

Android-SDK 打印机开发文档

(V1.0)

目录

1、简介.....	1
2、名词解释.....	1
2.1 名词解释.....	1
2.2 相关知识.....	1
3、相关类说明.....	2
3.1、PrinterInstance 打印核心类.....	2
3.1.1 获得打印机实体类相关方法.....	2
3.1.2 打印机通信相关方法.....	3
3.1.3 打印机设置相关方法.....	4
3.1.4 打印内容相关方法.....	6
3.1.5 标签打印相关方法.....	7
3.1.6 控制打印机硬件方法.....	13

3.1.7 获取打印机状态相关方法.....	13
3.1.8 连接打印调用示例.....	15
3.2 、Barcode 打印类.....	16
3.2.1 构造函数.....	16
3.2.2 条码打印调用示例.....	17
3.3 、Table 表格打印类.....	18
3.3.1 创建 Table 类实例.....	18
3.3.2 添加一行数据.....	18
3.3.3 设置 Table 中列对其方式.....	18
3.3.4 表格打印调用示例.....	18
3.4 、CanvasPrint 画布打印类.....	19
3.4.1 初始化画布.....	19
3.4.2 设置字体属性.....	19
3.4.3 画布内容相关方法.....	20
3.4.4 画布打印调用示例.....	21
3.5 、BitmapConvertor 类.....	22
3.5.1 彩色图转换为黑白二值 bitmap.....	22
3.6、PDF 文件打印相关类.....	22
3.6.1 打开 PDF 文件.....	22
3.6.2 获得 PDF 文件某一页对象.....	22
3.6.3 PDF 转换为 Bitmap.....	23
3.6.4 PDF 打印调用示例.....	23

4、附录.....	24
4.1、CODE128 码综述.....	24
4.2、字符集.....	24

1、简介

Android-SDK 是基于我司打印机开发的一套 Android 函数接口。此版本接口旨在帮助客户更便捷地使用我司打印机。注意此接口不再支持 SDK4.0 及更早版本，如使用 SDK4.0 及更早版本出现相关软件问题，请咨询我司售前。此 SDK 支持蓝牙，串口，WIFI，USB 通讯方式。相关类库说明如下：

ClassName	Discription
PrinterInstance	打印机核心类
Barcode	条码打印类
Table	表格打印类
CanvasPrint	画布打印类
PrinterConstants	相关常量类
BitmapConvertor	彩色 Bitmap 转换类
PdfContext, CodecDocument, CodecPage	Pdf 文件打印相关类

2、名词解释

2.1 名词解释

- 打印宽度：指打印机可支持的最大横向打印范围，由打印机本身决定。例如：对于 80mm 纸宽的打印机，打印最大有效宽度为 72mm(576 点)，对于 58mm 纸宽的打印机，打印最大有效宽度为 48mm(384 点)；
- 打印区域：打印区域可通过指令设置。打印区域一定小于等于打印宽度；
- 行高：字符行的高度，行高 = 字符高 + 行间距；
- 黑标纸：黑标是预印刷在黑标纸上的黑块，通过它可以实现定位；
- 分辨率 Dpi(dots per inch)：每英寸内打印的点数； 纵向或横向移动单位：默认一个移动单位就是一个打印点，横向移动单位为 1/8mm，纵向移动单位为 1/8mm；

2.2 相关知识

- 西文打印：我们规定的西文字符包含 ASCII 符和 CodePage(代码页)。其中 ASCII 符的范围为 0x20~0x7F，CodePage 的范围为 0x80~0xFF。西文语言（如德语和西班牙语）都有各自的单字节代码页。由于代码页的编码和汉

字编码有重叠部分，所以请在西文打印模式下打印代码页内容。常用的西文
字符点阵有：字体 A：12 × 24(点)、字体 B：9 × 17(点)；

- 中文打印：我们规定的中文字符包括简体中文和繁体中文。其中我们常用的
简体中文字符集有 GB2312、GB18030, 常用的繁体中文字符集为 BIG5。
常用的中文字符点阵有：24×24(点)；
- 倍高打印：字符高度为正常高度的两倍的打印方式；
- 倍宽打印：字符宽度为正常宽度的两倍的打印方式；

3、相关类说明

3.1、PrinterInstance 打印核心类

类说明：此类是打印的核心类，提供了获取打印机实体类对象方法，打印机
通信类方法，打印数据相关方法，打印机设置相关方法，标签打印相关方法，
获取打印机状态相关方法，以及控制打印机硬件相关方法。

3.1.1 获得打印机实体类相关方法

- 获得蓝牙打印机实体类对象

`PrinterInstance static synchronized getPrinterInstance (BluetoothDevice bthDevice,
Handler handler)`

参数：

`bthDeviceBluetoothDevice` 设备对象

`handlerHandler` 对象可以用来接收连接成功或连接失败的消息 返回值：

`PrinterInstance` 实体类对象

- 获得 USB 打印机实体类对象

`PrinterInstance static synchronized getPrinterInstance (Context
context, UsbDevice usbDevice, Handler handler)`

参数：

`context` 上下文对象

`usbDeviceUsbDeviceUSB` 设备对象

`handlerHandler` 对象可以用来接收连接成功或连接失败的消息 返回值：

`PrinterInstance` 实体类对象

- 获得 wifi 打印机实体类对象

`PrinterInstance` static synchronized `getPrinterInstance (String ipAddress, int portNumber, Handler handler)`

