

Mobile POS Printer SDK

1. Print by AIDL	2
2. Print by BlueTooth	8
3. Status Feedback	11
4. Reference.....	12

1 Print by AIDL

It's the way of AIDL to connect the printer. Android Interface Definition language (AIDL) is an interface for Android internal process communication, through which we can define the interface for communication between processes. AIDL provides packaged commonly used printing commands to facilitate rapid access to printer services by developers.

1.1 The use of AIDL

1.To add the AIDL into the developer's programming project:

1)SDK: com.iposprinter.iposprinterservice

2)File: There are two files in the aidl.rar: IPosPrinterService.aidl & IPosPrinterCallback.aidl

2. The command for connecting to the printer:
ServiceConnection

```
private ServiceConnection connectService = new
    ServiceConnection() {
        @Override
        public void onServiceConnected(ComponentName name, IBinder
            service) {
            mIPosPrinterService =
                IPosPrinterService.Stub.asInterface(service);
            setButtonEnable(true);
        }

        @Override
        public void onServiceDisconnected(ComponentName name) {
            mIPosPrinterService = null;
        }
    };
3. Invoke ApplicationContext.bindService(), and transmit in
    ServiceConnection
```

Note: Bindservice is a non blocking call, it means that the call is not immediately bound after completion, Standard must be serviceConnected

```
//Binding Service
Intent intent=new Intent();
intent.setPackage("com.iposprinter.iposprinterservice ");
intent.setAction("com.iposprinter.iposprinterservice.IPosPrintService");
bindService(intent, connectService, Context.BIND_AUTO_CREATE);
```

4. Once you bind the service, you can call various interfaces from the IPosPrintService for printing.

1.2 AIDL Reference:

1. Printer Initialization

Function	printerInit
Description	The printer power on and initializes the default settings
Return Value	void
Note	Before use, please determine the current status of the printer. If you are in the PRINTER_IS_BUSY state, please wait a second

2. Printer Status Checking

Function	getPrinterStatus
Description	Get the current Printer Status
Return Value	<p>Printer Status:</p> <p>0:PRINTER_NORMAL The printer is in free and normal. You can start a new print</p> <p>1:PRINTER_PAPERLESS Out of paper. If the current printing is not complete, please re-print it again after the new paper roll input.</p> <p>2:PRINTER_THP_HIGH_TEMPERATURE Printing head overheat, the current printing stop and will be continue after cooling.</p> <p>3:PRINTER_MOTOR_HIGH_TEMPERATURE Printer motor overheat, current printing stop. You need to initialize the printer after cooling.</p> <p>4:PRINTER_IS_BUSY Printer is printing something now.</p> <p>5:PRINTE_ERROR_UNKNOWN Unknown error occurred.</p>
Note	Please check the status before printing

3. Darkness Setting

Function	setPrinterPrintDepth
Description	Setting the Darkness of the printer
	depth: The level of darkness, between 1 – 10, standard value 6.
Return Value	void
Note	

4. Font Setting

Function	setPrinterPrintFontType
Description	Set font type of the printer
	typeface: Name of Character
Return Value	void
Note	

5. Font Size Setting

Function	setPrinterPrintFontSize
Description	Set font size of the printer
	fontsize: Supports 16、24、32、48 Pixel, Standard value is 24
Return Value	void
Note	

6. Printer Align Setting

Function	setPrinterPrintAlignment
Description	Printer Align Setting
	alignment: 0-- Align left, 1-- Center, 2-- Align Right
Return Value	void
Note	

7. Feed Paper

Function	printerFeedLines
Description	Paper Feed
	lines: The number of feed line
Return Value	void
Note	

8. Print Blank Lines

Function	printBlankLines
Description	Print blank lines After the previous printing, newlines and print blank lines. (now the data sent to printer is 0x00)
	lines: Number of blank lines (Max. 100) height: The height of the blank lines (Unit: Pixel)
Return Value	void
Note	Suggest to use this command to have blank between lines, do not recommend use printerFeedLines

9. Print Characters

Function	printText
Description	If the text width is full, printer will auto linefeed and continue printing
	text: The characters that need to be printed
Return Value	void
Note	

