

# **Android-SDK 打印机开发文档**

(V1.2)

文档信息

文档名称	Android-SDK 打印机开发文档			
作 者	付道辉，杨学志			
文件名称	Android-SDK 打印机开发文档.doc			
摘 要	本文档属于Android-SDK5.2 配套文档，该版本在SDK5.1版本基础上进行修改,修改了PDF打印方法，添加了打印直线虚线，优化打印图片，优化串口操作，文档中添加多国语言打印说明。			
修订历史 (REVISIONHISTORY)				
版本号	日期	修改人	审阅人	摘 要
1.0	2016-06-22	付道辉	秦正峰	初始创建本文档
1.1	2016-12-09	杨学志		1.增加TSPL指令标签打印相关方法 2.增加黑标定位功能函数 3.增加彩色位图转换为灰度图打印功能函数 4.文档中修改sendBytesData方法返回值添加-3的说明 5.串口通信增加须调用指定.so库说明。 6.所有方法添加超链接。

1.2	2017-07-26	付道辉		<ul style="list-style-type: none"><li>1、修改SDK中PDF打印</li><li>2、优化在非root设备上打开串口崩溃问题</li><li>3、添加了打印多国语言说明</li><li>4、优化打印图片方法</li><li>5、SDK中添加了打印直线，虚线方法</li><li>6、优化其他已知bug</li><li>7、文档中添加了快速阅读指南小节</li></ul>
-----	------------	-----	--	--

## 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
2.3 本文档速指南.....	2
3、相关类说明.....	3
3.1、PrinterInstance 打印核心类.....	3
3.1.1 获得打印机实体类相关方法.....	3
3.1.2 打印机通信相关方法.....	5
3.1.3 打印机设置相关方法.....	6
3.1.4 ESC/POS 指令打印相关方法.....	8
3.1.5 CPCL 指令标签打印相关方法.....	11
3.1.6 TSPL 指令标签打印相关方法.....	17
3.1.7 控制打印机硬件方法.....	32
3.1.8 查询打印机状态相关方法.....	33
3.1.9 连接打印调用示例.....	35
3.2、Barcode 条码打印类.....	36
3.2.1 构造函数.....	36
3.2.2 条码打印调用示例.....	37
3.3、Table 表格打印类.....	38

---

3.3.1 创建 Table 类实例 .....	38
3.3.2 添加一行数据 .....	38
3.3.3 设置 Table 中列对齐方式 .....	39
3.3.4 表格打印调用示例 .....	39
3.4、CanvasPrint 画布打印类 .....	39
3.4.1 初始化画布 .....	40
3.4.2 设置字体属性 .....	40
3.4.3 画布内容相关方法 .....	41
3.4.4 画布打印调用示例 .....	42
3.5、BitmapConvertor 类 .....	43
3.5.1 彩色图转换为黑白二值 bitmap .....	43
3.6、PDF 文件打印相关类 .....	44
3.6.1 PDF 打印调用示例 .....	44
3.7、CodePagePrinter 多国语言代码页打印类 .....	45
4、附录 .....	47
4.1、CODE128 码综述 .....	47
4.2、字符集 .....	48

## 1、简介

Android-SDK5.2 是基于我司打印机开发的一套 Android 函数接口。此版本接口在 SDK5.1 基础上修改了 PDF 文件打印；优化了在非 root 设备上打开串口崩溃的问题；增加了多国语言打印说明；优化了 cpcl 以及 esc/pos 指令下打印图片算法；增加了打印直线，虚线命令；优化其他已知 bug，旨在帮助客户更便捷地使用我司打印机。注意此版本不再支持 SDK4.0 及更早版本，兼容 5.0,5.1 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
<a href="#">PrinterInstance</a>	打印机核心类
<a href="#">Barcode</a>	条码打印类
<a href="#">Table</a>	表格打印类
<a href="#">CanvasPrint</a>	画布打印类
<a href="#">BitmapConvertor</a>	彩色 Bitmap 转换类
<a href="#">PdfContext,CodecDocument,CodecPage</a>	Pdf 文件打印相关类
<a href="#">PrinterConstants</a>	相关常量类

## 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；

## 2.3 本文档快速阅读指南

由于打印机机型不尽相同，使用指令也不相同，客户开发人员对打印机熟悉程度不一，为了让客户开发人员能更快速定位到调用的 API 方法，以下文字描述将会对您有所帮助。

无论您是什么机型，只要您需要安卓下通过蓝牙，USB，WIFI（或者有线网络），串口连接打印，均可以阅读 [3.1.1](#) 以及 [3.1.2](#) 小节查询获取打印机实体类，打开通信端口，读写数据，关闭端口等 API，另外 [3.1.9](#) 提供了以蓝牙为例，初始化打印机，连接打印机，打印数据，关闭等 API 的调用应用示例。