参数:

`ipAddress` 点分十进制的打印机 IP 地址字符串, 如: " 192. 168. 0. 100"

`portNumber` 打印机端口号, 默认是 " 9100"

`handlerHandler` 对象可以用来接收连接成功或连接失败的消息 返回值:

`PrinterInstance` 实体类对象

➤ 获得串口打印机实体类对象

`PrinterInstance` static synchronized `getPrinterInstance (File device, int baudrate, in flags, Handler handler)`

参数:

`device` 串口设备对象

`baudrate` 串口的波特率, 具体的波特率以打印机打印自检页为准

`flags` 串口的标志位, 默认为 0

`handlerHandler` 对象可以用来接收连接成功或连接失败的消息 返回值:

`PrinterInstance` 实体类对象

注: 以上所有构造方法的参数 `handler` 均用于接收打印机的连接状态, 对应的状态有:

`PrinterConstants. Connect. SUCCESS;` 连接成功

`PrinterConstants. Connect. FAILED;` 连接失败

`PrinterConstants. Connect. CLOSED;` 连接关闭

3.1.2 打印机通信相关方法

➤ 连接打印机

方法名: `public boolean openConnection ()`

方法描述: 连接打印机, 可用 `handler` 对象接收连接成功或者连接失败的消息

参数: 无

返回值: `true` 连接成功; `false` 连接失败

➤ 关闭与打印机的连接

方法名: `public void closeConnection ()`

方法描述：关闭与打印机的连接，可用 handler 对象接收关闭连接的消息

参数：无

返回值：无

➤ 向打印机发送十六进制数据

方法名：public int sendBytesData(byte[] srcData) 方法描述：发送十六进制指令至打印机，例如：若此 SDK 没有提供需要的功能，可参照指令手册调用此方法直接发送指令给打印机。如：在指令手册上查得设置打印位置居中的指令为：0x1B, 0x61, 0x01，代码可如下编写以实现设置打印位置居中效果：

```
byte[] command = newbyte[3];
command[0] = 0x1B;
command[1] = 0x61;
command[2] = 0x01;
sendByteData (command);
```

参数：srcData byte 数组 返回

值：

- > 0 成功发送到打印机的字节数
- 1 未初始化打印
- 2 srcData 为空或者 srcData 里没有数据。

➤ 读取打印机返回的数据 方法名：

public int read(byte[] buffer) 方法描

述：读取打印机返回的数据 参数：

用于接收读到字节的数组 返回值：

- > 0 成功读到的字节数
- 1 未初始化打印
- 2 srcData 为空或者 srcData 里没有数据。

3.1.3 打印机设置相关方法 注意：此设置方法在标签打

印机 CPCL 指令编程中无效

➤ 初始化打印机

方法名: `public void initPrinter()`

方法描述: 初始化打印机 参

数: 无

返回值: 无

➤ 设置打印机字体

方法名: `public void setFont(int mCharacterType, int mWidth, int mHeight, int`

`mBold,int mUnderline)` 方法描述: 设置打印机打印字体, 包括设置加粗,

倍高, 倍宽, 下划线, 标准 ASCII 字体 A (12×24), 压缩 ASCII 字体 B

(9×17)

参数:

mCharacterType: 0 表示 12×24 字体大小, 1 表示 9×16 字体大小, 此设置临时有效, 打印机不保存此设置, 即打印机重启后无效。如需打印 9×16 大小字体, 需要每次传入 1 手动设置。

mWidth: 倍宽, 范围 0~7

mHeight: 倍高, 范围 0~7 **mBold:**

0 不加粗, 1 加粗 **mUnderline:** 0 无

下划线, 1 下划线 返回值: 无

➤ 设置打印机打印

方法名: `public void setPrinter(in tcommand, int value)` 方法描述: 设置打印

机, 包括打印并走纸 n 点行; 打印并走纸 n 字符行; 设置字符对齐方式,

左对齐, 居中, 右对。

参数: **command**

PrinterConstants. Command. PRINT_AND_WAKE_PAPER_BY_LNCH 打印并走纸 value 点行

PrinterConstants. Command. PRINT_AND_WAKE_PAPER_BY_LINE

打印并走纸 value 字符行

PrinterConstants. Command. ALIGN 设置打印内容位置, Value 值可指定设置的具体位置, 可以为


```
PrinterConstants. Command.ALIGN_LEFT;
```

```
PrinterConstants.Command.ALIGN_CENTER;
```

PrinterConstants. Command.ALIGN_RIGHT

➤ 设置距离左边距点数

方法名: `public void setLeftMargin (in tdots)`

方法描述：设置距离左边距点数 参数：

dots	距离左边距点数
------	---------

返回值：无

3.1.4 打印内容相关方法

注意：以下方法在标签打印机处于 CPCL 指令模式下无效

► 打印文字

方法名: `public void printText(String content)`

方法描述：打印文字

参数: content 需要打印的文本内容

返回值：无

注意：打印数据会缓冲直到数据满一行或者 content 后面加上 “\n”

(printText(“helloworld\r\n”))才能打印出来content数据。

► 打印图片

方法名: public void printImage(Bitmap bitmap, PAlign alignType, int left, boolean isCompressed)

