulus
String exponent	CA Public Key exponent
String checkSum	CA Public Key checkSum
String expireDate	CA Public Key expireDate(MMYY)

Return Value:

Success	return capk list
Fail	return null

3.9.53 selectAidFirst

set which AID first select for contactless transaction

public int selectAidFirst(boolean enable, byte aidLen, byte[] aid);

parameter：

Attributes	Description
enable	Enable: true--first; false--default
aidLen	AID length
aid	AID

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.9.54 getSignNeed

get signature state

public boolean getSignNeed();

Return Value:

ture	need signature
false	not need signature

3.9.55 setPureKernelCapab

set pure kernel capability

int setPureKernelCapab(byte[] capab);

parameter：

Attributes	Description
capab	Capability, 5 bytes

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.Fail failure

3.10 setSystemClock

Set the system time.

public void setSystemClock (Context context, String datetime);

Parameters:

Parameter	Description
context	Context
datetime	Time format YYYYMMDDHHMMSS, the year in the range 1970-2049

Return Value: None

3.11 getDeviceInfo

get device information

public DeviceInfo getDeviceInfo();

DeviceInfo

Attributes	Description
String sn	Terminal serail number
String ksn	Custom Terminal serail number
String model	Terminal model , such as N5
String osVer	Os version ,such as 5.1.1
String sdkVer	Sdkversion, 2.0.7
String firmWareVer	firmWareversion
String kernelVer	linuxversion
String vendor	vendor ,such as Nexgo

Return Value:

successful return DeviceInfo

else return null

3.12 Serial class

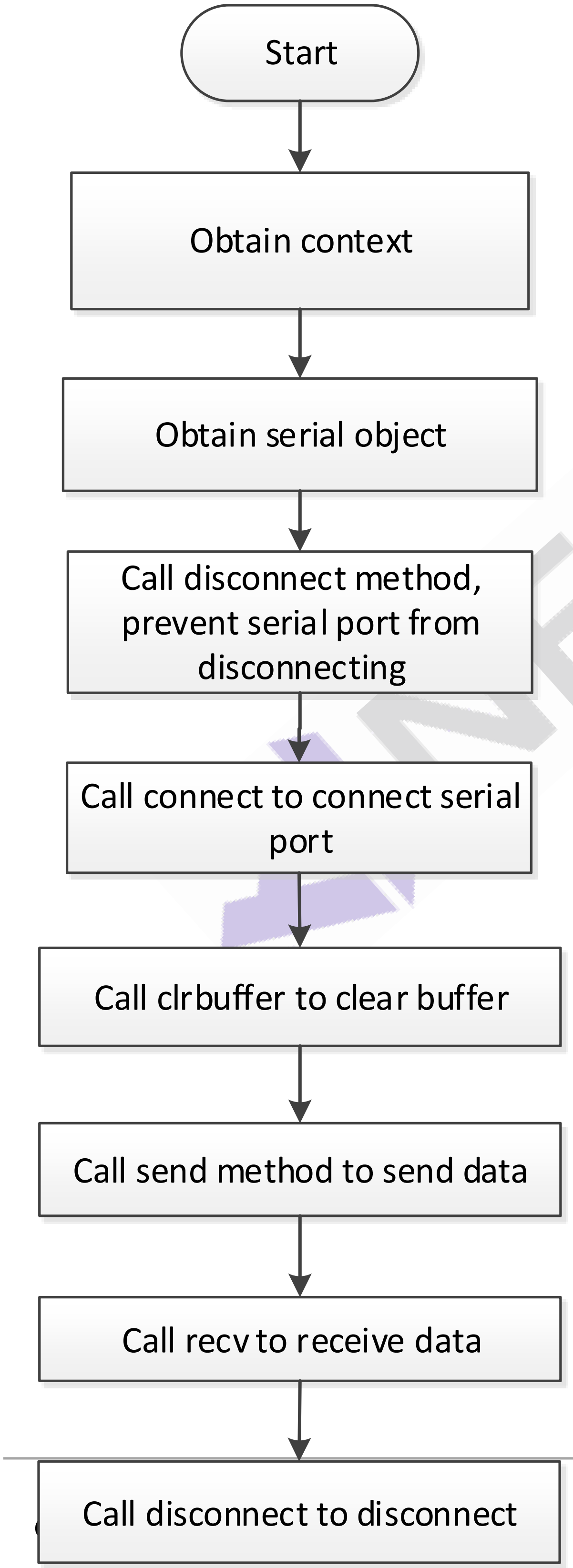
Serial class is responsible for managing POS serial port.

Get the serial class objects:

SerialPortDriver port = deviceEngine.getSerialPortDriver(int portNo);

Parameter	Description
portNo	Serial No. currently only supports the serial number 0

This module uses the basic flow chart:


Start
Obtain context
Obtain serial object
Call disconnect method, prevent serial port from disconnecting
Call connect to connect serial port
Call clrbuffer to clear buffer
Call send method to send data
Call recv to receive data
Call disconnect to disconnect

3.12.1 disconnect

Disconnect.

public int disconnect ();

Parameters: None

Return Value:

SdkResult.Success off successfully

SdkResult.SerialPort_Port_Not_Open serial port is not open

SdkResult.SerialPort_DisConnect_Fail serial chain disconnection failure

3.12.2 connect

Serial connection.

public int connect(SerialCfgEntity entity);

Parameters:

Parameter	Description
entity	SerialCfgEntity , Serial Info

SerialCfgEntity

Attributes	Description
int bauRate	The baud rate in the range of (bps): 110,300,600,1200,2400,4800, 9600,14400,56000,19200,38400,57600,115200,230400
int dataBits	Data Bits Range: 5, 6, 7, 8
char parity	Test methods in the range : 'o' odd , 'e' parity, 'n' no parity
int stopBits	Stop bit value range : 1, 2

Return Value:

SdkResult.Success serial connection success

SdkResult.Param_In_Invalid Parameter is null, illegal Parameter

SdkResult.SerialPort_Invalid_Communication_Parameter invalid communication Parameters

SdkResult.SerialPort_Connect_Fail serial connection failure

SdkResult.Fail other errors

3.12.3 clrBuffer

Clear the buffer.

public void clrBuffer ();

Parameters: None

Return Value: None

3.12.4 send

Send data.

public int send (byte [] data, int dataLen);

Parameters:

Parameter	Description
data	Input data
dataLen	Data length Range: 1-2048 bytes ; non-blocking send

Return Value:

SdkResult.Success sent successfully

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.SerialPort_Not_Open serial port is not open

SdkResult.SerialPort_Send_Fail serial data transmission failure

SdkResult.Fail other errors

3.12.5 recv

Receive data.

public int recv (byte [] buffer, int recvLen, long timeout);

Parameters:

Parameter	Description
buffer	Buffer to receive data
recvLen	The maximum length of buffer, which is 2048 bytes
timeout	Timeout in milliseconds; recommended value 3000

Return Value:

Successfully received returns the length of the received data

SdkResult.Param_In_Invalid illegal Parameter

SdkResult.SerialPort_Not_Open serial port is not open

SdkResult.SerialPort_Timeout_Receiving_Data serial data receive timeout

SdkResult.Fail other errors

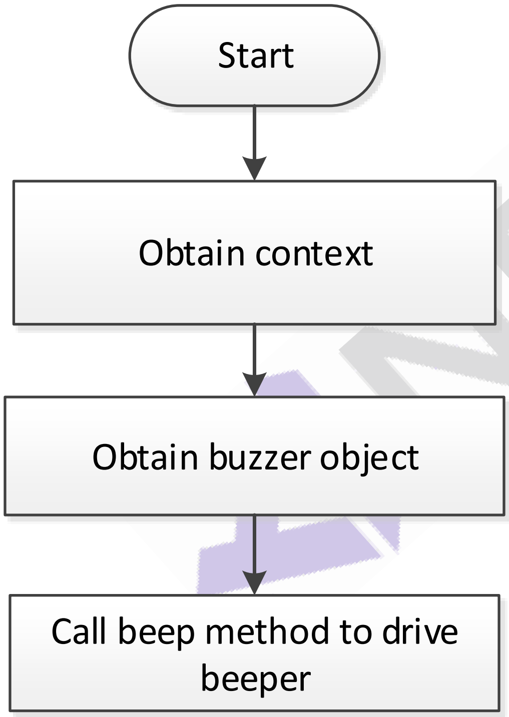
3.13 Buzzer class

Class is responsible for managing POS buzzer.

Get buzzer objects of class:

Beeper beep = deviceEngine.getBeeper();

This module uses the basic flow chart:



3.13.1 beep

Drive the buzzer sound duration specified length of time.

public void beep (int timeout);

Parameters:

Parameter	Description
timeout	Timeout in milliseconds. Zero immediately stops

Return Value: None

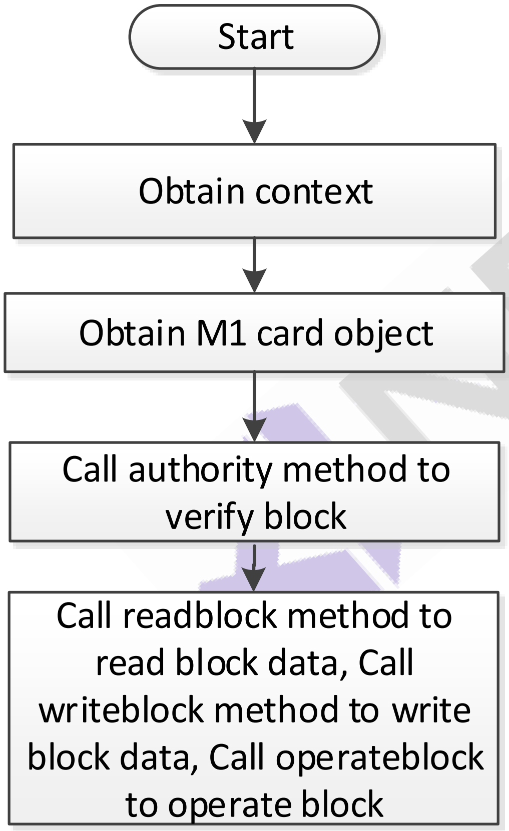
3.14 M1 Cards

M1 card class is responsible for managing M1 card.