如果您想在普通热敏纸上打印小票，您可以阅读 [3.1.4](#) 小节的内容。此小节 API 提供了打印文本，打印一维码，二维码，打印图片等相关的方法。

如果您想要打印标签，且您手中打印机支持 CPCL 指令，您可以阅读 [3.1.5](#) 小节。此小节提供了打印线条，矩形框，文本，区域中打印文本，条码等方法。

如果您想要打印标签，且您手中打印机支持 TSPL 指令，您可以阅读 [3.1.6](#) 小节。此小节提供了与 [3.1.5](#) 类似的方法，只不过实现的方式不同而已。

如果您有查询打印机状态的需求，可以阅读 [3.1.8](#) 小节。另外 [3.6](#) 小节提供 PDF 文件打印的一种方案。实际上您可能想打印更丰富的格式数据，画布打印方式可能比较适合您，请阅读 [3.4](#) 节；切纸，控制蜂鸣器，寻找黑标，

请阅读 [3.1.7](#) 小节。

注：关于网页打印需求的说明，打印机不能直接打印网页的数据，也无法识别网页中丰富多彩的格式。一般客户可以将网页数据转为 bitmap，在调用 `printImage(Bitmap bitmap, PAlign alignType, int left, boolean isCompressed)` 以图片的方式打印出来。

### 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。

<a href="#">获得打印机实体类相关方法</a>
<a href="#">打印机通信相关方法</a>
<a href="#">打印机设置相关方法</a>
<a href="#">打印内容相关方法</a>
<a href="#">CPCL 指令标签打印相关方法</a>
<a href="#">TSPL 指令标签打印相关方法</a>
<a href="#">控制打印机硬件相关方法</a>
<a href="#">获取打印机状态相关方法</a>
<a href="#">连接打印调用示例</a>

##### 3.1.1 获得打印机实体类相关方法

方法名	描述
<a href="#">getPrinterInstance</a>	获得蓝牙打印机实体类对象
<a href="#">getPrinterInstance</a>	获得 USB 打印机实体类对象
<a href="#">getPrinterInstance</a>	获得 WIFI 打印机实体类对象
<a href="#">getPrinterInstance</a>	获得串口打印机实体类对象

##### ➤ 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：

1、串口的使用必须在 demo 中加入 .so 库文件, PrintDemo/ libs/ armeabi /libserial\_port.so 文件。

2、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

方法名	描述
<a href="#">OpenConnection</a>	连接打印机
<a href="#">closeConnection</a>	关闭与打印机的连接
<a href="#">sendBytesData</a>	向打印机发送十六进制数据
<a href="#">read</a>	读取打印机返回的数据

#### ➤ 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：srcDatabyte 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
- 3        向打印机发送数据失败

#### ➤ 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

方法名	描述
<a href="#">initPrinter</a>	初始化打印机
<a href="#">setFont</a>	设置打印机字体
<a href="#">setPrinter</a>	设置打印机打印

<a href="#">setLeftMargin</a>	设置距离左边距点数
-------------------------------	-----------

➤ 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
```

#### 例 1：设置打印文本居中

```
mPrinter.setPrinter(Command.ALIGN, Command.ALIGN_CENTER);
mPrinter.printText("hello world!\n");
```

#### 例 2：设置打印一维码居中

```
mPrinter.setPrinter(Command.ALIGN, Command.ALIGN_CENTER);
Barcode barcode1 = new Barcode(BarcodeType.CODE128, 2, 150, 2, "123456");
mPrinter.printBarCode(barcode1);
```

#### 例 3：设置打印二维码居中

```
PrinterInstance.mPrinter.setPrinter(Command.ALIGN, Command.ALIGN_CENTER);
Barcode barcode2 = new Barcode(BarcodeType.QRCODE, 2, 3, 6, "123456");
PrinterInstance.mPrinter.printBarCode(barcode2);
```

注：以上居中对于打印图片 `printImage` 方法无效

#### ➤ 设置距离左边距点数

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

方法描述：设置距离左边距 inchs

参数：inchs 距离左边距英寸数

返回值：无

#### 例 1：设置距离左侧 8 个英寸

```
mPrinter.setLeftMargin(8);
mPrinter.printText("hello world!\n");
```

此方法对于打印一维码，二维码有效，对于打印图片 `printImage` 无效

### 3.1.4 ESC/POS 指令打印相关方法

注意：以下方法在标签打印机处于 CPCL 指令模式下无效

方法名	描述
<a href="#"><code>printText</code></a>	打印文字
<a href="#"><code>printImage</code></a>	打印黑白二值图片
<a href="#"><code>printColorImg2Gray</code></a>	打印灰度化的彩色图片

<a href="#">printTable</a>	打印表格
<a href="#">printBarcode</a>	打印条码
<a href="#">printhorizontalLine</a>	打印直线，虚线

➤ 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("hello world\r\n"))才能打印出来 content 数据。

➤ 打印黑白二值图片

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，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。

**例 1：以下描述设置打印图片居中**

已知打印纸宽：58mm，即横向有效打印点数 384 点