方法描述：距离左边距 left 点开始打印图片，注意：打印机只能打印黑白信息，传入彩色位图打印，打印仍然是黑白，但会导致打印的图片严重失真。

BitmapConvertor 类提供了将彩色图转为单色黑白图的方法，关于此方法请参见：BitmapConvertor 类介绍

参数：bitmap 要打印的单色位图，单色位图位深度为 1 alignType: 打印图片的位置，PAlign.START 居左，PAlign.CENTER 居中，PAlign.END 居右，PAlign.NONE 不指定打印位置。注意：当参数为 PAlign.NONE，可以通过设置 left 参数微调打印图片距离左边距的位置；其他参数时传入 left 将无效。如要设置居中，居右，居左时，调用此方法前

需要指定打印机打印宽度，即：PrinterConstants.paperWidth = 384(58mm 纸宽为 384，80mm 纸宽此值为 576，100mm 纸宽此值为 724)。

left: 当 PAlign.NONE 时，传入 Left 可以精细调节打印图片距离左边距的位置。alignType 不为 PAlign.NONE 时，此参数无效，left 必须是 8 的整倍数。

isCompressed: 是否使用压缩算法，true 使用；false 不使用。使用压缩算法 可以提高打印图片的速率。注：使用压缩方式打印图片需要定制打印机，一般打印机不支持此功能。

返回值：无

➤ 打印表格

方法说明： public void printTable(Table table) 方法描述：打印表格

参数：Table 实例化对象。具体实例化调用例子请参见：Table 类 3.3.4 节 返回值：无

➤ public void printBarCode(Barcode barcode)

方法描述：打印条码，包括一维码和二维码

参数：barcode 实例化对象，具体实例化例子请参见：Barcode 类 3.2.2 节 返回值：无

3.1.5 标签打印相关方法

注：以下方法适用于标签打印机，打印机需支持标准 CPCL 指令。

➤ 设置要打印区域的页宽和页高

方法名： public void pageSetup(LablePaperType paperWidth, int pageWidth, int pageHeight)

方法描述：设置要打印区域的页宽和页高

参数：

paperWidth 设置纸张类型，LablePaperType. Size_80mm, 80mm 纸；

LablePaperType. Size_58mm, 58mm 纸； LablePaperType. Size_100mm, 100mm 纸

pageWidth 页宽

pageHeight 页高

返回值：无

➤ 打印直线

方法名: `public void drawLine (int lineWidth, int startX, int startY, int endX, int endY, boolean isSolidLine)`

方法描述: 打印直线

参数:

<code>lineWidth</code>	打印线条的宽度
<code>startX</code>	开始点 X 坐标
<code>startY</code>	开始点 Y 坐标
<code>endX</code>	结束点 X 坐标
<code>startY</code>	结束点 Y 坐标
<code>isSolidLine</code>	<code>true</code> 实线 ; <code>false</code> 虚线

返回值: 无

➤ 打印框体

方法说明: `public void drawBorder (int lineWidth, int top_left_x, int top_left_y, int bottom_right_x, int bottom_right_y)` 方法描述: 打印框体

参数:

<code>lineWidth</code>	打印线条的宽度
<code>top_left_x</code>	框体左上角点 X 坐标
<code>top_left_y</code>	框体左上角点 Y 坐标
<code>bottom_right_x</code>	框体右下角点 X 坐标
<code>bottom_right_y</code>	框体右下角点 Y 坐标

返回值: 无

➤ 打印文本 text

方法说明: `public void drawText (int text_x, int text_y, String text, LableFontSize fontSize, PRotate rotate, int bold, int reverse, int underline)` 方

法描述: 在点 (`text_x`, `text_y`) 处打印文本 `text`

参数说明:

<code>text_x</code>	打印起始点 X 坐标
<code>text_y</code>	打印起始点 Y 坐标

text	要打印的文本内容	
fontSize	字体大小，支持的字体大小如下：	
LableFontSize. Size_16	16 点阵	
LableFontSize. Size_24	24 点阵	
LableFontSize. Size_32	32 点阵	
LableFontSize. Size_48	48 点阵	
LableFontSize. Size_64	64 点阵	
LableFontSize. Size_72	72 点阵	
LableFontSize. Size_96	96 点阵	
rotate	旋转角度	
PRotate. Rotate_0	旋转 0 度	
PRotate. Rotate_90	旋转 90 度	
PRotate. Rotate_180	旋转 180 度	
PRotate. Rotate_270	旋转 270 度	
bold	是否加粗	1 加粗 0 不加粗
reverse	是否反白	1 反白 0 不反白
underline	是否有下划线	1 有下划线 0 无下划线
返回值：无		

➤ 区域中打印文字

方法说明：drawText(int area_start_x, int area_start_y, int area_end_x, int area_end_y, PAlign xAlign, PAlign yAlign, String text, LableFontSize fontSize, int bold, int reverse, int underline, int deleteline, PRotate rotate)

