Internal documents

PED API Programming Guide

Version：V1.1.3

Revision History

No.	Change(+/-)Description	Author	Version	Date	Approve

Directory

1. Overview

This document is the POS product safety module interface specification documents, how to android application control through the upper preliminary description POS module functions for application developers to provide a reference.

2. Normative references

3. Terms and Abbreviations

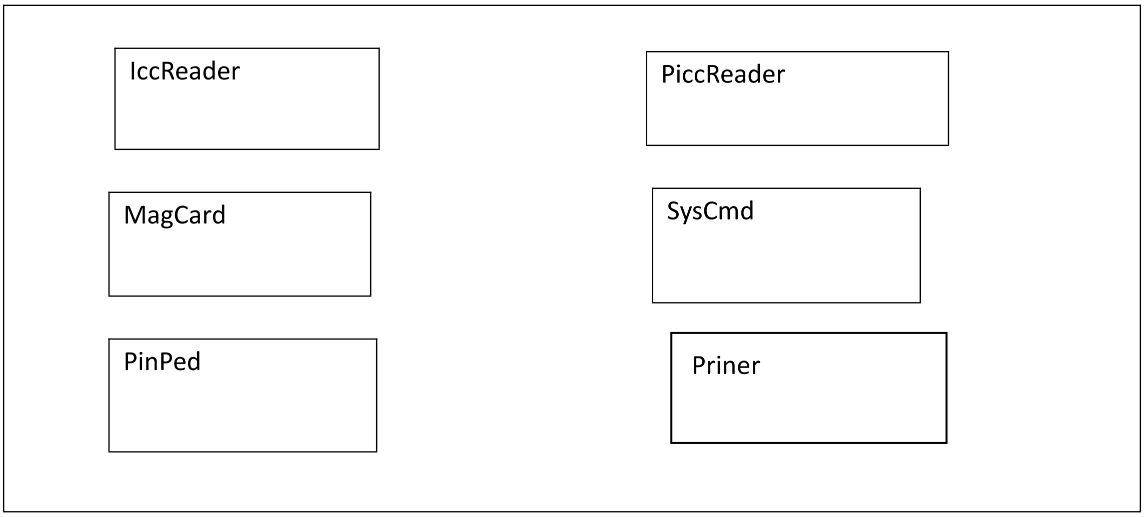
RspCode.RSPOK: command executed successfully

RspCode.RSPERR: command fails, the available SysCmd.GetLastErrNo () function to obtain the error code, the error code details please refer to appendix B.

4. POS hardware module description

4.1. POS Hardware Module Overview

There are 6 functional modules, For each sub-module, we define a class to achieve control of the underlying hardware.



5. POS module interface details

5.1. SpDev

5.1.1. Spdev.SpDevCreate

Function

POS device initialization, you have to call this function before calling anyother POS function, otherwise the follow-up operation will fail。

Parameter

Parameter name	Type	Description
null		

Return

Return Value	Description
RspCode.RspOk	Successfully initialized
RspCode.RspERR	Initialization failure

5.1.2. Spdev.SpDevRelease

Function

The release of the POS device, and Spdev.SpDevCreate () function in pairs. Generally invoked at program exit

Parameter

Parameter name	Type	Description
null		

Return

Return Value	Description
RspCode.RspOk	Successfully release
RspCode.RspERR	Release failure

5.2. IccReader

5.2.1. IccReader.SelectSlot

Function

Select the required operation of the card slot。

Parameter

Parameter name	Type	Description
Slot	byte	The selected slot: 0： Users slot 1： PSAM slot

Return

Return Value	Description
RspCode.RSPOK	Select the slot success
RspCode.RSPERR	Select the slot failed

5.2.2. IccReader.Detect

Function

To detect the ic card is inserted or not 。

Parameter

Parameter name	Type	Description
null		

Return

Return Value	Description
RspCode.RSPOK	Ic card is inserted
RspCode.RSPERR	No ic card is detected

5.2.3. IccReader.Reset

Function

Reset Card。

Parameter

Parameter name	Type	Description
null		

Return

Return Value	Description
RspCode.RSPOK	Reset Card Success
RspCode.RSPERR	Reset the card fails

5.2.4. IccReader.GetAtr

Function

Get the card ATR。

Parameter

Parameter name	Type	Description
pAtr	byte[]	The buffer used to store the ATR,encode as ascii code

Return

Return Value	Description
ilen	The length of ATR(ascii code)
RspCode.RSPERR	Failure to obtain the card ATR

5.2.5. IccReader.ApduTransmit

Function

Transfer APDU command。

Parameter

Parameter name	Type	Description
Apdu	byte[]	The command will be sent to card,encode as ascii code
apdulen	int	The length of apdu command
Rsp	byte[]	The response to the apdu command， encode as ascii code
Sw	byte[]	The StatusWord of the command， encode as ascii code

Return

Return Value	Description
ilen	The length of the response
RspCode.RSPERR	Apdu Command execution failed

5.2.6. IccReader.Eject

Function

Eject the IC card

Parameter

Parameter name	Type	Description
null		

Return

Return Value	Description
RspCode.RSPOK	Eject the IC card Success
RspCode.RSPERR	Eject the IC card Failed

5.2.7. IccReader.SetApduTimeOut

Function

To set the timeout for receiving response and it's default value is 10 seconds。 You can call this function to receive the response if the command have to take long time.

Parameter

Parameter name	Type	Description
ms	int	The timeout value (ms)

Return

Return Value	Description
void	

5.2.8. IccReader.StopApduRspRecv

Function

You can call this function while you need interrupt the receiving of the command response

Parameter

Parameter name	Type	Description
null		

Return

Return Value	Description
void	

5.3. PiccReader

5.3.1. PiccReader.OpenPicc

Function

Open the PICC module. After executing this command, the PICC module is working.。

Parameter

Parameter name	Type	Description
null		

Return

Return Value	Description
RspCode.RSPOK	Open the picc module Success
RspCode.RSPERR	Open the picc module failed

5.3.2. PiccReader.Detect

Function

To detect the Picc。

Parameter

Parameter name	Type	Description
mode	byte	0x00 Detect type A card and B card once , this model is suitable for the need to enhance multi detection function situation. The model is ISO14443 compliant card search mode； 0x01 Search type A card and B card once; this pattern card search mode is EMV, which is normally used in the model; ‘a’ or ‘A’ Only search for type A card at a time; ‘b’ or ‘B’ Only search for type B card at a time; ‘m’ or ‘M’ Only search for M1 card at a time;

Return

Return Value	Description
RspCode.RSPOK	Successfully detected PICC
RspCode.RSPERR	Failed detected PICC