```
PrinterConstants.paperWidth = 384;
```

```
mPrinter.printImage(bmp, PAlign.Command.ALIGN_CENTER, 0, false);
```

**例 2：以下描述距离左侧 24 点打印图片**

已知打印纸宽：58mm，即横向有效打印点数 384 点

```
PrinterConstants.paperWidth = 384;
```

```
mPrinter.printImage(bmp, PAlign.NONE, 24, false);
```

返回值：无

➤ **彩色图灰度化打印**

方法名：public void printColorImg2Gray (Bitmap bitmap, PAlign alignType, int left, boolean isCompressed)

方法描述：将彩色图灰度化并且距离左边 left 点打印

参数：bitmap 要打印的 bitmap 彩色图片

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，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

➤ 打印条码

方法说明：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

➤ 打印水平直线，虚线

方法说明：public int printhorizontalLine(int lineLength, int lineWith, boolean isSolidline, int interval)

方法描述：打印直线，虚线

参数：lineLength          直线的长度  
                  lineWith          直线的宽度  
                  isSolidline      true,直线，false，虚线  
                  interval          虚线的间距，可选参数有 4,8。数值越大，间距越大

返回值：无

### 3.1.5 CPCL 指令标签打印相关方法

注：以下方法适用于标签打印机，打印机需支持标准 CPCL 指令。

方法	描述
<a href="#">pageSetup</a>	设置要打印区域的页宽和页高
<a href="#">drawLine</a>	打印直线
<a href="#">drawBorder</a>	打印框体
<a href="#">drawText</a>	打印文本
<a href="#">drawText</a>	区域中打印文字
<a href="#">drawBarCode</a>	打印一维条码
<a href="#">drawBarCode</a>	区域中打印一维条码
<a href="#">drawQrCode</a>	打印二维码
<a href="#">drawGraphic</a>	打印位图

<a href="#">drawGraphic</a>	区域中打印 bitmap 位图
<a href="#">print</a>	打印

➤ 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：

lineWidth 打印线条的宽度

startX 开始点 X 坐标

startY 开始点 Y 坐标

endX 结束点 X 坐标

startY 结束点 Y 坐标

isSolidLinetrue 实线；false 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:

lineWidth          打印线条的宽度  
top\_left\_x          框体左上角点 X 坐标  
top\_left\_y          框体左上角点 Y 坐标  
bottom\_right\_x      框体右下角点 X 坐标  
bottom\_right\_y      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:

text\_x                  打印起始点 X 坐标  
text\_y                  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： `public void 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 `PAlign.START`；居中 `PAlign.CENTER`；居下 `PAlign.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：  
 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, PAlign xAlign, PAlign 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

lel                             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 bmp 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 bmp 位图

参数说明：

<code>area_start_x</code>	矩形区域的左上角 X 点坐标
<code>area_start_y</code>	矩形区域的左上角 Y 点坐标
<code>area_end_x</code>	矩形区域的右下角 X 点坐标
<code>area_end_y</code>	矩形区域的右下角 Y 点坐标
<code>xAlign</code>	水平位置，可以是居左 <code>PAlign.START</code> ，居中 <code>PAlign.CENTER</code> ，居右 <code>PAlign.END</code>
<code>yAlign</code>	竖直位置，参数如上
<code>bmp_size_x</code>	位图宽度
<code>bmp_size_y</code>	位图高度
<code>bmp</code>	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 TSPL 指令标签打印相关方法

注：以下方法适用于标签打印机，打印机需支持标准 TSPL 指令。

方法	描述
----	----

<a href="#">pageSetupTSPL</a>	设置页宽和页高
<a href="#">drawLineTSPL</a>	画线
<a href="#">drawBorderTSPL</a>	打印矩形框体
<a href="#">drawTextTSPL</a>	打印文字
<a href="#">drawTextTSPL</a>	区域中打印文字
<a href="#">drawBarCodeTSPL</a>	打印一维码
<a href="#">draw2DBarCodeTSPL</a>	打印二维码（TL21 机型不支持 DMATRIX 指令）
<a href="#">drawBitmapTSPL</a>	打印位图
<a href="#">drawBitmapTSPL</a>	区域中打印图片
<a href="#">printTSPL</a>	打印影像缓冲去内容
<a href="#">getPrinterStatusTSPL</a>	查询打印机状态
<a href="#">setCharsetNameTSPL</a>	设置字符集编码格式
<a href="#">setPaperbackOrPaperFeedTSPL</a>	控制进纸或退纸
<a href="#">reverseAreaTSPL</a>	指定区域内反相打印
<a href="#">eraseAreaTSPL</a>	清除影像缓冲区数据
<a href="#">setPrinterTSPL</a>	设置打印机相关功能（TL21 打印机不支持 SPEED, BLINE, SHIFT, REPRINT 指令, TL51 不支持 PEEL 指令）
<a href="#">openCashBoxTSPL</a>	开钱箱
<a href="#">getPrinterNameTSPL</a>	查询打印机型号
<a href="#">setGAPTSPL</a>	设置标签间垂直间距
<a href="#">selectCodePageTSPL</a>	选择字符代码页
<a href="#">selectCountryTSPL</a>	选择国际字符集（TL21 机型不支持该方法）
<a href="#">beepTSPL</a>	控制蜂鸣器响一声
<a href="#">downloadBitmap2PrinterTSPL</a>	下载位图到打印机（TL21 机型不支持该方法）

<a href="#">putBitmapTSPL</a>	将已下载位图放置的打印缓冲区中 (TL21 机型不支持该方法)
<a href="#">printSelfTestTSPL</a>	打印机打印自检页 (TL21 机型不支持该方法)
<a href="#">setLabelReferenceTSPL</a>	设置标签内容的参考坐标原点
<a href="#">sendStrToPrinterTSPL</a>	向打印机发送指令格式的字符串 (TL21 机型不支持 SET COUNTER Commands 指令)

以下所有方法出口均采用抛异常的方式。主要异常如下：

写异常 WriteException

读异常 ReadException

PrinterPort 为空异常 PrinterPortNullException

参数异常 ParameterErrorException

#### ➤ 设置页宽和页高

方法说明：public void pageSetupTSPL(int paperSizeType, int pageWith, int pageHeight)

throws WriteException ,PrinterPorNullException, ParameterErrorException

方法描述：设置页宽和页高,用户根据自己纸张是 2 寸，3 寸，4 寸给 paperSizeType 传入指定值，代码中根据 paperSizeType 大小维护了一个静态变量，当打印线条，框体需判断传入的坐标和 paperSizeType 比较以确定是否超过纸宽。

参数描述：

paperSizeType 纸张大小类型

0 表示选择 2 寸打印纸(20mm~56mm)

1 表示选择 3 寸打印纸(20mm~80mm)

2 表示选择 4 寸打印纸(20mm~108mm)

**pageWidth** 页宽，即要打印内容区域宽度（dot），超出区域宽度将打印出错

**pageHeight** 页高，即要打印内容区域高度（dot），超出区域高度将打印出错；

返回值：无

#### ➤ 画线

**public void drawLineTSPL(int startX, int startY, int lineLength, int lineWidth)**  
throws WriteException, PrinterPortNullException, PrinterPortNullException, Exception

方法描述：从指定坐标画线

参数描述：

**startX** 左上角水平方向起点，以点（dot）表示(不可超过标签宽度)

**startY** 左上角垂直方向起点，以点（dot）表示(不可超过标签高度)

**lineHeight** 线高，以点（dot）表示

**lineWidth** 线宽，以点（dot）表示

返回值：无

#### ➤ 打印矩形框体

**public void drawBorderTSPL(int lineWidth, int top\_left\_x, int top\_left\_y, int bottom\_right\_x, int bottom\_right\_y)**  
throws WriteException, PrinterPortNullException, ParameterErrorException

方法描述：打印框体

参数描述：

**lineWidth** 矩形线宽，以点（dot）表示

**top\_left\_x** 矩形水平方向左上角 x 起始位置以点（dot）表示

**top\_left\_y** 矩形垂直方向左上角 y 起始位置以点（dot）表示

**bottom\_right\_x** 矩形水平方向右下角 x 结束位置以点（dot）表示

**bottom\_right\_y** 矩形垂直方向右下角 y 结束位置以点（dot）表示

返回值：无

#### ➤ 打印文字