Get M1 card class objects:

M1CardHandler m1Card = deviceEngine.getM1CardHandler();

This module uses the basic flow chart:


Start
Obtain context
Obtain M1 card object
Call authority method to verify block
Call readblock method to read block data, Call writeblock method to write block data, Call operateblock to operate block

3.14.1 authority

Block certification.

public int authority (Authentity entity) ;

Parameters:

Parameter	Description
entity	AuthEntity, the authentication information class

AuthEntity

Attributes	Description
int blkNo	Block number
M1KeyTypeEnum keyType	Key type enumeration
byte [] pwd	Password authentication
String m1SN	M1 card uid

M1KeyTypeEnum

Enumeration Name	Description
KEYTYPE_A	KEY A
KEYTYPE_B	KEY B

Return Value:

SdkResult.Success success

SdkResult.Device_Not_Ready device is not ready

SdkResult.Param_In_InValid Parameter is not legitimate

SdkResult.M1Card_Verify_Err M1 card authentication failure

SdkResult.Fail other errors

3.14.2 readBlock

Read block data.

public int readBlock (Blockentity entity) ;

Parameters:

Parameter	Description
entity	BlockEntity block Info

BlockEntity

Attributes	Description
M1CardOperTypeEnum operType	Operation enumeration
byte [] blkData	Data to be operated
int BLKNO	Block number to be operated
int desBlkNo	Destination block number

M1CardOperTypeEnum

Enumeration Name	Description
INCREMENT	Increment
DECREMENT	Decrement operation
BACKUP	Backup

Return Value:

SdkResult.Success success

SdkResult.Device_Not_Ready device is not ready

SdkResult.Param_In_InValid Parameter is not legitimate

SdkResult.Fail other errors

3.14.3 readBlockValue

Read block value

public int readblockValue(Blockentity entity);

Parameters:

Parameter	Description
entity	BlockEntity block Info

BlockEntity

Attributes	Description
M1CardOperTypeEnum operType	Operation enumeration
byte[] blkData	Data to be operated
int blkValue	Read and write block data values in M1 card data format
int BLKNO	Block number to be operated
int desBlkNo	Destination block number

M1CardOperTypeEnum

Enumeration Name	Description
INCREMENT	Increment
DECREMENT	Decrement operation
BACKUP	Backup

Return Value:

SdkResult.Success success

SdkResult.Device_Not_Ready device is not ready

SdkResult.Param_In_InValid Parameter is not legitimate

SdkResult.Fail other errors

3.14.4 writeBlock

Write block data.

public int writeBlock (Blockentity entity) ;

Parameters:

Parameter	Description
entity	BlockEntity block Info

BlockEntity

Attributes	Description
M1CardOperTypeEnum operType	Operation enumeration
byte [] blkData	Data to be operated
int BLKNO	Block number to be operated
int desBlkNo	Destination block number

M1CardOperTypeEnum

Enumeration Name	Description
INCREMENT	Increment
DECREMENT	Decrement operation
BACKUP	Backup

Return Value:

SdkResult.Success success

SdkResult.Device_Not_Ready device is not ready

SdkResult.Param_In_InValid Parameter is not legitimate

SdkResult.Fail other errors

3.14.5 writeBlockValue

Write block value

public int writeblock(Blockentity entity);

Parameters:

Parameter	Description
entity	BlockEntity block Info

BlockEntity

Attributes	Description
M1CardOperTypeEnum operType	Operation enumeration
byte[] blkData	Data to be operated
int blkValue	Read and write block data values in M1 card data format
int BLKNO	Block number to be operated
int desBlkNo	Destination block number

M1CardOperTypeEnum

Enumeration Name	Description
INCREMENT	Increment
DECREMENT	Decrement operation
BACKUP	Backup

Return Value:

SdkResult.Success success

SdkResult.Device_Not_Ready device is not ready

SdkResult.Param_In_InValid Parameter is not legitimate

SdkResult.Fail other errors

3.14.6 operateBlock

Operation block data.

public int operateblock (Blockentity entity) ;

Parameters:

Parameter	Description
entity	BlockEntity block Info

BlockEntity

Attributes	Description
M1CardOperTypeEnum operType	Operation enumeration
byte [] blkData	Data to be operated
int BLKNO	Block number to be operated
int desBlkNo	Destination block number

M1CardOperTypeEnum

Enumeration Name	Description
INCREMENT	Increment
DECREMENT	Decrement operation
BACKUP	Backup

Return Value:

SdkResult.Success success

SdkResult.Device_Not_Ready device is not ready

SdkResult.Param_In_InValid Parameter is not legitimate

SdkResult.Fail other errors

3.15 MemoryCard

MemoryCard class is responsible for managing MemoryCard.

Get MemoryCard card class objects:

MemoryCard memoryCard = deviceEngine.getMemoryCardHandler (CardSlotTypeEnum slotType);

Parameters:

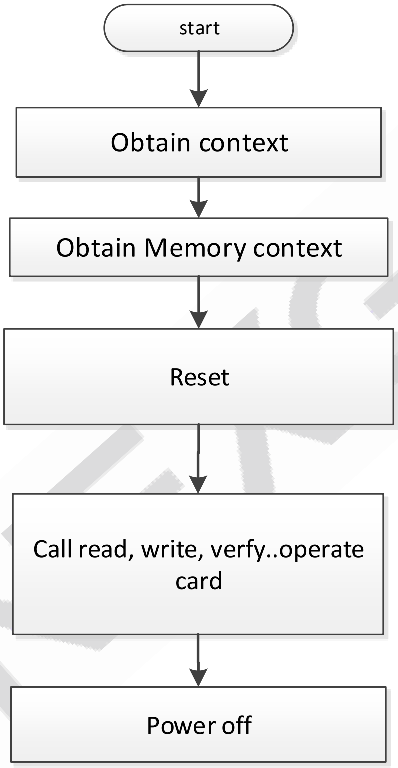
Parameter	Description
slotTypes	Slot enumerated type CardSlotTypeEnum; supports a variety of combinations of slots

CardSlotTypeEnum

Enumeration Name	Description
ICC1	Default IC card slot
ICC2	Unavailable
ICC3	Unavailable
PSAM1	PSAM slot 1
PSAM2	PSAM slot 2
PSAM3	Unavailable

RF	Non-access card slot
SWIPE	Magnetic stripe card slot

This module uses the basic flow chart:



3.15.1 reset

reset

public int reset(CardTypeEnum cardType);

Parameters:

Parameter	**Description**
cardType	Card type

CardTypeEnum

Enumeration Name	**Description**

AT24C01	
AT24C02	
AT24C04	
AT24C08	
AT24C16	
AT24C32	
AT24C64	
AT88SC101	
AT88SC102	
IS23SC1604	
AT88SC153	
AT88SC1608	
SLE4442	
SLE4428	

Return Value:

SdkResult.Success success

SdkResult.Fail other errors

3.15.2 read

read

public byte[] read(ReadEntity read);

Parameters:

Parameter	**Description**
read	ReadEntity

ReadEntity

attribute	**Description**
CardTypeEnum cardType	Card type

int zone	SLE4428 means protection bit mode 0: no protection bit, 1: protection bit SLE4442 means the storage area, 0: main storage area, 1: protection area AT88SC153,AT88SC1608 means the partition number IS23SC1604 means area code
int address	The starting address, starting at 0
int readLen	Len of read data

CardTypeEnum

Enumeration Name	Description
AT24C01	
AT24C02	
AT24C04	
AT24C08	
AT24C16	
AT24C32	
AT24C64	
AT88SC101	
AT88SC102	
IS23SC1604	
AT88SC153	
AT88SC1608	
SLE4442	
SLE4428	

Return Value:

Success return byte[]

Fail return null

3.15.3 write

write

public int write(WriteEntity write);

Parameters:

Parameter	Description
write	WriteEntity

WriteEntity

attribute	Description
CardTypeEnum cardType	Card type
int zone	SLE4428 means protection bit mode 0: no protection bit, 1: protection bit SLE4442 means the storage area, 0: main storage area, 1: protection area AT88SC153,AT88SC1608 means the partition number IS23SC1604 means area code
int address	The starting address, starting at 0
byte[] writeData	Write data
int writeLen	Len of write data

CardTypeEnum

Enumeration Name	Description
AT24C01	
AT24C02	
AT24C04	
AT24C08	
AT24C16	
AT24C32	
AT24C64	
AT88SC101	
AT88SC102	
IS23SC1604	
AT88SC153	
AT88SC1608	
SLE4442	
SLE4428	

Return Value:

SdkResult.Success success

SdkResult.Fail other errors

3.15.4 erase

erase

public int earse(EraseEntity erase);

Parameters:

Parameter	Description
erase	EraseEntity

EraseEntity

attribute	Description
CardTypeEnum cardType	Cardtype ,only support IS23SC1604,AT88SC101,AT88SC102
int address	The starting address, starting at 0
int eraseLen	Erase data length, unit byte
int zone	Zone number

CardTypeEnum

Enumeration Name	Description
AT24C01	
AT24C02	
AT24C04	
AT24C08	
AT24C16	
AT24C32	
AT24C64	
AT88SC101	
AT88SC102	
IS23SC1604	
AT88SC153	
AT88SC1608	
SLE4442	
SLE4428	

Return Value:

SdkResult.Success success

SdkResult.Param_In_Invalid invaild param

SdkResult.Fail other errors

3.15.5 verify

verify card pin.

public int verify(VerifyEntity verify);

Parameters:

Parameter	Description
verify	VerifyEntity

VerifyEntity

attribute	Description
CardTypeEnum cardType	cardtype ,only support SLE4428,SLE4442,AT88SC153,AT88SC1608,IS23SC1604,AT88SC101,AT88SC102
byte[] pwd	password
int mode	AT88SC153,AT88SC1608 means check mode, 0: read check ;1: write check AT88SC101 AT88SC102, IS23SC1604 means check content 0: security code ;1: erase the password
int zone	AT88SC153,AT88SC1608 means password index AT88SC101 AT88SC102, IS23SC1604 means area ,0: the whole storage area; 1 ~ n: application code

CardTypeEnum

Enumeration Name	Description
AT24C01	
AT24C02	
AT24C04	
AT24C08	
AT24C16	
AT24C32	
AT24C64	
AT88SC101	
AT88SC102	
IS23SC1604	
AT88SC153	
AT88SC1608	
SLE4442	
SLE4428	

Return Value:

SdkResult.Success the remaining password verification times

SdkResult.Param_In_Invalid invaild param

SdkResult.Fail other errors

3.15.6 readEC

Read remaining password check times.

public int readEC(ReadECEntity readEC);

Parameters:

Parameter	Description
readEC	ReadECEntity

ReadECEntity

attribute	Description
CardTypeEnum cardType	card type , only support SLE4428,SLE4442,AT88SC153,AT88SC1608,IS23SC1604,AT88SC101,AT88SC102
int mode	AT88SC153,AT88SC1608 means check mode, 0: read check ;1: write check AT88SC101 AT88SC102, IS23SC1604 means check content 0: security code ;1: erase the password
int zone	AT88SC153,AT88SC1608 means password index AT88SC101 AT88SC102, IS23SC1604 means area ,0: the whole storage area; 1 ~ n: application code

CardTypeEnum

Enumeration Name	Description
AT24C01	
AT24C02	
AT24C04	
AT24C08	
AT24C16	
AT24C32	

AT24C64	
AT88SC101	
AT88SC102	
IS23SC1604	
AT88SC153	
AT88SC1608	
SLE4442	
SLE4428	

Return Value:

SdkResult.Success the remaining password verification times

SdkResult.Param_In_Invalid invaild param

SdkResult.Fail other errors

3.15.7 updateEC

Modify card password

public int updateEC(UpdateECEntity updateEC);

Parameters:

Parameter	Description
readEC	UpdateECEntity

UpdateECEntity

attribute	Description
CardTypeEnum cardType	Card type, only support SLE4428,SLE4442,AT88SC153,AT88SC1608,IS23SC1604,AT88SC101,AT88SC102
byte[] pwd	password
int mode	AT88SC153,AT88SC1608 means check mode, 0: read check ;1: write check AT88SC101 AT88SC102, IS23SC1604 means check content 0: security code ;1: erase the password
int zone	AT88SC153,AT88SC1608 means password index AT88SC101 AT88SC102, IS23SC1604 means area ,0: the whole storage area; 1 ~ n: application code

CardTypeEnum

Enumeration Name	Description
AT24C01	
AT24C02	
AT24C04	
AT24C08	
AT24C16	
AT24C32	
AT24C64	
AT88SC101	
AT88SC102	
IS23SC1604	
AT88SC153	
AT88SC1608	
SLE4442	
SLE4428	

Return Value:

SdkResult.Success

SdkResult.Param_In_Invalid invaild param

SdkResult.Fail other errors

3.15.8 powerOff

poweroff

public void powerOff();

Return Value:

None

3.16 Desfire Cards

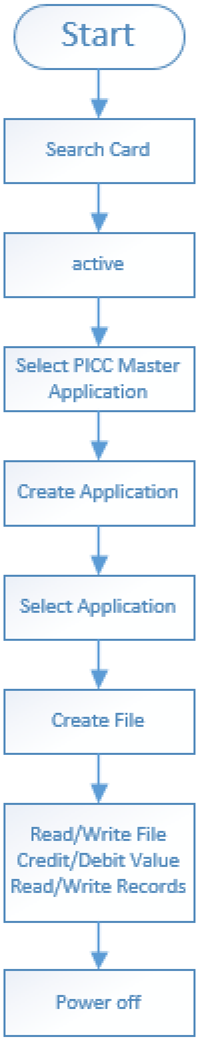
DesfireHandler is responsible for interacting with Desfire card.

