Ciontek

CS50C SDK Instruction

V1.0.4

Updated on
2022/04/12

Chapter 1 Overview

1.1. Introduction

This document is the instruction of all API defined by Ciontek for developer to progra m own Android application upon this android based smart POS. while the MCU version need to be updated to the latest version to match the SDK version. Along with this instruction document, usually one SDK file and a ZIP file of Demo code will b e given. The Android firmware no earlier than a52c_v0.12_20220412 and MCU version no earlier than V1.3.1 .
The EMV kernal related API is introduced in another instruction file accordingly. Pleas e contact your sales contact for updated version before your integration efforts.

Support : CS50C Android11.0

1.2. Modify records

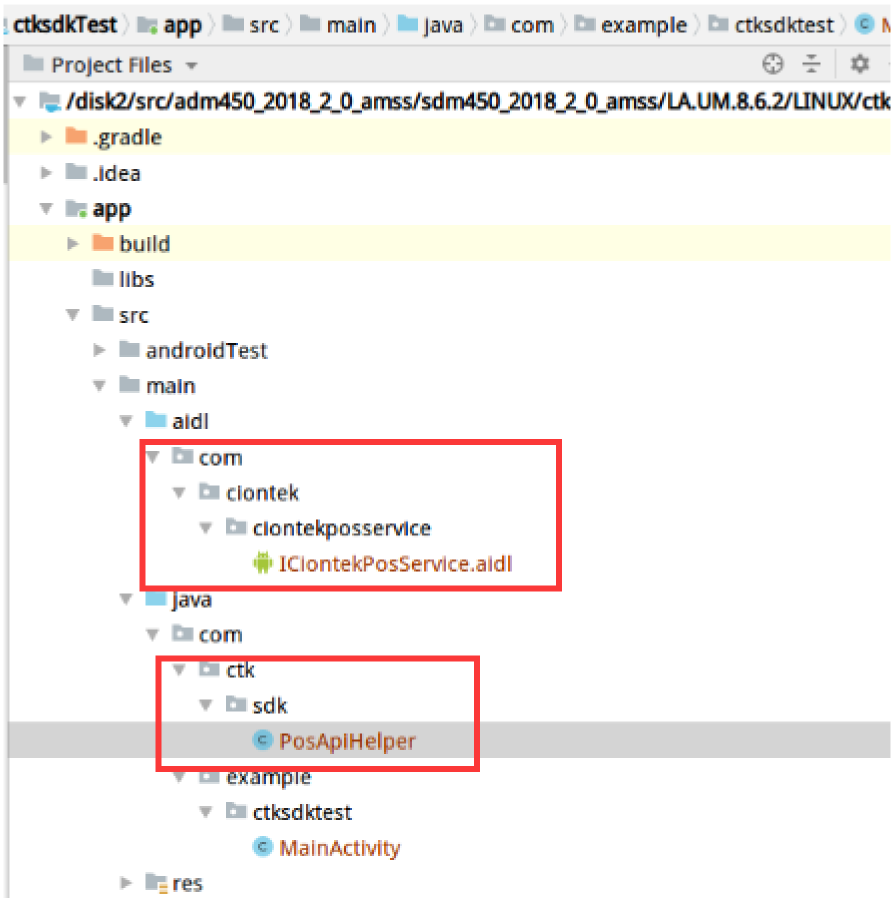
version	author	date	remarks
V1.0.1	Tao	2021-03-11	
V1.0.2	Tao	2021-6-23	1. Add Print API: PrintSetUnderline PrintSetReverse PrintSetBold PrintLogo 2.Discarded the old APP White List APIs that in V1.0.1, And replace it use the new APIs : enableAppInstallWhiteList disableAppInstallWhiteList addAppToInstallWhiteList delAppFromInstallWhiteList getAppInstallWhiteList 3.Add APP black list enableAppUninstallBlackList disableAppUninstallBlackList addAppToUninstallBlackList delAppFromUninstallBlackList getAppUninstallBlackList
V1.0.3	Tao	2021-7-27	Delete the caution of print APIs
V1.0.4	Tao	2022-4-12	1. Add PrintLabLocate for support print label