```
public void drawTextTSPL(int start_x, int start_y, boolean isSimplifiedChinese,
    int xMultiplication, int yMultiplication, PRotate rotate, String content)
throws WriteException, PrinterPortNullException, ParameterErrorException
```

方法描述：打印文字

参数描述：

start\_x            文字 X 方向起始点坐标(dot)

start\_y            文字 Y 方向起始点坐标(dot)

isSimplifiedChinese            true 简体中文 24×24Font(GB 码);false 繁体中文 24×24Font(大五码), 注意英文和数字都是 12\*24。isSimplifiedChinese 为 false 时 content 中的中文必须是繁体中文同时设置字符集为 BIG5，否则打印出的中文乱码

xMultiplication    文字宽度的放大倍率，范围 1~4

yMultiplication    文字高度的放大倍率，范围 1~4

注：使用 xMultiplication，yMultiplication 可以设定字体大小

rotate            顺时针方向 旋转，PRotate.Rotate\_0 不旋转；

PRotate.Rotate\_90 旋转 90 度；PRotate.Rotate\_180 旋转 180 度；

PRotate.Rotate\_270 旋转 270 度

content    要打印的文本内容

#### ➤ 区域中打印文字

```
public void drawTextTSPL(int area_start_x, int area_start_y, int area_end_x, int
    area_end_y, PAlign xAlign, PAlign yAlign, boolean isSimplifiedChinese, int
    xMultiplication, int yMultiplication, PRotate rotate, String content)
throws WriteException, PrinterPortNullException,
ParameterErrorException
```

方法描述：区域中打印文字,可以在区域中指定水平方向和竖直方向的位置。

水平方向：居左，居中，居右；竖直方向：居上，居中，居下；

参数描述：

lineWidth            矩形线宽(dot)

top\_left\_x            区域的左上角 X 坐标(dot)

top\_left\_y            区域的左上角 Y 坐标(dot)

bottom\_right\_x    区域的左上角 Y 坐标(dot)

bottom\_right\_y 区域的右下角 Y 坐标(dot)

xAlign X 轴方向位置设置, 居左, PAlign.START;居中, PAlign.CENTER;居右, PAlign.END

yAlign Y 轴方向位置设置, 居上, PAlign.START;居中, PAlign.CENTER;居下, PAlign.END

注: PAlign 为自定义枚举类型。包括: START, CENTER, END, NONE。为 NONE 时候应以实际传入的坐标打印, 不计算处理。下同。

isSimplifiedChinese true 简体中文 24×24Font(GB 码);false 繁体中文

24×24Font, isSimplifiedChinese 为 false 时 content 中的中文必须是繁体中文同时设置字符集为 BIG5, 否则打印出的中文乱码

xMultiplication 文字宽度的放大倍率, 范围 1~4

yMultiplication 文字高度的放大倍率, 范围 1~4

注: 使用 xMultiplication, yMultiplication 可以设定字体大小

rotate 顺时针方向 旋转, PRotate.Rotate\_0 不旋转;

PRotate.Rotate\_90 旋转 90 度; PRotate.Rotate\_180 旋转 180 度;

PRotate.Rotate\_270 旋转 270 度

content 要打印的文本内容

#### ➤ 打印一维码

```
public void drawBarCodeTSPL(int start_x, int start_y, PBarcodeType type,
int height, boolean isReadable, int narrowWidth ,
int wideWidth, PRotate rotate, String content)
```

throws WriteException, PrinterPortNullException, ParameterErrorException

方法描述: 打印一维码

参数描述:

start\_x X 方向起始点坐标(dot)

start\_y Y 方向起始点坐标(dot)

type 条码的类型, 支持的条码类型如下:

PPBarcodeType. 128 128 条码

PPBarcodeType. EAN128 EAN128

PPBarcodeType. 39 39 条码

PPBarcodeType. 93	93 条码
PPBarcodeType. EAN13	EAN13 条码
PPBarcodeType. EAN8	EAN8 条码
PPBarcodeType. CODA	CODA 条码
PPBarcodeType. UPCA	UPCA 条码
PPBarcodeType. UPCE5	UPCE+5 条码
height	条形码高度，以点（dot）表示
isReadable	0，人眼不可识；1，人眼可识
narrowWidth	条码窄条宽 [0~5]
wideWidth	条码宽条宽 $0 \leq \text{wideWidth}/(\text{narrowWidth}+1) < 5$ 或 $20 \leq \text{wideWidth}/(\text{narrowWidth}+1) \leq 30$

注：通过 narrowWidth 和 wideWidth 可以设置条码窄条和宽条宽度,从而可以适当扩大条码宽度。

rotate 顺时针方向 旋转，PRotate.Rotate\_0 不旋转；  
 PRotate.Rotate\_90 旋转 90 度；PRotate.Rotate\_180 旋转 180 度；  
 PRotate.Rotate\_270 旋转 270 度  
 content 条码内容

返回值：无

#### ➤ 打印二维码

```
public void draw2DBarcodeTSPL(int start_x, int start_y, TwoDarCodeType
type , int param1, int param2, PRotate rotate, String content)
```

throws WriteException, PrinterPortNullException, ParameterErrorException

方法描述：打印二维码

参数描述：

start_x	X 方向起始点坐标(dot)
start_y	Y 方向起始点坐标(dot)
type	二维码条码类型（QRCode, DataMatrix）
TwoDarCodeType.QR	QRcode
TwoDarCodeType. DataMatrix	DataMatrix

**param1**            当条码类型是 DataMatrix，此参数无作用；当条码类型为 QRCode,此参数表示 QR 码的纠错等级，0 表示 L 7%，1 表示 M 15%，2 表示 Q 25%，3 表示 H 30%

**param2**            参数表示二维码方块宽度（1~6）

**rotate**            条码的旋转角度，顺时针方向。0，不旋转；90，顺时针方向旋转 90 度；180，顺时针方向线旋转 180 度；270，顺时针方向线旋转 180 度。

注：当条码类型是：DataMatrix，此参数不起作用，随意传入

**content**            二维码的内容

返回值：无

#### ➤ 打印位图

`public void drawBitmapTSPL(int start_x, int start_y, int mode, Bitmap bmp)`

throws WriteException, PrinterPortNullException, ParameterErrorException

方法描述：打印位图

参数描述：

**start\_x**            点阵影像的水平起始位置(dot)

**start\_y**            点阵影像的垂直起始位置(dot)

**mode**              影像绘制模式 0 OVERWRITE ; 1 OR ; 2 XOR

**bmp**                位图对象(单色 bitmap)

返回值:无

#### ➤ 区域内打印图片

`public void drawBitmapTSPL(int area_start_x, int area_start_y, int area_end_x,`

`int area_end_y, PAlign xAlign, PAlign yAlign, int mode, Bitmap bmp)`

throws WriteException, PrinterPortNullException, ParameterErrorException

方法描述：在矩形区域内打印文字，可以在水平位置指定居中，居右，居中；竖直位置指定文字居上，居中，居下。

参数描述：

**area\_start\_x**            区域左上角 x 坐标(dot)

**area\_start\_y**            区域左上角 Y 坐标(dot)

**area\_end\_x**            区域的右下角 X 坐标(dot)

area_end_y	区域的右下角 y 坐标(dot)
xAlign	水平方向的位置
yAlign	竖直方向的位置
mode	影像绘制模式 0 OVERWRITE;1 OR; 2 XOR
bmp	要打印的 bitmap(单色 bitmap)

返回值:无

➤ 打印影像缓冲区内容