Note: Currently, only N5 can support desfire card.

Get Desfire card handler Object:

DesfireHandler desfireHandler = deviceEngine.getDesfireHandler();

This module uses the basic flow chart:


Start
Search Card
active
Select PICC Master Application
Create Application
Select Application
Create File
Read/Write File
Credit/Debit Value
Read/Write Records
Power off

3.16.1 Authenticate

Prototype	int authenticate(byte keyNo, byte[] key);	
Function	confirms that both entities (PICC and PCD) can trust each other, DES/3DES algorithm	
Parameters	keyNo	the key no used to authentication process
	Key	key used for authentication, 16bytes need. if the actual key is only 8bytes long, then should extended to 16bytes: key[0...7] || key[0...7].
Return	SdkResult.Success	Success
	SdkResult.Fail	Fail
	SdkResult.Param_In_Invalid	Parameter is Invalid
	SdkResult.TimeOut	TimeOut

3.16.2 AuthenticateIso

Prototype	int authenticateIso(byte keyNo, byte[] key);	
Function	confirms that both entities (PICC and PCD) can trust each other, DES/3DES /3KDES algorithm	
Parameters	keyNo	the key no used to authentication process
	Key	key used for authentication, 24bytes need. if the actual key is only 8bytes long, then should extended to 24bytes: key[0...7] || key[0...7] || key[0...7]

		if the actual key is only 16bytes long, then should extended to 24bytes: key[0...7] | | key[8...15] | | key[0...7]
Return	SdkResult.Success	Success
	SdkResult.Fail	Fail
	SdkResult.Param_In_Invalid	Parameter is Invalid
	SdkResult.TimeOut	TimeOut

3.16.3 AuthenticateAes

Prototype	int authenticateAes(byte keyNo, byte[] key);	
Function	confirms that both entities (PICC and PCD) can trust each other, AES algorithm	
Parameters	keyNo	the key no used to authentication process
	Key	key used for authentication, 16bytes need.
Return	SdkResult.Success	Success
	SdkResult.Fail	Fail
	SdkResult.Param_In_Invalid	Parameter is Invalid
	SdkResult.TimeOut	TimeOut

3.16.4 changeKeySettings

Prototype	int changeKeySettings(byte keySettings);	
Function	Changes the master key configuration settings depending on the currently selected AID.	
Parameters	keySettings	------ for PICC master key: bit7~bit4: 0000 RFU

		bit3: whether a change of the PICC master key settings is allowed bit2: whether PICC master key authentication is needed before Create- / DeleteApplication bit1: whether PICC master key authentication is needed for application directory access bit0: whether the PICC master key is changeable ------ for Application master key: bit7~bit4: hold the Access Rights for changing application keys (ChangeKey command) 0x0: Application master key authentication is necessary to change any key (default) 0x1~0x0D: Authentication with the specified key is necessary to change any key. 0x0E: Authentication with the key to be changed (same KeyNo) is necessary to change a key. 0x0F: All Keys (except application master key, see Bit0) within this application are frozen. bit3: whether a change of the application master key settings is allowed bit2: whether application master key authentication is needed before CreateFile / DeleteFile bit1: whether application master key authentication is needed for file directory access bit0: whether the application master key is changeable
Return	SdkResult.Success	Success
	SdkResult.Fail	Fail

	SdkResult.Param_In_Invalid	Parameter is Invalid
	SdkResult.TimeOut	TimeOut

3.16.5 getKeySettings

Prototype	KeySettingsEntity getKeySettings();	
Function	get configuration information on the PICC and application master key configuration settings, and get maximum number of keys which can be stored within the selected application. Depending on the master key settings, a preceding authentication with the master key is required.	
Parameters	null	
Return	KeySettingsEntity	getKeySettings current master key setting
		getMaxKeyNum maximum number of keys which can be stored within the selected application

3.16.6 changePiccMasterkey

Prototype	int changePiccMasterkey(KeyTypeEnum masterKeyType, byte[] key, byte aesVersion);
Function	change PICC master key 1. according to PICC master key setting, a authentication with PICC master key is necessary

KeyTypeEnum

Enumeration Name	Description

	2. After a successful change of the key used to reach the current authentication status, this authentication is invalidated i.e. an authentication with the new key is necessary for subsequent operations.	
Parameters	masterKeyType	PICC master key type
	key	key information (16/24 bytes)
	aesVersion	key version, only valid when type = {@link KeyTypeEnum#AES}.
Return	SdkResult.Success	Success
	SdkResult.Fail	Fail
	SdkResult.Param_In_Invalid	Parameter is Invalid
	SdkResult.TimeOut	TimeOut

DES_TDES_128	16bytes DES/3DES key
KEYTYPE_B	24bytes 3KDES key
AES	16bytes AES key

3.16.7 changeAppKey

Prototype	int changeAppKey(KeyTypeEnum appKeyType, byte keyNo, byte[] oldKey, byte[] newKey, byte aesVersion);	
Function	change application master key 1. according to application master key setting, a authentication with specified key is necessary 2. After a successful change of the key used to reach the current authentication status, this authentication is invalidated i.e. an authentication with the new key is necessary for subsequent operations.	
Parameters	appKeyType	app key type{ KeyTypeEnum}
	keyNo	the key to change
	oldKey	old key value

	newKey	new key value
	aesVersion	key version, only valid when type = {@link KeyTypeEnum#AES}.
Return	SdkResult.Success	Success
	SdkResult.Fail	Fail
	SdkResult.Param_In_Invalid	Parameter is Invalid
	SdkResult.TimeOut	TimeOut

3.16.8 getKeyVersion

Prototype	byte getKeyVersion(byte keyNo);	
Function	read out the current key version of any key stored on the PICC This command can be issued without valid authentication.	
Parameters	keyNo	key no
Return	key version of this key	

3.16.9 createApplication

Prototype	int createApplication(ApplicationEntity application);
Function	create new applications on the PICC. 1. Application Identifier 0x00 00 00 is reserved as a reference to the PICC itself. 2. After application be created, All keys are initialised with a string consisting of 0x00 bytes 3. Before any setup of a file system, it is recommemded to configure the whole card using the command 'SetConfiguration'. This command will initialize all keys of any created application to a specified value which is taken out of the default key

ApplicationEntity

	and default version from the 'SetConfiguration' command. Without this command all keys are consisting of 0x00 bytes.	
Parameters	application	{@link ApplicationEntity}
Return	SdkResult.Success	Success
	SdkResult.Fail	Fail
	SdkResult.Param_In_Invalid	Parameter is Invalid
	SdkResult.TimeOut	TimeOut

Attributes	Description
byte[] aid	application identifier
byte[] isoFid	2 byte ISO/IEC 7816-4 File Identifies for files within the application
byte[]dfName	"DF-name" used in 7816-4 mode in combination with the ISO SELECT command
Byte masterKeySetting	Application master key setting
byte numberOfKey	Number of keys that can be stored within the application for cryptographic purposes. A maximum of 14 keys can be stored within an application of MIFARE DESFire EV1. One can also create an application with no keys
KeyTypeEnum keyType	the key type of application
boolean isSupFid	whether use File Identifies
boolean isSupIsoFid	whether use of 2 byte ISO/IEC 7816-4 File Identifies for files within the Application

3.16.10 deleteApplication

Prototype	int deleteApplication(byte[] aid);
Function	allows to permanently deactivate applications on the PICC

	1. Depending on the PICC master key settings, an PICC master key authentication is required. 2. The AID allocation is removed, therefore it is possible to create a new application with the deleted application's AID. However, the deleted memory blocks can only be recovered by using the FormatPICC command which erases the full user memory of the PICC.	
Parameters	aid	application Identifies (3bytes)
Return	SdkResult.Success	success
	SdkResult.Fail	fail
	SdkResult.Param_In_Invalid	Parameter is not legitimate
	SdkResult.TimeOut	timeout

3.16.11 getAids

Prototype	List<byte[]> getAids();	
Function	returns the Application IDentifiers of all active applications on a PICC. Depending on the PICC master key settings a successful authentication with the PICC master key might be required to execute this command.	
Parameters	null	
Return	if error return empty list.	

3.16.12 getDfNames

Prototype	List<DfNameEntity> getDfNames();
Function	Returns the ISO/IEC 7816-4 DF-Names of all active applications on a PICC Depending on the PICC master key settings a successful authentication with the PICC master key might be required to execute this command.

Parameters	null	
Return	if error return empty list.	

3.16.13 selectApplication

Prototype	int selectApplication(byte[] aid);	
Function	select one specific application for further access. 1. each SelectApplication command invalidates the current authentication status. 2. If this pAID is 0x00 00 00, the PICC level is selected and any further operations (typically commands like CreateApplication, DeleteApplication) are related to this level. 3. If an application with the specified AID is found in the application directory of the PICC, the subsequent commands interact with this application.	
Parameters	aid	Application Identifier (3bytes)
Return	SdkResult.Success	success
	SdkResult.Fail	fail
	SdkResult.Param_In_Invalid	Parameter is not legitimate
	SdkResult.TimeOut	timeout

3.16.14 formatPicc

Prototype	int formatPicc();
Function	This command releases the PICC user memory. 1. The FormatPICC Command releases all allocated user memory on the PICC. 2. All applications are deleted and all files within those applications are deleted.

	3. This command always requires a preceding authentication with the PICC master key. 4. The PICC master key and the PICC master key settings keep their currently set values, they are not influenced by this command.	
Parameters	null	
Return	SdkResult.Success	success
	SdkResult.Fail	fail
	SdkResult.Param_In_Invalid	Parameter is not legitimate
	SdkResult.TimeOut	timeout

3.16.15 getVersion

Prototype	VersionEntity getVersion();	
Function	Return manufacturing related data of the PICC.	
Parameters	null	
Return	VersionEntity	{@link VersionEntity }

VersionEntity

Attributes	**Description**
byte hwVendorId	codes the vendor ID (0x04 for PHILIPS)
byte hwType	codes the type (here 0x01)
byte hwSubType	codes the subtype (here 0x01)
byte hwMajorVer	codes the major version number
byte hwMinorVer	codes the minor version number
byte hwSize	codes the storage size (here 0x1A = 8192 bytes)

byte hwProtocol	codes the communication protocol type (here 0x05 meaning ISO 14443-2 and -3)
byte swVendorId	codes the vendor ID (here 0x04 for PHILIPS)
byte swType	codes the type (here 0x01)
byte swSubType	codes the subtype (here 0x01)
byte swMajorVer	codes the major version
byte swMinorVer	codes the minor version
byte swSize	codes the storage size (here 0x1A = 8192 bytes)
byte swProtocol	codes the communication protocol type (here 0x05 meaning ISO 14443-3 and -4)
byte[] uid	code the unique serial number
byte[] batchNo	code the production batch number
byte weekOfProduction	codes the calendar week of production
byte yearOfProduction	codes the year of production

3.16.16 getFreeMemory

Prototype	int getFreeMemory();	
Function	Returns the available bytes on the PICC	
Parameters	null	
Return	SdkResult.Success	success
	SdkResult.Fail	fail
	SdkResult.Param_In_Invalid	Parameter is not legitimate
	SdkResult.TimeOut	timeout

3.16.17 setConfiguration

Prototype	int setConfiguration(byte option, byte[] info);	
Function	set PICC configuration PICC master key authentication on card level needs to be performed prior to this command.	
Parameters	option	configuration type, value as following 0x00: info data is the configuration byte 0x01: info data is the default key version and default key all applications will be personalized during creation with this default key and version instead of 0x00 0x02: info data is the user defined ATS 0xxx: RF
	info	configuration information, according to option if option = 0x00, the configuration byte showed as following: bit0 = 0 Format card enabled bit0 = 1 Format card disabled;can not be reset bit1 = 0 Random ID disabled bit1 = 1 Random ID enabled; can not be reset if option = 0x01, the *info should be 24bytes key and 1byte default version
Return	SdkResult.Success	success
	SdkResult.Fail	fail
	SdkResult.Param_In_Invalid	Parameter is not legitimate
	SdkResult.TimeOut	timeout

3.16.18 getCardUid

Prototype	byte[] getCardUid();	
Function	return the uid of PICC 1. An authentication with any key needs to be performed prior to this command 2. This command is only available when {@link DesfireHandler#authenticateIso(byte, byte[])} or {@link DesfireHandler#authenticateAes(byte, byte[])} called	
Parameters	null	
Return	uid information of PICC (7bytes)	

3.16.19 getFids

Prototype	List<byte[]> getFids();	
Function	returns the File IDentifiers of all active files within the currently selected application. 1. Depending on the application master key settings, a preceding authentication with the application master key might be required. 2. Each File ID is coded in one byte and is in the range from 0x00 to 0x1F.	
Parameters	null	
Return	if error return empty list.	