10. Print the text that specifies the type and size of the font

Function	printSpecifiedTypeText
Description	If the text width is full, printer will auto linefeed and continue printing
	text: The Characters that need to be printed typeface: Font name fontsize: Font size. Supports 16, 24, 32, 48
Return Value	void
Note	

11. Print sheet text

Function	printColumnsText
Description	To print a row of a sheet, you can specify column widths and column alignment
	colsTextArr: Array of text strings colsWidthArr: Column width Total width cannot be larger than $((384 / \text{fontsize}) \ll 1) - (\text{column} + 1)$ colsAlign: Column alignment (0=left, 1=center, 2=right) isContinuousPrint: Continue printing form? 1: Continue printing

	0: Do not continue
Return Value	void
Note	The array length of the three parameters should be consistent. If the content width of colsTextArr[i] is larger than colsWidthArr[i], the text will linefeed

12. Print Image

Function	printBitmap
Description	Print bmp file alignment: 0=left, 1=center, 2=right bitmapSize: Bitmap size between 1 -- 8, out of range it would be return to defaults, 6 Unit: 48bit mBitmap: Bitmap Object (max. width 384 pixels)
Return Value	void
Note	

13. Print Barcode

Function	printBarCode
Description	data: Content of the barcode symbology: Barcode type 0 -- UPC-A 1 -- UPC-E 2 -- JAN13(EAN13) 3 -- JAN8(EAN8) 4 -- CODE39 5 -- ITF 6 -- CODABAR 7 -- CODE93 8 -- CODE128 height: Height of the barcode, from 1 -- 8, Defaults 3, Each unit high 48 pixels width: Width of the barcode, from 1 -- 8, Defaults 6, Each unit wide 48 pixels textposition: Position of the Characters 0- Don't print characters 1- Characters at the above of barcode 2- Characters at the bottom of barcode 3- Characters print at the above & bottom of the barcode
Return Value	Void
Note	

14. Print 2D Code

Function	printQRCode
Description	Data: Date of the QR Code Modulesize: Size of the QR Code (unit: Pixel, from 1 – 8) , defaults 6
Return Value	void
Note	

15. Print Byte Data

Function	printRawData
Description	Print raw byte data rawPrintData: Byte data
Return Value	void
Note	

16. ESC/POS

Function	sendCMDRawData
Description	To control the printer by ESC/POS command list data: ESC/POS Command
Return Value	void
Note	

17. Implement Setting of the Printer

Function	printerPerformPrint
Description	To active all setting like font size, align, darkness....
Return Value	Void
Note	After completed the printing setting like font size, align..., you need to run this command to implement the printer. But before the execution, please check the printer status, if the printer is in PRINTER_NORMAL, this command works effective, otherwise this command do not run.

2. Print by Bluetooth

2.1 Open the Bluetooth Printer

When switch on the Bluetooth, you will find that there is a Matched Bluetooth Device “IposPrinter” appeared. It’s a virtual Bluetooth printer created by Systems and connect it to the printer that built-in the machine. This Bluetooth printer supports ESC/POS command.

2.2 How to work with the Bluetooth Printer

1. Switch on the Bluetooth and connect with the virtual Bluetooth Printer
2. Send the ESC/POS command and contents to the “IposPrinter” through Bluetooth.
3. Contents print out and mission accomplished.

P.S: There is a class “BluetoothUtil” in the Demo application. Before you use this demo to do testing with the printer, please press “Load in Driver for Bluetooth printer” first.

BluetoothUtil, the standard Bluetooth connection application:

```
public class BluetoothUtil{
    private static final String TAG = "BluetoothUtil";
    private static final UUID IPOSPRINTER_UUID =
    UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");
    private static final String IPosPrinter_Address = "00:AA:11:BB:22:CC";
    public static BluetoothAdapter getBluetoothAdapter(){
        return BluetoothAdapter.getDefaultAdapter();
    }
    public static BluetoothDevice getIposPrinterDevice(BluetoothAdapter
    mBluetoothAdapter){
        BluetoothDevice IPosPrinter_device = null;
        Set<BluetoothDevice> devices =
        mBluetoothAdapter.getBondedDevices();
        for (BluetoothDevice device : devices){
            if(device.getAddress().equals(IPosPrinter_Address))
            {
                IPosPrinter_device =device;
                break;
            }
        } return IPosPrinter_device;
    }
    public static BluetoothSocket getSocket(BluetoothDevice mDevice)
    throws IOException
    {
        BluetoothSocket socket =
```