方法描述：在点区域中打印文字

参数说明：

area_start_x	矩形区域的左上角 X 点坐标
area_start_y	矩形区域的左上角 Y 点坐标
area_end_x	矩形区域的右下角 X 点坐标
area_end_y	矩形区域的右下角 Y 点坐标
xAlign	水平位置，可以是居左 PAlign. START；居中 PAlign. CENTER；居右 PAlign. END

yAlign 竖直位置，可以是居上 **PAAlign. START**；居中
PAAlign. CENTER；居下 **PAAlign. END**
fontSize 参见上面 **drawText** 方法描述：
bold 是否加粗 1 加粗 0 不加粗
reverse 是否反白 1 反白 0 不反白
underline 是否有下划线 1 有下划线 0 无下划线
deleteline 是否有删除线 1 有删除线 0 无删除线
rotate 旋转角度，参见如上方法
 返回值：无

➤ 打印一维条码

方法说明： **public void drawBarCode (int start_x, int start_y, String text, PBarcodeType type, int linewidth, int height, PRotate rotate)** 方法描述：打印一维条码

参数说明：

start_x 开始位置 x 坐标
start_y 开始位置 y 坐标
text 要打印的条码数据内容
type 一维条码类型，支持类型如下：
PAAlign. CODE128, **PAAlign. JAN3_EAN13**, **PAAlign. JAN8_EAN8**, **PAAlign. CODE93**, **PAAlign. JAN3_EAN13**,
PAAlign. UPCE, **PAAlign. ITF**, **PAAlign. UPC_E**
linewidth 条码窄条宽
height 条码高度
rotate 条码的旋转角度，即： **PRotate. Rotate_0**,
PRotate. Rotate_90, **PRotate. Rotate_180**, **PRotate. Rotate_270** 返
 回值：无

➤ 区域内打印一维条码

方法说明： **drawBarCode (int area_start_x, int area_start_y, int area_end_x, int area_end_y, PAAlign xAlign, PAAlign yAlign, int start_x, int**

start_y, String text, PBarcodeType type, int linewidth, int height, PRotate rotate)

方法描述：区域内打印一维条码

参数说明：

area_start_x	矩形区域位置左上角 x 坐标
area_start_y	矩形区域位置左上角 y 坐标
area_end_x	矩形区域位置右下角 x 坐标
area_end_y	矩形区域位置右下角 y 坐标
xAlign	水平位置，可以是居左 PAlign. START，居中 PAlign. CENTER，居右 PAlign. END
yAlign	竖直位置，参数如上
start_x	区域内条码起始位置 X 坐标
start_y	区域内条码起始位置 Y 坐标
text	要打印的条码数据内容
type	一维条码类型，支持类型如下： PBarcodeType. CODE128, PBarcodeType. JAN3_EAN13, PBarcodeType. JAN8_EAN8, PBarcodeType. CODE93, PBarcodeType. JAN3_EAN13, PBarcodeType. UPCAPBarcodeType. ITF, PBarcodeType. UPC_E
linewidth	条码窄条宽
height	条码高度
rotate	条码的旋转角度，即：PRotate. Rotate_0, PRotate. Rotate_90, PRotate. Rotate_180, PRotate. Rotate_270 返回

返回值：无

➤ 打印二维码

方法说明：public void drawQrCode (int start_x, int start_y, String text, PRotate rotate, int ver, int lel)

方法描述：打印二维码

参数：	start_x 二维码起始 X 位置
start_y	二维码起始 Y 坐标
text	二维条码的内容数据

rotate 和打印一维码旋转角度相同

ver QRCode 放大倍数 (1-6) 默认是 6

lcl 纠错等级, 0 纠错等级 L; 1 纠错等级 M, 2 纠错等级 Q, 3 纠错等级 H

返回值: 无

➤ 打印位图

方法说明: `public void drawGraphic (int start_x, int start_y, int bmp_size_x, int bmp_size_y, Bitmap bmp)`

方法描述: 在指定位置打印位图

参数说明:

start_x 打印图片位置的 X 坐标

start_y 打印图片位置的 Y 坐标

bmp_size_x 位图宽度

bmp_size_y 位图高度

bmp bitmap 对象

➤ 区域内打印 bitmap 位图

方法说明: `public void drawGraphic (int area_start_x, int area_start_y, int area_end_x, int area_end_y, PAlign xAlign, PAlign yAlign, int bmp_size_x, int bmp_size_y, Bitmap bmp)`

方法描述: 在区域内打印 bitmap 位图

参数说明:

area_start_x 矩形区域的左上角 X 点坐标

area_start_y 矩形区域的左上角 Y 点坐标

area_end_x 矩形区域的右下角 X 点坐标

area_end_y 矩形区域的右下角 Y 点坐标

xAlign 水平位置, 可以是居左 PAlign. START, 居中 PAlign. CENTER, 居右 PAlign. END

yAlign 竖直位置, 参数如上

bmp_size_x 位图宽度

bmp_size_y 位图高度

bmp bitmap 对象

返回值：无

➤ 打印

方法说明：public void print(PRotate rotate, int skip) 方法描述：打印标签，前面调用的打印文字，打印直线等方法不会立即打印。直到调用本方法才会最终打印出来。

参数说明：

rotate 0：正常打印，不旋转；1：整个页面顺时针旋转 180° 后，再打印

skip: 0：打印结束后不定位，直接停止；1：打印结束后定位到标签分割线，如果无缝隙，最大走纸 30cm 后停止

返回值：无

3.1.6 控制打印机硬件方法

➤ 切纸

方法说明：public void cutPaper(in cutterType, int n)

