

Android-SDK 开发文档 v1.6

一、 简介

1. 软件包名称:android_print_sdk

2. 类库名称:

ClassName	Discription
BluetoothPrinter	蓝牙打印机类
USBPrinter	Usb 打印机类
WiFiPrinter	WiFi 打印机类
Barcode	条码打印类
Table	表格打印类
CanvasPrint	画布打印类
PdfContext, CodecDocument, CodecPage	Pdf 文件打印相关类

二、 BluetoothPrinter 类:

1. 实例化 :

a) `BluetoothPrinter(BluetoothDevice device);` 参数为 `BluetoothDevice`, 开发人员将获得的 `BluetoothDevice` 传入以初始化。

b) `BluetoothPrinter(String macAddress, int flag)` 参数 1 为蓝牙设备的 MAC 地址, 参数 2 未使用。

c) `BluetoothPrinter(String deviceName)` 参数为蓝牙设备的名称, 只能初始化已配对的蓝牙设备。且如果有名称相

同的多个设备，则只打开第一个找到的设备 (此方法已过期)。

不推荐使用此方法实例化打印机。

注：若手动指定蓝牙地址或蓝牙设备名称实例化打印机,有可能实例化打印机为空,为避免报错,建议实例化之后使 `isPrinterNull()` 方法判断打印机操作类是否为空。

2. 打 开 与 关 闭 连 接

a) `Void openConnection()` 与蓝牙设备建立连接

`Void closeConnection()` 断开与蓝牙设备的连接

结合 `handler` 返回的蓝牙打印机连接信息，包括连接中，连接成功，连接失败，连接关闭。

```
Handler_Connect_Connecting,  
  
Handler_Connect_Success ,  
  
Handler_Connect_Failed,  
  
Handler_Connect_Closed,  
  
Handler_Message_Error,  
  
Handler_Message_Read。
```

b) `Boolean Open ()` 与蓝牙设备建立连接 （新 API）

返回值:`true/false`

`Void Close()` 断开与蓝牙设备的连接

三、 **USBPrinter** 类:

1. 实例化:

```
USBPrinter(Context mContext);
```

2. 打开与关闭连接

```
boolean openConnection(UsbDevice device) 打开 USB 连接
```

```
void closeConnection() 关闭连接
```

四、 WiFiPrinter 类:

1. 实例化:

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

ipAddress:打印机 IP 地址

portNumber:端口号

2. 打开与关闭连接

```
boolean open() 打开连接
```

```
void close() 关闭连接
```

五、 打印方法

1. ESC/POS 指令打印（一般用于普通热敏纸）

```
a) printText(String content)
```

打印普通文本，参数为要打印的文本内容 (send() 方法已过期)，若要换行，在需要换行的字符串后拼接“\n”。

```
b) printByteData(byte[] content)
```

打印 byte 数据，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] = 27;  
command[1] = 97;  
command[2] = 49;  
  