			2. Add Magnetic card APIs

1.3. Usage

1.3.1. Import the SDK for Android studio

Unzip the sdk “ciontek – cs50c - SDK - v *.zip” and merger the sdk files to your android studio project.
NOTE: Keep “ICiontekPosService.aidl” in path “aidl/com/ciontek/ciontekposservice”, And “PosApiHelper.java” in path “com/ctk/sdk”



```
ctksdkTest 〉 app 〉 src 〉 main 〉 java 〉 com 〉 example 〉 ctksdktest 〉 M
Project Files ▾
▾ /disk2/src/adm450_2018_2_0_amss/sdm450_2018_2_0_amss/LA.UM.8.6.2/LINUX/ctk
  ▸ .gradle
  ▸ .idea
  ▾ app
    ▸ build
      libs
    ▾ src
      ▸ androidTest
      ▾ main
        ▾ aidl
          ▾ com
            ▾ ciontek
              ▾ ciontekposservice
                  ICiontekPosService.aidl
        ▾ java
          ▾ com
            ▾ ctk
              ▾ sdk
                  PosApiHelper
            ▾ example
              ▾ ctksdktest
                  MainActivity
        ▸ res
```

Class PosApiHelper describe APIs for Ciontek CS50C, more detailed introduction please see the APIs list in PosApiHelper.java

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

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

1.3.2. Runtime environment

For CS50C, Please make sure the android build number is a52c_v0.12_20220412 or after

Special instructions:
Because of the SDK APIs may be a time consuming call, So you need create a new thread to invoke them

Chapter 2 Contact Type IC Card

2.1. IccCheck

Function prototype	public int IccCheck(byte slot)

Parameter description	slot cassette No.：
	0x00—IC Card Channel;
	0x01—PSAM1 Card Channel;
	0x02—PSAM2 Card Channel;

Return	int 0 ： The card was detected and inserted Other： failure
Function description	Check if there is a card in the specified cassette
Example	ret = posApiHelper.IccCheck(1);

2.2. IccOpen

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

Parameter description	Slotcassette No.：
	0x00—IC Card Channel;
	0x01—PSAM1 Card Channel;
	0x02—PSAM2 Card Channel;

	VCC_Mode Read Card Voltage：
	1---5V;
	2---3V;
	3---1.8V;
	ATR：
	Card reset response. (at least 32+1bytes of space). The contents are length (1 byte) + reset response content

Return	int 0 Initialization success. (-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.
Function description	Initialize the IC card and return the response content of the card

Example	byte ATR[] = new byte[41];
	ret = posApiHelper.IccOpen(1, 1, ATR);

2.3. IccCommand

Function prototype	public int IccCommand(byte slot, byte[] apduSend, byte[] apduResp)
Parameter description	Slotcassette No.： 0x00—IC Card Channel; 0x01—PSAM1 Card Channel; 0x02—PSAM2 Card Channel; ApduSend： sent to the card's apdu ApduResp： Receive the card's apdu of returned
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
Function description	Read and Write IC Card If both LC and LE exist, you should set "LC = X; LE = 0x01"

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(slot, ApduSend.getBytes(), resp);

2.4. IccClose

Function prototype	public int IccClose(byte slot)
Parameter description	Slotcassette No.： 0x00—IC and Channel 0x01—PSAM1 and Channel 0x02—PSAM2 and Channel
Return	int 0 : successfully Other :failure
Function description	Close IC Card
Example	ret = posApiHelper.IccClose(1);

Chapter 3 Print

Herein the APIs are defined for the integrated printer of CS50C. If label printer feature required please contact sales person to ensure the Hardware version support or not.
The Bluetooth connection/ESC protocol printer feature is also supported by default firmware .