3.16.20 getIsoFids

Prototype	List<byte[]> getIsoFids();

Function	Returns the 2 byte ISO/IEC 7816-4 File IDentifiers of all active files within the currently selected application 1. Depending on the application master key settings, a preceding authentication with the application master key might be required. 2. Each ISO File ID is coded in two byte .	
Parameters	null	
Return	if error return empty list.	

3.16.21 getFileSettings

Prototype	FileSettingsEntity getFileSettings(byte fileNo);	
Function	get information on the properties of a specific file. 1. This file number must be in the range between 0x00 and 0x1F. 2. Depending on the application master key settings, a preceding authentication with the application master key might be required. 3. After updating a value file's value but before issuing the CommitTransaction command, the GetFileSettings command will always retrieve the old, unchanged limit for the limited credit value.	
Parameters	fileNo	the specific file no, value 0~0x1F allowed
Return	FileSettingsEntity	

FileSettingsEntity

Attributes	Description
byte fileType	DESfire file type: 0x00 --- Standard Data Files 0x01 --- Backup Data Files 0x02 --- Value Files wih Backup 0x03 --- Linear record Files with Backup 0x04 --- Cyclic Record Files with Backup

byte commSettings	0x00 or 0x02 --- Plain communication 0x01 --- Plain communication secured by MACing 0x03 --- Fully enciphered communication
byte readAccessRightKeyNum	Access right capability, 0x0E means free access, and 0x0F means deny access. the reference number of the key which needs to be authenticated prior to Read Access and Read&Write Access
byte writeAccessRightKeyNum	the reference number of the key which needs to be authentication prior to Write Access and Read&Write Access
Byte readAndWriteAccessRightKeyNum	the reference number of the key which needs to be authentication prior to Read&Write Access
byte changeAccessRightKeyNum	the reference number of the key,which is necessary to be authenticated with in order to change the access rights for the file and to link each access right to key numbers
int fileSize	the user file size in bytes, only available when file_type = 0x00 or file_type = 0x01
int lowerLimit	lower limit of the value file ,only available when file_type = 0x02
int upperLimit	upper limit of the value file,only available when file_type = 0x02
the current maximum" limited credit" value	limitedCreditValue,only available when file_type = 0x02
boolean limitedCreditEnabled	if the LimitedCredit command is allowed for this file,only available when file_type = 0x03 or file_type = 0x04
int recordSize	the size of one single record (as deefined at file creation),only available when file_type = 0x03 or file_type = 0x04
int maxNumberOfRecords	the maximum number of records within the record file (as defined at file creation),only available when file_type = 0x03 or file_type = 0x04
int currentNumberOfRecords	the current number of records within the record file,only available when file_type = 0x03 or file_type = 0x04

3.16.22 changeFileSettings

Prototype	int changeFileSettings(byte fileNo, byte commSettings, byte newReadAccessKeyNum, byte newWriteAccessKeyNum, byte newReadAndWriteAccessKeyNum, byte newChangeAccessKeyNum);	
Function	changes the access parameters of an existing file 1. This change only succeeds if the current "Change Access Right" is different from "never", that is old_change_access_keyno != 0x0E 2. To guarantee that the ChangeFileSettings command is coming from the same party which did the preceding authentication, it is necessary to apply basically the same security mechanism as used with the ChangeKey command	
Parameters	fileNo	the specific file no, value 0~0x1F allowed
	commSettings	0x00 or 0x02 --- Plain communication 0x01 --- Plain communication secured by MACing 0x03--- Fully enciphered communication
	newReadAccessKeyNum	new Read Access Right Key No
	newWriteAccessKeyNum	new Write Access Right Key No
	newReadAndWriteAccessKeyNum	new Read and Write Access Right Key No
	newChangeAccessKeyNum	new Change Access Right Key No
Return	SdkResult.Success	success
	SdkResult.Fail	fail
	SdkResult.Param_In_Invalid	Parameter is not legitimate
	SdkResult.TimeOut	timeout

3.16.23 createStdDataFile

Prototype	int createStdDataFile(byte fileNo, DataFileEntity dataFile);

Function	create files for the storage of plain unformatted user data within an existing application on the PICC	
Parameters	fileNo	the specific file no, value 0~0x1F allowed
	dataFile	file settings {@link DataFileEntity}
Return	SdkResult.Success	success
	SdkResult.Fail	fail
	SdkResult.Param_In_Invalid	Parameter is not legitimate
	SdkResult.TimeOut	timeout

DataFileEntity

Attributes	Description
boolean isoFidEnable	whether ISO/IEC 7816-4 File IDentifiers enabled (0x00 - disabled, 0x01-enabled)
byte[] isoFid	2bytes ISO/IEC 7816-4 File IDentifiers
commSettings	0x00 or 0x02 --- Plain communication 0x01 --- Plain communication secured by MACing 0x03--- Fully enciphered communication
byte readAccessRightKeyNum	Access right capability, 0x0E means free access, and 0x0F means deny access. the reference number of the key which needs to be authenticated prior to Read Access and Read&Write Access
byte writeAccessRightKeyNum	the reference number of the key which needs to be authentication prior to Write Access and Read&Write Access
Byte readAndWriteAccessRightKeyNum	the reference number of the key which needs to be authentication prior to Read&Write Access
byte changeAccessRightKeyNum	the reference number of the key,which is necessary to be authenticated with in order to change the access

	rights for the file and to link each access right to key numbers
int fileSize	the user file size in bytes, only available when file_type = 0x00 or file_type = 0x01

3.16.24 createBackupDatafile

Prototype	int createBackupDatafile(byte fileNo, DataFileEntity dataFile);	
Function	create files for the storage of plain unformatted user data within an existing application on the PICC, additionally supporting the feature of an integrated backup mechanism 1. Due to the mirror image a BackupDataFile always consumes DOUBLE the NV-memory on the PICC compared to a StdDataFile with the same specified FileSize. 2. Every Write command is done in a independent mirror image of this file. To validate a write access to this file type, it is necessary to confirm it with a CommitTransaction command. If no CommitTransaction command is send by the PCD, only the mirror image is changed, the original data remains unchanged and valid.	
Parameters	fileNo	the specific file no, value 0~0x1F allowed
	dataFile	file settings {@link DataFileEntity}
Return	SdkResult.Success	success
	SdkResult.Fail	fail
	SdkResult.Param_In_Invalid	Parameter is not legitimate
	SdkResult.TimeOut	timeout

3.16.25 createValueFile

Prototype	int createValueFile(byte fileNo, ValueFileEntity valueFile);

Function	create files for the storage and manipulation of 32bit signed integer values within an existing application on the PICC ValueFiles feature always the integrated backup mechanism. Therefore every access changing the value needs to be validated using the CommitTransaction command 1. It is necessary to validate the updated value with a CommitTransaction command. An AbortTransaction command will invalidate all changes 2. The value modifications of Credit, Debit and LimitedCredit commands are cumulated until a CommitTransaction command is issued.	
Parameters	fileNo	the specific file no, value 0~0x1F allowed
	valueFile	file settings {@link ValueFileEntity }
Return	SdkResult.Success	success
	SdkResult.Fail	fail
	SdkResult.Param_In_Invalid	Parameter is not legitimate
	SdkResult.TimeOut	timeout

ValueFileEntity

Attributes	Description
commSettings	0x00 or 0x02 --- Plain communication 0x01 --- Plain communication secured by MACing 0x03--- Fully enciphered communication
byte readAccessRightKeyNum	Access right capability, 0x0E means free access, and 0x0F means deny access. the reference number of the key which needs to be authenticated prior to Read Access and Read&Write Access
byte writeAccessRightKeyNum	the reference number of the key which needs to be authentication prior to Write Access and Read&Write Access

Byte readAndWriteAccessRightKeyNum	the reference number of the key which needs to be authentication prior to Read&Write Access
byte changeAccessRightKeyNum	the reference number of the key,which is necessary to be authenticated with in order to change the access rights for the file and to link each access right to key numbers
int lowerLimit	lower limit of the value file, only available when file_type = 0x02
int upperLimit	upper limit of the value file, only available when file_type = 0x02
int initValue	the initial value of this value file, only available when file_type = 0x02
boolean limitedCreditEnabled	if the LimitedCredit command is allowed for this file, only available when file_type = 0x02

3.16.26 createLinearRecordFile

Prototype	int createLinearRecordFile(byte fileNo, RecordFileEntity recordFile);	
Function	create files for multiple storage of structural data, for example for loyalty programs, within an existing application on the PICC 1. Once the file is filled completely with data records, further writing to the file is not possible unless it is cleared, see command ClearRecordFile. 2. Linear Record Files feature always the integrated backup mechanism. Therefore every access appending a record needs to be validated using the CommitTransaction command	
Parameters	fileNo	the specific file no, value 0~0x1F allowed
	valueFile	file settings {@link ValueFileEntity }
Return	SdkResult.Success	success
	SdkResult.Fail	fail
	SdkResult.Param_In_Invalid	Parameter is not legitimate

	SdkResult.TimeOut	timeout

RecordFileEntity

Attributes	Description
boolean isoFidEnable	whether ISO/IEC 7816-4 File IDentifiers enabled (0x00 - disabled, 0x01-enabled)
byte[] isoFid	2bytes ISO/IEC 7816-4 File IDentifiers
commSettings	0x00 or 0x02 --- Plain communication 0x01 --- Plain communication secured by MACing 0x03--- Fully enciphered communication
byte readAccessRightKeyNum	Access right capability, 0x0E means free access, and 0x0F means deny access. the reference number of the key which needs to be authenticated prior to Read Access and Read&Write Access
byte writeAccessRightKeyNum	the reference number of the key which needs to be authentication prior to Write Access and Read&Write Access
Byte readAndWriteAccessRightKeyNum	the reference number of the key which needs to be authentication prior to Read&Write Access
byte changeAccessRightKeyNum	the reference number of the key,which is necessary to be authenticated with in order to change the access rights for the file and to link each access right to key numbers
int recordSize	the size of one single record (as deefined at file creation), only available when file_type = 0x03 or file_type = 0x04
int maxNumberOfRecords	the maximum number of records within the record file (as defined at file creation) , only available when file_type = 0x03 or file_type = 0x04
byte specifiesRandomWriteAccessOption	whether specifies Random write access option, (0x00 - not, 0x01 - yes), only available when file_type = 0x03 or file_type = 0x04
boolean allowedRandomWriteAccess	whether allowed Random write access, only available when file_type = 0x03 or file_type = 0x04

3.16.27 createCyclicRecordFile

Prototype	int createCyclicRecordFile(byte fileNo, RecordFileEntity recordFile);	
Function	create files for multiple storage of structural data, for example for loyalty programs, within an existing application on the PICC 1. Once the file is filled completely with data records, further writing to the file is not possible unless it is cleared, see command ClearRecordFile. 2. Linear Record Files feature always the integrated backup mechanism. Therefore every access appending a record needs to be validated using the CommitTransaction command	
Parameters	fileNo	the specific file no, value 0~0x1F allowed
	valueFile	file settings {@link ValueFileEntity }
Return	SdkResult.Success	success
	SdkResult.Fail	fail
	SdkResult.Param_In_Invalid	Parameter is not legitimate
	SdkResult.TimeOut	timeout

3.16.28 deleteFile

Prototype	int deleteFile(byte fileNo);
Function	permanently deactivates a file within the file directory of the currently selected application. 1. The operation of this command invalidates the file directory entry of the specified file which means that the file can't be accessed anymore.