5.3.3. PiccModule.ApduTransmit

Function

Send APDU to the Picc

Parameter

Parameter name	Type	Description
cmd	Byte[]	The APDU will be sent to the picc, command format with 7816 specification.(BCD FORMAT)
cmdlen	Int	Length of the command
rsp	Byte[]	The response from the card (BCD FORMAT)
sw	Byte[]	The StatusWord from the card(2bytes,BCD FORMAT)

Return

Return value	Description
len	The length of the respone,don’t include the status
RspCode.RSPERR	Error occurs

5.3.4. PiccReader.GetCardTypeFunction

Function

To get the Type of the Card detected。

Parameter

Parameter name	Type	Description
null		

Return

Return value	Description
Cardtype	Card type (asc)

5.3.5. PiccReader.GetCardChanleFunction

Function

To get the LogicChannel of the card

Parameter

Parameter name	Type	Description
null		

Return

Return value	Description
LogicChannel	Logic channel，(bcd)

5.3.6. PiccReader.GetCardSNFunction

Function

To get the Card Sn of the card

Parameter

Parameter name	Type	Description
SN	byte []	The sn of the card (bcd)

Return

Return Value	Description
len	The length of the sn

5.3.7. PiccReader.GetCardInfoFunction

Function

Get the CardInfo of the picc

Parameter

Parameter name	Type	Description
Cardinfo	byte []	Card information， bcd

Return

Return value	Description
len	The length of the card information

5.3.8. PiccReader.Remove

Function

Remove the card 。

Parameter

Parameter name	Type	Description
mode	byte	'h' or ‘H' is 'HALT, exit stop live instruction only to the card after sending; the processdoes not execute the card removed detection 'r 'or'R' REMOVE, stop live instruction to the card sending, and

		executes the cardremoved detection; 'e' or 'E' to meet the specification of the shift mode reset carrier non EMV, and execute the card removed detection.
logicchannel	byte	The logic channel which is got when you detected the card

Return

Return value	Description
len	The length of the information

5.3.9. PiccReader.Close

Function

Close the picc module.

Parameter

Parameter name	Type	Description
null		

Return

Return Value	Description
RspCode.RSPOK	Close PICCmodule succesful
RspCode.RSPERR	Close PICCmodule failed

5.3.10. PiccReader.KeyAuth_M1

Function

Key of M1 authentication。

Parameter

Parameter name	Type	Description
keyType	int	Key type: 'A' or 'a':KEY A 'B'or 'b': KEY B
blkNo	int	the number of block which need to read
keylen	int	Length of the key
keyBuf	byte[]	Data of the key
iSeriNumLen	int	Length of the serial number
SeriNum	byte[]	Serial number

Return

Return Value	Description
RspCode.RSPOK	Authentication is ok
RspCode.RSPERR	Authentication is failed

5.3.11. PiccReader.ReadBlk_M1

Function

Read M1 card data

Parameter

Parameter name	Type	Description
blkNo	int	the number of block which need to read
pReadBuf	byte[]	Used to save the data

Return

Return Value	Description
iLen	The length of the data which we read from the card。

RspCode.RSPERR	Read data failed

5.3.12. PiccReader.WriteBlk_M1

Function

Write data into card

Parameter

Parameter name	Type	Description
blkNo	int	Number of the Block
iLenWriteBuf	int	The length of the data will be writen into card
pWriteBuf	byte[]	The data will be writen into card

Return

Return Value	Description
RspCode.RSPOK	Write ok
RspCode.RSPERR	Write failed

5.3.13. PiccReader.OperateBlk_M1

Function

To do Charge / impairment / backup operation to the specified data block and put the new value to another specify block。

Attation：
1 if BlkNo and UpdateBlkNo block number equal, if the "+" or "-" operation, then the appreciation or depreciation of the block number.
2 if BlkNo and UpdateBlkNo block number not equal, "+" or "-" operation, BlkNo stored in the same amount, the amount of UpdateBlkNo is equal to the corresponding appreciation or depreciation amount of BlkNo.
3 if the ">" operation, then value does not play a role, will be in the amount of BlkNo stored in UpdateNo.

The block number must be numerical block, and BlkNo and UpdateBlkNo must be in the same sector.

Parameter

Parameter name	Type	Description
operateType	int	Operation tyoe
BlkNo	int	The block will be operated
iMoney	int	Amount of money
UpdateBlkNo	int	The block used to write data into

Return

Return Value	Description
RspCode.RSPOK	successful
RspCode.RSPERR	failed

5.4. MagCard

5.4.1. MagCard.InitDev

Function

Initializing magnetic stripe card reader, execute this command, the magnetic stripe card reader in a state of waiting for the user swipe.

Parameter

Parameter name	Type	Description
null		

Return

Return Value	Description
RspCode.RSPOK	Initialize magnetic stripe reader success.
RspCode.RSPERR	Failed to initialize magnetic stripe reader

5.4.2. MagCard.Check

Function

Check whether a user is detected magnetic stripe card swipe action.

Parameter

Parameter name	Type	Description
null		

Return

Return Value	Description
RspCode.RSPOK	A user card detected
RspCode.RSPERR	A user card is not detected

5.4.3. MagCard.GetAllStripInfo

Function

Get information on the various tracks (1、2、3) 。

Parameter

Parameter name	Type	Description
Info	byte[]	A magnetic stripe for storing the acquired information card 3 tracks Format is:

		The first byte indicates the track number, The second byte is the number of bytes of data tracks N, The third byte is a byte to the N +3 track data; Track data in ASCII string; Three tracks of information sequentially combined (TLV format). TAG Definition: 01 = track 1, 02 = track 2, 03 = track 3 00 = a swipe action, but failed reading tracks

Return

Return Value	Description
Info_len	The length of the information

5.4.4. MagCard.GetSingleStripInfo

Function

Gets the specified track magnetic stripe card (1/2/3) information。

Parameter

Parameter name	Type	Description
Strip	int	Specified the track of magnetic stripe cards
Info	byte[]	For storing magnetic stripe cards to obtain information on the designated track，encode as ascii code

Return

Return Value	Description
Info_len	The length of the information

5.4.5. MagCard.CancelDev

Function

Closed magnetic stripe reader modules.。

Parameter

Parameter name	Type	Description
null		

Return

Return Value	Description
RspCode.RSPOK	Closed magnetic stripe reader module successfully
RspCode.RSPERR	Closed magnetic stripe reader module failure

5.5. SysCmd

5.5.1. SysCmd.SysUnLockFunction

Function

Unlock the module

Parameter

Parameter name	Type	Description
NULL		

Return

Return Value	Description
RspCode.RSPOK	Successful

RspCode.RSPERR	Failed

5.5.2. SysCmd.SysSelfCheckFunction

Function

A self-examination of the module

Parameter

Parameter name	Type	Description
CheckByte	byte	The dev module will be checked,you can only check one module by once Mag: 0x01 Icc : 0x02 Picc : 0x04 Flash: 0x08 KEY: 0x10 ALL: 0XFF,

