MUNBYN

IPDA098 Android POS

SDK Manual

1.00

MUNBYN

Version History

Date	By	Changes	Version
2021-8-23	Byron Zhong, William	• Initial Version	1.00
		•	
		•	
		•	
		•	
		•	
		•	
		•	

MUNBYN

Contents

1.0. Overview

1.1. Introduction

The manual is the instruction of all API defined by MUNBYN for android application development based on IPDA098 smart POS. Along with this instruction document, usually one lib file and one ZIP file of demo cod will be given.

1.2. Usage

1.2.1. Import the SDK for Android Studio

The SDK includes two files. The "ICiontekPosService.aidl" file must be kept in path "aidl/com/ciontek/ciontekposservice", and the "posApiHelper.java" in path "com/ctk/sdk".

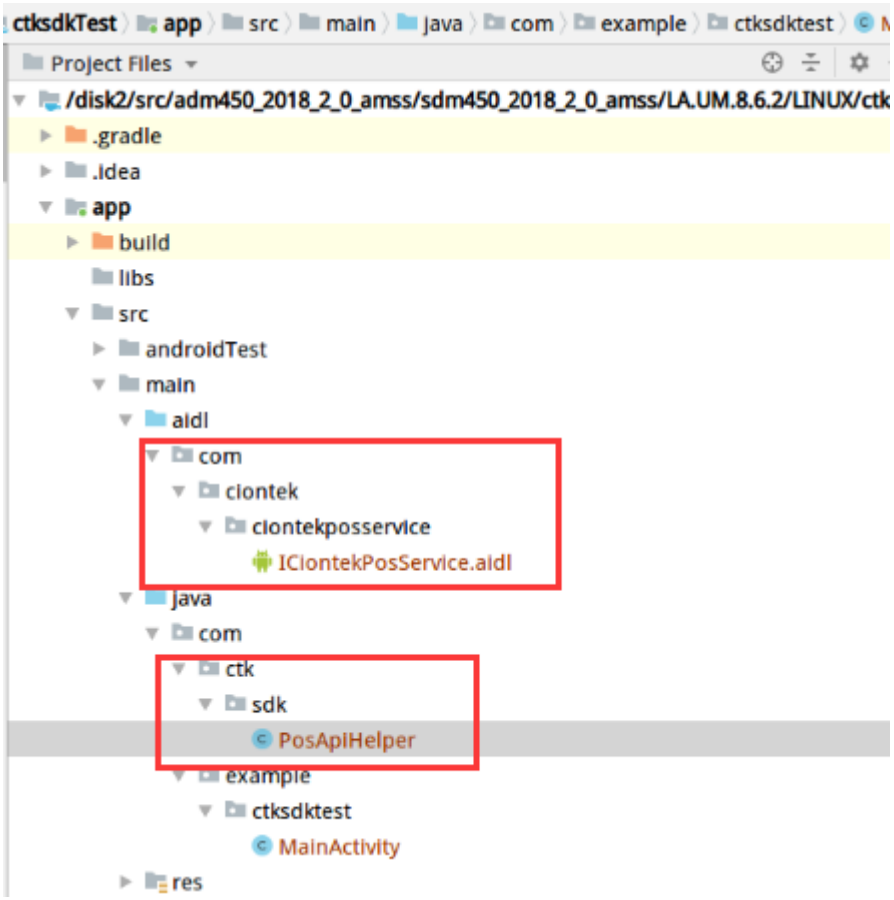

Figure 1-1 The Library Path

Class PosApiHelper describe APIs for IPDA098, more detailed introduction in the APIs list in posApiHelper.java.

By get a PosApiHelper instance to call APIs, for example:

```
PosApiHelper posApiHelper = posApiHelper.getInstance();
posApiHelper.SysGetVersion(version);
```

1.2.2. Runtime Environment

For IPDA098, please make sure the android build number is higher or equal to a51_v0.13_20210625g.

Hints:

Most of the APIs are time consuming call, so it is recommended to create a new thread to invoke them.