```
public void printTSPL(int mSets, int mCopys)
```

throws WriteException, PrinterPortNullException,

ParameterErrorException

方法描述：打印影像缓冲区命令，调用打印线条，打印文字，条码等，打印机会把打印内容缓冲到打印机中，调用本方法会将缓冲区数据全部打印出来,通过参数 mSets 指定打印多少份，通过 mCopys 指定每一份中每一页标签打印多少份。

参数描述：

mSets            打印的份数

mCopys    每一份中每一标签页打印多少份

返回值： 无

➤ 查询打印机状态

```
public int getPrinterStatusTSPL()
```

throws WriteException, PrinterPortNullException, ReadExcetion,

ParameterErrorException

方法描述：查询打印机状态，包括缺纸，开盖，及其他影响正常打印错误状态。

参数:无

返回值：

0 状态正常

-1 缺纸

-2 开盖

-3 其他错误

## ➤ 设置字符集编码格式

```
public void setCharsetNameTSPL(String charsetName)
```

throws `ParameterErrorException`

方法描述：设置字符集编码格式,注：此方法适用于 ESC/POS,CPCL,TSPL 指令集。

参数：

`charsetName` 字符集名；简体中文，“GBK”；繁体中文，“BIG5”

注：当使用 `drawTextTSPL` 方法打印繁体中文，一定先调用此方法将字符集设置为“BIG5”

返回值：无

## ➤ 控制进纸或退纸

```
public void setPaperbackOrPaperFeedTSPL(boolean isFeedBack, int mDot)
```

throws `WriteException`, `PrinterPortNullException`, `ParameterErrorException`

方法描述：控制进纸或退纸

参数：

`isFeedBack` 是否退纸 `true` 退纸；`false` 进一张纸

`mDot` 退纸点数；当选择进纸时，此参数无效

返回值：无

## ➤ 指定区域内反相打印

```
public void reverseAreaTSPL(int start_x, int start_y, int width, int heigth)
```

throws `WriteException`, `PrinterPortNullException`, `ParameterErrorException`

方法描述：指定区域内反相打印

参数：

`start_x` 区域左上角 X 坐标（dot）

`start_y` 区域内左上角 Y 坐标（dot）

`width` 区域的宽度（dot）

`height` 区域的高度（dot）

返回值：无

## ➤ 清除影像缓冲区数据