Return

Return Value	Description
DevStatus	If self check is successful,the bit of the dev should be '1',or it's failed or unavalibale
RspCode.RSPERR	Self check failed

5.5.3. SysCmd.SysSetSpTime

Function

Set the time of Pos module

Parameter

Parameter name	Type	Description

time	byte[]	The time value will be set， it must be 6bytes Such as 13/10/1 13:05:10, the buffer will be 131001130510

Return

Return Value	Description
RspCode.RSPOK	Set successful
RspCode.RSPERR	Set failed

5.5.4. SysCmd.SysGetSpTime

Function

To get the time of pos module

Parameter

Parameter name	Type	Description
time	byte[]	The buffer used to srore the time ,it's 6 bytes Such as 131001130510 ,

Return

Return Value	Description
RspCode.RSPOK	successful
RspCode.RSPERR	failed

5.5.5. SysCmd.SysGetSpVersion

Function

Read the version of software

Parameter

Parameter name	Type	Description
SpVersion	byte[]	the buffer used to store the data， ASCII

Return

Return Value	Description
n	The length of the version
RspCode.RSPERR	failed

5.5.6. SysCmd.GetLastErrNo

Function

Get the error no

Parameter

Parameter name	Type	Description
NULL		

Return

Return Value	Description
ERRNO	The error number
RspCode.RSPOK	No error occurs

5.5.7. SysCmd.GetLastErrString

Function

Get error info

Parameter

Parameter name	Type	Description
the_errno	int	Error number

Return

Return Value	Description
Errinfo	Eror info

5.5.8. SysCmd.SysSensorCheckEXT

Function

Check Seneor

Parameter

Parameter name	Type	Description
pBuf	Byte[]	The result of the checking,4bytes length

Return

Return Value	Description
RspCode.RSPOK	successful
RspCode.RSPERR	failed

5.5.9. SysCmd.SysActSecurityEXT

Function

Activate the security mechanism

Parameter

Parameter name	Type	Description
channel	String	indicate which channel you want to close by a String，if you want to close channel 1、 2、 8， it is ”1,2,8”
pBuf	Byte[]	Indicate the status of every channel,4 bytes length

Return

Return Value	Description
RspCode.RSPOK	successful
RspCode.RSPERR	failed

5.5.10. SysCmd.SetCmdSendMaxWT

Function

Set the max send time， the default value is 1 second

Parameter

Parameter name	Type	Description
sendmxwt	int	The value will be set, ms

Return

Return Value	Description
RspCode.RSPOK	successful
RspCode.RSPERR	failed

5.5.11. SysCmd.SetRspRecvMaxWT

Function

Set the max wait time for response ,default vaule is 10 seconds

Parameter

Parameter name	Type	Description
recvmxwt	int	The max waittime for the response of command, ms

Return

Return Value	Description
RspCode.RSPOK	successful
RspCode.RSPERR	failed

5.5.12. SysCmd.InterruptCmd

Function

Interrupt reception

Parameter

Parameter name	Type	Description
NULL		

Return

Return Value	Description
void	

5.6. Ped

5.6.1. Data type definitions

#define PED_TLK	0x01	TLK (Loading Key)
#define PED_TMK	0x02	TMK (master key)
#define PED_TPK	0x03	PIN Key
#define PED_TAK	0x04	MAC Key
#define PED_TDK	0x05	DES Key for encryption
#define PED_TEK	0X06	DES Key for decryption
#define PED_TTK	0x09	Dukpt Key

5.6.2. Ped.WriteKey

Function

Write key into pos

Parameter

Parameter name	Type	Description
srcKeyType	int	Source key type: It's 0x00 if you want to write the plaintext of the key (only when you dont inject the TLK you can write thr plaintext) 2、PED_TLK 3、PED_TMK
srcKeyIdx	int	Source key index 1、 It's 0x00 if you want to write the plaintext of the key (only when you dont inject the TLK you can write thr plaintext) 2、PED_TLK:0x01 3、PED_TMK :0x01~0x40
dstKeyType	int	Target key type: PED_TLK PED_TMK PED_TPK PED_TAK PED_TDK PED_TEK PED_TTK

		PS:dstKeyType>= srcKeyType
dstKeyIdx	int	Target key index: PED_TLK:0x01 PED_TMK:1~64 When using the same key type and the same key index as the source key writing PED_TPK, PED_TAK, PED_TDK, PED_TEK, PED_TTK, the index of the target key should be different
algthflag	int	Algorithm identifies 0x00:3DES 0x10:AES 0x20:SM4
dstKeyLen	int	Length of target key: TLK:16 or 24bytes ucDstKeyLen only be 8/16/24, and ,when ucDstKeyLen is 8,the key should only used for DES;ucDstKeyLen is 16 or 24,can used for TDES
bufIn	Byte[]	The plaintext or ciphertext of the target key
iCheckMode	int	Authentication Mode：0、1、2、3
checkbuflen	int	The length of checkvalue
aucCheckBuf	Byte[]	It is the Checkvalue aucCheckBuf: iCheckMode=0: aucCheckBuf is invalid,we will not check the value iCheckMode=1： To do TDES to 8 bytes 0x00 with target key,and the first 4bytes are the key check value iCheckMode=2 do odd parity to the key and then do same things as mode=1 iCheckMode=3 do even parity to the key and then do same things as mode=1

Return

Return Value	Description
RspCode.RSPOK	Inject key successful
RspCode.RSPERR	Inject key failed

5.6.3. Ped.WriteTIK

Function

Write the TIK

Parameter

Parameter name	Type	Description
GroupIdx	int	[1~10] DUKPT key group index
SrcKeyIdx	int	Source key index 0~1, =0， the key is the plaintxt ,only when TLK is not exist we can write the plaintxt =1， TLK is used to decrypt the ciphertext for TIK
KeyLen	int	Length of TIK,Now DUKPT algorithm supports key length in bytes 8/16
KeyValueIn	Byte[]	The plaintxt or ciphertext of TIK
KsnIn	byte	Initialization KSN (key sequence number), 10 bytes
mode	int	checkmode， Please refer WriteKey()
checkbuflen	int	Length of key check value
aucCheckBufIn	Byte[]	Data of key checkvalue

Return

Return Value	Description
RspCode.RSPOK	Write TIK successful
RspCode.RSPERR	Write TIK failed