方法描述：切纸，可全切，半切 参数说明：

cutterType 切纸类型

48 直接全切

49 直接半切

65 进纸到(切纸位置+[n × (纵向移动单位)])并且全切

66 进纸到(切纸位置+[n × (纵向移动单位)])并且半切

注：纵向移动单位为 1 点

n: 纵向移动点数, 当 cutterType 为 65 有效

返回值：无

➤ 控制蜂鸣器响

方法说明：public void ringBuzzer(int time)

方法描述：控制蜂鸣器响 time 秒 参数说明：

time 蜂鸣器响秒数 返回值：无

3.1.7 获取打印机状态相关方法

➤ 获取打印机当前状态

方法名: `public int getCurrentStatus ()` 方法描述: 获得打印机当前的状态, 包括打印机状态正常, 缺纸, 开盖, 纸将尽, 通信异常。只有部分机型有纸将尽检测, 检测纸将尽, 需要手动 开启纸将尽使能功能。

参数说明: 无

返回值:

- 0 打印状态正常
- 1 打印机通信异常
- 2 缺纸
- 3 纸将尽
- 4 打印机开盖

➤ 打印机打印是否完成

方法说明: `public int getPrintingStatus (String BufferMsg, int timeout)`

方法描述: 判断打印机打印是否完成

参数说明: `Msg` 接收打印是否完成信息, 具体值和本方法的返回值意义一一对应。即, ”当前打印已经完成”, “未知异常”, “打印未完成, 缺纸”等。

`timeout` 预估打印一单需要的时间

返回值:

- 0 当前打印已经完成
- 1 未知异常
- 2 当前打印机正在打印中
- 3 打印未完成, 缺纸
- 4 打印未完成, 纸舱盖开盖
- 5 打印未完成, 与打印机通信失败
- 6 数据发送失败, 通信异常
- 7 接收数据格式不正确

➤ 判断打印机是否缺纸 方法名:

`public boolean isPaperOut ()` 方法描

述: 判断打印机是否缺纸

参数说明：无

返回值：true 缺纸；false 不缺纸

➤ 判断打印机是否纸将尽

方法名：public boolean isPaperWillOut() 方法描述：判断打印机是否纸将尽，只有部分机型有纸将尽传感器，获取 纸将尽状态前需要开启纸将尽使能功能

参数说明：无

返回值：true 纸将尽，false 纸足够

3.1.8 连接打印调用示例

//1、创建 handler 对象用于接收连接成功或者连接失败的消息

```
privateHandlermHandler = newHandler() {
@Override
publicvoidhandleMessage(Messagemsg)
{ switch (msg.what)
{ caseConnect.SUCCESS:
isConnected = true;//连接成功
break;
caseConnect.FAILED:
isConnected = false;
Toast.makeText(mContext, R.string.conn_failed, Toast.LENGTH_SHORT).show();
Log.i(TAG, "连接失败!");
break;
caseConnect.CLOSED:
isConnected = false;
Toast.makeText(mContext, R.string.conn_closed, Toast.LENGTH_SHORT).show();
Log.i(TAG, "连接关闭!");
break;
caseConnect.NODEVICE:
isConnected = false;
Toast.makeText(mContext, R.string.conn_no, Toast.LENGTH_SHORT).show();
break;
default:
break;
```

```

}

//2、实例化蓝牙通信方式的打印机实体类,其他通信方式实例化此对象请参照 demo
BluetoothDevice mDevice =
BluetoothAdapter.getDefaultAdapter().getRemoteDevice(devicesAddress); //devicesAddress 为蓝
牙的 mac 地址
PrinterInstance mPrinter = PrinterInstance.getPrinterInstance(mDevice, mHandler);

//3、打开连接
mPrinter.openConnection();

//4、开始打印数据
//打印前判断连接是否正常,且打印机不缺纸,不开盖
if (mPrinter != null && mPrinter.getCurrentStatus() == 0) { mPrinter.setFont(0, 1, 1, 1, 1); //设
置字体, 倍高, 倍宽, 加粗, 下划线 mPrinter.setPrinter(Command.ALIGN,
Command.ALIGN_CENTER); //设置字体居中 mPrinter.printText("printTest!" +
"\r\n"); //打印文本 printTest!
mPrinter.setPrinter(Command.PRINT_AND_WAKE_PAPER_BY_LINE, 2);
mPrinter.setFont(1, 1, 1, 1, 1); //设置 9*17 压缩字体, 倍高, 倍宽, 加粗, 下划线
mPrinter.printText("printTest!" + "\r\n"); //打印文本 printTest!
} else {
//这里打印机状态异常: 通信异常, 缺纸, 开盖, 纸将尽
Toast.makeText(mContext, "printerstatusisnotnormal!", 1).show();
}

//5、关闭打印机连接
mPrinter.closeConnection(); //断开连接

```

3.2、Barcode 打印类

3.2.1 构造函数

➤ 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。

param1，param2，param3 为具体条码参数：

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. 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。

3. 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。

3.2.2 条码打印调用示例

- 打印一维码调用示例，以 code128 为例

```
Barcodebarcode1 = newBarcode(BarcodeType.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

```
Barcodebarcode2 = newBarcode(BarcodeType.QRCODE, 2, 3, 6, "123456");
```

`mPrinter.printBarCode(barcode1);` // `mPrinter` 实例化的打印机实体类，且打印机已经连接

注：更详尽的条码打印调用请参见 `BarcoePrintActivity.java` 类

3.3、Table 表格打印类

3.3.1 创建 Table 类实例