	2. Depending on the application master key settings, a preceding authentication with the application master key is required. 3. Allocated memory blocks associated with the deleted file are not set free. The FileNo of the deleted file can be re-used to create a new file within that application. 4. To release memory blocks for re-use, the whole PICC user NV-memory needs to be erased using the FormatPICC command.	
Parameters	fileNo	the file number within the file directory of the currently selected application.
Return	SdkResult.Success	success
	SdkResult.Fail	fail
	SdkResult.Param_In_Invalid	Parameter is not legitimate
	SdkResult.TimeOut	timeout

3.16.29 readData

Prototype	byte[] readData(byte fileNo, byte commSettings, int offset, int len);	
Function	Read data from Standard Data Files or Backup Data Files 1. This offset has to be in the range from 0 to file size -1. 2. If the len is coded as 0, the entire data file, starting from the position specified in the offset value, is read. 3. If Backup Data Files are read after writing to them, but before issuing the CommitTransaction command, the ReadData command will always retrieve the old, unchanged data stored in the PICC. All data written to a Backup Data File is validated and externally "visible" for a ReadData command only after a CommitTransaction command. 4. The Read command requires a preceding authentication either with the key specified for "Read" or "Read&Write" access	
	fileNo	the file number

Parameters	commSettings	0x00 or 0x02 --- Plain communication 0x01 --- Plain communication secured by MACing 0x03 --- Fully enciphered communication
	offset	the starting position for the read operation within the file
	len	the number of data bytes want to be read
Return	return the out data	

3.16.30 writeData

Prototype	int writeData(byte fileNo, byte commSettings, int offset, byte[] data);	
Function	Write data to Standard Data Files and Backup Data Files. 1. The Write command requires a preceding authentication either with the key specified for "Write" or "Read&Write" access. 2. If the WriteData operation is performed on a Backup Data File, it is necessary to validate the written data with a CommitTransaction command. An AbortTransaction command will invalidate all changes.	
Parameters	fileNo	the file number
	commSettings	0x00 or 0x02 --- Plain communication 0x01 --- Plain communication secured by MACing 0x03 --- Fully enciphered communication
	offset	the starting position for the write operation within the file
	data	Data to send
Return	SdkResult.Success	Success
	SdkResult.Fail	Fail

	SdkResult.Param_In_Invalid	Parameter is invalid
	SdkResult.TimeOut	timeout

3.16.31 getValue

Prototype	int getValue(byte fileNo, byte commSettings);	
Function	Read the currently stored value from Value Files. 1. The GetValue command requires a preceding authentication with the key specified for Read, Write or Read&Write access 2. After updating a value file's value but before issuing the CommitTransaction command, the GetValue command will always retrieve the old, unchanged value which is still the valid one.	
Parameters	fileNo	the file number
	commSettings	0x00 or 0x02 --- Plain communication 0x01 --- Plain communication secured by MACing 0x03 --- Fully enciphered communication
Return	SdkResult.Success	Success
	SdkResult.Fail	Fail
	SdkResult.Param_In_Invalid	Parameter is invalid
	SdkResult.TimeOut	timeout

3.16.32 credit

Prototype	int credit(byte fileNo, byte commSettings, int value);
Function	Increase a value stored in a Value File.

	1. Credit commands do NEVER modify the Limited Credit Value of a Value file. However, if the Limited Credit Value needs to be set to 0, a LimitedCredit with value 0 can be used. 2. The Credit command requires a preceding authentication with the key specified for "Read&Write" access.	
Parameters	fileNo	the file number
	commSettings	0x00 or 0x02 --- Plain communication 0x01 --- Plain communication secured by MACing 0x03 --- Fully enciphered communication
	value	the value which will be subtracted from the current value stored in the file. Only positive values are allowed for the Credit command.
Return	SdkResult.Success	Success
	SdkResult.Fail	Fail
	SdkResult.Param_In_Invalid	Parameter is invalid
	SdkResult.TimeOut	timeout

3.16.33 debit

Prototype	int debit(byte fileNo, byte commSettings, int value);
Function	Decrease a value stored in a Value File. 1. The Debit command requires a preceding authentication with one of the keys specified for Read, Write or Read&Write access. 2. If the usage of the LimitedCredit feature is enabled, the new limit for a subsequent LimitedCredit command is set to the sum of Debit commands within one transaction before issuing a CommitTransaction command. This assures that a LimitedCredit command can not re-book more values than a debiting transaction deducted before.

Parameters	fileNo	the file number
	commSettings	0x00 or 0x02 --- Plain communication 0x01 --- Plain communication secured by MACing 0x03 --- Fully enciphered communication
	value	the value which will be subtracted from the current value stored in the file. Only positive values are allowed for the Credit command.
Return	SdkResult.Success	Success
	SdkResult.Fail	Fail
	SdkResult.Param_In_Invalid	Parameter is invalid
	SdkResult.TimeOut	timeout

3.16.34 limitedCredit

Prototype	int limitedCredit(byte fileNo, byte commSettings, int value);	
Function	Allows a limited increase of a value stored in a Value File without having full Read&Write permissions to the file. This feature can be enabled or disabled during value file creation. 1. The LimitedCredit command requires a preceding authentication with the key specified for "Write" or "Read&Write" access. 2. The value for LimitedCredit is limited to the sum of the Debit commands on this value file within the most recent transaction containing at least one Debit. After executing the LimitedCredit command the new limit is set to 0 regardless of the amount which has been re-booked. Therefore the LimitedCredit command can only be used once after a Debit transaction.	
Parameters	fileNo	the file number
	commSettings	0x00 or 0x02 --- Plain communication 0x01 --- Plain communication secured by MACing

		0x03 --- Fully enciphered communication
	value	the value which will be subtracted from the current value stored in the file. Only positive values are allowed for the Credit command.
Return	SdkResult.Success	Success
	SdkResult.Fail	Fail
	SdkResult.Param_In_Invalid	Parameter is invalid
	SdkResult.TimeOut	timeout

3.16.35 writeRecord

Prototype	int writeRecord(byte fileNo, byte commSettings, int offset, int len, byte[] record);
Function	The WriteRecord command allows to write data to a record in a Cyclic or Linear Record File. 1. The WriteRecord command appends one record at the end of the linear record file, it erases and overwrites the oldest record in case of a cyclic record file if it is already full. The entire new record is cleared before data is written to it. 2. If no CommitTransaction command is sent after a WriteRecord command, the next WriteRecord command to the same file writes to the already created record. After sending a CommitTransaction command, a new WriteRecord command will create a new record in the record file. An AbortTransaction command will invalidate all changes 3. After issuing a ClearRecordFile command, but before a CommitTransaction / AbortTransaction command, a WriteRecord command to the same record file will fail. 4. The WriteRecord command requires a preceding authentication either with the key specified for "Write" or "Read&Write" access.

Parameters	fileNo	the file number
	commSettings	0x00 or 0x02 --- Plain communication 0x01 --- Plain communication secured by MACing 0x03 --- Fully enciphered communication
	offset	the offset within one single record, the value has to be in therange from 0 to record size - 1.
	len	the length of data which is to be written to the record file, the value has to be in the range from 1 to record size - offset.
	record	Record Information
Return	SdkResult.Success	Success
	SdkResult.Fail	Fail
	SdkResult.Param_In_Invalid	Parameter is invalid
	SdkResult.TimeOut	timeout

3.16.36 readRecords

Prototype	byte[] readRecords(byte fileNo, byte commSettings, int recordSize, int first, int num);	
Function	The ReadRecords command allows to read out a set of complete records from a Cyclic or Linear Record File. 1. In cyclic record files the maximum number of stored valid records is one less than the number of records specified in the CreateCyclicRecordFile command. 2. A ReadRecords command on an empty record file (directly after creation or after a committed clearance will result in an error. 3. The ReadRecords command requires a preceding authentication either with the key specified for "Read" or "Read&Write" access.	
	fileNo	the file number

Parameters	commSettings	0x00 or 0x02 --- Plain communication 0x01 --- Plain communication secured by MACing 0x03 --- Fully enciphered communication
	recordSize	the size of single record
	first	the first record which is read out. In case of 0x00 the latest record is read out. The value must be in the range from 0x00 to number of existing records - 1.
	num	the number of records to be read from the PICC. Records are always transmitted by the PICC in chronological order (= starting with the oldest, which is number of records ¨C 1 before the one addressed by the given offset). If this parameter is set to 0x00 then all records, from the oldest record up to and including the newest record(given by the offset parameter) are read.
Return	return Record Information	

3.16.37 clearRecordFile

Prototype	int clearRecordFile(byte fileNo);
Function	The ClearRecordFile command allows to reset a Cyclic or Linear Record File to the empty state. 1. After executing the ClearRecordFile command but before CommitTransaction, all subsequent WriteRecord commands will fail. 2. The ReadRecords command will return the old still valid records. 3. After the CommitTransaction command is issued, a ReadRecords command will fail, WriteRecord commands will be successful.

	4. An AbortTransaction command (instead of CommitTransaction) will invalidate the clearance	
Parameters	fileNo	the file number
Return	SdkResult.Success	Success
	SdkResult.Fail	Fail
	SdkResult.Param_In_Invalid	Parameter is invalid
	SdkResult.TimeOut	timeout

3.16.38 commitTransaction

Prototype	int commitTransaction();	
Function	The CommitTransaction command allows to validate all previous write access on Backup Data Files, Value Files and Record Files within one application. The CommitTransaction is typically the last command of a transaction before the ISO 14443-4 Deselect command or before proceeding with another application (SelectApplication command).	
Parameters	null	
Return	SdkResult.Success	Success
	SdkResult.Fail	Fail
	SdkResult.Param_In_Invalid	Parameter is invalid
	SdkResult.TimeOut	timeout

3.16.39 abortTransaction

Prototype	int abortTransaction();
Function	The AbortTransaction command allows to invalidate all previous write access on Backup Data Files, Value Files and Record Files within one application.

	This is useful to cancel a transaction without the need for re-authentication to the PICC, which would lead to the same functionality.	
Parameters	null	
Return	SdkResult.Success	Success
	SdkResult.Fail	Fail
	SdkResult.Param_In_Invalid	Parameter is invalid
	SdkResult.TimeOut	timeout

3.17 Mifare Ultralight card

The Ultralight module class is responsible for managing operate the Mifare Ultralight card.