<pre> mDevice.createRfcommSocketToServiceRecord(IPOSPRINTER_UUID); socket.connect(); return socket; } } </pre>
<p>Get the Bluetooth printer and connect to it:</p>
<pre> // 1: Get BluetoothAdapter mBluetoothAdapter = BluetoothUtil.getBluetoothAdapter(); if(mBluetoothAdapter == null) { return; } //2: Get bluetoothPrinter Devices mBluetoothPrinterDevice = BluetoothUtil.getIposPrinterDevice(mBluetoothAdapter); if(mBluetoothPrinterDevice == null) { return; } //3: Get conect Socket try { socket = BluetoothUtil.getSocket(mBluetoothPrinterDevice); } catch (IOException e) { e.printStackTrace(); return; } </pre>
<p>Note: You need to add the station of Bluetooth permission in the app before you use the printer.</p>
<pre> <manifest> <uses-permission android:name="android.permission.BLUETOOTH"></uses-permission> <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"></uses-permissio n> </manifest> </pre>

3. Status Feedback

3.1 Get the status feedback from the Printer

The user needs to set up a broadcast receiver to monitor the printer's current status

```

//Printer in normal & free
private final String PRINTER_NORMAL_ACTION =
"com.iposprinter.iposprinterservice.NORMAL_ACTION";

//Out of paper
private final String PRINTER_PAPERLESS_ACTION =
" com.iposprinter.iposprinterservice.PAPERLESS_ACTION";

//Paper on the printer
private final String PRINTER_PAPEREXISTS_ACTION =

```

```
"com.iposprinter.iposprinterservice.PAPEREXISTS_ACTION";
```

```
//Printing head overheat
```

```
private final String PRINTER_THP_HIGHTEMP_ACTION =  
"com.iposprinter.iposprinterservice.THP_HIGHTEMP_ACTION";
```

```
//Printing head is normal
```

```
private final String PRINTER_THP_NORMALTEMP_ACTION =  
"com.iposprinter.iposprinterservice.THP_NORMALTEMP_ACTION";
```

```
//Printing motor overheat
```

```
private final String PRINTER_MOTOR_HIGHTEMP_ACTION =  
"com.iposprinter.iposprinterservice.MOTOR_HIGHTEMP_ACTION";
```

```
//Printer is busy (Printing)
```

```
private final String PRINTER_BUSY_ACTION =  
"com.iposprinter.iposprinterservice.BUSY_ACTION";
```

P.S: To get the status forwardly by user, please use command
"getPrinterStatus" and refer to 1.2 AIDL Reference

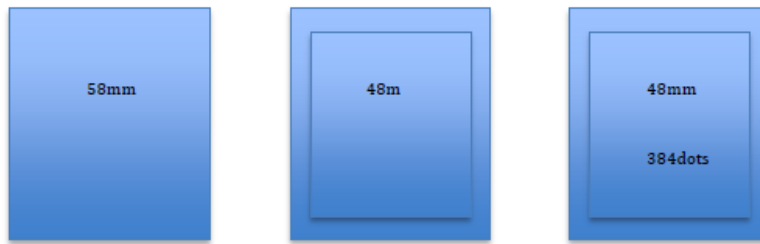
3.2 Feedback of instruction callback

There is two feedback of the instruction callback:

Function	Feedback	Result
onRunResult	boolean is Success	True False
onReturnString	final String result	True False

4. Print Parameter Reference

4.1 Paper



The printer supports paper wide 58mm, with a valid printing width of 48mm, 384 pixels per line.

4.2 Printer Resolution

Printer resolution is 205DPI

$\text{DPI} = 384\text{dots} / 48\text{mm} = 8\text{dots} / 1\text{mm} = 205\text{dots} / \text{in} = 205$

4.3 Font

Defaults Font size is 24, 24 * 24 for Chinese and 12*24 for English.

4.4 2D(QR) Code

To print the QR Code, each CODE should larger than 48 Pixels. (Less than 48 Pixels cannot be analyzed)

4.5 Print Image

The Max width of the imange should be 384 Pixels, if the width of the image larger than 384 pixels, please try to reduce.