2.0. API Description

2.1. Contact IC Card

2.1.1. ICC Check

/**********

*** Description:** Check if there is a Contact IC Card in the specified Slot

*** Input:** Slot, the Slot number

0x00, IC Card Slot

0x01, PSAM1 Slot, for IPDA098, there is only PSAM1 Slot exist

0x02, PSAM2 Slot

*** Output:** None

*** Return:** 0, there is a card or SAM inserted in the specified Slot.

Other, failed

**********/

```
public int IccCheck(byte Slot)
```

Example:

```
ret = posApiHelper.IccCheck(1);
```

2.1.2. ICC Open

/**********

*** Description:** Resets the Contact IC card and return the ATR

*** Input:** Slot:

0x00, IC Card Slot

0x01, PSAM1 Slot, for IPDA098, there is only PSAM1 slot exist

0x02, PSAM2 Slot

vccMode, Card voltage selection

1, 5V; 2, 3V; 3, 1.8V.

*** Output:** atr, answer to reset, get the ATR message from the card reset

*** Return:** 0, Succeed.

(-2403) Channel Error

(-2405) The card is pulled out or not

(-2404) Protocol error

(-2500) Voltage mode error of IC card reset

(-2503) Communication failure.

**********/

```
public int IccOpen(byte slot, byte vccMode, byte[] atr)
```

Example:

```
byte ATR[] = new byte[41];
```

```
ret = posApiHelper.IccOpen(1, 1, ATR);
```

2.1.3. ICC Command

```
/*****************************************************************************************************
```

*** Description:** Read and Write IC Card

* Input: Slo No.:

0x00—IC Card Channel;

0x01—PSAM1 Card Channel;

0x02—PSAM2 Card Channel;

ApduSend:

APDU command

If both LC and LE exist, you should set "LC = X; LE = 0x01"

*** Output:** APDU response

*** Return:** int

0 Execute successfully

(-2503) Communication timeout

(-2405) The cards are put out in the transaction

(-2401) Parity error

(-2403) Select Channel error

(-2400) Sending data too long（LC）

(-2404) The Protocol error（is Not T = 0 or T = 1）

(-2406) No reset card

```
*****************************************************************************************************/
public int IccCommand(byte Slot, byte[] apduSend, byte[] apduResp)
```

Example:

```
byte cmd[] = new byte[4];
cmd[0] = 0x00;          //0-3 cmd
cmd[1] = (byte) 0x84;
cmd[2] = 0x00;
cmd[3] = 0x00;
short lc = 0x00;
short le = 0x04;
String sendmsg = "";
byte [] dataIn = sendmsg.getBytes();
APDU_SEND ApduSend = new APDU_SEND(cmd, lc, dataIn, le);
APDU_RESP ApduResp = null;
byte[] resp = new byte[516];
ret = posApiHelper.IccCommand(Slo, ApduSend.getBytes(), resp);
```

2.1.4. ICC Close

```
/*****************************************************************************************************
* Description: Close IC Card
* Input: Slot No.:
        0x00—IC and Channel
        0x01—PSAM1 and Channel
        0x02—PSAM2 and Channel
* Output: None
* Return:  int
            0: successfully
           Other : failure
*****************************************************************************************************/
public int IccClose(byte Slot)
```

Example:

```
ret = posApiHelper.IccClose(1);
```

2.2. NFC

Please refer to the standard android NFC development.

https://developer.android.com/guide/topics/connectivity/nfc?hl=en

2.3. Printer

2.3.1. Print Init

```
/*****************************************************************************************************
* Description: Initialize the printing parameters and load font
* Input: gray, the level of density, 1-high, 2-medium, 3-low
        fontHeight, the height of font, 16 or 24.
        fontWidth, the width of font, 16 or 24
        fontZoom, the zoom of font, 0x00 or 0x33
* Output: None
* Return:  0, initial succeed.
          Others, failed
          -4001, print busy
          -4002, print no paper
          -4003, print data error
          -4004, print fault
          -4005, print too heat
          -4006, print not finished
```

