

Ciontek

CS30Pro SDK Instruction

V1.0.1

Updated on
2021/03/18

Chapter 1 Overview	4
1.1. Introduction	4
1.2. Modify records	4
1.3. Usage.....	5
1.3.1. Import the SDK for Android studio.....	5
1.3.2. Runtime environment.....	6
Chapter 2 Contact Type IC Card	6
2.1. IccCheck	6
2.2. IccOpen	6
2.3. IccCommand.....	7
2.4. IccClose.....	8
Chapter 3 Print.....	8
Caution:.....	8
3.1. PrintInit	9
3.2. PrintStart	10
3.3. PrintCheckStatus	10
3.4. PrintSetVoltage.....	11
3.5. PrintSetGray	11
3.6. PrintSetFont	11
3.7. PrintSetMode	12
3.8. PrintStr	12
3.9. PrintBmp	12
3.10. PrintBarcode.....	13
3.11. PrintQrCode_Cut	14
3.12. PrintCutQrCode_Str	14
Chapter 4 Generic APIs	15
4.1. SysUpdate.....	15
4.2. SysGetRand	15
4.3. SysGetVersion.....	16
4.4. SysReadChipID.....	16
4.5. SysWriteSN	16
4.6. SysReadSN	17
Chapter 5 Barcode Scan	17
5.1. Start Scan	17
5.2. Stop Scan.....	18
5.3. Get scan results	18
5.4. Scan settings.....	19
Chapter 6 App White List	20
6.1. AppInstallEnableAll	21
6.2. AppInstallDisableAll.....	21
6.3. AppInstallConfig	21
6.4. AppInstallReadConfig	22
Chapter 7 Android OS API	22
7.1. installRomPackage.....	23

7.2. getOSVersion	23
7.3. getDeviceId.....	23
Chapter 8 Serial Port module.....	24
8.1. fiscalOpen.....	24
8.2. fiscalClose.....	24
8.3. fiscalWrite	25
8.4. fiscalRead	26

Chapter 1 Overview

1.1. Introduction

This document is the instruction of all API defined by Ciontek for developer to program 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 be given.

The EMV kernel related API is introduced in another instruction file accordingly. Please contact your sales contact for updated version before your integration efforts.

Support : CS30Pro Android10.0

1.2. Modify records

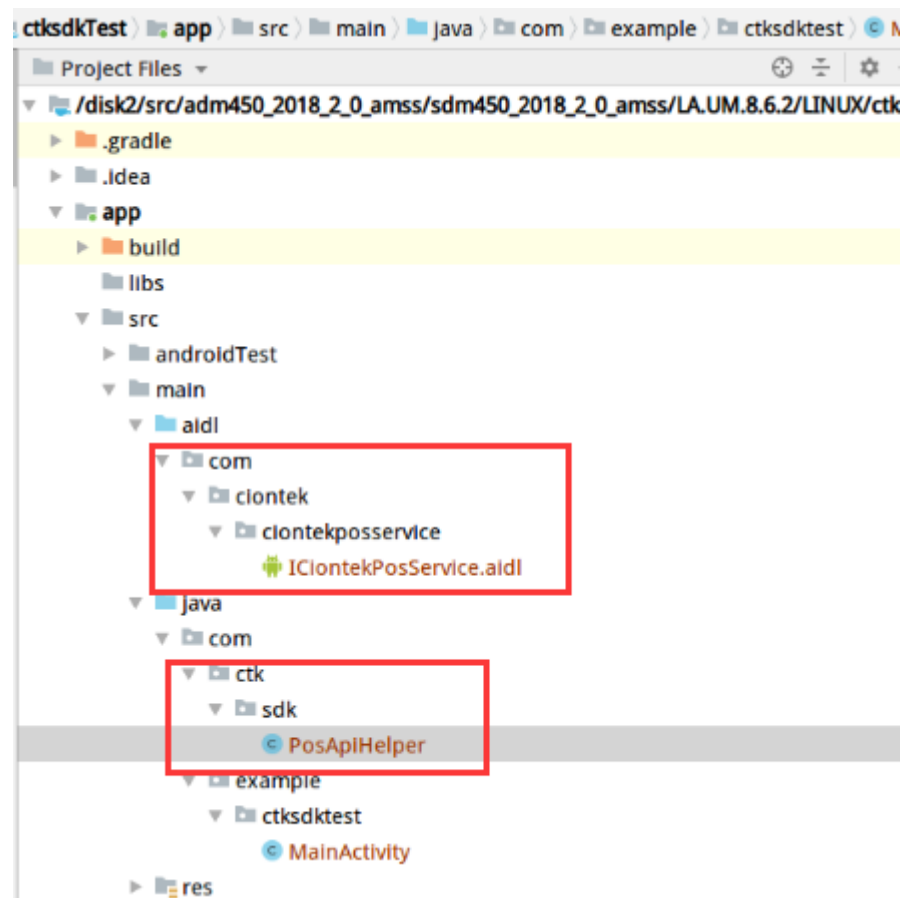
version	author	date	remarks
V1.0.1	Tao	2021-03-11	

1.3. Usage

1.3.1. Import the SDK for Android studio

Unzip the sdk “ciontek - cs30Pro - 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”



Class PosApiHelper describe APIs for Ciontek CS30Pro, 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 CS30Pro, Please make sure the android build number is a51_v0.08_20210324c or after.

