

AndroidPrinterSDK Development Document

V3.2

一、 Introduction

This development kit includes Bluetooth, USB and WiFi.

1. Package name: com.android.print.sdk
2. Class library name:

Class Name	Discription
Barcode	Barcode printing class
Table	Form printing class
PrinterConstants	Related constants
PrinterInstance	Printer class (related to text attributes, such as font size, bold, underline, etc.)

PrinterInstance The methods provided by the class are as follows:

1. Constructor PrinterInstance:

- a) // Bluetooth interface printer constructor

```
PrinterInstance(Context context, BluetoothDevice bluetoothDevice, Handler handler)
```

- b) // USB interface printer constructor

```
PrinterInstance(Context context, UsbDevice usbDevice, Handler handler)
```

- c) // WiFi interface printer constructor

```
PrinterInstance(String ipAddress, int portNumber, Handler handler)
```

handler Used to receive the connection status of the printer, the

corresponding status is: PrinterConstants.Connect.SUCCESS;

PrinterConstants.Connect.FAILED;

PrinterConstants.Connect.CLOSED;

2. Open and close the connection:

- a) `openConnection()`
- b) `closeConnection()`

3. Common method:

- a) Initialize the printer

```
init();
```

- b) Print normal text

```
printText(String content);
```

- c) Send Byte data

```
sendByteData(byte[] content);
```

If the instructions in the development manual are not provided in the SDK, the developer can use this method to send instructions directly to the printer. For example, the command to set the print position to be centered is:

```
byte[] command = new byte[3];
```

```
command[0] = 0x1B;
```

```
command[1] = 0x31;
```

```
command[2] = 49;
```

```
sendByteData (command);
```

- d) Print picture

bitmap is the picture; *left* is the left margin of the picture;

```
printImage(Bitmap bitmap);
```

```
printImage(String bitmap, int left);
```

- e) Print the form

```
printTable(Table table);
```

- f) Print barcode

```
PrintBarCode(Barcode barcode);
```

- g) Cutter

```
cutPaper()
```

- h) Buzzer

ringBuzzer(byte number, byte time)

- i) Open the cash box

openCashbox(boolean cashbox1, boolean cashbox2)

- 4. The setting method is as follows:

- a) Set the character encoding format of the printed text, the default is "gbk"

setEncoding(String encoding);

- b) Set character magnification, parameter x is width and y is height. The value of x, y is [0,2], 0 is the default width and height

setCharacterMultiple(int x, int y);

- c) Set the left margin of the printing area, the value is (nL + nH*256) * horizontal unit.

Generally only pass nL, nH pass 0. setLeftMargin(int nL, int nH);

- d) Set print mode

setPrintModel(boolean smallFont, boolean isBold, boolean isDoubleHeight, boolean isDoubleWidth, boolean isUnderLine);

smallFont: smallFont

isBold: Whether it is bold

isDoubleHeight: Is it double high

isDoubleWidth: Is it double width

isUnderLine: Whether to underline

- e) Set up the printer (command constants start with

PrinterConstants.Command.)

setPrinter(int command);

command: Command for printer:

INIT_PRINTER: Initialize the printer (a separate method init() has been proposed)

WAKE_PRINTER: Wake up the printer

PRINT_AND_RETURN_STANDARD: Print in page mode and return to standard mode

PRINT_AND_NEWLINE: Print and wrap

PRINT_AND_ENTER: Print and wrap

MOVE_NEXT_TAB_POSITION: Move the printing position to the next horizontal tab position

DEF_LINE_SPACING: Restore the default row height

setPrinter(int command, int value);

value Is the value corresponding to the command:

PRINT_AND_WAKE_PAPER_BY_LNCH: Print and feed paper value height (inch)

PRINT_AND_WAKE_PAPER_BY_LINE: Print and feed the value line

CLOCKWISE_ROTATE_90: Rotate 90 degrees clockwise 0-false, 1-true

SELECT_OR_CANCEL_THE_UNDERLINE_MODE: Select/cancel underline mode

SET_LINE_SPACE: Set line spacing

SET_RIGHT_SPACE: Set the right spacing of characters

ALIGN: Alignment mode. The 3 alignment variables are as follows:

*ALIGN_LEFT*Align left; *ALIGN_CENTER*Centered; *ALIGN_RIGHT*Align right

二、 Table Class

1. Constructor

Table(String column, String regular, int[] columnWidth);

1.The parameter *column* is the header of the table separated by the parameter *regular*.

Like "serial number, unit price, quantity, amount"

2.The parameter *regular* is the separator of the string in the table. As the above is ",".

3.The parameter *columnWidth* is the character width of each column of the table. The calculation method of the default font size is Chinese 2

One, one in English, and then add up. For example, the width of "Serial Number" is 4, and the default is 8.

2. Add a row of data

addRow(String row);

Add a row of data. The data format is consistent with the header format. If the data in a cell exceeds the limited character width, it will automatically wrap and print. If you need to manually wrap, you can add "\n" where you need to wrap.

3. Set the alignment of the column data in the Table, the default is right-aligned

setColumnAlignLeft(boolean left);

三、 Barcode Class

1. Constructor

`Barcode(byte barcodeType);`

`Barcode(byte barcodeType, int param1, int param2, int param3);`

`Barcode(byte barcodeType, int param1, int param2, int param3, String content);`

- i. The parameter *barcodeType* is the barcode type:

Type constant start with `PrinterConstants.BarcodeType.:`

One-dimensional barcode: `UPC_A`, `UPC_E`, `JAN13`, `JAN8`,
`CODE39`, `ITF`, `CODABAR`, `CODE93`, `CODE128` .

- ii. Parameters *param1*, *param2*, *param3*

Barcodeparameter: When bar code type type is a
one-dimensional bar code, three parameters indicate:

param1: Barcode horizontal width $2 \leq n \leq 6$, the default is 2.

param2: Barcode height $1 \leq n \leq 255$, default 162

param3: Bar code comment position, 0 is not printed, 1 is above, 2 is
below, and 3 is both above and below