3.1. PrintInit

Function prototype	public int PrintInit(int gray, int fontHeight, int fontWidth, int fontZoom)
Parameter description	Gray: the grad density. 1-high density, 2-medium,3-low Fontheight: font height. 16 or 24 Fontwidth: font width. 16 or 24 Fontzoom: bolt font, 0x00 or 0x33
Return	0: successfully Other value: failure For example: -4001 : PRINT BUSY -4002 : PRINT NOPAPER -4003 : PRINT DATAERR -4004 : PRINT FAULT -4005 : PRINT TOOHEAT -4006 : PRINT UNFINISHED -4007 : PRINT NOFONTLIB -4008 : PRINT BUFFOVERFLOW -4009 : PRINT SETFONTERR -4010 : PRINT GETFONTERR
Function description	Initialize printer function parameter and load font

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();`
	`}`

3.2. PrintStart

Function prototype	public int PrintStart()
Parameter description	none
Return	0： success； -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；
Function description	Start print
Example	ret = posApiHelper.PrintStart();

3.3. PrintCheckStatus

Function prototype	public int PrintCheckStatus()
Parameter description	None
Return	0 –success ； -1 –need paper -2 –high temperature ； -3 –Low battery voltage
Function description	Check printer status
Example	ret = posApiHelper.PrintCheckStatus();

3.4. PrintSetVoltage

Function prototype	public int PrintSetVoltage(int voltage)

Parameter description	voltage： current battery voltage*10
Return	0 –successfully Other -failure
Function description	Set voltage

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

3.5. PrintSetGray

Function prototype	public int PrintSetGray(int nLevel)
Parameter description	nLevel： density level, value 1~5 1:Lowest 3： medium 5： Highest
Return	0 –successfully Other -failure
Function description	Set print density
Example	ret = posApiHelper.PrintSetGray (2);

3.6. PrintSetFont

Function prototype	public int PrintSetFont(byte fontHeight, byte fontWidth, byte zoom)
Parameter description	asciiFontHeight： font dot matrix height, value 16 or 24 extendFontHeight： font dot matrix width, value 16 or 24 Zoom： Font set as bold and bigger, value 0x00 or 0x33
Return	0 –success Other -failure
Function description	Set print font size

Example	posApiHelper.PrintSetFont((byte)16, (byte)16, (byte)0x33);
	posApiHelper.PrintSetFont((byte)24, (byte)24, (byte)0x00);

3.7. PrintSetMode

Function prototype	public int PrintSetMode (int mode)
Parameter description	mode： 0 -> print a receipt (default) 1 -> print a label
Return	0 –successfully Other -failure
Function description	set print mode for receipt or label
Example	ret = posApiHelper. PrintSetMode (1);

3.8. PrintStr

Function prototype	public int PrintStr(String str)
Parameter description	str: print content
Return	0 –successfully -4002 –need paper -4003 –data error
Function description	Set print content
Example	posApiHelper.PrintStr("POS SALES SLIP\n");

3.9. PrintBmp

Function prototype	public int PrintBmp(Bitmap bitmap)
Parameter description	bitmap： BMP photo data
Return	0 –successfully Other -failure Such as: -4003 PRN_DATAERR -4004 PRN_FAULT -4008 PRN_BUFFOVERFLOW

Function description	Set BMP photo print content (size requirement width <=384,height <=500)

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

3.10. PrintBarcode

Function prototype	public int PrintBarcode(String contents, int desiredWidth, int desiredHeight, String barcodeFormat);
Parameter description	contents： barcode content desiredWidth： barcode width desiredHeight： barcode height barcodeFormat： barcode standard CODE_128 , CODE_39, EAN_8, QR_CODE PDF_417 , ITF
Return	0 –successfully Other -failure
Function description	Set barcode print content

Example	posApiHelper.PrintBarcode("12345678", 360, 120, BarcodeFormat.EAN_8);
	posApiHelper.PrintBarcode("12345678", 360, 120, BarcodeFormat.ITF);
	posApiHelper.PrintBarcode("12345678", 360, 240, BarcodeFormat.PDF_417);
	posApiHelper.PrintBarcode("12345678",360,120,”CODE_128”);
	posApiHelper.PrintBarcode("12345678",360,120,”CODE_39”);
	posApiHelper.PrintBarcode("12345678",240,240,”*QR_CODE”*);