```
public void eraseAreaTSPL(int start_x, int start_y, int width, int heigth)
```

throws WriteException, PrinterPortNullException, ParameterErrorException

方法描述：清除此影像缓冲区数据

参数：

start\_x 区域左上角 X 坐标 (dot)

start\_y 区域内左上角 Y 坐标 (dot)

width 区域的宽度 (dot)

height 区域的高度 (dot)

返回值：无

#### ➤ 设置打印机相关功能

public void setPrinterTSPL(CommandTSPL command,int value)

throws WriteException, PrinterPortNullException,

ParameterErrorException

方法描述：设置打印机相关功能

参数：

command

CommandTSPL.DIRECTION 设置打印方向； value 为 0 方向和字体相同，  
1 相反

CommandTSPL.FEED 控制进纸设置；  $1 \leq \text{value} \leq 1000$  表示点数

CommandTSPL.REPRINT 启用错误重新打印命令； value 为 0 关闭此功能；  
1 开启此功能

CommandTSPL.SPEED 指定打印速度； value 为每秒走纸英寸数

CommandTSPL.DENSITY 指定打印浓度；  $0 \leq \text{value} \leq 15$ , 数字越大浓度越  
大

CommandTSPL.SHIFT 走纸偏移 value 点； value > 0, 走纸方向和打印方向  
相同； < 0 相反；

CommandTSPL.FORMFEED 控制打印机进一张纸； value 值随意传入

CommandTSPL.HOME 寻起始位置，使缝隙对准撕纸片； value 值随意  
传入

CommandTSPL.PRINTKEY 设定按键打印命令(禁止，启用，自动设定  
按键打印功能)

`CommandTSPL . KEY1` 启用按键 KEY1 预设功能命令(此按键功能是暂停还是进纸)

`CommandTSPL . KEY2` 启用按键 KEY1 预设功能命令(此按键功能是暂停还是进纸)

`CommandTSPL . TEAR` 启用或关闭走到撕纸位置命令(缝隙对准撕纸位)

`CommandTSPL.PEEL` 启用剥离模式

`Value` 具体含义与 `command` 对应

返回值：无

➤ 开钱箱

`public void openCashBoxTSPL()`

throws `WriteException`, `PrinterPortNullException`,  
`ParameterErrorException`

方法描述：开钱箱

参数：无

返回值：无

➤ 查询打印机型号

`public String getPrinterNameTSPL()`

throws `WriteException`, `PrinterPortNullException`,  
`ParameterErrorException`, `ReadException`

方法描述：查询打印机型号

参数：无

返回值：打印机具体型号

➤ 设置标签纸间垂直间距

`public void setGAPTSPL(int value)`

throws `WriteException`, `PrinterPortNullException`,  
`ParameterErrorException`

方法描述：设置标签纸间垂直间距

参数：

`value` 标签纸间垂直间距（单位：mm），mm

$(0 \leq m \leq 1(\text{inch}) , 0 \leq m \leq 25.4(\text{mm})$

返回值：无

## ➤ 选择字符代码页

`public void selectCodePageTSPL(int value)`  
 throws `WriteException`, `PrinterPortNullException`,  
`ParameterErrorException`

方法描述：选择字符代码页

参数：

Value 值	437: United States
	850: Multilingual
	852: Slavic
	860: Portuguese
	863: Canadian/French
	865: Nordic
	857: Turkish
	1250: Central Europe
	1252: Latin I
	1253: Greek
	1254: Turkish

传入其他值：默认为 437

返回值：无

## ➤ 选择国际字符集

`public void selectCountryTSPL(int value)`  
 throws `WriteException`, `PrinterPortNullException`,  
`ParameterErrorException`

方法描述：选择字符代码页

参数：

Value	1: USA
	2: Canadian-French
	3: Spanish (Latin America)
	33: French (France)
	34: Spanish (Spain)
	39: Italian
	42: Slovak

44: United Kingdom  
 45: Danish  
 46: Swedish  
 47: Norwegian  
 49: German  
 61: English (International)

传入其他值：默认为 1

返回值：无

➤ 控制蜂鸣器响一声

`public void beepTSPL(int level,int interval)`