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	<pre>byte ATR[] = new byte[41]; ret = posApiHelper.IccOpen(1, 1, ATR);</pre>

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	<pre>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();</pre>

	<pre> 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); </pre>
--	--

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

Caution:

1, Please don't interrupt the printing process. Since the printer module will maintain communication with the host CPU during this process, it is recommended to block the Return, Power, Home buttons and other events that can suddenly interrupt printing, such as a mask pop-up.

For example: to block the Power button: disableFunctionLaunch(true);

To activate the Power button: disableFunctionLaunch(false)

For more examples please refer to the PrintActivity.Java in SDK Demo

source code.

2, for long sheet print such as 80k byte[], it will take around 5 seconds to start.

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	<pre>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);</pre>

	<pre> posApiHelper.PrintStr("CODE_128 : " + "123456789" + "\n\n"); posApiHelper.PrintStr("QR_CODE : " + "123456789" + "\n\n"); posApiHelper.PrintStr(" \n"); posApiHelper.PrintStart(); } </pre>
--	--

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	<code>ret = posApiHelper.PrintCheckStatus();</code>
----------------	---

3.4. PrintSetVoltage

Function prototype	<code>public int PrintSetVoltage(int voltage)</code>
Parameter description	voltage: current battery voltage*10
Return	0 –successfully Other -failure
Function description	Set voltage
Example	<code>//Set current voltage as 7.5V ret = posApiHelper.PrintSetVoltage(75);</code>

3.5. PrintSetGray

Function prototype	<code>public int PrintSetGray(int nLevel)</code>
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	<code>ret = posApiHelper.PrintSetGray (2);</code>

3.6. PrintSetFont

Function prototype	<code>public int PrintSetFont(byte fontHeight, byte fontWidth, byte zoom)</code>
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	bitmap:

description	BMP photo data
Return	0 –successfully Other -failure Such as: -4003 PRN_DATAERR -4004 PRN_FAULT -4008 PRN_BUFFEROVERFLOW
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://{\"PayeeMemberUid\": \"a3d7fe8e-873d-499b-9f11-000000000000\", \"PayerMemberUid\": 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	<pre>String content = "com.chips.ewallet.scheme://{\"PayeeMemberUid\": \"a3d7fe8e-873d- 499b-9f11- 000000000000\", \"PayerMemberUid\": null, \"TotalAmount\": \"900\", \"Pa yeeSiteUuid\": null, \"PayeeTransId\": \"100101-084850- 6444\", \"PayeeSiteReference\": \"\", \"PayeeDescription\": null, \"Confirmati onUuid\": 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);</pre>

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 description	buf: firmware no.
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 broadcast receiver:

```
mFilter= new IntentFilter("ACTION_BAR_SCAN");
mContext.registerReceiver(mReceiver, mFilter);
```

unregister broadcast 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

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.

6.1. AppInstallEnableAll

Function prototype	public int AppInstallEnableAll ()
Parameter description	
Return	true : success false : fail
Function description	allow all app to be install
Example	posApiHelper.AppInstallEnableAll ();

6.2. AppInstallDisableAll

Function prototype	public int AppInstallDisableAll ()
Parameter description	
Return	true : success false : fail
Function description	Restrict all app to install
Example	posApiHelper.AppInstallDisableAll ();

6.3. AppInstallConfig

Function prototype	public int AppInstallConfig(String packageNameList)
Parameter description	packageNameList: the package name list, separate by "\r\n"
Return	true : success

	false : fail
Function description	allow apps that specified to install
Example	<pre>String packageNameList = "com.app.package.name1\r\n" + "com.app.package.name2\r\n" + "com.app.package.name3\r\n" posApiHelper.AppInstallConfig (packageNameList);</pre>

6.4. AppInstallReadConfig

Function prototype	public List<String> AppInstallReadConfig()
Parameter description	
Return	Return a package name list that can be install "*" : All APP can be install "" : All APP will be restrict "package name " : specify package name can be install
Function description	read App white list
Example	<pre>List<String> packageList = posApiHelper.AppInstallReadConfig(); for (int i = 0; i < packageList.size(); i++) { Log.d(TAG,"App install config package " +i + " :"+ packageList.get(i)); }</pre>

Chapter 7 Android OS API

This API is available for Ciontek CS30Pro.

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

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 description	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 description	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	<code>posApiHelper.fiscalClose();</code>

8.3. fiscalWrite

Function prototype	<code>public int fiscalWrite(byte[] data)</code>
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	<pre> 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); </pre>

8.4. fiscalRead

Function prototype	<code>int fiscalRead(byte[] buffer, int bufLen, int timeout)</code>
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	<pre>byte[] buffer = new byte[36]; readCount = posApiHelper.fiscalRead(buffer,36,500);</pre>