-4007, print no font lib

-4008, print buff overflow

-4009, print set font error

-4010, print get font error

<=0, failed

**********/

```
public int PrintInit(int gray, int fontHeight, int fontWidth, int fontZoom)
```

Example:

```
void testApiSimple()
{
    int ret = posApiHelper.PrintInit(2, 24, 24, 0x33);
    if (ret != 0)   return;
    posApiHelper.PrintStr("Print Tile\n");
    if (ret != 0)   return;
    posApiHelper.PrintStr("--------------------\n");
    posApiHelper.PrintStr("  Print Str2\n");
    posApiHelper.PrintBarcode("123456789", 360, 120, BarcodeFormat.CODE_128);
    posApiHelper.PrintBarcode("123456789", 240, 240, BarcodeFormat.QR_CODE);
    posApiHelper.PrintStr("CODE_128:" + "123456789" + "\n\n");
    posApiHelper.PrintStr("QR_CODE:" + "123456789" + "\n\n");
    posApiHelper.PrintStr("----------------------\n");
    posApiHelper.PrintStart();
}
```

2.3.2. Print Start

/**********

*** Description:** Start Print

*** Input:** None

*** Output:** None

*** Return:** 0, succeed.

Others, failed

-1001/1001, send fail

-1002/1002, receive timeout

-1, short of paper

-2, the temperature is too high

-3, the voltage is too low

8/9, Instruction reply disorder

-1023, status error

-1021, short of paper

-1000/-1016/-1001/-1002/-1003/-1004/-1019/-1017/-1018/-1020, print timeout

-1007/-1008/-1009/-1010/-1011/-1012, print times exceeds limit

-1022, heat error

-1015/-1014, short of paper

**********/

```
public int PrintStart();
```

Example:

```
ret = posApiHelper.PrintStart();
```

2.3.3. Print Check Status

/**********

*** Description:** Check the printer status

*** Input:** None

*** Output:** None

*** Return:** 0, succeed

Others, failed

-1, need paper

-2, high temperature

-3, low battery voltage

**********/

```
public int PrintCheckStatus();
```

Example:

```
ret = posApiHelper.PrintCheckStatus();
```

2.3.4. Print Set Voltage

/**********

*** Description:** Set voltage

*** Input:** voltage, unit 0.1V

*** Output:** None

*** Return:** 0, succeed

Others, failed

**********/

```
public int PrintSetVoltage(int voltage)
```

Example:

```
//Set current voltage as 7.5V
ret = posApiHelper.PrintSetVoltage(75);
```

2.3.5. Print Set Gray

```
/*****************************************************************************************************
```

*** Description:** Set printing density

*** Input:** nLevel, density level, range from 1 to 5

*** Output:** None

*** Return:** 0, succeed

Others, failed

```
*****************************************************************************************************/
```

```
public int PrintSetGray(int nLevel)
```

Example:

```
ret = posApiHelper.PrintSetGray(2);
```

2.3.6. Print Set Font

```
/*****************************************************************************************************
```

*** Description:** Set print font size.

*** Input:** fontHeight, the height of dot matrix font, 16 or 24.

fontWidth, the width of dot matrix font, 16 or 24

zoom, font set as bold or bigger, 0x00 or 0x33

*** Output:** None

*** Return:** 0, succeed

Others, failed

```
*****************************************************************************************************/
```

```
public int PrintSetFont(byte fontHeight, byte fontWidth, byte zoom)
```

Example:

```
ret = posApiHelper.PrintSetFont(16,16,0x33);
ret = posApiHelper.PrintSetFont(24,24,0x00);
```

2.3.7. Print Set Mode

```
/*****************************************************************************************************
```

*** Description:** Set print mode for receipt or label

*** Input:** mode,

0, receipt printing (default)

1, label printing

*** Output:** None

*** Return:** 0, succeed

Others, failed

```
*****************************************************************************************************/
```

```
public int PrintSetMode(int mode)
```