5.6.4. Ped.GetPinBlock

Function

Get the pin block 。 This function must be called in the thread。

Parameter

Parameter name	Type	Description
mContext	Activity	Context of current activity
pwdText	EditText	The edittext used to indicate the user input
KeyIdx	Byte	[1~64]， index of TPK
ExpPinLenIn	String	The string used to indicate the length of the Pin， such as”0,4,6,8”,it means that you can input the pin with length of 0/4/6/8 The max length of the pin is 12
pan	String	pan
fmtdata	Byte[]	8-byte format data, which may be a random number, transaction serial number or timestamp,But each byte high 4bits and low 4bits, must be between 0xA ~ 0xF
mode	Byte	0x00 ISO9564 format 0 0x01 ISO9564 format 1 0x02 ISO9564 format 3 0x03 HK EPS special format
Timeout_Ms	long	Timeout ,the max value is 300000ms
outdata	Byte[]	pinblk Format is len+value

Return

Return Value	Description
RspCode.KEY_OK	Get pinblock successful
RspCode.KEY_OKDIRECT	Users simply press the Enter key
RspCode.KEY_CANCEL	user presses the Cancel button
RspCode.ERR_PARA	Parameter error
RspCode.RSPERR	Get pinblock failed
other	Other error

5.6.5. Ped.GetMac

Function

MAC calculation(not use DUKPT)。

Parameter

Parameter name	Type	Description
KeyIdx	Byte	The index of TAK 1~64
DataInLen	Short	The length of the data input，<=1024bytes
DataIn	Byte[]	The data used to calculate the MAC packet
MacOut	Byte[]	The mac value
mode	Byte	Operation mode 0x00: The MAC key BLOCK1 do with DES / TDES encryption, encryption results BLOCK2 bitwise XOR TAK do then with DES / TDES encryption, in turn get an 8-byte encryption result 0x01: The BLOCK1 and BLOCK2 perform bitwise exclusive OR, XOR result BLOCK3 XOR bit by bit, in turn, finally got an 8-byte XOR result, the results were used TAK DES / TDES encryption algorithms 0x02: ANSIX9.19 specification, will BLOCK1 TAK do with DES encryption (just take the first 8 bytes of key), the encryption result BLOCK2 bitwise XOR then TAK do with DES encryption, in turn get 8 bytes encryption result, until the last time using DES / TDES encryption

Return

Return Value	Description
RspCode.RSPERR	MAC calculation failed
N(>0)	The length of the MAC value

5.6.6. Ped.CalcDES

Function

Conduct des encryption and decryption(not DUKPT)。

Parameter

Parameter name	Type	Description
KeyIdx	Byte	Key index number, Encryption: TEK index number; Decryption: TDK index number.
DataInLen	Short	The length of the input data, needs to be divisible by 8,and it must be <=1024
DataIn	Byte[]	The data used to des
DataOut	Byte[]	The buffer used to store the result
Mode	Byte	Operation mode ECB decryption: 0x00 ECB encryption: 0x01 CBC decryption: 0x02 CBC encryption: 0x03

Return

Return Value	Description
RspCode.RSPERR	Des failed
N(>0)	The length of the result

5.6.7. Ped.GetPinblkDukpt(DUKPT)

Function

Get pinblock with dukpt key。This function must be called in the thread。

Parameter

Parameter name	Type	Description
mContext	Activity	Context of current activity
pwdEdit	EditText	The edittext used to indicate the user input
GroupIdx	Byte	the ID of the DUKPT group，1~10
ExpPinLenIn	String	The string used to indicate the length of the Pin， such as”0,4,6,8”,it means that you can input the pin with length of

		0/4/6/8 The max length of the pin is 12
pan	String	pan
fmtdata	Byte[]	8-byte format data, which may be a random number, transaction serial number or timestamp,But each byte high 4bits and low 4bits, must be between 0xA ~ 0xF
mode	Byte	PIN BLOCK' format, 00, ISO9564 format 0, KSN+=1 01, ISO9564 format 1, KSN+=1 02, ISO9564 format 2,KSN+=1 03 format HK EPS, KSN+=1 20, ISO9564 format 0, KSN unchanged 21, ISO9564 format 1 ,KSN unchanged 22, ISO9564 format 2, KSN unchanged 23, format HK EPS , KSN unchanged
TimeoutMs	long	Timeout ,the max value is 300000ms
KsnOut	Byte[]	KSN,format is len+value
PinBlockOut	Byte[]	PinBlock, format is len+value

Return

Return Value	Description
RspCode.KEY_OK	Get pinblock successful
RspCode.KEY_OKDIRECT	Users simply press the Enter key
RspCode.KEY_CANCEL	user presses the Cancel button
RspCode.ERR_PARA	Parameter error
RspCode.RSPERR	Get pinblock failed
other	Other error

5.6.8. Ped.GetMacDukpt

Function

MAC calculation(use DUKPT)。

Parameter

Parameter name	Type	Description
GroupIdx	Byte	ID of DUKPT group ， 1~10
DataInLen	Short	Length of input,<=1024
DataIn	Byte[]	the package used to calcilate MAC
MacOut	Byte[]	The MAC value
KsnOut	Byte[]	The KSN value
Mode	Byte	MAC key request and response 00: The BLOCK1 do with the MAC key TDES encryption, encryption results BLOCK2 bitwise XOR MAC before making TDES encryption key, in turn get an 8-byte encryption result. KSN is automatically incremented 01: The BLOCK1 and BLOCK2 by-bit exclusive OR, XOR result BLOCK3 XOR bit by bit, in turn, finally got an 8-byte XOR result, the results of the MAC key TDES encryption algorithms. KSN is automatically incremented 02: ANSIX9.19 specification, will BLOCK1 to do with the MAC key DES encryption (just take the first 8 bytes of key), the encryption result BLOCK2 bitwise XOR MAC before making DES encryption key, in turn get 8 bytes of the encryption result, until the last use TDES encryption. KSN is automatically incremented 20: The BLOCK1 do with the MAC key TDES encryption, encryption results BLOCK2 bitwise XOR MAC before making TDES encryption key, in turn get an 8-byte encryption result. KSN does not automatically add 1 21: The BLOCK1 and BLOCK2 by-bit exclusive OR, XOR result BLOCK3 XOR bit by bit, in turn, finally got an 8-byte XOR result, the results of the MAC key TDES encryption algorithms. KSN does not automatically add 1 22: ANSIX9.19 specification, will BLOCK1 to do with the MAC key DES encryption (just take the first 8 bytes of key), the encryption result BLOCK2 bitwise XOR MAC before making DES encryption key, in turn get 8 bytes of the encryption result, until the last use TDES encryption. KSN does not automatically add 1 Other values are reserved extended MAC algorithm. MAC key of answer 40: The BLOCK1 do with the MAC key TDES encryption, encryption results BLOCK2 bitwise XOR MAC before making TDES encryption key, in turn get an 8-byte encryption result. KSN does not automatically add 1 41: The BLOCK1 and BLOCK2 by-bit exclusive OR, XOR result

		BLOCK3 XOR bit by bit, in turn, finally got an 8-byte XOR result, the results of the MAC key TDES encryption algorithms. KSN does not automatically add 1 42: ANSIX9.19 specification, will BLOCK1 do DES encryption with MAC key (just take the first 8 bytes of key), the encryption result BLOCK2 bitwise XOR then TAK do with DES encryption, in turn get 8 characters encryption result section, until the last use TDES encryption. KSN does not automatically add 1 Other values are reserved extended MAC algorithm.