throws `WriteException`, `PrinterPortNullException`, `ParameterErrorException`

方法描述：设置蜂鸣器响的级别和间隔，设置完成后需要发送 BEEP，蜂鸣器才能响

参数：

Level                      蜂鸣器响的级别

Interval                    蜂鸣器响的时间

返回值：无

➤ 下载位图到打印机

`public void downloadBitmap2PrinterTSPL(boolean isMove2Flash, InputStream in, String fileName)`

throws `WriteException`, `PrinterPortNullException`,  
`ParameterErrorException`

方法描述：下载 bitmap 到打印机中

参数：

isMove2Flash false 下载 bitmap 到打印机内存中（内存中数据掉电会丢失）;true 下载到 Flash（使用 MOVE 指令进 RAM 中数据搬移到 FLASH 中）。

in 要下载的 raw 目录下的 bitmap 对应的输入流。

filename 下载到内存或者打印机 FLASH 中，并指定其名称，文件名称最长是 8 个字符。

返回值：无

注：位图数据取点按照 BITMAP 指令规则取点。

- 将已下载位图放置到打印缓冲区中

`public void putBitmapTSPL(String fileName, int start_x, int start_y)`

throws WriteException, PrinterPortNullException, ParameterErrorException

方法描述：将已经下载到打印机内存或者 FLASH 中的位图数据放在打印缓冲区坐标（start\_x,int start\_y）处

参数：

filename 已经下载到打印机内存中的位图名称，必须和下载到打印机内存中或 FLASH 中 Bitmap 名称一样，否则可能打印不出来。即

downloadBitmap2Printer 方法将位图下载到打印机中，putBitmap 将此位图放入打印缓冲区中。

start\_x 位图数据水平 X 坐标

start\_y 位图数据水平 Y 坐标

返回值：无

- 打印机打印自检页

`public void printSelfTestTSPL()`

throws WriteException, PrinterPortNullException,

ParameterErrorException

方法描述：打印机发送指令打印自检页

参数：无

返回值：无

- 设置标签内容的参考坐标原点

`public void setLabelReferenceTSPL (int left, int top)`

throws PrinterPortNullException,ParameterErrorException

方法描述：设置标签的参考坐标原点，调整标签内容在标签纸上的位置

参数：left 标签内容距离标签纸的左边距（mm）

top 标签内容距离标签纸的上边距（mm）

返回值：无

- 向打印机发送指令格式的字符串

方法说明：`public void sendStrToPrinterTSPL(String str)`

throws PrinterPorNullException,WriteException,  
ParameterErrorException

方法描述：向打印机直接发送为封装到 SDK 接口中的 TSPL 指令。

参数说明：str 符合 TSPL 指令格式的字符串

返回值：无

### 3.1.7 控制打印机硬件方法

方法	描述
<a href="#">cutPaper</a>	切纸
<a href="#">ringBuzzer</a>	控制蜂鸣器响
<a href="#">blackLableFind</a>	黑标定位

#### ➤ 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

#### ➤ 黑标定位

方法说明：public void blackLableFind()

方法描述：打印机走纸到黑标位置（打印机需支持黑标定位功能，即有黑标传感器。同时只有在黑标模式下才能实现黑标定位功能，如果找不到黑标，则打印机走纸 30 才 cm 停止）（如何开启打印机黑标定位功能，请联系我司售前技术，索取设置工具，设置相关功能使能即可。）

参数说明：无

返回值：无

### 3.1.8 查询打印机状态相关方法

方法	描述
<a href="#">getCurrentStatus</a>	获取打印机当前状态
<a href="#">getPrintingStatus</a>	打印机打印是否完成
<a href="#">isPaperOut</a>	判断打印机是否缺纸
<a href="#">isPaperWillOut</a>	判断打印机是否纸将尽

#### ➤ 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。（如何开启打印机纸将尽功能，请联系我司售前技术，索取设置工具，设置相关功能使能即可。）

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(ms)

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

**原理说明：**打印机单片机 CPU 执行数据是顺序执行，先发送打印数据，在发送查询打印是否完成的指令，按照“后进后执行数据”的原则，执行到打印完成判断指令，一定执行完打印的数据。

### 例 1：查询打印是否完成

//1、先判断打印机当前状态，状态正常方可发送数据打印