Example:

```
ret = posApiHelper.PrintSetMode(1);
```

2.3.8. Print String

/**********

*** Description:** Set print content.

*** Input:** str, the print content

*** Output:** None

*** Return:** 0, succeed

Others, failed

-4002, need paper

-4003, data error

**********/

```
public int PrintStr(String str)
```

Example:

```
ret = posApiHelper.PrintStr("Android POS IPDA098\n");
```

2.3.9. Print BMP

/**********

*** Description:** Set BMP print content (width <=384, height <=500)

*** Input:** bitmap, BMP photo data

*** Output:** None

*** Return:** 0, succeed

Others, failed

-4003, data error

-4004, fault error

-4008, buff overflow

**********/

```
public int PrintBmp(Bitmap bitmap)
```

Example:

```
Bitmap bmp = BitmapFactory.decodeResource(PrintActivity.this.getResources(),
R.drawable.mbmp);  // R.drawable.mbmp -photo path
ret = posApiHelper.PrintBmp(bmp);
```

2.3.10. Print 1D Bar code

/**********

*** Description:** Set barcode print content

*** Input:** contents, barcode content

desiredWidth, barcode width

desiredHeigth, barcode height

barcodeFormat, barcode standard,

" ODE_128", "CODE_39", "EAN_8", "QR_CODE", "PDF_417", "ITF"

*** Output:** None

*** Return:** 0, succeed

Others, failed

```
*****************************************************************************************************/
public int PrintBarcode(String contents, int desiredWidth, int desiredHeight,
String barcodeFormat)
```

Example:

```
posApiHelper.PrintBarcode("123456789", 360, 120, BarcodeFormat.EAN_8);
posApiHelper.PrintBarcode(("123456789", 240, 240, BarcodeFormat.QR_CODE);
```

2.3.11. Print 2D Bar Code

```
/*****************************************************************************************************
```

*** Description:** Print 2D Bar code

*** Input:** content of the dr code;

desiredWidth, Width;

desiredHeight, Heigh;

barcodeFormat, Coding format;

*** Output:** None

*** Return:** 0, successfully

Others -failure

```
*****************************************************************************************************/
public int PrintQrCode_Cut (String contents, int desiredWidth, int desiredHeight,
String barcodeFormat);
```

Example:

```
String content =
"com.chips.ewallet.scheme://{\"PayeeMemberUuid\":\"a3d7fe8e-873d-499b-9f11-0000
00000000\",\"PayerMemberUuid\":null,\"TotalAmount\":\"900\",\"PayeeSiteUuid\":n
ull,\"PayeeTransId\":\"100101-084850-6444\",\"PayeeSiteReference\":\"\",\"Payee
Description\":null,\"ConfirmationUuid\":null,\"StpReference\":null}";
        posApiHelper.PrintStr("QR_CODE display " );
      posApiHelper.PrintQrCode_Cut(content, 360, 360, BarcodeFormat.QR_CODE);
      posApiHelper.PrintStr("PrintCutQrCode_Str display " );
      posApiHelper.PrintCutQrCode_Str(content,"PK TXT adsad adasd sda",5, 300,
300,"QR_CODE");
```

2.3.12. Print 2D Bar Code with Characters

```
/*****************************************************************************************************
```

*** Description:** print QR code, also print characters below the QR code

*** Input:** qrContent, Content of the dr code;
printTxt, Character next to the qr code;
Distance, Line spacing for input data of "printTxt ";
desiredWidth, Width;
desiredHeight, Heigh;
barcodeFormat, Coding format;

*** Output:** None

*** Return:** 0, successfully

Others -failure

```
**********************************************************************************************************/

public int PrintCutQrCode_Str (String contents, String printTxt, int
distance, int desiredWidth,int desiredHeight, String
barcodeFormat);
```

Example:

```
String content =
"com.chips.ewallet.scheme://{\"PayeeMemberUuid\":\"a3d7fe8e-873d-499b-9f11-0000
00000000\",\"PayerMemberUuid\":null,\"TotalAmount\":\"900\",\"PayeeSiteUuid\":n
ull,\"PayeeTransId\":\"100101-084850-6444\",\"PayeeSiteReference\":\"\",\"Payee
Description\":null,\"ConfirmationUuid\":null,\"StpReference\":null}";

       posApiHelper.PrintStr("QR_CODE display " );
       posApiHelper.PrintQrCode_Cut(content, 360, 360, "QR_CODE");
       posApiHelper.PrintStr("PrintCutQrCode_Str display " );
       posApiHelper.PrintCutQrCode_Str(content,"PK TXT adsad adasd sda",5, 300,
300, BarcodeFormat.QR_CODE);
```