UltralightCCardHandler ultralightCCardHandler = deviceEngine.getUltralightCCardHandler();

3.17.1 authority

Block certification.

public int authority(byte[] keyData);

Parameters:

Parameter	Description
keyData	Password authentication, 16 bytes(hex)

Return Value:

SdkResult.Success success

SdkResult.Fail other errors

3.17.2 readBlock

Read block data.

public byte[] readBlock(byte blockNum);

Parameters:

Parameter	Description
blockNum	block number

Return Value:

Success, return block data

Failed, return null

3.17.3 writeBlock

Write block data.

public int writeBlock(byte blockNum, byte[] writeData);

Parameters:

Parameter	Description
blockNum	Block number
writeData	Write data

Return Value:

SdkResult.Success success

SdkResult.Fail other errors

3.17.4 exchangeCmd

Exchange data with command(use original command communicate with the card directly)

public byte[] exchangeCmd(byte[] cmdData);

Parameters:

Parameter	Description
cmdData	Command data send to card

Return Value:

Success, return response data

Failed ,return null

4 Callback information

4.1 OnPrintListener

Responsible for managing the printer class callback interface.

4.1.1 onPrintResult

After executing the startPrint method, callback to print results.

public void onPrintResult (int retCode);

Parameters:

Parameter	Description
retCode	Print callback Results : SdkResult.Success success SdkResult.Printer_Print_Fail failed to print SdkResult.Printer_PaperLack out of paper SdkResult.Printer_UnFinished print unfinished SdkResult.Printer_TooHot printer is overheating

Return Value: None

4.2 OnPinPadInputListener

PIN pad class is responsible for the management callback interface.

4.2.1 onInputResult

After inputOnlinePin or inputOffinePin method is executed, it callback to show pin result.

public void onInputResult (int retCode, byte [] data);

Parameters:

Parameter	Description
retCode	Enter the result :

	SdkResult.Success success SdkResult.Fail failure SdkResult.PinPad_Input_Timeout input timeout SdkResult.PinPad_Input_Cancel cancel input SdkResulr.PinPad_No_Pin_Input no password entered, press Enter directly
data	When retCode == SdkResult.Success return pin Otherwise return null

Return Value: None

4.2.2 onSendKey

After inputText, inputOnlinePin, inputOfflinePin method, this is executed, callback the key input, when input password, number key will return KEYCODE_STAR.

public void onSendKey (byte keyCode);

Parameters:

Parameter	Description
keyCode	Input key value

Key-Value

Constant Name	Constant Value	Description
KEYCODE_0	0x30 (byte)	0
KEYCODE_1	0x31 (byte)	1
KEYCODE_2	0x32 (byte)	2
KEYCODE_3	0x33 (byte)	3
KEYCODE_4	0x34 (byte)	4
KEYCODE_5	0x35 (byte)	5
KEYCODE_6	0x36 (byte)	6
KEYCODE_7	0x37 (byte)	7
KEYCODE_8	0x38 (byte)	8
KEYCODE_9	0x39 (byte)	9
KEYCODE_ the STAR	0x2a (byte)	*
KEYCODE_OCTOTHORPE	0x23 (byte)	#
KEYCODE_CANCEL	0x18 (byte)	Cancel key
KEYCODE_BACKSPACE	0x08 (byte)	Backspace
KEYCODE_CLEAR	0xfe (byte)	Clear key

KEYCODE_CONFIRM	0x0d (byte)	Enter

Return Value: None

4.3 OnScanner Listener

Responsible for managing the camera scan code results callback.

4.3.1 onInitResult

Initialize the camera configuration callback.

public void onInitResult (int retCode);

Parameters:

Parameter	Description
retCode	Enter the result: SdkResult.Success success SdkResult.Fail failure

Return Value: None

4.3.2 onScannerResult

Scan code results callback.

public void onScannerResult (int retCode, String data);

Parameters:

Parameter	Description
retCode	Enter the result : SdkResult.Success success SdkResult.Fail failure SdkResult.Param_In_Invalid Parameter error SdkResult. TimeOut scan code timeout SdkResult. Scanner_Customer_Exit voluntary user withdrawal
data	Scan code Results

Return Value: None

4.4 OnCardInfoListener

Reader class is responsible for managing the callback interface.

4.4.1 onCardInfo

After executing the searchCard method, callback reader results.

public void onCardInfo (int retCode, CardInfoEntity cardInfo);

Parameters:

Parameter	Description
retCode	Enter the result : SdkResult.Success success SdkResult.Fail failure SdkResult.TimeOut timeout
cardInfo	When retCode == SdkResult.Success return card information Otherwise return null

CardInfoEntity

Attributes	Description
String cardNo	Card number
CardSlotTypeEnum cardExistslot	CardSlotType
RfCardTypeEnum rfCardType	RfCardTyp
String tk1	track one
String tk2	tracks two
String tk3	tracks three
String expiredDate	Card is valid
String serviceCode	Service Code
boolean isTk1Valid	A track LRC is correct
boolean isTk2Valid	Two tracks LRC is correct
boolean isTk3Valid	Three tracks LRC is correct
boolean isICC	If mag card has chip flag
String csn	Card serial number, only returnd in OnEmvProcessListener.onConfirmCardNo

CardSlotTypeEnum

Enumeration Name	Description
ICC1	The I C slot 1
ICC2	The I C slot 2
ICC3	The I C slot 3

PSAM1	PSAM slot 1
PSAM2	PSAM slot 1
PSAM3	PSAM slot 1
RF	Contactless card slot
SWIPE	Magnetic stripe card slot

RfCardTypeEnum

Enumeration Name	Description
TYPE_A_CPU	
TYPE_B_CPU	
S50	
FELICA	
S70	
ULTRALIGHT	
MEMORY_OTHER	
S50_PRO	
S70_PRO	

Return Value: None

4.4.2 onSwipeIncorrect

After executing searchCard method, will be callback when a swipe error occurs. This callback is a process callback, not a result callback

```
  public  void onSwipeIncorrect();
```

Return Value: None

4.4.3 onMultipleCards

After executing searchCard method，will be callback when find multiple contactless cards. This callback is a process callback, not a result callback

```
public  void onMultipleCards();
```

Return Value: None

4.5 OnEMVProcessListener2

Responsible for managing the EMV class callback interface.

4.5.1 onSelApp

After EmvProcess executed, if card have multi-application, onSelApp callback will be executed .it will show app-list to let the user to select the application. Then call EmvHandler2.onSetSelAppResponse.

public void onSelApp (List <String> appNameList, List <CandidateAppInfoEntity> appInfoList, boolean isFirstSelect);

Parameters:

Parameter	Description
appNameList	Application displays a list of names
appInfoList	Candidate application information card
isFirstSelect	Whether making the selection for the first time

CandidateAppInfoEntity

Attributes	Description
byte [] aid	AID
byte [] appLabel	Apply the label
byte [] preferName	Application Preferred Name
byte priority	Application Priority Indicator
byte [] langPrefer	The preferred language
byte icti	Issuer Code Table Index

Return Value: None

4.5.2 onTransInitBeforeGPO

After EmvProcess executed, before excute GPO, the EMV will callback this method. User can call setTlv method to set personalized tags. User can set it or not.(This is suit for both EMV contact and contactless flow), then call EmvHandler2. onSetContactlessTapCardResponse.

public void onTransInitBeforeGPO ();

Return Value: None

4.5.3 onConfirmCardNo

After EmvProcess executed, confirm the card number, then call EmvHandler2.onSetConfirmCardNoResponse.

public void onConfirmCardNo (String cardNo);

Parameters:

Parameter	Description
cardNo	card number

Return Value: None

4.5.4 onCardHolderInputPin

EmvProcess executing the method, enter the password, to be called EmvHandler2.onSetPinInputResponse.

public void onCardHolderInputPin (boolean isOnlinePin, int leftTimes);

Parameters:

Parameter	Description
isOnlinePin	Is online password
leftTimes	Enter the remaining number of times for the offline PIN

Return Value: None

4.5.5 onContactlessTapCardAgain

EmvProcess executing the method, Callback the second read card .(When host response the script with contactless EMV transaction.) The application should re-search contactless card, then call

EmvHandler2. onSetContactlessTapCardResponse to notify EMV continue procress.

public void onContactlessTapCardAgain ();

Return Value: None

Note: for amex contactless, the method will called by kernel, when process case “please see phone”

4.5.6 onOnlineProc

EmvProcess executing the method, means EMV kernel request online process, Then the application should call method getTlv to get the EMV tags, then send request message to the host. After host response, the application should call EmvHandler2.onSetOnlineProcResponse to notify the EMV kernel to do the second auth.

public void onOnlineProc ();

Parameters: None

Return Value: None

4.5.7 onPrompt

EmvProcess executing the method, notify the application prompt information to the user, Then the application should call EmvHandler2. onPromptResponse to notify the EMV kernel to continue the flow.

public void onPrompt(PromptEnum prompt);

Parameters:

Parameters	**Description**
prompt	enum

PromptEnum

enum	**Description**
APP_SELECTION_IS_NOT_ACCEPTED	Application is not accepted, please try again
OFFLINE_PIN_INCORRECT_TRY_AGAIN	Offline pin incorrect, please try again
OFFLINE_PIN_INCORRECT	Offline pin incorrect
OFFLINE_PIN_CORRECT	Offline pin correct

Return Value: None

4.5.8 onRemoveCard

EmvProcess executing the method, notify the application contactless card can be remove from the card reader, Then call EmvHandler2. onSetRemoveCardResponse to notify the EMV kernel to continue the flow.

public void onRemoveCard ();

Parameters: None

Return Value: None

4.5.9 onFinish

EmvProcess executing the method, means all the EMV flow is finish. The retcode will indicate the EMV transaction result.

public void onFinish (int retCode, EmvProcessResultEntity entity);

Parameters:

Parameter	Description
retCode	Enter the result : SdkResult.Success success SdkResult.Fail failure Other return results please refer to the EMV class table 3.7 or consult Appendix 5
entity	When retCode == SdkResult.Success return data Otherwise return null

EmvProcessResultEntity

Attributes	Description
byte [] scriptResult	Script execution results
List <EMVCardLogEntity> EMVlog	Cards Blog List
byte [] ecBalance	Electronic cash balance

EmvCardLogEntity

Attributes	Description
boolean isAmtExist	Whether the amount of presence
String amt	Amount of money
boolean isOtherAmtExist	Whether the existence of other
String otherAmt	Other Amount
boolean isDateExist	Date of the transaction if there are
String transDate	transaction date
boolean isTimeExist	The existence of transactions
String transTime	transaction hour
boolean isCntCodeExist	Whether there is a country code
String cntCode	country code

boolean isCurExist	Currency code if there
String curCode	Currency code
boolean isAtcExist	The existence of the transaction counter
String atc	Transaction Counter
boolean is9CExist	9c transaction type whether there
String serveType	Transaction Type
boolean isMerNameExist	The existence of a business name
String merName	Business Name

Return Value: None

4.6 OnEMVProcessListener Deprecated

Please note: All the Emvhandler method, do not recommend use it anymore.

Responsible for managing the EMV class callback interface.