3.11. PrintQrCode_Cut

Function prototype	public int PrintQrCode_Cut (String contents, int desiredWidth, int desiredHeight, String barcodeFormat);
Parameter description	Input: Contents:Content of the dr code; desiredWidth:Width;

	desiredHeight:Heigh;
	barcodeFormat:Coding format;
	Output:no;

Return	0 –successfully Other -failure
Function description	Print QR code

Example	String content = "com.chips.ewallet.scheme://{\"PayeeMemberUuid\":\"a3d7fe8e-873d-499b-9f11-000000000000\",\"PayerMemberUuid\":null,\"TotalAmount\":\"900\",\"PayeeSiteUuid\":null,\"PayeeTransId\":\"100101-084850-6444\",\"PayeeSiteReference\":\"\",\"PayeeDescription\":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");

3.12. PrintCutQrCode_Str

Function prototype	public int PrintCutQrCode_Str (String qrContent, String printTxt ,int distance, int desiredWidth,int desiredHeight, String barcodeFormat);

Parameter description	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:no;

Return	0 –successfully Other -failure
Function description	print QR code, also print characters on the side.
Example	String content = "com.chips.ewallet.scheme://{\"PayeeMemberUuid\":\"a3d7fe8e-873d-499b-9f11-000000000000\",\"PayerMemberUuid\":null,\"TotalAmount\":\"900\",\"PayeeSiteUuid\":null,\"PayeeTransId\":\"100101-084850-

	6444\",\"PayeeSiteReference\":\"\",\"PayeeDescription\":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);

3.13. PrintSetUnderline

Function prototype	public int PrintSetUnderline(int x);
Parameter description	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
Return	0 –successfully Other -failure
Function description	Set the lines and width of underline
Example	posApiHelper. PrintSetUnderline (0x1F);

3.14. PrintSetReverse

Function prototype	public int PrintSetReverse (int x);
Parameter description	x: * 0(default) -> normal * 1 -> reverse
Return	0 –successfully Other -failure
Function description	Set the font display reverse mode
Example	posApiHelper. PrintSetReverse (1);

3.15. PrintSetBold

Function prototype	public int PrintSetBold (int x);
Parameter description	mode: * 0(default) -> normal * 1 -> Bold
Return	0 –successfully Other -failure
Function description	Set the font display Bold mode
Example	posApiHelper. PrintSetBold (1);

3.16. PrintLogo

Function prototype	public int PrintLogo (byte[] logo);
Parameter description	byte[] logo: the byte[] for a picture
Return	0 -successfully Other -failure
Function description	print a picture by a byte[]
Example	posApiHelper. PrintLogo (logo);

3.17. PrintLabLocate

Function prototype	public int PrintLabLocate (step);
Parameter description	reserved
Return	0 -successfully Other -failure

Function description	Determine the print position for print a label PS: The printer needs to support label print
Example	posApiHelper. PrintLabLocate (0);

Chapter 4 Generic APIs

4.1. SysUpdate

Function prototype	public int SysUpdate()
Parameter description	None
Return	0 successfully Other failure
Function description	Payment module firmware upgrade
Example	int ret = posApiHelper.SysUpdate ();

4.2. SysGetRand

Function prototype	Public int SysGetRand(byte[] rnd)
Parameter description	byte[] rnd: The random number returned by the MCU
Return	0 successfully Other failure
Function description	To get 8 byte random number

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

4.3. SysGetVersion

Function prototype	public int SysGetVersion (byte[] buf)
Parameter	buf: firmware no.

description	
Return	0 successfully Other failure
Function description	Read firmware version

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

4.4. SysReadChipID

Function prototype	public int SysReadChipID (byte[] buf, int len)
Parameter description	buf: IC card ID no. len: length
Return	0 successfully Other failure
Function description	Get IC card ID no.

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

4.5. SysWriteSN

Function prototype	public int SysWriteSN (byte[] SN)
Parameter description	SN: 16 byte serial no.
Return	0 successfully Other failure
Function description	Write serial no.