2.3.13. Print Set Underline

```
/**********************************************************************************************************
```

*** Description:** Set the lines and width of underline

*** Input:** x, the value is in HEX format,
The upper four digits are the number of underlined lines, greater than 2 is 2 lines, and less than 2 is 1 line
The lower four bits are the width

*** Output:** None

*** Return:** 0, successfully

Others -failure

```
**********************************************************************************************************/

public int PrintSetUnderline(int x);
```

Example:

```
posApiHelper.PrintSetUnderline (0x1F);
```

2.3.14. Print Set Reverse

```
/**********************************************************************************************************
```

*** Description:** Set the font display reverse mode

*** Input:x:**

0(default) -> normal
1 -> reverse

*** Output:** None

*** Return:** 0, successfully

Others -failure

```
**********************************************************************************************************/

public int PrintSetReverse(int x);
```

Example:

```
posApiHelper.PrintSetReverse(1);
```

2.3.15. Print SetBold

```
/*****************************************************************************************************
```

*** Description:** Set the font display Bold mode

*** Input:** mode:
* 0(default) -> normal
* 1 -> Bold

*** Output:** None

*** Return:** 0, successfully
Others -failure

```
*****************************************************************************************************/
public int PrintSetBold(int x);
```

Example:

```
posApiHelper. PrintSetBold(1);
```

2.3.16. Print Logo

```
/*****************************************************************************************************
```

*** Description:** print a picture by a byte[]

*** Input:** byte[] logo, the byte[] for a picture

*** Output:** None

*** Return:** 0, successfully
Others -failure

```
*****************************************************************************************************/
public int PrintLogo(byte[] logo);
```

Example:

```
posApiHelper.PrintLogo(logo);
```

2.4. Generic APIs

2.4.1. Sys Update

```
/*****************************************************************************************************
```

*** Description:** Payment module firmware upgrade

*** Input:** None

*** Output:** None

*** Return:** 0 successfully
Other failure

```
*****************************************************************************************************/
public int SysUpdate()
```

Example:

```
int ret = posApiHelper.SysUpdate();
```

2.4.2. Sys Get Rand

```
/*****************************************************************************************************
```

*** Description:** To get 8 bytes random number

*** Input:** byte[] rnd,
The random number returned by the MCU

*** Output:** None

*** Return:** 0 successfully
Other failure

**********/

```
public int SysGetRand(byte[] rnd)
```

Example:

```
Byte[] random = new byte[8];
int ret = posApiHelper.SysGetRand(random);
```

2.4.3. Sys Get Version

/**********

*** Description:** Read firmware version

*** Input:** buf, firmware version no.

*** Output:** None

*** Return:** 0 successfully
Other failure

**********/

```
public int SysGetVersion(byte[] buf)
```

Example:

```
byte buf[] = new byte[9];
 ret= posApiHelper.SysGetVersion(buf);
```

2.4.4. Sys Read Chip ID

/**********

*** Description:** Get IC card ID.

*** Input:** buf,
IC card ID.
Len,
length

*** Output:** None

*** Return:** 0 successfully
Other failure

**********/

```
public int SysReadChipID (byte[] buf, int len)
```

Example:

```
byte chipIdBuf[] = new byte[16];
  int ret = posApiHelper.SysReadChipID(chipIdBuf, 16);
```

2.4.5. Sys Write SN

/**********

*** Description:** Write serial no.

*** Input:** SN:
16 byte serial no.

*** Output:** None

*** Return:** 0 successfully
Other failure

```
*****************************************************************************************************************/
public int SysWriteSN(byte[] SN)
```

Example:

```
byte SN[] = new byte[32];
  int ret = posApiHelper.SysWriteSN(SN);
```

2.4.6. Sys Read SN

```
/*****************************************************************************************************************
```

*** Description:** Read serial no.

*** Input:** SN, 16 bytes serial no.

*** Output:** None

*** Return:** 0 successfully
Other failure

```
*****************************************************************************************************************/
public int SysReadSN(byte[] SN)
```

Example:

```
byte SN[] = new byte[32];
       ret= posApiHelper.SysReadSN(SN);
```