printByteData(command);
```

c) `printImage(String path/Bitmap bm)`

打印图片，参数为图片路径或者 bitmap 文件。

d) `printImage(String path/Bitmap bm , int left)`

打印图片，参数 1 为图片路径或者 bitmap 文件。参数 2 为图片距离左边距离。

e) `PrintBarcode(String code)`

打印条形码，参数为条码号。条码设置可调用 `setBarcode()` 法。

f) `PrintTable(Table table)`

打印表格数据，参数为实例化的 Table 类。

g) `PrintTitle()`

打印标题，设置标题调用 `setTitle()` 方法。

h) `cutPaper()`

切纸。

2. CPCL 指令打印（一般用于标签纸）

a) 打印表格：`drawBox (int lineWidth, int top_left_x, int top_left_y, int bottom_right_x, int bottom_right_y)`

b) 打印直线: drawLine(int lineWidth, int start_x, int start_y,int end_x, int end_y)

c) 打印文字: drawText (int bold, int fontNaID, int fontSzID, int x, int y, String text)

Bold (0,1) : 0 不加粗, 1 加粗。

fontNaID: 字体号 (16x16 点阵为 55, 24x24 点阵为 24)

fontSzID: 字体大小 (取值为下图 size)

size	纵向放大	size	横向放大
0	1 (正常)	0	1 (正常)
1	2 (2倍高)	10	2 (2倍宽)
2	3	20	3
3	4	30	4
4	5	40	5
5	6	50	6
6	7	60	7
7	8	70	8

X:

x 轴正反向距离

y: y 轴负方向距离

d) 打印文字 drawText (int text_x, int text_y, String text,int fontSize, int rotate, int bold, boolean reverse, boolean underline);

text_x 起始横坐标

text_y 起始纵坐标

text 字符串

fontSize 字体大小:

20:20 点阵;

1:16 点阵;

2:24 点阵;

3:32 点阵;

4:24 点阵放大一倍;

5:32 点阵放大一倍;

6:24 点阵放大两倍;

7:32 点阵放大两倍;

其他:24 点阵;

Rot4ate 旋转角度 0:不旋转;1:90度;2:180°;3:270°;

bold 是否粗体 0:取消;1:设置;

underline 是否有下划线 false:没有;true:有;

reverse 是否反白 false:不反白;true:反白;

e) 打印文本框内容(支持自动换行)drawText (int text_x, int
text_y, String text, int fontSize, int rotate, int
bold, boolean reverse, boolean underline);

text_x 文字起始 x 坐标

text_y 文字起始 y 坐标

width 文本框宽度

height 文本框高度

str 文本内容

fontsize 字体大小

rotate 旋转度数

bold 加粗

reverse 反显

underline 下划线

(参数上同)

f) 一维条码: drawBarCode (int var1, int var2, int height, int x, int y, String text)

Var1: 条码窄条宽宽度

Var2: 宽条宽和窄条宽比率

Height: 条码高度

x: 起始 x 坐标

y: 起始 y 坐标

Text: 条码数据

g) drawBarCode 打印一维码 (int start_x, int start_y, String text, int type, boolean rotate, int linewidth, int height);

start_x 打印的起始横坐标

start_y 打印的起始纵坐标

text 字符串

type 条码类型:

0:CODE39;

1:CODE128;

2:CODE93;

3:CODEBAR;

4:EAN8;

5:EAN13;

6:UPCA;

7:UPC-E;

8:ITF;

rotate 旋转角度 0:不旋转;1 : 旋转°;

linewidth 条码宽度 (0-6)

height 条码高度

h) 二维条码: drawQrCode (int x, int y, int Mn,int Un,
String text, int rotate)

x: 水平方向起始位置

y: 垂直方向起始位置

Mn: QR 条码模式.n 范围:1 或者 2。

Un: 放大倍数。n 范围: 1-15 (部分型号仅支持 0-6)。

Text: 条码数据

Rotate: 旋转度数。(目前选定为 PRotate.Rotate_0)

i) 打印图片 drawGraphic (int start_x, int start_y, int
bmp_size_x, int bmp_size_y, Bitmap bmp)

start_x: 水平起始位置

start_y: 垂直起始位置

bmp_size_x: 0

bmp_size_y: 0

Bmp: 图片对象

j) 打印反显 inverse (int x1, int y1, int x2, int y2,
int width)

x1: 左上角 x 坐标

y1: 左上角 y 坐标

x2: 右上角 x 坐标

y2: 右上角 y 坐标

Width: 反显内容宽度

注: 在反显所选择的区域内, 将区域内生成的内容, 黑色的区域描绘成白色, 白色的区域描绘成黑色. 使用顺序: 先打印字体, 在打印反显.

k) 设置纸高宽 `setPage (int width, int hight)`

Width: 页宽

Hight: 页高

l) 打印结尾和旋转 `lablePrint (int horizontal, int skip)`

horizontal: 取 1 时 顺时针旋转 90°

skip: 取 1 时 缝隙定位

3. 设置方法

a) `setHandler (Handler handler)`

设置 Handler 接收 SDK 返回的蓝牙打印机连接信息, 包括连接中; 连接成功; 连接失败; 连接关闭。对应常量为:

```
Handler_Connect_Connecting,  
Handler_Connect_Success ,  
Handler_Connect_Failed,  
Handler_Connect_Closed,  
Handler_Message_Error,  
Handler_Message_Read.
```

b) `setEncoding (String encoding)`

设置打印文本的字符编码格式,默认为 GBK.

c) `setTitle (String title, String subTitle, Bitmap logo)`

设置打印公司抬头,顺序参数为:主标题,副标题,公司 logo

d) `setCurrentPrintType(PrinterType currentPrintType)`

设置打印机类型,数为枚举类型 `PrinterType`,M21,ML31 等.

e) `setCharacterMultiple(int x, int y)`

设置字符倍数,参数 x 为宽,y 为高.x,y 取值为[0-7],0 为默认宽高.

f) `setLeftMargin(int nL, int nH)`

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

g) `setPrintModel(Boolean Bold, Boolean sDouble eight, boolean isDoubleWidth, boolean isUnderLine)`

设置打印模式,顺序参数为:

IsBold:否粗体

IsDoubleHeight:否倍高

IsDoubleWidth:否倍宽

IsUnderLine:否下划线

h) `setPrinter(int command)`

设置打印机，参数为打印机命令。全部命令常量如下：

COMM_INIT_PRINTER:始化打印机(已提出单独方法

init())

COMM_WAKE_PRINTER:唤醒打印机

COMM_PRINT_AND_RETURN_STANDARD:页模式下打印并返回

标准模式

COMM_PRINT_AND_NEWLINE:打印并换行

COMM_PRINT_AND_ENTER: 打印并回车

COMM_MOVE_NEXT_TAB_POSITION: 移动打印位置到下一个
水平制表符位置

COMM_DEF_LINE_SPACING: 恢复默认行高

i) setPrinter(int command, int value)

设置打印机,参数 command 为打印机命令, value 为命令对应的
值。

COMM_PRINT_AND_WAKE_PAPER_BY_LNCH: 打印并进纸

value 高度(英寸)

COMM_PRINT_AND_WAKE_PAPER_BY_LINE: 打印并走纸

value 行

COMM_CLOCKWISE_ROTATE_90: 顺时针旋转 90 度

COMM_LINE_HEIGHT: 设置行高

COMM_CHARACTER_RIGHT_MARGIN: 字符右间距

COMM_ALIGN: 对齐模式。 3 种对齐方式变量如下:

COMM_ALIGN_LEFT: 左对齐

COMM_ALIGN_RIGHT: 右对齐

COMM_ALIGN_CENTER: 居中对齐

j) setAutoReceiveDate(boolean auto)

设置是否自动接收返回值（打印机状态），若 auto 为 true，需要设置 handler，即调用 sethandler 方法，则收到返回值时，会发送 handler_message_read 消息，其中 msg.arg1 是接收数据的长度，msg.obj 是具体的数据。

```
Int length = msg.arg1;
```

```
byte[] bytes = (byte[]) msg.obj;
```

```
String recStr = new String(bytes);
```

Handler_Message_Error 消息表示接收数据出错，一般情况是连接出现问题。

k) setPrintDensity(int density)

设置打印机浓度 1-5

l) Int getPrinterStatus()

获取打印机状态

返回值：0：打印机正常

1：打印机缺纸

2：打印机开上盖

3：缺纸和上盖开

4：缺纸和切刀错误

5：缺纸和过温

6：缺纸和切刀错误和过温

7：切刀错误和上盖开

8：过温和上盖开

9：切刀错误和过温和上盖开

10：缺纸和开盖和切刀错误

11：缺纸和上盖和过温

12：缺纸和上盖和过温和切刀错误

13：切刀错误

14：过温

15：切刀错误和过温

16：打印机通讯异常（无返回值，可以结合调用

`isConnected（）`来判断打印机是否断开连接。）

m) `setNeedVerify(Boolean need)`

设置蓝牙连接是否校验，调用后只可连接本公司打印机。

六、 **CanvasPrint** 画布打印类提供的方法如下：

简要说明：以图形方式打印非常用的语言或者自定义的排版。可以在画布上画文字（可以调用第三方字体文件设置字体），画条码，画图形等。最终转换 bitmap，调用 `printImage` 完成画布的打印。

1. 初始化画布

方法名：`public void init(PrinterType printerType)`

方法描述：初始化操作画布

参数说明：参数为打印机类型，如传入 `PrinterType.M31` 指定画布为 80mm 打印 纸纸宽，`PrinterType.M21` 指定画布的宽度为 58mm 打印纸纸宽

返回值：无

2. 设置字体属性

方法名: `public void setFontProperty(FontProperty fp)`

方法描述: 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。

参数说明: `FontProperty` 类实例

返回值: 无

注: 调用此方法需要显示实例化 `FontProperty` ;

`setFont(booleanbBold,booleanbItalic,boolean
bUnderLine,boolean bStrikeout,int iSize,Typefaces
Face)` 方法顺序参数为:

`bBold` `true` 粗体; `false` 正常字体

注: 单独设置中文为粗体, 打印不出来

`bItalic` `true` 斜体, `false` 正常字体

`bUnderLine` `true` 有下划线, `false` 无下划线

`bStrikeout` `true` 有删除线, `false` 无删除线

`iSize` 字体大小 (取值为一整数)

`bItalic` `true` 斜体, `false` 正常字体

`sFace` 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:

a) 1 `setLineWidth(floatw);` 画笔宽度

- b) `1 setTextSize(intsize);` 字体大小
- c) `1 setItalic(booleanitalic);` 是否斜体
- d) `1 setStrikeThruText(booleanstrike);` 是否删除线
- e) `1 setUnderlineText(booleanunderline);` 是否有下列线
- f) `1 setFakeBoldText(booleanfakeBold);` 是否粗体

3. 画布内容相关方法

- a)** 画字符串，`x, y` 为字符串的左下角坐标, `nStr` 为所要画的字符串

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

- b)** 绘制直线。

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

- c)** 绘制矩形，参数为左上角，右下角坐标

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

- d)** 绘制椭圆，参数为椭圆外切矩形的左上角，右下角坐标

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

- e)** 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(Bitmapimage);  
drawImage(float left, Bitmap image);  
drawImage(float left, float top, Bitmap image);
```