4.6.1 onSelApp

After EmvProcess executed, callback app-list to select the application, then call EmvHandler.onSetSelAppResponse.

public void onSelApp (List <String> appNameList, List <CandidateAppInfoEntity> appInfoList, boolean isFirstSelect);

Parameters:

Parameter	Description
appNameList	Application displays a list of names
appInfoList	Candidate application information card
isFirstSelect	Whether making the selection for the first time

CandidateAppInfoEntity

Attributes	Description
byte [] aid	AID
byte [] appLabel	Apply the label
byte [] preferName	Application Preferred Name

byte priority	Application Priority Indicator
byte [] langPrefer	The preferred language
byte icti	Issuer Code Table Index

Return Value: None

4.6.2 onAfterFinalSelectedApp

After EmvProcess executed, before excute GPO, the EMV will callback this method. User can call setTlv method to set personalized tags. User can set it or not.(This is suit for EMV contactless flow), then call EmvHandler. onSetAfterFinalSelectedAppResponse.

public void onAfterFinalSelectedApp ();

Return Value: None

4.6.3 onRequestAmount

After EmvProcess executed, callback request input amount (triggered when when the transaction amount has not been input), then call EmvHandler.onSetRequestAmountResponse.

public void onRequestAmount ();

Return Value: None

4.6.4 onConfirmEcSwitch

After EmvProcess method executed, whether use electronic cash (triggered when the transaction is set to support electronic cash, and the card also support e-cash), then call EmvHandler.onSetConfirmEcSwitchResponse.

public void onConfirmEcSwitch ();

Return Value: None

4.6.5 onConfirmCardNo

After EmvProcess executed, confirm the card number, then call EmvHandler.onSetConfirmCardNoResponse.

public void onConfirmCardNo (String cardNo);

Parameters:

Parameter	Description
cardNo	card number

Return Value: None

4.6.6 onCardHolderInputPin

EmvProcess executing the method, enter the password, to be called EmvHandler.onSetPinInputResponse.

public void onCardHolderInputPin (boolean isOnlinePin, int leftTimes);

Parameters:

Parameter	Description
isOnlinePin	Is online password
leftTimes	Enter the remaining number of times for the offline PIN

Return Value: None

4.6.7 onCertVerify

EmvProcess executing the method of confirming documents, later to be called EmvHandler.onSetCertVerifyResponse.

public void onCertVerify (String certName, String certInfo);

Parameters:

Parameter	Description
certName	the name of your ID
certInfo	identity information

Return Value: None

4.6.8 onReadCardAgain

EmvProcess executing the method, Callback the second read card .(When host response the script with contactless EMV transaction.) The application should re-search contactless card, then call

EmvHandler.onSetReadCardAgainResponse to notify EMV continue procress.

public void onReadCardAgain();

Return Value: None

Note: for amex contactless, the method will called by kernel, when process case “please see phone”

4.6.9 onOnlineProc

EmvProcess executing the method, means EMV kernel request online process, Then the application should call method getTlv to get the tags, then send request message to the host. After host response, the application should call EmvHandler.onSetOnlineProcResponse to notify the EMV kernel to do the second auth.

public void onOnlineProc ();

Parameters: None

Return Value: None

4.6.10 onPrompt

EmvProcess executing the method, notify the application prompt information to the user, Then the application should call EmvHandler. onPromptResponse to notify the EMV kernel to continue the flow.

public void onPrompt(PromptEnum prompt);

Parameters:

Parameters	**Description**
prompt	enum

PromptEnum

enum	**Description**
APP_SELECTION_IS_NOT_ACCEPTED	Application is not accepted, please try again
OFFLINE_PIN_INCORRECT_TRY_AGAIN	Offline pin incorrect, please try again
OFFLINE_PIN_INCORRECT	Offline pin incorrect
OFFLINE_PIN_CORRECT	Offline pin correct

Return Value: None

4.6.11 onRemoveCard

EmvProcess executing the method, notify the application contactless card can be remove from the card reader, Then call EmvHandler. onSetRemoveCardResponse to notify the EMV kernel to continue the flow.

public void onRemoveCard ();

Parameters: None

Return Value: None

4.6.12 onFinish

EmvProcess executing the method, means all the EMV flow is finish. The retcode will indicate the EMV transaction result.

public void onFinish (int retCode, EmvProcessResultEntity entity);

Parameters:

Parameter	Description
retCode	Enter the result : SdkResult.Success success SdkResult.Fail failure Other return results please refer to the EMV class table 3.7 or consult Appendix 5
entity	When retCode == SdkResult.Success return data Otherwise return null

EmvProcessResultEntity

Attributes	Description
byte [] scriptResult	Script execution results
List <EMVCardLogEntity> EMVlog	Cards Blog List
byte [] ecBalance	Electronic cash balance

EmvCardLogEntity

Attributes	Description
boolean isAmtExist	Whether the amount of presence
String amt	Amount of money
boolean isOtherAmtExist	Whether the existence of other
String otherAmt	Other Amount

boolean isDateExist	Date of the transaction if there are
String transDate	transaction date
boolean isTimeExist	The existence of transactions
String transTime	transaction hour
boolean isCntCodeExist	Whether there is a country code
String cntCode	country code
boolean isCurExist	Currency code if there
String curCode	Currency code
boolean isAtcExist	The existence of the transaction counter
String atc	Transaction Counter
boolean is9CExist	9c transaction type whether there
String serveType	Transaction Type
boolean isMerNameExist	The existence of a business name
String merName	Business Name