```
public Table(String column, String regularExpression, int columnWidth[])
```

方法描述：创建 Table 类实例

参数说明：column 参数 column 为以参数 regular 分隔的表头。形如” 序号, 单价, 数量, 金额”

regularExpression 为表内字符串的分隔符。如上面的是” , ”

columnWidth 为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 个字符，英文 1 个字符，然后相加，如” 序号” 的宽度为 4 个字符。

返回值：Table 类实例

3.3.2 添加一行数据

```
public void addRow(String row)
```

方法描述：添加一行数据 参

数说明：row 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”。

返回值：无

3.3.3 设置 Table 中列对其方式 方法名：

```
setColumnAlignLeft(boolean left);
```

 方法描述：设置 Table

中列数据的对其方式，默认右对齐 参数说明：left 设置

Table 中列数据的对其方式为左对齐 返回值：无

3.3.4 表格打印调用示例

```
Stringcolumn =” 品名;数量;单价;金额” ;
```

```
Tabletable = newTable(column, “;”, newint[] { 14, 6, 6, 6});
```

```
table.addRow(“保鲜袋”+ “;10.00;1;10.00”);
```

```
table.addRow(“铁丝挂钩” + “;5.00;2;10.00”);
```

```
table.addRow("雨伞"+ " ";5.00;3;15.00");
```

mPrinter.printTable(table); //mPrinter 是实例化的 PrinterInstance 对象，且连接上了打印机

注：更详尽的 pdf 打印调用请参见 TextPrintActivity.java 中 print_note 点击事件打印小票示例。

3.4、CanvasPrint 画布打印类

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 2.1.4 中 printImage 完成画布的打印。主要方法如下：

3.4.1 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.T9 指定画布为 80mm 打印 纸纸宽，PrinterType.TIII 指定画布的宽度为 58mm 打印纸纸宽

返回值：无

3.4.2 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，然后调用 FontProperty 类

setFont(boolean bBold, boolean bItalic, boolean bUnderLine, boolean

bStrikeout, int iSize, Typefaces Face)

setFont 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 ture 斜体，false 正常字体

bUnderLine ture 有下划线，false 无下划线

bStrikeout ture 有删除线，false 无删除线

iSize 字体大小（取值为一整数）

<code>bItalic</code>	<code>true</code> 斜体, <code>false</code> 正常字体
<code>sFace</code>	字体类型 (一般设置为 <code>null</code> , 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:

<code>setLineWidth(floatw);</code>	画笔宽度
<code>setTextSize(intsize);</code>	字体大小
<code>setItalic(booleanitalic);</code>	是否斜体
<code>setStrikeThruText(booleanstrike);</code>	是否删除线
<code>setUnderlineText(booleanunderline);</code>	是否有下划线
<code>setFakeBoldText(booleanfakeBold);</code>	是否粗体 返

返回值: 无

3.4.3 画布内容相关方法

➤ 画字符串

方法名: `public void drawText(String nStr);`

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

`public void drawText(float x, float y, String nStr);` 参数说明: `x`, `y` 为字符串的左下角坐标, `nStr` 为所要画的字符串 返回值: 无

➤ 绘制直线。

方法名: `publi void drawLine(float startX, float startY, float stopX, float stopY);`

参数说明: 参数 `startX`, `startY` 为起始坐标, `stopX`, `stopY` 为结束坐标。

返回值: 无

➤ 绘制矩形。方法名: `drawRectangle(float left, float top, float right, float bottom);`

参数说明: 参数为左上角, 右下角坐标

返回值: 无

➤ 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);`

➤ 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。

```
getCanvasImage();
```

- 设置文本是否靠右，针对一些特殊文字，如阿拉伯文。

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

- 文本超出是否换行；

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

- 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);
```