Return

Return Value	Description
RspCode.RSPOK	Calculation failed
N(>0)	MAC length and KSN length, high four bits is MAC length ,lower four for KSN

5.6.9. Ped.VerifyPlainPin

Function

Offline plaintext PIN authentication。This function must be called in the thread。

Parameter

Parameter name	Type	Description
mContext	Activity	Context of current activity
pwdText	EditText	The edittext used to indicate the user input
IccSlot	Byte	0x00:IC card 0x10:PICC card
ExpPinLenIn	String	The string used to indicate the length of the Pin , such as”0,4,6,8”,it means that you can input the pin with length of 0/4/6/8 The max length of the pin is 12
Mode	Byte	0x00,
TimeoutMs	long	Timeout ,the max value is 300000ms

IccRespOut	Byte[]	Len+value,the status words of command

Return

Return Value	Description
RspCode.KEY_OK	Command execute successful,you can get the status word
RspCode.KEY_OKDIRECT	Users simply press the Enter key
RspCode.KEY_CANCEL	user presses the Cancel button
RspCode.KEY_NOICCARD	NO IC CARD
RspCode.RSPERR	ERROR
other	

5.6.10. Ped.VerifyCipherPin

Function

Offline ciphertext PIN authentication。This function must be called in the thread。

Parameter

Parameter name	Type	Description
mContext	Activity	Context of current activity
pwdEdit	EditText	The edittext used to indicate the user input
IccSlot	Byte	0x00:IC card 0x10:PICC card
ExpPinLenIn	String	The string used to indicate the length of the Pin， such as”0,4,6,8”,it means that you can input the pin with length of 0/4/6/8 The max length of the pin is 12
modlen	Int	module length of the public key
mod	Byte[]	module of the public key
Explen	int	exponent length of the public key
exp	Byte[]	exponent of the public key
iccrandomlen	Int	Length of the Random generated by icc
iccrandom	Byte[]	the Random generated by icc

Mode	Byte	0x00 : EMV2000IC command
TimeoutMs	long	Timeout ,the max value is 300000ms
IccRespOut	Byte[]	Len+value,the status words of command

Return

Return Value	Description
RspCode.KEY_OK	Command execute successful,you can get the status word
RspCode.KEY_OKDIRECT	Users simply press the Enter key
RspCode.KEY_CANCEL	user presses the Cancel button
RspCode.KEY_NOICCARD	NO IC CARD
RspCode.RSPERR	ERROR
other	

5.6.11. Ped.Erase

Function

Erase all keys in PED。 。

Parameter

Parameter name	Type	Description
NULL		

Return

Return Value	Description
RspCode.RSPOK	Erase successful
RspCode.RSPERR	Erase failed

5. 6. 12. Ped. DukptDes

Function

Calculation of DES using the DUKPT key。

Parameter

Name	Type	Description
GroupIdx	byte	DUKPT group ID： 1~10
KeyVarType	byte	Key type： 0x00 : Use the DUKPT MAC key to calculate DES 0x01 : Use the DUKPT DES key to calculate DES
IvLen	byte	The initial vector length : =0, pucIV should be set to null, the function will automatically be initial vector set to 8 0x00; =8, pucIV cannot be null
pucIV	Byte[]	The initial vectors of 8 bytes, the need for CBC encryption and decryption.
DataInLen	Short	Encryption and decryption of data length
DataIn	Byte[]	Encryption and decryption of data
DataOut	Byte[]	The first two bytes is the output length, the follow-up is the output data
KsnOut	Byte[]	The first byte is the length of the value 10, the follow-up is generated KSN (10 bytes)
Mode	Byte	Calculation model： 0x00:EBC Decrypt 0x01:EBC Encryption 0x02:CBC Decrypt 0x03:CBC Encryption

Return

Return Value	Description
RspCode.RSPOK	success
RspCode.RSPERR	fail

5.6.13. Ped.GetDukptKSN

Function

Read the KSN of next calculation.

Parameter

Name	Type	Description
GroupIdx	byte	DUKPT group ID ： 1~10
KsnOut	Byte[]	Len+value

Return

Return Value	Description
RspCode.RSPOK	successful
RspCode.RSPERR	failed

5.6.14. Ped.DukptIncreaseKsn

Function

KSN+1。If function return PED_RET_ERR_DUKPT_NEED_INC_KSN,you should call this function .

Parameter

Name	Type	Description
GroupIdx	byte	DUKPT group ID ： 1~10

Return

Return Value	Description
RspCode.RSPOK	successful
RspCode.RSPERR	failed

5.6.15. Ped.GetRandom

Function

Get random

Parameter

Name	Type	Description
ilen	int	The length of the random that you want to get
random	Byte[]	The buffer used to store the data

Return

Return Value	Description
RspCode.RSPERR	Failed
n	The length of the random

5.7. Printer

5.7.1. Printer.GetPrinterType

Function

Get the type of the printer

Parameter

Name	Type	Description
NULL		

Return

Return Value	Description
RspCode.RSPERR	Get the type failed
0	The printer is not exist
1	It is Thermal Printer

5.7.2. Printer.GetPrinterName

Function

Get the name of the printer

Parameter

Name	Type	Description
name	Byte[]	Used to store the printer name

Return

Return Value	Description
RspCode.RSPERR	Get the name failed
n	The length of the buffer

5.7.3. Printer.GetPrinterPix_X

Function

Get the number of pixels in the X-axis of the printer

Parameter

Name	Type	Description
NULL		

Return

Return Value	Description
RspCode.RSPERR	failed
n	The number of Pixels

5.7.4. Printer.GetPrinterPix_Y

Function

Get the number of pixels in the Y-axis of the printer

Parameter

Name	Type	Description
NULL		

Return

Return Value	Description
RspCode.RSPERR	failed
n	The number of Pixels

5.7.5. Printer.GetPrinterTemp

Function

Getting Printer current temperature

Parameter

Name	Type	Description
NULL		

Return

Return Value	Description
RspCode.RSPERR	failed
n	The temperature

5.7.6. Printer.SetPrinterPara

Function

Setting up the printer gray

Parameter

Name	Type	Description
gray	short	The gray

Return

Return Value	Description
RspCode.RSPERR	failed
RspCode.RSPOK	successful

5.7.7. Printer.SetFontPath

Function

Set the font,the defalut is ”DroidSansFallback.ttf”