2.5. Barcode Scan

2.5.1. Start Scan

```
/*****************************************************************************************************************
```

You can start scan through send a broadcast “ACTION_BAR_TRIGSCAN”, when scan is triggered, the scanner will emit red light for 6 seconds by default, then stop scanning if time out. The timeout index may be configured as below

```
*****************************************************************************************************************/
```

Example:

```
Intent intent = new Intent ("ACTION_BAR_TRIGSCAN");
       mContext.sendBroadcast(intent);

Or:(add timeout)
       Intent intent = new Intent("ACTION_BAR_TRIGSCAN");
       intent.putExtra("timeout", 4);// Units per second, and Maximum 9
        mContext.sendBroadcast(intent);
```

2.5.2. Stop Scan

```
/*****************************************************************************************************************
```

"You can start scan through Send a broadcast of "**ACTION_BAR_TRIGSTOP**".

**********/

Example:

```
intent = new Intent();
        intent.setAction("ACTION_BAR_TRIGSTOP");
        mContext.sendBroadcast(intent);
```

2.5.3. Get scan results

/**********

There are two manners of scan result output, directly fill and API transfer.

In directly fill manner, the return value will be filled directly to "Editview", and you can read the content of Editview as well.

In API transfer manner, you can get the scan results by registering a broadcast receiver "**ACTION_BAR_SCAN**", This broadcast has 3 Parameters.

Parameters: The parameter 1 " **EXTRA_SCAN_DATA**" is the bar code value, of which the data type is String or byte[].

The parameter 2 "**EXTRA_SCAN_LENGTH**" is the bar code data length, of which the data type is int.

The parameter 3 "**EXTRA_SCAN_ENCODE_MODE**" is the coding type of result, value may be 1,2,3 and means UTF-8,GBK, and raw value accordingly.

The parameter 4 "**EXTRA_SCAN_BARTYPE**" is the barcode type, of which the data type is int

**********/

Example:

```
Register broadcase receiver:
        mFilter= new IntentFilter("ACTION_BAR_SCAN");
        mContext.registerReceiver(mReceiver, mFilter);

unregister broadcase receiver:
        mContext.unregisterReceiver(mReceiver);

obtain scan results:
public static final int ENCODE_MODE_UTF8 = 1;
public static final int ENCODE_MODE_GBK = 2;
public static final int ENCODE_MODE_NONE = 3;

String scanResult=""

mReceiver= new BroadcastReceiver()

{
public void onReceive(Context context, Intent intent)

    {
        int length = intent.getIntExtra("EXTRA_SCAN_LENGTH",0);
        int encodeType= intent.getIntExtra("EXTRA_SCAN_ENCODE_MODE",1);
        if (encodeType  ==  ENCODE_MODE_NONE ){
        byte[] data = intent.getByteArrayExtra("EXTRA_SCAN_DATA");
         scanResult= new String (data ,0,length ,Encode);//Encode is the coding type
returned.
```

```
    }
else {
       scanResult=intent.getStringExtra("EXTRA_SCAN_DATA");
     }
    }

 };
```

2.5.4. Scan Settings

```
/*************************************************************************************************************
```

All config may be set in “Setting-Scanner” manually or by sending broadcast “ACTION_BAR_SCANCFG”

```
*************************************************************************************************************/
```

The parameters are defined as follows:

Table 2-1 ACTION_BAR List

parameter	data type	Remarks
EXTRA_SCAN_POWER	INT	= 0 disable scanning = 1 enable scanning Explanation: when the scan head is enabled, system will initialize the scan head.It will take some time, and the relevant scan request is ignored
EXTRA_TRIG_MODE	INT	= 0 as a normal trigger mode = 1 as continuous trigger mode
EXTRA_SCAN_MODE	INT	Filling = 1 : The scan results are filled directly into the editview Api = 2 : The scan results are output by a broadcast
EXTRA_SCAN_AUTOENT	INT	= 0 = 1 Automatically add “Enter” characters after scan
EXTRA_SCAN_NOTY_SND	INT	= 0 close scanning sound = 1 open scanning sound
EXTRA_SCAN_NOTY_VIB	INT	= 0 close Scanning vibration = 1 open Scanning vibration
EXTRA_SCAN_NOTY_LED	INT	= 0 close scanning indicator light = 1 open scanning indicator light