3.4.4 画布打印调用示例

```
CanvasPrintcp=newCanvasPrint(); //创建画布
```

```
BitmapbitmapCODE39 = createBitmapQR_CODE("123456789", 270, 270); //生成二 维 码
```

```
cp. init(PrinterType. T9);
```

```
//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();
```

```
//字体属性赋值 此处参数个数根据 SDK 版本不同，有略微差别，酌情增减。
```

```
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(), PAlign.NONE, 0); 注:

更详尽的画布打印调用请参见 [PicturePrintActivity.java](#) 中

btn_canvas_print 打印画布点击事件。

3.5、BitmapConvertor 类

类说明: 此类主要用来处理彩色图转为黑白二值图

3.5.1 彩色图转换为黑白二值 bitmap

方法名: public Bitmap convertBitmap(Bitmap inputBitmap) 方法描述: 彩色图转为黑白二值 bitmap, 支持 JPG, PNG, Bitmap 等常见彩色图 片

参数说明:

inputBitmap 彩色 bitmap, 可使用 BitmapFactory.decodeResource 方法将

JPG,PNG,Bitmap 等常见彩色图片等转为 inputBitmap, inputBitmap 的宽度应该 是小于等于打印机最大有效打印宽度的, 超过打印纸宽度会导致打印图片不全, inputBitmap 太大会导致转换较慢, 应先压缩处理。

返回值: 无

注: 彩色图一般为 24, 32 位深度, 转变为位深度为 1 的单色 bitmap 时会有不同程度的失真。如需要打印清晰的图片, 需要用户自己制作清晰的单色 bitmap, 调用 PrinterInstance 方法的 printImage(Bitmapbitmap) 完成。

3.6、PDF 文件打印相关类

类说明: PDF 文件打印相关类主要有 PdfContext,CodecDocument,CodecPage, 这些类将 PDF 文件处理成 bitmap,调用 PrinterInstance 中 printImage, 使打印机能够正常打印 PDF 文件, 此 PDF 文件应该是黑白 PDF 文件, 彩色 PDF 文件打印将会有不同程度的失真。

3.6.1 打开 PDF 文件

方法名: public CodecDocument openDocument(String fileName)

方法描述: 打开 **PDF** 文件返回 CodecDocument 对象 参数说

明: fileName PDF 文件的绝对路径 返回值:

CodecDocument 对象

3.6.2 获得 PDF 文件某一页对象

方法名: `public CodecPage getPage (int pageNumber)`

方法描述: 返回 PDF 文件某一页 `CodecPage` 对象

参数说明: `pageNumber` PDF 文件页码

返回值: 无

注: 此方法由 `CodecDocument` 对象调用, 由 `openDocument` 方法返回获得

3.6.3 PDF 转换为 Bitmap

方法名: `public Bitmap renderBitmap (int width, int height, RectF pageSliceBounds)`

方法描述: 返回 `Bitmap` 对象

参数说明: `width` 需要转换的 `Bitmap` 宽度, `height` 需要转换 `height` 的高度,

`pageSliceBounds` `RectF` 对象, `pageSliceBounds` 有四个属性,

`top`, `left`, `bottom`, `right`, 例如 `top=0.5`, `left=0.5`, `right=1.0`, `1.0` 则截取 PDF 文件的右下角四分之一的部分。

返回值: `Bitmap` 对象

注: 此方法由 `CodecPage` 对象调用, 由 `getPage` 方法返回获得

3.6.4 PDF 打印调用示例

`CodecDocument d = pdf_conext.openDocument("/sdcard/Download/test3.pdf");` //打开 pdf 文件, 如此路径文件不存在会报错

`for (inti = 0; i<d.getPageCount(); i++) {` //获取 pdf 文件有多少页

`vuPage = d.getPage(i);`

`rf = newRectF();`

`rf.top = 0;`

`rf.left = 0;`

`rf.bottom = (float) 1.0;`

`rf.right = (float) 1.0;`

// 参数一二为生成图片的宽高, 参数三 `Rect` 的 `top`, `left`, `bottom`, `right` 为截取部分在该页中的百分比位置

// 例如 `top=0.5`, `left=0.5`, `right=1.0`, `1.0` 则截取右下角四分之一的部分

`bitmap = vuPage.renderBitmap((int) screen_width, (int) screen_height, rf);`

`newInnerAsyncTask().execute();`

附: `InnerAsyncTask` 类如下:

```
private class InnerAsyncTask extends AsyncTask<Void, Void, Void> {
```

```

@Override
protected void doInBackground(Void... params)
{
    PrinterInstance mPrinter.printImage(bitmap, PAlign.NONE,
    0); return null;
}
}

```

注：更详尽的 **PDF** 打印调用请参见 PdfPrintActivity.java 调用

4、附录

4.1、CODE128 码综述

CODE 128 码通过交替使用字符集 A、字符集 B 和字符集 C，能够对 128 个 ASCII 字符和 00~99 的 100 个数字以及一些特殊字符进行编码。每个字符集编码的字符如下：

- 字符集 A：ASCII 字符 00H 到 5FH
- 字符集 B：ASCII 字符 20H 到 7FH
- 字符集 C：00~99 的 100 个数字

CODE128 码也能对下列特殊字符进行编码：

- SHIFT 字符

“SHIFT”能使条码符号 SHIFT 字符后边第一个字符从字符集 A 转换到字符集 B，或从字符集 B 转换到字符集 A，从第二个字符开始恢复到 SHIFT 以前所用的字符集。“SHIFT”字符仅能在字符集 A 和字符集 B 之间转换使用，它无法使当前的编码字符进入或退出字符集 C 的状态。

- 字符集选择字符（CODE A、CODE B、CODE C） 这些字符能将其后边的编码字符转换到字符集 A、B 或 C。
- 功能字符（FNC1、FNC2、FNC3、FNC4） 这些功能符的用处取决于应用软件。在字符集 C 中，只有 FNC1 可用。

4.2、字符集

字符集 A

字符	发送数据		字符	发送数据		字符	发送数据	
	Hex	Decimal		Hex	Decimal		Hex	Decimal
NULL	00	0	&	26	38	L	4C	76
SOH	01	1	,	27	39	M	4D	77
STX	02	2	(28	40	N	4E	78
ETX	03	3)	29	41	O	4F	49
EOT	04	4	*	2A	42	P	50	80
ENQ	05	5	+	2B	43	Q	51	81
ACK	06	6	,	2C	44	R	52	82
BEL	07	7	—	2D	45	S	53	83
BS	08	8	.	2E	46	T	54	84
HT	09	9	/	2F	47	U	55	85
LF	0A	10	0	30	48	V	56	86
VT	0B	11	1	31	49	W	57	87
FF	0C	12	2	32	50	X	58	88
CR	0D	13	3	33	51	Y	59	89
SO	0E	14	4	34	52	Z	5A	90
SI	0F	15	5	35	53	[5B	91
DLE	10	16	6	36	54	\	5C	92
DC1	11	17	7	37	55]	5D	93
DC2	12	18	8	38	56	^	5E	94
DC3	13	19	9	39	57	—	5F	95
DC4	14	20	:	3A	58	FNC1	7B, 31	123, 49
NAK	15	21	;	3B	59	FNC2	7B, 32	123, 50
SYN	16	22	<	3C	60	FNC3	7B, 33	123, 51
ETB	17	23	=	3D	61	FNC4	7B, 34	123, 52
CAN	18	24	>	3E	62	SHIFT	7B, 53	123, 83
EM	19	25	?	3F	63	CODEB	7B, 42	123, 66
SUB	1A	26	@	40	64	CODEC	7B, 43	123, 67
ESC	1B	27	A	41	65			
FS	1C	28	B	42	66			
GS	1D	29	C	43	67			
RS	1E	30	D	44	68			
US	1F	31	E	45	69			
SP	20	32	F	46	70			
!	21	33	G	47	71			
"	22	34	H	48	72			
#	23	35	I	49	73			
\$	24	36	J	4A	74			
%	25	37	K	4B	75			

字符集 B

字符	发送数据		字符	发送数据		字符	发送数据	
	Hex	Decimal		Hex	Decimal		Hex	Decimal
SP	20	32	F	46	70	l	6C	108
!	21	33	G	47	71	m	6D	109
"	22	34	H	48	72	n	6E	110
#	23	35	I	49	73	o	6F	111
\$	24	36	J	4A	74	p	70	112
%	25	37	K	4B	75	q	71	113
&	26	38	L	4C	76	r	72	114
'	27	39	M	4D	77	s	73	115
(28	40	N	4E	78	t	74	116
)	29	41	O	4F	79	u	75	117
*	2A	42	P	50	80	v	76	118
+	2B	43	Q	51	81	w	77	119
,	2C	44	R	52	82	x	78	120
-	2D	45	S	53	83	y	79	121
.	2E	46	T	54	84	z	7A	122
/	2F	47	U	55	85	{	7B, 7B	123, 123
0	30	48	V	56	86		7C	124
1	31	49	W	57	87	}	7D	125
2	32	50	X	58	88	—	7E	126
3	33	51	Y	59	89	DEL	7F	127
4	34	52	Z	5A	90	FNC1	7B, 31	123, 49
5	35	53	[5B	91	FNC2	7B, 32	123, 50
6	36	54	\	5C	92	FNC3	7B, 33	123, 51
7	37	55]	5D	93	FNC4	7B, 34	123, 52
8	38	56	^	5E	94	SHIFT	7B, 53	123, 83
9	39	57	_	5F	95	CODEA	7B, 41	123, 65
:	3A	58	`	60	96	CODEC	7B, 43	123, 67
;	3B	59	a	61	97			
<	3C	60	b	62	98			
=	3D	61	c	63	99			
>	3E	62	d	64	100			
?	3F	63	e	65	101			
@	40	64	f	66	102			
A	41	65	g	67	103			
B	42	66	h	68	104			
C	43	67	i	69	105			
D	44	68	j	6A	106			
E	45	69	k	6B	107			

字符集 C

字符	发送数据		字符	发送数据		字符	发送数据	
	Hex	Decimal		Hex	Decimal		Hex	Decimal
0	00	0	38	26	38	76	4C	76
1	01	1	39	27	39	77	4D	77
2	02	2	40	28	40	78	4E	78
3	03	3	41	29	41	79	4F	79
4	04	4	42	2A	42	80	50	80
5	05	5	43	2B	43	81	51	81
6	06	6	44	2C	44	82	52	82
7	07	7	45	2D	45	83	53	83
8	08	8	46	2E	46	84	54	84
9	09	9	47	2F	47	85	55	85
10	0A	10	48	30	48	86	56	86
11	0B	11	49	31	49	87	57	87
12	0C	12	50	32	50	88	58	88
13	0D	13	51	33	51	89	59	89
14	0E	14	52	34	52	90	5A	90
15	0F	15	53	35	53	91	5B	91
16	10	16	54	36	54	92	5C	92
17	11	17	55	37	55	93	5D	93
18	12	18	56	38	56	94	5E	94
19	13	19	57	39	57	95	5F	95
20	14	20	58	3A	58	96	60	96
21	15	21	59	3B	59	97	61	97
22	16	22	60	3C	60	98	62	98
23	17	23	61	3D	61	99	63	99
24	18	24	62	3E	62	FNC1	7B, 31	123, 49
25	19	25	63	3F	63	CODEA	7B, 41	123, 65
26	1A	26	64	40	64	CODEB	7B, 42	123, 66
27	1B	27	65	41	65			
28	1C	28	66	42	66			
29	1D	29	67	43	67			
30	1E	30	68	44	68			
31	1F	31	69	45	69			
32	20	32	70	46	70			
33	21	33	71	47	71			
34	22	34	72	48	72			
35	23	35	73	49	73			
36	24	36	74	4A	74			
37	25	37	75	4B	75			