Parameter

Name	Type	Description
fontpath	string	Font name ,the suffix is “.ttf” , the font should exist

		in”system\font\”

Return

Return Value	Description
RspCode.RSPERR	failed
RspCode.RSPOK	successful

5.7.8. Printer.SetFontSize

Function

Set the fontsize

Parameter

Name	Type	Description
fontsize	int	The fontsize

Return

Return Value	Description
void	

5.7.9. Printer.SetFont

Function

set the font and the fontsize

Parameter

Name	Type	Description
fontpath	string	Font name ,the suffix is “.ttf” , the font should exist in”system\font\”

fontsize	int	fontsize

Return

Return Value	Description
RspCode.RSPERR	failed
RspCode.RSPOK	successful

5.7.10. Printer.SetLineSpace

Function

Set the line space,the unit is pixels,the default value is 1 pix

Parameter

Name	Type	Description
linespace	int	Line space

Return

Return Value	Description
void	

5.7.11. Printer.SetProperty

Function

Sets the printing effects, italic and bold effects currently supported

Parameter

Name	Type	Description

property	int	0:no effect 1:bold 2:italic

Return

Return Value	Description
void	

5.7.12. Printer.SetCharSpace

Function

Set word spacing， default value is 5 pixes

Parameter

Name	Type	Description
charspace	int	The word spaceing

Return

Return Value	Description
void	

5.7.13. Printer.ClearPrintData

Function

Clear the contents of the current print buffer

Parameter

Name	Type	Description

null		

Return

Return Value	Description
Void	

5.7.14. Printer.GetCursorX

Function

Get the current cursor position in X axis.

Parameter

Name	Type	Description
null		

Return

Return Value	Description
x	Current cursor position in x axis

5.7.15. Printer.GetCursorY

Function

Get the current cursor position in Y axis.

Parametr

Name	Type	Description
NULL		

Return

Return Value	Description
y	Current cursor position in y axis

5.7.16. Printer.AddString

Function

Add buffer to print from (x,y)

Parameter

Name	Type	Description
data	string	The data used to printed
x	int	x
y	int	y

Return

Return Value	Description
RspCode.RSPERR	Failed
RspCode.RSPOK	Successful

5.7.17. Printer.AddString

Function

Add buffer to print from current cursor

Parameter

Name	Type	Description
data	string	The data used to printed

Return

Return Value	Description
RspCode.RSPERR	failed
RspCode.RSPOK	successful

5.7.18. Printer.AddBmpData

Function

Add bmp to print from (x,y).Image format must be monochrome bmp images

Parameter

Name	Type	Description
bmpPath	string	The bmp path
x	int	x
y	int	y

Return

Return Value	Description
RspCode.RSPERR	failed
RspCode.RSPOK	successful

5.7.19. Printer.AddBmpData

Function

Add bmp to print from current cursor.Image format must be monochrome bmp images

Parameter

Name	Type	Description
bmpPath	string	The bmp path

Return

Return Value	Description
RspCode.RSPERR	failed
RspCode.RSPOK	Successful

5.7.20. Printer.Print

Function

print the data in the print buffer

Parameter

Name	Type	Description
null		

Return

Return Value	Description
RspCode.RSPERR	failed
RspCode.RSPOK	successful

5.7.21. Printer.GetLineSpace

Function

Get current line space

Parameter

Name	Type	Description
null		

Return

Return Value	Description
n	current line space

5.7.22. Printer.GetCharSpace

Function

Get current char space

Parameter

Name	Type	Description
null		

Return

Return Value	Description
n	current char space

5.7.23. Printer.GetFontSize

Function

Get current fontsize

Parameter

Name	Type	Description
null		

Return

Return Value	Description
n	Curent font size

5.7.24. Printer.GetLastCursor_X

Function

Get the last character of the X-axis where the pixel position

Parameter

Name	Type	Description
NULL		

Return

Return Value	Description
n	The x position of last character

5.7.25. Printer.GetLastCursor_Y

Function

Get the last character of the Y-axis where the pixel position

Parameter

Name	Type	Description
NULL		

Return

Return Value	Description
n	The y position of last character

5.7.26. Printer.SetCursor

Function

Set the current cursor pixel position

Parameter

Name	Type	Description
x	int	The x postion
y	int	The y postion

Return

Return Value	Description
void	

5.7.27. Printer.PrintString

Function

Print the string ,and will add the “\r\n” automatically

Parameter

Name	Type	Description
dataString	String	The string will be printed

Return

Return Value	Description
RspCode.RSPERR	falied
RspCode.RSPOK	successful

5.7.28. Printer.PrintBmp

Function

Print the bmp immediacy

Parameter

Name	Type	Description
bmppath	String	The bmp path,Image format must be monochrome bmp images

Return

Return Value	Description
RspCode.RSPERR	failed
RspCode.RSPOK	successful

6. Appendix

6.1. MiscDevice

This module is used to power up and power down POS 。

6.1.1. MiscDevice.setPinHigh

Function

Set the pin to high.。

Parameter

Name	Type	Description
pinId	Int	pin

Return

Return Value	Description
0	
other	

6.1.2. MiscDevice.setPinLow

Function

Set the pin to low。

Parameter

Name	Type	Description
pinId	Int	pin

Return

Return Value	Description
0	OK
other	FAILED

6.1.3. Sample Code

```
MiscDevice        mMiscDevice=newMiscDevice(MiscUtil.POS_MISC_DEV,MiscUtil.POS_MISC_IO);

//power                                                                                    on
mMiscDevice.setPinHigh(MiscUtil.POS_PIN_PWR);
mMiscDevice.setPinHigh(MiscUtil.POS_PIN_PWD);

//power                                                                                    off
mMiscDevice.setPinLow(MiscUtil.POS_PIN_PWR);
mMiscDevice.setPinLow(MiscUtil.POS_PIN_PWD);
```