Example:

```
disable scanning
        Intent intent = new Intent ("ACTION_BAR_SCANCFG");
        intent.putExtra("EXTRA_SCAN_POWER", 0);
        mContext.sendBroadcast(intent);

        Enable scanning
        Intent intent = new Intent ("ACTION_BAR_SCANCFG");
        intent.putExtra("EXTRA_SCAN_POWER", 1);
        mContext.sendBroadcast(intent);
```

Or

```
//SCAN_MODE : Api mode
        Intent intent = new Intent ("ACTION_BAR_SCANCFG");
        intent.putExtra-("EXTRA_SCAN_MODE", 2);
        intent.putExtra("EXTRA_SCAN_AUTOENT", 1);
        mContext.sendBroadcast(intent);
```

2.6. App White List & Black List

2.6.1. Enable App Install White List

```
/**********************************************************************************************************
```

*** Description:** enable the function of App White list

*** Input:** None

*** Output:** None

*** Return:** true: success
false: fail

```
**********************************************************************************************************/
public boolean enableAppInstallWhiteList()
```

Example:

```
posApiHelper.enableAppInstallWhiteList();
```

2.6.2. Disable App Install White List

```
/**********************************************************************************************************
```

*** Description:** disable the function of App White list

*** Input:**None

*** Output:** None

*** Return:** true : success
false : fail

```
**********************************************************************************************************/
public boolean disableAppInstallWhiteList()
```

Example:

```
posApiHelper.disableAppInstallWhiteList();
```

2.6.3. Add App To Install White List

```
/**********************************************************************************************************
```

*** Description:** add an app to white list

*** Input:** pkgName,
the APP package name

* **Output:** None

* **Return:** true: success
false: fail

**********/

```
public boolean addAppToInstallWhiteList(String pkgName)
```

Example:

```
String packageNameList = "com.app.package.name"
       posApiHelper.addAppToInstallWhiteList(packageNameList);
```

2.6.4. DelApp From Install White List

/**********

* **Description:** delete an app from white list

* **Input:** None

* **Output:** None

* **Return:** true : success
false : fail

**********/

```
public boolean delAppFromInstallWhiteList(String pkgName)
```

Example:

```
String packageNameList = "com.app.package.name"
       posApiHelper.delAppFromInstallWhiteList(packageNameList);
```

2.6.5. Get App Install White List

/**********U

* **Description:** get the APP white list

* **Input:** None

* **Output:** None

* **Return:** The app white list

**********/

```
public List<String> getAppInstallWhiteList()
```

Example:

```
posApiHelper.getAppUninstallBlackList();
```

2.6.6. Disable App Uninstall Black List

/**********

* **Description:** disable the function of App black list

* **Input:**None

* **Output:** None

* **Return:** true : success
false : fail

**********/

```
Public boolean enableAppUninstallBlackList()
```

Example:

posApiHelper. enableAppUninstallBlackList ();

2.6.7. Enable App Uninstall Black List

```
/*************************************************************************************************************
* Description: disable the function of App black list
* Input: None
* Output: None
* Return: true: success
          false: fail
*************************************************************************************************************/
Public boolean disableAppUninstallBlackList()
Example:
posApiHelper.disableAppUninstallBlackList ();
```

2.6.8. Add App To Uninstall Black List

```
/*************************************************************************************************************
* Description: add an app to black list
* Input: None
* Output: None
* Return: true: success
          false: fail
*************************************************************************************************************/
Public boolean addAppToUninstallBlackList (String pkgName)
Example:
posApiHelper.addAppToUninstallBlackList (pkgName);
```

2.6.9. Del App From Uninstall Black List

```
/*************************************************************************************************************
* Description: delete an app from black list
* Input:None
* Output: None
* Return: true : success
          false : fail
*************************************************************************************************************/
Public boolean delAppFromUninstallBlackList(String pkgName)
Example:
posApiHelper.delAppFromUninstallBlackList (pkgName);
```

2.6.10. Get App Uninstall Black List

```
/*************************************************************************************************************
* Description: get the APP black list
```