Return Value: None

Appendix

Return Value Description

```
public class SdkResult {
        public final static int Success = 0;
        public final static int Fail = -1;
        public final static int Param_In_Invalid = -2;
        public final static int TimeOut = -3;
        / * Device not signed * /
        public final static int Device_Not_Ready = -4;
        // ---- Printer Error -----
        private final static int Printer_Base_Error = -1000;
        / ** * Print failed /
        public final static int Printer_Print_Fail = Printer_Base_Error -1;
```

```
/ * Failed to set string buffer * /
public final static int Printer_AddPrnStr_Fail = Printer_Base_Error -2;
/ ** * Set picture buffer failure /
public final static int Printer_AddImg_Fail = Printer_Base_Error -3;
/ ** * Printer Busy /
public final static int Printer_Busy = Printer_Base_Error - 4;
/ ** * The printer is out of paper /
public final static int Printer_PaperLack = Printer_Base_Error - 5;
/ ** * Wrong packet format print /
public final static int Printer_Wrong_Package = Printer_Base_Error - 6;
/ ** * Printer Fault /
public final static int Printer_Fault = Printer_Base_Error - 7;
/ ** * Printer overheating /
public final static int Printer_TooHot = Printer_Base_Error - 8;
/ ** * Print the unfinished /
public final static int Printer_UnFinished = Printer_Base_Error - 9;
/ ** Other exception error * /
public final static int Printer_Other_Error = Printer_Base_Error-999;
// ---- Scanner Error -----
private final static int Scanner_Base_Error = -2000;
/ ** * Button to exit the user /
public final static int Scanner_Customer_Exit = Scanner_Base_Error-1;
/ ** Other exception error * /
public final static int Scanner_Other_Error = Scanner_Base_Error-999;
// ---- SerialPort Error -----
private final static int SerialPort_Base_Error = -4000;
/ ** * Serial connection failure /
public final static int SerialPort_Connect_Fail = SerialPort_Base_Error - 1;
/ ** * Serial data transmission failure /
public final static int SerialPort_Send_Fail = SerialPort_Base_Error - 2;
/ ** Fd error * /
```

```
public final static int SerialPort_Fd_Error = SerialPort_Base_Error - 3;
/ ** * Unopened serial /
public final static int SerialPort_Port_Not_Open = SerialPort_Base_Error - 4;
/ ** * Serial scission failure /
public final static int SerialPort_DisConnect_Fail = SerialPort_Base_Error - 5;
/ ** Transmit buffer is not empty (the remaining data to be transmitted) * /
public final static int SerialPort_Sending_Buf_IsNot_Null = SerialPort_Base_Error - 6;
/ ** Invalid channel number * /
public final static int SerialPort_Invalid_Channel = SerialPort_Base_Error - 7;
/ ** Channel is not open and no communication with any physical port * /
public final static int SerialPort_Channel_Isnot_Open = SerialPort_Base_Error - 8;
/ ** Transmit buffer error (continue 500ms at full state) * /
public final static int SerialPort_Sending_Buffer_Error = SerialPort_Base_Error - 9;
/ ** No available physical port * /
public final static int SerialPort_No_Available_Ports = SerialPort_Base_Error - 10;
/ ** Device enumeration and configuration process is not completed (USB DEV dedicated * /
public final static int SerialPort_Conf_Process_Error = SerialPort_Base_Error - 11;
/ ** Equipment de-energized and the host loses connection (USB DEV dedicated * /
public final static int SerialPort_Device_Lost_Power = SerialPort_Base_Error - 12;
/ ** From the host device and then plug plucking (USB DEV dedicated) * /
public final static int SerialPort_Unplug_Error = SerialPort_Base_Error - 13;
/ ** Device is off (USBDEV dedicated) * /
public final static int SerialPort_Device_Is_Off = SerialPort_Base_Error - 14;
/ ** * Data receive timeout /
public final static int SerialPort_Timeout_Receiving_Data = SerialPort_Base_Error - 15;
/ *** Channel is being occupied by the system * /
public final static int SerialPort_Channle_Is_Occupied = SerialPort_Base_Error - 16;
/ ** Invalid communication Parameters, communication Parameters do not meet the rules for strings or data beyond the normal range. * /
public final static int SerialPort_Invalid_Communication_Parameter = SerialPort_Base_Error - 17;
/ ** USB to serial device mounted unsuccessful (the Return Value only FIDI USB to serial use) * /
```

```
public final static int SerialPort_Usb_Mounted_Unsuccessful = SerialPort_Base_Error - 18;
/ ** Usb to serial device error (only FTDI USB serial port using the Return Value of re-exports * /
public final static int SerialPort_Reset_Usb_Error = SerialPort_Base_Error - 19;
/ ** Device USB to serial chip traffic congestion (only FIDI USB serial adapter used in the Return Value) * /
public final static int SerialPort_Devices_Error = SerialPort_Base_Error - 20;
/ ** Other exception error * /
public final static int SerialPort_Other_Error = SerialPort_Base_Error-999;
// ---- MagCardReader Error -----
private final static int MagCardReader_Base_Error = -5000;
/ ** * No credit card /
public final static int MagCardReader_No_Swiped = MagCardReader_Base_Error -1;
/ ** Other exception error * /
public final static int MagCardReader_Other_Error = MagCardReader_Base_Error -999;
// ---- IccCardReader Error -----
private final static int IccCardReader_Base_Error = -6000;
public final static int IccCardReader_Read_CardType_Error = IccCardReader_Base_Error-1;
public final static int IccCardReader_CardInit_Error = IccCardReader_Base_Error-2;
/ ** Other exception error * /
public final static int IccCardReader_Other_Error = IccCardReader_Base_Error-999;
// ---- PinPad Error ----
private final static int PinPad_Base_Error = -7000;
/ ** * Key does not exist /
public final static int PinPad_No_Key_Error = PinPad_Base_Error - 1;
/ ** Wrong key index, the index is not within the Parameters range * /
public final static int PinPad_KeyIdx_Error = PinPad_Base_Error - 2;
When / ** key writing, the key level is lower than the source object key * /
public final static int PinPad_Derive_Error = PinPad_Base_Error - 3;
/ ** * Key verification failed /
public final static int PinPad_Check_Key_Fail = PinPad_Base_Error - 4;
/ ** Did not enter PIN * /
```

```
public final static int PinPad_No_Pin_Input = PinPad_Base_Error - 5;
/ ** Cancel Enter PIN * /
public final static int PinPad_Input_Cancel = PinPad_Base_Error - 6;
/ ** Function call is less than the minimum interval * /
public final static int PinPad_Wait_Interval = PinPad_Base_Error - 7;
/ ** KCV wrong mode is not supported * /
public final static int PinPad_Check_Mode_Error = PinPad_Base_Error - 8;
/ ** No right to use this key, when the key label does not appear, or write key, the key values of the source type is larger than the object of the key type, the key will return * /
public final static int PinPad_No_Right_Use = PinPad_Base_Error - 9;
/ ** * Wrong key type /
public final static int PinPad_Key_Type_Error = PinPad_Base_Error - 10;
/ ** Desired PIN length string wrong * /
public final static int PinPad_ExpLen_Error = PinPad_Base_Error - 11;
/ ** Objective The key index was wrong, is not within the range * /
public final static int PinPad_Dstkey_Idx_Error = PinPad_Base_Error - 12;
/ ** Source wrong key index, is not within the range * /
public final static int PinPad_SrcKey_Idx_Error = PinPad_Base_Error - 13;
/ ** * Key length wrong /
public final static int PinPad_Key_Len_Error = PinPad_Base_Error - 14;
/ ** Enter PIN Timeout * /
public final static int PinPad_Input_Timeout = PinPad_Base_Error - 15;
/ ** IC card does not exist * /
public final static int PinPad_No_Icc = PinPad_Base_Error - 16;
/ ** IC card is not initialized * /
public final static int PinPad_Icc_No_Init = PinPad_Base_Error - 17;
/ ** DUKPT wrong group index * /
public final static int PinPad_Group_Idx_Error = PinPad_Base_Error - 18;
/ ** * Pointer argument is null illegal /
public final static int PinPad_Param_Ptr_Null = PinPad_Base_Error - 19;
/ ** PED locked * /
```

```
public final static int PinPad_Locked = PinPad_Base_Error - 20;
/ ** PED generic error * /
public final static int PinPad_Ret_Error = PinPad_Base_Error - 21;
/ ** No free buffer * /
public final static int PinPad_Nomore_Buf = PinPad_Base_Error - 22;
/ ** * The need to obtain advanced permission /
public final static int PinPad_Need_Admin = PinPad_Base_Error - 23;
/ ** DUKPT has overflowed * /
public final static int PinPad_Dukpt_Overflow = PinPad_Base_Error - 24;
/ ** KCV check failed * /
public final static int PinPad_Kcv_Check_Fail = PinPad_Base_Error - 25;
/ ** * Source wrong key type /
public final static int PinPad_SrcKey_Type_Error = PinPad_Base_Error - 26;
/ ** * Command is not supported /
public final static int PinPad_Unsupport_Cmd = PinPad_Base_Error - 27;
/ ** Communication error * /
public final static int PinPad_Comm_Error = PinPad_Base_Error - 28;
/ ** No public key user authentication * /
public final static int PinPad_No_Uapuk = PinPad_Base_Error - 29;
/ ** * Failure to take the system-sensitive services /
public final static int PinPad_Admin_Error = PinPad_Base_Error - 30;
/ ** PED is downloaded inactive * /
public final static int PinPad_Download_Disactive = PinPad_Base_Error - 31;
/ ** KCV odd parity failed * /
public final static int PinPad_Kcv_Odd_Check_Fail = PinPad_Base_Error - 32;
/ ** * Failed to read data PED /
public final static int PinPad_Ped_Data_Rw_Fail = PinPad_Base_Error - 33;
/ ** Card operation error (offline plaintext, ciphertext password authentication) * /
public final static int PinPad_Icc_Cmd_Error = PinPad_Base_Error - 34;
/ ** Press CLEAR key to exit the user input PIN * /
public final static int PinPad_Input_Clear = PinPad_Base_Error - 35;
```

```
/ ** PED sufficient storage space * /
public final static int PinPad_No_Free_Flash = PinPad_Base_Error - 36;
/ ** DUKPT KSN need to add 1 * /
public final static int PinPad_Dukpt_Need_Inc_Ksn = PinPad_Base_Error - 37;
/ ** KCVMODE error * /
public final static int PinPad_Kcv_Mode_Error = PinPad_Base_Error - 38;
/ ** NO KCV * /
public final static int PinPad_Dukpt_No_Kcv = PinPad_Base_Error - 39;
/ ** Press FN / ATM4 cancel PIN input * /
public final static int PinPad_Pin_Bypass_ByFunKey = PinPad_Base_Error - 40;
/ ** Data MAC checksum error * /
public final static int PinPad_Mac_Error = PinPad_Base_Error - 41;
/ ** Data CRC error * /
public final static int PinPad_Crc_Error = PinPad_Base_Error - 42;
/ ** * Password wrong type of keyboard /
public final static int PinPad_Type_Error = PinPad_Base_Error - 43;
/ ** Other exception error * /
public final static int PinPad_Other_Error = PinPad_Base_Error-999;
// ---- EMVHandler Error ----
private final static int EMVHandler_Base_Error = -8000;
/ ** <Try other communication interface * /
public final static int EMV_Other_Interface = EMVHandler_Base_Error - 1;
/ ** <Contactless transactions offline approved * /
public final static int EMV_Qpboc_Offline = EMVHandler_Base_Error - 2;
/ ** <Contactless union pay online transaction * /
public final static int EMV_Qpboc_Online = EMVHandler_Base_Error - 3;
/ ** <Contactless PBOC online transaction , Abolished * /
public final static int EMV_Pboc_Online = EMVHandler_Base_Error - 4;
/ ** <Contactless MSD online transaction, Abolished * /
public final static int EMV_MSD_Online = EMVHandler_Base_Error - 5;
/ ** <Offline electronic cash acceptance, Abolished * /
```

```
public final static int EMV_Ec_Accept = EMVHandler_Base_Error - 6;
/ ** <contact transaction Offline approved * /
public final static int EMV_Offline_Accept = EMVHandler_Base_Error - 7;
/ ** <Transaction card is removed * /
public final static int EMV_Card_Removed = EMVHandler_Base_Error -8;
/ ** <Reader failed * /
public final static int EMV_Command_Fail = EMVHandler_Base_Error -9;
/ ** <Card is Blocked * /
public final static int EMV_Card_Block = EMVHandler_Base_Error -10;
/ ** <Parameters wrong * /
public final static int EMV_PARA_ERR = EMVHandler_Base_Error -11;
/ ** <No common application * /
public final static int EMV_Candidatelist_Empty = EMVHandler_Base_Error -12;
/ ** <Application locked * /
public final static int EMV_App_Block = EMVHandler_Base_Error -13;
/ ** <Transaction fallback , need to swipe card* /
public final static int EMV_FallBack = EMVHandler_Base_Error -14;
/ ** <Data authentication has failed * /
public final static int EMV_Auth_Fail = EMVHandler_Base_Error -15;
/ ** <Application has not yet entered into force * /
public final static int EMV_App_Ineffect = EMVHandler_Base_Error -16;
/ ** <Application has expired * /
public final static int EMV_App_Expired = EMVHandler_Base_Error -17;
/ ** <Cardholder verification failed * /
public final static int EMV_Cvm_Fail = EMVHandler_Base_Error -18;
/ ** <* Transactions should the online,  Abolished /
public final static int EMV_Online = EMVHandler_Base_Error -19;
/ ** <Cancel the transaction * /
public final static int EMV_Cancel = EMVHandler_Base_Error -20;
/ ** <Transaction online decline * /
public final static int EMV_Declined = EMVHandler_Base_Error -21;
```

```
/ ** <Issuer Authentication failed * /
public final static int EMV_Arpc_Fail = EMVHandler_Base_Error -22;
/ ** <Issuer Script execution failed * /
public final static int EMV_Script_Fail = EMVHandler_Base_Error -23;
/ ** <Applications are not accepted, you can re-select * /
public final static int EMV_App_NoAccept = EMVHandler_Base_Error -24;
/ ** <Electronic cash offline decline * /
public final static int EMV_Ec_Decliend = EMVHandler_Base_Error -25;
/ ** <Successful transaction, Issuer Authentication failed * /
public final static int EMV_Sucess_Arpc_Fail = EMVHandler_Base_Error -26;
/ ** <plese see phone* /
public final static int Emv_Plz_See_Phone = EmvHandler_Base_Error - 27;
/ ** < Transaction Terminate * /
public final static int Emv_Terminate = EmvHandler_Base_Error - 28;
/ ** < Transaction Communicate Timeout * /
public final static int Emv_Communicate_Timeout = EmvHandler_Base_Error - 29;
/ ** < Use other card * /
public final static int Emv_USE_OTHER_CARD = EmvHandler_Base_Error - 30;
/ ** * Other error exception /
public final static int EMV_Other_Error = EMVHandler_Base_Error -999;
// ---- CardHandler Error -----
// ---- ~ -10000 -19900 Allocated to card manipulation
// ---- -10.1 Thousand representatives contact CPU card -10200-- non-contact CPU card supports a total of 99 kinds of card types
private final static int CardHandler_Base_Error = -10000;
// ---- Contactless IC card return code segment
public final static int Icc_Base_Error = CardHandler_Base_Error -100;
/ ** Transaction card dialed * /
public final static int Icc_PullOut_Card = Icc_Base_Error - 1;
/ ** Parity error * /
```

```
public final static int Icc_Parity_Err = Icc_Base_Error - 2;
/ ** Select the channel error * /
public final static int Icc_Channel_Err = Icc_Base_Error - 3;
/ ** Transmit data too long (LC) * /
public final static int Icc_Data_Len_TooLong = Icc_Base_Error - 4;
/ ** Card Error protocol (T = 0 or not T = 1) * /
public final static int Icc_Protocol_Err = Icc_Base_Error - 5;
/ ** * Not reset the card /
public final static int Icc_No_Reset_Card = Icc_Base_Error - 6;
/ ** The dead can not communicate or * /
public final static int Icc_Not_Call = Icc_Base_Error - 7;
/ ** Other exception error * /
public final static int Icc_Other_Error = Icc_Base_Error - 99;
// ---- Contactless IC card return code segment
private final static int Picc_Base_Error = CardHandler_Base_Error -200;
/ ** * RF module is not turned on /
public final static int Picc_Not_Open = Picc_Base_Error - 1;
/ ** Not find the card (the sensor area no specific type of card) * /
public final static int Picc_Not_Searched_Card = Picc_Base_Error - 2;
/ ** Card induction area too (there is a communication conflict) * /
public final static int Picc_Card_Too_Many = Picc_Base_Error - 3;
/ ** Protocol error (data in violation of the agreement appears response card) * /
public final static int Picc_Protocol_Data_Err = Picc_Base_Error - 4;
/ ** * Card not active /
public final static int Picc_Card_No_Activation = Picc_Base_Error - 5;
/ ** * Conflict Doka /
public final static int Picc_Muti_Card_Err = Picc_Base_Error - 6;
/ ** Protocol error * /
public final static int Picc_Protocol_Err = Picc_Base_Error - 7;
/ ** Communications transmission error * /
public final static int Picc_Io_Err = Picc_Base_Error - 8;
```

```
        / ** * Card is still the sensor area /
        public final static int Picc_Card_Sense_Err = Picc_Base_Error - 9;
        / ** Card status error (such as A / B card calling card interface M1, or M1 card call PiccIsoCommand Interface) * /
        public final static int Picc_Card_Status_Err = Picc_Base_Error - 10;
        / ** Interface chip does not exist or abnormal * /
        public final static int Picc_Not_Call = Picc_Base_Error - 11;
        / ** * Other error exception /
        public final static int Picc_Other_Error = Picc_Base_Error - 99;
        / ** M1 card section * /
        public final static int M1Card_Base_Error = CardHandler_Base_Error -300;
        / ** M1 card authentication failed * /
        public final static int M1Card_Verify_Err = M1Card_Base_Error - 1;
        / ** * Sector unauthenticated /
        public final static int M1Card_Fan_Not_Verify = M1Card_Base_Error - 2;
        / ** Numeric data block format is wrong * /
        public final static int M1Card_Data_Block_Err = M1Card_Base_Error - 3;
        / ** * Module unopened /
        public final static int M1Card_Not_Open = M1Card_Base_Error - 4;
        / ** * Card not active /
        public final static int M1Card_Card_Not_Activation = M1Card_Base_Error - 5;
        / ** Wrong type of card operations for operateBlock the Senate operType Check * /
        public final static int M1Card_Card_OperType_Error = M1Card_Base_Error - 6;
        / ** * Other error exception /
        public final static int M1Card_Other_Error = M1Card_Base_Error - 99;
}
```