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

4.6. SysReadSN

Function prototype	public int SysReadSN (byte[] SN)
Parameter description	SN: 16 byte serial no.
Return	0 successfully Other failure
Function description	Write serial no.

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

Chapter 5 Barcode Scan

5.1. Start Scan

Description:
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

For 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);
```

5.2. Stop Scan

Description:
You can start scan through Send a broadcast “**ACTION_BAR_TRIGSTOP**”.

For example:

```
          intent = new Intent();
```

```
                intent.setAction(ACTION_SCANNER_CANCEL);
                mContext.sendBroadcast(intent);
```

5.3. Get scan results

Description :

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.
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

For 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");
    }
    }
    };
```

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：

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

For 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);
```

For example：

Set scan to API output mode， and Automatically add “Enter” characters after scan

//SCAN_MODE : Fill mode

```
        Intent intent = new Intent ("ACTION_BAR_SCANCFG");
        intent.putExtra("EXTRA_SCAN_MODE", 1);
        intent.putExtra("EXTRA_SCAN_AUTOENT", 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);
```

Chapter 6 App White List& Black List

The white list is used to restrict the APP that can be loaded. Only applications in the white list can be loaded into the system to ensure the security of the system. On the contrary the black list is used to restrict the APP that can not be loaded.

APIs for App White list:

6.1. enableAppInstallWhiteList

Function prototype	public boolean enableAppInstallWhiteList()

Parameter description	
Return	true : success false : fail
Function description	enable the function of App White list
Example	posApiHelper. enableAppInstallWhiteList ();

6.2. disableAppInstallWhiteList

Function prototype	public boolean disableAppInstallWhiteList ()
Parameter description	
Return	true : success false : fail
Function description	disable the function of App White list
Example	posApiHelper. disableAppInstallWhiteList ();

6.3. addAppToInstallWhiteList

Function prototype	public boolean addAppToInstallWhiteList (String pkgName)
Parameter description	pkgName： the APP package name
Return	true : success false : fail
Function description	add an apk to white list
Example	String packageNameList = "com.app.package.name" posApiHelper. addAppToInstallWhiteList (packageNameList);

6.4. delAppFromInstallWhiteList

Function prototype	public boolean delAppFromInstallWhiteList (String pkgName)
Parameter descriptio n	
Return	true : success false : fail
Function descriptio n	delete an apk from white list
Example	String packageNameList = "com.app.package.name" posApiHelper. delAppFromInstallWhiteList (packageNameList);

6.5. getAppInstallWhiteList

Function prototype	public List<String>getAppInstallWhiteList ()
Parameter descriptio n	
Return	The app white list
Function descriptio n	get the APP white list
Example	posApiHelper. getAppUninstallBlackList ();

APIs for App black list:

6.6. enableAppUninstallBlackList

Function prototype	Public boolean enableAppUninstallBlackList ()
Parameter description	
Return	true : success false : fail
Function description	enable the function of App black list
Example	posApiHelper. enableAppUninstallBlackList ();

6.7. disableAppUninstallBlackList

Function prototype	Public boolean disableAppUninstallBlackList ()
Parameter description	
Return	true : success false : fail
Function description	disable the function of App black list
Example	posApiHelper. disableAppUninstallBlackList ();

6.8. addAppToUninstallBlackList

Function prototype	Public boolean addAppToUninstallBlackList (String pkgName)
Parameter	

description	
Return	true : success false : fail
Function description	add an apk to black list
Example	posApiHelper. addAppToUninstallBlackList (pkgName);

6.9. delAppFromUninstallBlackList

Function prototype	Public boolean delAppFromUninstallBlackList(String pkgName)
Parameter description	
Return	true : success false : fail
Function description	delete an apk from black list
Example	posApiHelper. delAppFromUninstallBlackList (pkgName);

6.10. getAppUninstallBlackList

Function prototype	Public List<String>getAppUninstallBlackList ()