6.2. Error code

```
public class RspCode
{
    public static final int   RSPOK =     0;
    public static final int   RSPERR      =     -1;
    /***/
    public static final int   PED_RET_ERR_START    =( -300);
    /**the key is not exist */
    public static final int   PED_RET_ERR_NO_KEY        =(PED_RET_ERR_START-1);
    /**the index of the key is error*/
    public static final int   PED_RET_ERR_KEYIDX_ERR        =(PED_RET_ERR_START-2);

    /*the type of the source key is error or it's level is lower than the    destnatiion key's when
writing the key   */
    public static final int   PED_RET_ERR_DERIVE_ERR       =(PED_RET_ERR_START-3);
    /**check the key fail*/
    public static final int   PED_RET_ERR_CHECK_KEY_FAIL =(PED_RET_ERR_START-4);
    /**do not input the pin*/
    public static final int   PED_RET_ERR_NO_PIN_INPUT      =(PED_RET_ERR_START-5);
```

```
        /**pin input is canceled*/
        public static final int    PED_RET_ERR_INPUT_CANCEL      =(PED_RET_ERR_START-6);

        /*time interval is not enough */
        public static final int    PED_RET_ERR_WAIT_INTERVAL    =(PED_RET_ERR_START-7);

        /**KCV model error,unspported*/
        public static final int    PED_RET_ERR_CHECK_MODE_ERR =(PED_RET_ERR_START-8);

        /**you have no right to use the key, it is not equal between the PED current key label value
and the key tag value will be used */
        public static final int    PED_RET_ERR_NO_RIGHT_USE      =(PED_RET_ERR_START-9);
        /**the type of the key is wrong*/
        public static final int    PED_RET_ERR_KEY_TYPE_ERR      =(PED_RET_ERR_START-10);
        /**the len expected of the pin is wrong */
        public static final int    PED_RET_ERR_EXPLEN_ERR          =(PED_RET_ERR_START-11);
        /**destination key index is wrong, not in range*/
        public static final int    PED_RET_ERR_DSTKEY_IDX_ERR =(PED_RET_ERR_START-12);

        /**The    index of the source key is wrong, not in range,or source key type values greater
than objective key type*/
        public static final int    PED_RET_ERR_SRCKEY_IDX_ERR =(PED_RET_ERR_START-13);
        /**length of the key is wrong*/
        public static final int    PED_RET_ERR_KEY_LEN_ERR    =(PED_RET_ERR_START-14);
        /**timeout when input pin*/
        public static final int    PED_RET_ERR_INPUT_TIMEOUT    =(PED_RET_ERR_START-15);
        /**ic card is not exist*/
        public static final int    PED_RET_ERR_NO_ICC                   =(PED_RET_ERR_START-16);
        /**ic card is not initialized*/
        public static final int    PED_RET_ERR_ICC_NO_INIT        =(PED_RET_ERR_START-17);
        /**the index of the dukpt group is wrong*/
        public static final int    PED_RET_ERR_GROUP_IDX_ERR =(PED_RET_ERR_START-18);
        /**point is null*/
        public static final int    PED_RET_ERR_PARAM_PTR_NULL        =(PED_RET_ERR_START-19);
        /**PED is attacked*/
        public static final int    PED_RET_ERR_TAMPERED =(PED_RET_ERR_START-20);
        /**PED err*/
        public static final int    PED_RET_ERROR =(PED_RET_ERR_START-21);
        /**memory is not enough*/
        public static final int    PED_RET_ERR_NOMORE_BUF =(PED_RET_ERR_START-22);
        /**need permission*/
        public static final int    PED_RET_ERR_NEED_ADMIN =(PED_RET_ERR_START-23);
        /**DUKPT has over*/
        public static final int    PED_RET_ERR_DUKPT_OVERFLOW =(PED_RET_ERR_START-24);
```

```
     /**KCV error*/
    public static final int    PED_RET_ERR_KCV_CHECK_FAIL =(PED_RET_ERR_START-25);
     /**key type and source key type ID does not match the source key */
     public static final int    PED_RET_ERR_SRCKEY_TYPE_ERR =(PED_RET_ERR_START-26);
    /**command is not supported*/
    public static final int    PED_RET_ERR_UNSPT_CMD    =(PED_RET_ERR_START-27);
    /**communition error*/
    public static final int    PED_RET_ERR_COMM_ERR =(PED_RET_ERR_START-28);
    /**have no user author key*/
    public static final int    PED_RET_ERR_NO_UAPUK =(PED_RET_ERR_START-29);
    /**get system server failed*/
    public static final int    PED_RET_ERR_ADMIN_ERR      =(PED_RET_ERR_START-30);
    /**ped is inactivity*/
    public static final int    PED_RET_ERR_DOWNLOAD_INACTIVE =(PED_RET_ERR_START-31);
     /**KCV    Odd parity failed*/
    public static final int    PED_RET_ERR_KCV_ODD_CHECK_FAIL =(PED_RET_ERR_START-32);
    /**read PED data failed*/
    public static final int    PED_RET_ERR_PED_DATA_RW_FAIL =(PED_RET_ERR_START-33);
    /**operate ic card failed*/
    public static final int    PED_RET_ERR_ICC_CMD_ERR =(PED_RET_ERR_START-34);

    /**key value is wrong*/
    public static final int    PED_RET_ERR_KEY_VALUE_INVALID =(PED_RET_ERR_START-35);
    /**the key is not exist*/
    public static final int    PED_RET_ERR_KEY_VALUE_EXIST =(PED_RET_ERR_START-36);
    /** commution parmter is invalid*/
    public static final int    PED_RET_ERR_UART_PARAM_INVALID =(PED_RET_ERR_START-37);

    /**the index of the key selected is wrong*/
    public    static    final    int          PED_RET_ERR_KEY_INDEX_NOT_SELECT_OR_NOT_MATCH =(PED_RET_ERR_START-38);

    /**user press the “clear”*/
    public static final int    PED_RET_ERR_INPUT_CLEAR      =(PED_RET_ERR_START-39);

    /***/
    public static final int    PED_RET_ERR_LOAD_TRK_FAIL      =(PED_RET_ERR_START-40);
    /***/
    public static final int    PED_RET_ERR_TRK_VERIFY_FAIL      =(PED_RET_ERR_START-41);

    public static final int    PED_RET_ERR_MSR_STATUS_INVALID      =(PED_RET_ERR_START-42);

    public static final int    PED_RET_ERR_NO_FREE_FLASH =(PED_RET_ERR_START-43);
    /**DUKPT KSN need add 1*/
    public static final int PED_RET_ERR_DUKPT_NEED_INC_KSN =(PED_RET_ERR_START-44);
    /**KCV MODE error */
    public static final int  PED_RET_ERR_KCV_MODE_ERR    =(PED_RET_ERR_START-45);
    /**NO KCV*/
    public static final int  PED_RET_ERR_DUKPT_NO_KCV    =(PED_RET_ERR_START-46);
    // pin input is canceled by pressing FN/ATM4
    public static final int  PED_RET_ERR_PIN_BYPASS_BYFUNKEY  = (PED_RET_ERR_START-47);
    //check MAC error
    public static final int PED_RET_ERR_MAC_ERR =(PED_RET_ERR_START-48);