获得画布上绘制的图像, 发送给打印机打印。

- f)** 获得画布上绘制的图像, 发送给打印机打印。

```
getCanvasImage();
```

- g)** 设置文本是否靠右, 针对一些特殊文字, 如阿拉伯文。

```
setTextAlignRight(boolean alignRight);
```

- h)** 文本超出是否换行;

```
setTextExceedNewLine(boolean newLine);
```

- i)** 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);
```

- j)** 画布打印调用示例

```
CanvasPrintcp=newCanvasPrint();//创建画布  
Bitmap bitmapCODE39 =  
createBitmapQR_CODE("123456789",270,270);//生  
成二维码 cp.init(PrinterType.M31);
```



```

//将二维码画到画布上(0,0)处坐标
cp.drawImage(0, 0, bitmapCODE39);
cp.drawImage(0, 0, bitmapCODE39);

//创建字体
FontPropertyfp=newFontProperty();
fp.setFont(true, false, false, false, 40, null);

//设置字体
cp.setFontProperty(fp);

//将文字画到画布上指定坐标处

cp.drawText(250,80,"扫一扫 升级");

cp.drawText(250, 120, "您的智能车生活");

cp.drawText(250,180,"彩码头客服电话");
cp.drawText(250, 220, "4008317317");

//将画布保存成图片并进行打印
mPrinter.printImage(cp.getCanvasImage());

```

七、 Table 类

1. Table(String column, String regular,int[]
columnWidth)

column :以 regular 分隔的表头。形如“序号,单价,数量,
金额”

regular: 表内字符串的分隔符。如上面的是“,”。

ColumnWidth: 是表格每一列的字符宽度。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. `add(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. `get(int location)`

获得某行数据，0 行为表头。

4. `getListTableData()`

获得表格数据的一个 list 集合。

5. `setHasSeparator(boolean hasSeparator)`

设置打印表格的前后是否有“====”分隔线。

6. `hasSeparator()`

判断当前打印表格是否已设置打印分隔线。

八、Barcode 类

1. 创建 Barcode 实例

方法名：`Barcode(byte barcodeType, int param1, int param2, in param3, String content);`

方法描述：创建 Barcode 实例

参数说明：`barcodeType` 为条码类型，类型常量如下：

一维条码：`UPC_A`，`UPC_E`，`JAN13`，`JAN8`，`CODE39`，`ITF`，`CODABAR`，`CODE93`，`CODE128`。

二维条码：`PDF417`，`DATAMATRIX`，`QRCODE`。

param1, param2, param3 为具体条码参数:

a) 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。

b) 条码类型 type 为二维条码时, 三个参数表示不同的意思:

i. 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。

ii. DATAMATRIX :

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。

iii. QRCODE:

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。content 为条码数据

2. 条码打印调用示例

打印一维码调用示例, 以 code128 为例

```
Barcode barcode1 = new Barcode(BluetoothPrinter.  
BAR_CODE_TYPE_CODE128, 2, 150, 2, "123456");  
  
mPrinter.printBarCode(barcode1); //mPrinter 实例化的打  
印机实体类，且打印机已经连接
```

打印二维码调用示例，以 QRCode 为例

```
Barcode barcode2 = new Barcode(BluetoothPrinter.  
BAR_CODE_TYPE_QRCODE, 2, 3, 6, "123456");  
  
mPrinter.printBarCode(barcode2); //mPrinter 实例化的打  
印机实体类，且打印机已经连接
```