*** Input:** None

*** Output:** None

*** Return:** None

**********/

```
Public List<String> getAppUninstallBlackList()
```

Example:

```
posApiHelper.getAppUninstallBlackList();
```

2.7. Android OS API

2.7.1. Install Rom Package

/**********

*** Description: API for Android firmware update, useful for client want to deploy its own OTA system.**

*** Input**: context , Context
romFilePath , rom file path

*** Output:** None

*** Return:** 0 : success
!0 : fail

**********/

```
public int installRomPackage(String romFilePath)
```

Example:

```
String path = "/storage/emulated/0/update.zip";
        File mOsFile=new File(path);
        if(!mOsFile.exists()){
        //TODO
          return;
        }
        boolean flag = posApiHelper.installRomPackage(path);
```

2.7.2. Get OS Version

/**********

*** Description:** Get OS version

*** Input:** None

*** Output:** None

*** Return:** String: the OS version

**********/

```
public String getOSVersion()
```

Example:

```
String osVersion = posApiHelper.getOSVersion();
```

2.7.3. Get Device Id

/**********

*** Description:** Get the device serial number

*** Input:** None

*** Output:** None

*** Return:** String: the device serial number

**********/

```
public String getDeviceId ()
```

Example:

```
String osVersion = posApiHelper.getDeviceId ();
```

2.8. Fiscal module

2.8.1. Fiscal Open

/**********

*** Description:** Power on the fiscal module and open the serial port

*** Input**: baudrate, the baudrate of serial port
size, data bits of serial port
stop, stop bits of serial port
parity, parity bit of serial port
cflow, control options serial port

*** Output:** None

*** Return:** 0: success
-1: fail
-2: uninitialized
-3: parameter error
-4: timeout
-5: init UART port error
-6: read error
-7: write error

**********/

```
public int fiscalOpen(int baudrate, int size, int stop, char parity, char cflow)
```

Example:

```
posApiHelper.fiscalOpen (115200,8,1,'N','N');
```

2.8.2. Fiscal Close

/**********

*** Description:** power off the fiscal and close the UART port

*** Input:**None

*** Output:** None

*** Return:** 0: success
-1: fail
-2: uninitialized
-3: parameter error
-4: timeout
-5: init UART port error
-6: read error
-7: write error

```
**********************************************************************************************************/
public int fiscalClose()
Example:
posApiHelper.fiscalClose();
```

2.8.3. Fiscal Write

```
/**********************************************************************************************************
```

*** Description:** Write data to fiscal by the serial port

*** Input:data**

*** Output:** None

*** Return:** 0: success
-1: fail
-2: uninitialized
-3: parameter error
-4: timeout
-5: init UART port error
-6: read error
-7: write error

```
**********************************************************************************************************/
public int fiscalWrite(byte[] data)
Example:
byte[] cmd = new byte[6];
        cmd[0] = (byte)0x04;
        cmd[1] = (byte)0x01;
        cmd[2] = (byte)0x00;
        cmd[3] = (byte)0x30;
        cmd[4] = (byte)0xff;
        cmd[5] = (byte)0xcd;
         ret = posApiHelper.fiscalWrite(cmd);
```

2.8.4. Fiscal Read

```
/**********************************************************************************************************
```

*** Description:** read data from fiscal by the serial port

*** Input:** Buffer,the buffer for data form serial port
bufLen, the length of the buffer
timeout, timeout for read, unit: ms

*** Output:** None

*** Return:** >0 : the counts read form serial port
<0: read fail
-1: fail
-2: uninitialized
-3: parameter error
-4: timeout
-5: init UART port error
-6: read error
-7: write error

```
**********************************************************************************************************/
int fiscalRead(byte[] buffer,int bufLen,int timeout)
Example:
```

```
byte[] buffer = new byte[36];
readCount = posApiHelper.fiscalRead(buffer,36,500);
```

SCAN THE QR CODE FOR FACEBOOK ONLINE CHAT

Contact us

Email: support@munbyn.com

Whatsapp: +86-17817881067

Skype: live:munbyn

If you meet any problem during using the thermal printer, please contact us.