    //check CRC error
    public static final int PED_RET_ERR_CRC_ERR =  (PED_RET_ERR_START-49);

    //-352 pin error
    public static final int  PED_RET_ERR_PWD_ERR    =(PED_RET_ERR_START-52);
    //-353 pin need to  reset
    public static final int  PED_RET_ERR_PWD_NEEDRESET    =(PED_RET_ERR_START-53);
    //-354 pin enter time out
    public static final int  PED_RET_ERR_PWD_ENTERTIMEOUT    =(PED_RET_ERR_START-54);


    public static final int  ERR_BASE =     (-60000);
    public static final int  ERR_UARTSEND =(ERR_BASE-1);//uart send failed
    public static final int  ERR_UARTSENDTIMEOUT  =(ERR_BASE-2);//uart recv ack failed
    public static final int  ERR_UARTRECVRSPTIMEOUT  =(ERR_BASE-3);//uart recv response failed
    public static final int  ERR_UARTRSPLRC=(ERR_BASE-4);//response lrc error
    public static final int  ERR_UARTCMDLRC  =(ERR_BASE-5);//cmd lrc error
    public static final int  ERR_UARTRSPDATAFMT=(ERR_BASE-6);//cmmand format error
    public static final int  ERR_USERINTERRUPTRECV=(ERR_BASE-7);//user interrupt receive the response
    public static final int  ERR_MEMFAILED=(ERR_BASE-8);//memory is not enough
    public static final int  ERR_BMPBMERR=(ERR_BASE-9);//not BMP
    public static final int   ERR_BMPWIDTHERR=(ERR_BASE-10);//the width of the bmp is not384
    public static final int  ERR_BMPCLORERR  =(ERR_BASE-11);//it’s not monochrome bitmap
    public static final int  ERR_FILEOPENERR=(ERR_BASE-12);//file open failed
    public static final int  ERR_PARA =(ERR_BASE-13);//paramter error
    public static final int  ERR_FILENOTEXIST =(ERR_BASE-14);//file not exist

    public static final int SC_VCCERR        =(-2100);     //    voltage model eror
    public static final int SC_SLOTERR       =(-2101);     //    card slot error
    public static final int SC_PARERR        =(-2102);     //    odd-even error
    public static final int SC_PARAERR         =(-2103);     //   paramter null
    public static final int SC_PROTOCALERR =(-2104);         //   protocal error
    public static final int SC_DATALENERR   =(-2105);        //     data len error
    public static final int SC_CARDOUT       =(-2106);       //   card has ejected
    public static final int SC_NORESET        =(-2107);      //   card is poweroff
    public static final int SC_TIMEOUT        =(-2108);      //   card communition timeout
    public static final int SC_PPSERR          =(-2109);       //PPS error
    public static final int SC_ATRERR          =(-2110);     //     ATR error
    public static final int SC_APDUERR        =(-2111);      //   card communition failed

    public static final int RET_RF_START                      =(-3000);
    public static final int RET_RF_ERR_PARAM                       =(RET_RF_START-1); /*paramter
error*/
    public static final int RET_RF_ERR_NO_OPEN                      =(RET_RF_START-2); /*rf model
is not opened*/

    public static final int RET_RF_ERR_NOT_ACT                        =(RET_RF_START-3); /*card is
not activated*/
    public static final int RET_RF_ERR_MULTI_CARD                 =(RET_RF_START-4); /*confict*/
    public static final int RET_RF_ERR_TIMEOUT                                 =(RET_RF_START-5);
/*timeout*/
    public static final int RET_RF_ERR_PROTOCOL                     =(RET_RF_START-6); /*protocal
error*/

    public static final int RET_RF_ERR_TRANSMIT                     =(RET_RF_START-7); /*transmit
error*/
    public static final int RET_RF_ERR_AUTH                              =(RET_RF_START-8); /*M1 auth
failed*/
    public static final int RET_RF_ERR_NO_AUTH                     =(RET_RF_START-9); /**/
    public static final int RET_RF_ERR_VAL                           =(RET_RF_START-10); /**/

    public static final int RET_RF_ERR_CARD_EXIST                     =(RET_RF_START-11); /*card
always in the filed*/
    public static final int RET_RF_ERR_STATUS                             =(RET_RF_START-12); /*card
status error*/

    public static final int RET_RF_ERR_OVERFLOW                  =(RET_RF_START-13);
    public static final int RET_RF_ERR_FAILED                      =(RET_RF_START-14); /**/

    public static final int RET_RF_ERR_COLLERR                    =(RET_RF_START-15);
    public static final int RET_RF_ERR_FIFO                          =(RET_RF_START-16); /**/
    public static final int RET_RF_ERR_CRC                            =(RET_RF_START-17);
    public static final int RET_RF_ERR_FRAMING                    =(RET_RF_START-18);
    public static final int RET_RF_ERR_PARITY                       =(RET_RF_START-19);
    public static final int RET_RF_ERR_DES_VAL                     =(RET_RF_START-20); /**/
    public static final int RET_RF_ERR_NOT_ALLOWED                  =(RET_RF_START-21); /**/

    public static final int RET_RF_ERR_CHIP_ABNORMAL               =(RET_RF_START-100); /**/

    public static final int RET_RF_DET_START =(RET_RF_START-200);
    public static final int RET_RF_DET_ERR_INVALID_PARAM      =(RET_RF_DET_START-1);
    public static final int RET_RF_DET_ERR_NO_POWER              =(RET_RF_DET_START-2);
    public static final int RET_RF_DET_ERR_NO_CARD               =(RET_RF_DET_START-3);
    public static final int RET_RF_DET_ERR_COLL                  =(RET_RF_DET_START-4);
    public static final int RET_RF_DET_ERR_ACT                   =(RET_RF_DET_START-5);
    public static final int RET_RF_DET_ERR_PROTOCOL              =(RET_RF_DET_START-6);

    public static final int RET_RF_CMD_START =(RET_RF_START-300);
    public static final int RET_RF_CMD_ERR_INVALID_PARAM      =(RET_RF_CMD_START-1);
    public static final int RET_RF_CMD_ERR_NO_POWER              =(RET_RF_CMD_START-2);
    public static final int RET_RF_CMD_ERR_NO_CARD               =(RET_RF_CMD_START-3);
    public static final int RET_RF_CMD_ERR_TX                    =(RET_RF_CMD_START-4);
    public static final int RET_RF_CMD_ERR_PROTOCOL              =(RET_RF_CMD_START-5);

    public static final int KEY_OK = -70000;
    public static final int KEY_CANCEL = KEY_OK-1;
    public static final int KEY_OKDIRECT = KEY_OK-2;
    public static final int KEY_NOICCARD = KEY_OK-3;


}
```

7. References

None