

Android-SDK 开发文档

V3.0

一、简介

此开发包包括蓝牙，USB 和 WiFi。

1. 软件包名称: com.android.print.sdk
2. 类库名称:

Class Name	Discription
Barcode	条码打印类
Table	表格打印类
PrinterConstants	相关常量
PrinterInstance	打印机类
CanvasPrint	图形方式打印
FontProperty	图形方式打印中的字体属性设置

PrinterInstance 类提供的方法如下:

1. 构造函数 **PrinterInstance**:

- a) //蓝牙接口打印机构造函数

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

- b) //USB 接口打印机构造函数

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

- c) //WiFi 接口打印机构造函数

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

handler 用于接收打印机的连接状态，对应的状态为:

```
PrinterConstants.Connect.SUCCESS;
```

```
PrinterConstants.Connect.FAILED;
```

```
PrinterConstants.Connect.CLOSED;
```

2. 打开与关闭连接:

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

3. 常用方法:

- a) 初始化打印机

```
init();
```

- b) 打印普通文本

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

- c) 发送 Byte 数据

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

若 SDK 中未提供开发手册中的指令，开发者可以使用此方法直接给打印机发送指令。如发送设置打印位置居中的指令为：

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

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

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

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

```
sendByteData (command);
```

- d) 打印图片

bitmap 为图片； *left* 为图片左边距； *multiple* 为针打图形放大倍数

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

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

```
printImage(Bitmap bitmap, int multiple);
```

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

- e) 打印表格

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

- f) 打印条码

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

- g) 切刀

```
cutPaper()
```

- h) 响蜂鸣器

`ringBuzzer(byte time)`

- i) 开钱箱

`openCashbox(boolean cashbox1, boolean cashbox2)`

4. 设置方法，所：

- a) 设置打印文本的字符编码格式，默认为“gbk”

`setEncoding(String encoding);`

- b) 设置字符放大倍数，参数 x 为宽，y 为高。x, y 取值为[0,7]，0 为默认宽高

`setCharacterMultiple(int x, int y);`

- c) 设置打印区域左边距，值为 $(nL + nH * 256) * \text{横向单位}$ 。一般只传 nL, nH 传 0。

`setLeftMargin(int nL, int nH);`

- d) 设置打印模式

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

isBold: 是否粗体

isDoubleHeight: 是否倍高

isDoubleWidth: 是否倍宽

isUnderLine: 是否下划线

- e) 设置打印机(命令常量以 `PrinterConstants.Command` 开头)

`setPrinter(int command);`

command 为打印机命令：

`INIT_PRINTER`: 初始化打印机(已提出单独方法 `init()`)

`WAKE_PRINTER`: 唤醒打印机

`PRINT_AND_RETURN_STANDARD`: 页模式下打印并返回标准模式

`PRINT_AND_NEWLINE`: 打印并换行

`PRINT_AND_ENTER`: 打印并回车

`MOVE_NEXT_TAB_POSITION`: 移动打印位置到下一个水平制表符位置

`DEF_LINE_SPACING`: 恢复默认行高

`setPrinter(int command, int value);`

value 为命令对应的值：

`PRINT_AND_WAKE_PAPER_BY_LNCH`: 打印并进纸 *value* 高度(英寸)

PRINT_AND_WAKE_PAPER_BY_LINE: 打印并走纸 value 行

CLOCKWISE_ROTATE_90: 顺时针旋转 90 度 0-false, 1-true

LINE_HEIGHT: 设置行高

CHARACTER_RIGHT_MARGIN: 字符右间距

ALIGN: 对齐模式。3 种对齐方式变量如下:

ALIGN_LEFT 左对齐; *ALIGN_CENTER* 居中; *ALIGN_RIGHT* 右对齐

二、 Table 类

1. 构造函数

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

参数 *column* 为以参数 *regular* 分隔的表头。形如“序号, 单价, 数量, 金额”

参数 *regular* 为表内字符串的分隔符。如上面的是“, ”。

参数 *columnWidth* 是表格每一列的字符宽度。默认字体大小的计算方法是中文 2 个, 英文 1 个, 然后相加, 如“序号”的宽度为 4, 默认为 8。

2. 添加一行数据

addRow(String row);

添加一行数据。数据格式与表头格式一致。若某一单元格的数据超出限定的字符宽度, 会自动换行打印, 若需要手动换行, 可在需要换行处加“\n”。

3. 设置 Table 中列数据的对其方式, 默认右对齐

setColumnAlignLeft(boolean left);

三、 Barcode 类

1. 构造函数

`Barcode(byte barcodeType);`

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

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

i. 参数 `barcodeType` 为条码类型:

类型常量以 `PrinterConstants.BarcodeType` 开头:

一维条码: `UPC_A`, `UPC_E`, `JAN13`, `JAN8`, `CODE39`,

`ITF`, `CODABAR`, `CODE93`, `CODE128`。

二维条码: `PDF417`, `DATAMATRIX`, `QRCODE`。

ii. 参数 `param1`, `param2`, `param3` 为条码参数:

条码类型 `type` 为一维条码时, 三个参数表示:

`param1`: 条码横向宽度 $2 \leq n \leq 6$, 默认为 2。

`param2`: 条码高度 $1 \leq n \leq 255$, 默认 162

`param3`: 条码注释位置, 0 不打印, 1 上方, 2 下方, 3 上下方均有

条码类型 `type` 为二维条码时, 三个参数表示不同的意思:

1. PDF417

`param1`: 表示每行字符数, $1 \leq n \leq 30$ 。

`param2`: 表示纠错等级, $0 \leq n \leq 8$ 。

`param3`: 表示纵向放大倍数。

2. DATA MATRIX

`param1`: 表示图形高, $0 \leq n \leq 144$ (0:自动选择)。

`param2`: 表示图形宽, $8 \leq n \leq 144$ (`param1` 为 0 时,无效)。

`param3`: 表示纵向放大倍数。

3. QR CODE

`param1`: 表示图形版本号, $1 \leq n \leq 30$ (0:自动选择)。

`param2`: 表示纠错等级, $n = 76, 77, 81, 72$ (L:7%,M:15%,Q:25%,H:30%)。

`param3`: 表示纵向放大倍数。

iii. 参数 `content` 为条码数据。

四、CanvasPrint 类

1. 以图形方式打印非常用的语言或者自定义的排版。主要方法如下：

初始化操作，参数为打印机类型，使用此方法时默认初

始化画布宽度为最大宽度。如 T9 的 72mm。其他 47mm。

`init(PrinterType printerType);`

2. 设置字体属性。参数为 `FontProperty` 类型，`FontProperty` 类是字体属性的一个集合，包括加粗，斜体等。

`setFontProperty(FontProperty fp);`

`setFont` 方法顺序参数为：

第 1 个参数：是否粗体（取值为 `true/false`）

（经测试，若单独设置中文为粗体，打印不出来）

第 2 个参数：是否斜体（取值为 `true/false`）

第 3 个参数：是否下划线（取值为 `true/false`）

第 4 个参数：是否删除线（取值为 `true/false`）

第 5 个参数：字体大小（取值为一整数）

第 6 个参数：字体类型（一般设置为 `null`，表示使用系统默认字体）

若不用此方法，也可单独进行设置，方法如下：

`setLineWidth(float w);` 画笔宽度

`setTextSize(int size);` 字体大小

`setItalic(boolean italic);` 是否斜体

`setStrikeThruText(boolean strike);` 是否删除线

`setUnderlineText(boolean underline);` 是否有下划线

`setFakeBoldText(boolean fakeBold);` 是否粗体

3. 画字符串，参数 `x`, `y` 为字符串的左下角坐标，`nStr` 为所要画的字符串。

`drawText(String nStr);`

`drawText(float x, String nStr);`

`drawText(float x, float y, String nStr);`

4. 绘制直线。参数 `startX`, `startY` 为起始坐标，`stopX`, `stopY` 为结束坐标。

`drawLine(float startX, float startY, float stopX, float stopY);`

5. 绘制矩形。参数为左上右下边距离打印左边距或上边距的距离

`drawRectangle(float left, float top, float right, float bottom);`

6. 绘制椭圆，参数为椭圆外切矩形的左上右下边距离打印左边距或上边距的距离

`drawEllips(float left, float top, float right, float bottom);`

7. 绘制图片，参数 `left` 与 `top` 为图片的左上角坐标。`bitmap` 为图片文件

`drawImage(Bitmap image);`

`drawImage(float left, Bitmap image);`

`drawImage(float left, float top, Bitmap image);`

8. 获得画布上绘制的图像，发送给打印机打印。

`getCanvasImage();`

9. 设置文本是否靠右，针对一些特殊文字，如阿拉伯文。

`setTextAlignRight(boolean alignRight);`

10. 文本超出是否换行。

`setTextExceedNewLine(boolean newLine);`

11. 文本超出后换行是否使用分隔字符串，避免一个单词被拆分。默认按空格拆分。

`setUseSplit(boolean useSplit);`

`setUseSplitAndString(boolean useSplit, String splitStr);`