Parameter description	
Return	
Function description	get the APP black list
Example	posApiHelper. getAppUninstallBlackList ();

Chapter 7 Android OS API

This APIs is availabel for Ciontek POS.

7.1. installRomPackage

Function prototype	public int installRomPackage(String romFilePath)
Parameter description	context :Context romFilePath : rom file path
Return	0 : success !0 : fail
Function description	API for Android firmware update, useful for client want to deploy its own OTA system.

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

7.2. getOSVersion

Function prototype	public String getOSVersion()
Parameter description	
Return	String: the OS version
Function description	Get OS version
Example	String osVersion = posApiHelper.getOSVersion();

7.3. getDeviceId

Function prototype	public String getDeviceId ()
Parameter description	
Return	String: the device serial number
Function description	Get the device serial number
Example	String osVersion = posApiHelper.getDeviceId ();

Chapter 8 Serial Port module

The serial port at the bottom of the device use for the fiscal module.

8.1. fiscalOpen

Function prototype	public int fiscalOpen(int baudrate, int size, int stop, char parity, char cflow)
Parameter descriptio n	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
Return	0: success -1: fail -2: uninitialized -3: parameter error -4: timeout -5: init uart port error -6: read error -7: write error
Function descriptio n	Power on the fiscal module and open the serial port

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

8.2. fiscalClose

Function prototype	public int fiscalClose ()
Parameter description	
Return	0: success -1: fail -2: uninitialized -3: parameter error -4: timeout -5: init uart port error -6: read error -7: write error
Function description	power off the fiscal and close the uart port
Example	posApiHelper.fiscalClose();

8.3. fiscalWrite

Function prototype	public int fiscalWrite(byte[] data)
Parameter description	data
Return	0: success -1: fail -2: uninitialized -3: parameter error -4: timeout -5: init uart port error -6: read error -7: write error
Function description	Write data to fiscal by the serial port

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);`

8.4. fiscalRead

Function prototype	int fiscalRead(byte[] buffer, int bufLen, int timeout)
Parameter description	Buffer: the buffer for data form serial port bufLen: the length of the buffer timeout: timeout for read, unit: ms
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
Function description	read data from fiscal by the serial port

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

Chapter 9 Magnetic card

9.1. McrOpen

Function prototype	public int McrOpen()
Parameter description	
Return	0 : success !0:fail
Function description	To open the Magnetic card reader
Example	posApiHelper.McrOpen();

9.2. McrClose

Function prototype	public int McrClose()
Parameter description	
Return	0 : success !0:fail
Function description	To close the Magnetic card reader
Example	posApiHelper. McrClose();

9.3. McrReset

Function prototype	public int McrReset()
Parameter description	
Return	0 : success !0:fail

Function description	Magnetic head restoration,clear magnetic relief data
Example	posApiHelper. McrReset();

9.4. McrCheck

Function prototype	public int McrCheck()
Parameter description	
Return	0 : success !0:fail
Function description	Check have Magnetic card swiped
Example	posApiHelper. McrCheck();

9.5. McrRead

Function prototype	public int McrRead(byte keyNo, byte mode, byte[] track1, byte[] track2, byte[] track3)
Parameter description	keyNo: DES key index no, value from 0~4, should be the same AUTHDESK key index no in authentication mode: Magnetic head mode mode -> 0：Unencrypted mode -> 1：encrypted track1: store track 1 data’s pointer [Application layer relief area should set as 256] track2: store track 2 data’s pointer [Application layer relief area should set as 256] track3: store track 3 data’s pointer [Application layer relief area should set as 256]
Return	0 : swipe card error (>0)： bit0 = 1 read track 1 data correctly bit1 = 1 read track 2 data correctly bit2 = 1 read track 3 data correctly bit4 = 1 checkout wrong in track 1 data bit5 = 1 checkout wrong in track 2 data bit6 = 1 checkout wrong in track 3 data

	other value
Function description	Read magnetic card track 1,2,3 data from relief area
Example	ret = posApiHelper.McrRead((byte)0, (byte)0, track1, track2, track3);