Android SDK Instruction

Version	Date	Note
V1.001	2023-2-23	First version
V1.002	2023-3-2	Revised Unified Interface
V1.003	2023-6-9	1 POSPrinter add void printNormal(int iStation, byte[] bData) 2 add void printBitmap(int station,Bitmap bmp, int width, int alignment)
V1.0031	2023-7-17	1 modify printPDF

Content

Brief introduction

This SDK realizes the operation of printer, magnetic stripe card, cash drawer, LCD according to the methods specified in UnifiedPOS V1.141, and the printing class POSPrinter in the SDK corresponds to CHAPTER 30 POS Printer; The magnetic card class MSR in the SDK corresponds to CHAPTER 25 MSR – Magnetic Stripe Reader; The LCD class LineDisplay in the SDK corresponds to CHAPTER 22 Line Display; The cash drawer class CashDrawer in the SDK corresponds to CHAPTER 8 Cash Drawer; The serial port class is Serialport. For details about the functions implemented by each device, see SDK Interface Instruction.

一、 Interface File

possdk.aar store files in the libs folder under Android project
minSdkVersion 21 targetSdkVersion 28
dependencies add implementation fileTree(include: ['*.aar'], dir: 'libs')

二、 SDK Interface Instruction

1. Printer

import upos.POSException;
import upos.POSPrinter;
import upos.POSPrinterConst;
import upos.events.StatusUpdateEvent;
import upos.events.StatusUpdateListener;

1) Constructor

POSPrinter ()

2) Printer status value callback

void addStatusUpdateListener (StatusUpdateListener l)
void statusUpdateOccurred(StatusUpdateEvent e);
void removeStatusUpdateListener(StatusUpdateListener l)

3) Open printer

void open(String logicalDeviceName) throws POSException
Parameter logicalDeviceName PTR_CP_2INCH :2inch ; PTR_CP_3INCH :3inch
It will throw POSException(POS_FAILED) when open failed.

4) Close printer

void close()

5) Cutter

void cutPaper(int percentage)
Parameter percentage PTR_CP_FULLCUT: full cutting;

PTR_CP_PARTIALCUT: half cutting

6) Print bitmap

void printBitmap(int station, String filePath, int width, int alignment)
Parameter station ignore
filePath path of bitmap
width Print width of bitmap
alignment alignment mode PTR_BC_LEFT Left
PTR_BC_CENTER Center
PTR_BC_RIGHT Right

void printMemoryBitmap(int station, byte[] imagedata, int type, int width, int alignment)
Parameter station ignore
imagedata bitmap's data
type PTR_BMT_BMP PTR_BMT_JPEG
width bitmap's width
alignment alignment mode PTR_BC_LEFT Left
PTR_BC_CENTER Center
PTR_BC_RIGHT Right

void printBitmap(int station,Bitmap bmp, int width, int alignment)
Parameter station ignore
bmp bitmap
width bmp's width
alignment alignment mode PTR_BC_LEFT Left
PTR_BC_CENTER Center
PTR_BC_RIGHT Right

7) Send ESC data

void printNormal(int iStation, String strData)
Parameter: iStation ignore
strData ESC data

void printNormal(int iStation, byte[] bData)
Parameter: iStation ignore
bData ESC data bytes

8) Print barcode

void printBarCode(int station, String strData, int symbology, int height, int width, int alignment, int textPosition)

Parameter station ignore
strData barcode data
symbology barcode type PTR_BCS_UPCA, PTR_BCS_UPCE…PTR_BCS_***
height barcode height 1<=height<=255
width barcode width 2<=width<=6
alignment alignment mode PTR_BC_LEFT PTR_BC_CENTER PTR_BC_RIGHT
textPosition text position PTR_BC_TEXT_NONE: not printing
PTR_BC_TEXT_ABOVE: top of barcode
PTR_BC_TEXT_BELOW: bottom of barcode
PTR_BC_TEXT_BOTH: print both top and bottom

2. Serial port

import upos.POSException;
import upos.Serialport;
import upos.events.DataEvent;
import upos.events.DataReceiveListener;

1) Constructor

Serialport ()

2) Serial port data callback

void addDataReceiveListener(DataReceiveListener l)
void dataReceiveOccurred(DataEvent e)
void remeveDataReceiveListener(DataReceiveListener l)

3) Open serial port

void open(String logicalDeviceName) throws POSException
Parameter: logicalDeviceName serial port path, Baudrate eg. “/dev/ttyS1,9600”
It will throw POSException(POS_FAILED) when open failed.

4) Close serial port

```
void close()
```

5) Send data

```
void writeData(byte[] bData,int iLen)
```

Parameter: bData   data to send
iLen   data length

6) Start and stop reading serial port data

```
void startReadData(boolean bStart)
```

Parameter: bStart true start to read serial port thread; false stop reading data thread
Data read will be called back by listener

3. Magnetic Stripe Reader

```
import upos.POSException;
import upos.MSR;
import upos.events.DataEvent;
import upos.events.DataReceiveListener;
```

1) Constructor

```
MSR()
```

2) Magcard data callback

```
void addDataReceiveListener(DataReceiveListener l)
void dataReceiveOccurred(DataEvent e);
void remeveDataReceiveListener(DataReceiveListener l)
```

3) Open magcard device

```
void open(String logicalDeviceName) throws POSException
```

Parameter logicalDeviceName ignore
It will throw POSException(POS_FAILED) when open failed.

4) Close magcard device

void close()

5) Start to read magcard data

void authenticateDevice(byte[] buffer)
Parameter: buffer ignore
Magcad data read will be called back by dataReceiveOccurred

6) Stop reading magcard data thread

void deauthenticateDevice(byte[] buffer)
Parameter: buffer ignore

4. LCD

import upos.POSException;
import upos.LineDispaly;
import upos.LineDisplayConst ;

1) Constructor

LineDispaly()

2) Open LCD

void open(String logicalDeviceName) throws POSException

Parameter logicalDeviceName:	LINEDISPLAY_TYPE_CD6	9VFD;
	LINEDISPLAY_TYPE_CD9	32X144;
	LINEDISPLAY_TYPE_CD10	65X132;
	LINEDISPLAY_TYPE_CD8	24X128;
	LINEDISPLAY_TYPE_CD11	32X240;
	LINEDISPLAY_TYPE_COG	65X132COG

It will throw POSException(POS_FAILED) when open failed.

3) Close LCD

void close()

4) Set LCD display parameter

void setDescriptor(int iDescriptor,int iValue)
Parameter: iDescriptor
LINEDISPLAY_DESCRIPTOR_BACKLIGHT open and close backlight
iValue 0 close, 1 open
LINEDISPLAY_DESCRIPTOR_CONTRAST set contrast iValue 0-9

5) Clear LCD Screen

void clearText ()

6) Display string

void displayText(String strText,int iFontSize)
Parameter: strText character data
iFontSize font size

7) Display bitmap

void displayBitmap(String path,int iWidth,int iAlignX,int iAlignY)
Parameter path bitmap path; ignore other parameter

5. Drawer

import upos.POSException;
import upos.CashDrawer;

1) Constructor

CashDrawer ()

2) Open drawer device

void open(String logicalDeviceName) throws POSException
Parameter logicalDeviceName ignore

It will throw POSException(POS_FAILED) when open failed.

3) Close drawer device

void close()

4) Open drawer

void openDrawer ()