```
if(getCurrentStatus() != 0){
    return ;
}
```

//2、发送打印数据

```
mPrinter.printerText("hello world!");
StringBuffer str = new StringBuffer("");
int timeout = 2000;//预计打印上述文字的时间
```

//3、判断打印是否完成

```
int ret = mPrinter.getPrintingStatus(str, timeout);
if (ret == 0) {
    Log.i("sprt", "text print finished!");
}
```

➤ 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

### 3.1.9 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
```

上述以蓝牙通信方式调用为例，其他通信方式的连接调用请参见"PrintDemoV5.2"。

3.2、Barcode 条码打印类

<a href="#">构造函数</a>
<a href="#">条码打印调用示例</a>

3.2.1 构造函数

方法	描述
<a href="#">Barcode</a>	创建 Barcode 实例

➤ 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

[创建 Table 类实例](#)

[添加一行数据](#)

[设置 Table 中列对齐方式](#)

[表格打印调用示例](#)

#### 3.3.1 创建 Table 类实例

方法	描述
<a href="#">Table</a>	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

方法	描述
<a href="#">addRow</a>	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 中列对齐方式

方法	描述
<a href="#">setColumnAlignLeft</a>	设置 Table 中列对齐方式

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：更详尽的表格打印调用请参见 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：

<a href="#">初始化画布</a>
<a href="#">设置字体属性</a>
<a href="#">画布内容相关方法</a>
<a href="#">画布打印调用示例</a>

### 3.4.1 初始化画布

方法	描述
<a href="#"><u>init</u></a>	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	描述
<a href="#">setFontProperty</a>	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）
-------	--------------

bItalic                      ture 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：

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

返回值：无

### 3.4.3 画布内容相关方法

方法	描述
<a href="#"><code>drawText</code></a>	画字符串
<a href="#"><code>drawLine</code></a>	绘制直线
<a href="#"><code>drawRectangle</code></a>	绘制矩形
<a href="#"><code>drawEllips</code></a>	绘制椭圆
<a href="#"><code>drawImage</code></a>	绘制图片
<a href="#"><code>getCanvasImage</code></a>	获得画布上绘制的图像，发送给打印机打印
<a href="#"><code>setTextAlignRight</code></a>	设置文本是否靠右
<a href="#"><code>setTextExceedNewLine</code></a>	文本超出是否换行
<a href="#"><code>setUseSplit</code></a>	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。
<a href="#"><code>setUseSplitAndString</code></a>	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。用指定符号拆分。

#### ➤ 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: `public 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

[彩色图转换为黑白二值 bitmap](#)

#### 3.5.1 彩色图转换为黑白二值 bitmap

方法	描述
<a href="#">convertBitmap</a>	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(Bitmap bitmap)` 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 `MuPDFCore`，此类将 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 PDF 打印调用示例

1、在代码工程中添加处理 PDF 文件 so 库，即：`libmupdf.so`

2、代码处理流程

//1、实例化 PDF 文件处理核心类

```
MuPDFCore core = new MuPDFCore(this, filePath);
```

//2、获取 PDF 文件页数

```
int count = core.countPages();
```

//3、获得 PDF 页的宽高

```
PointF pageSize = core.getPageSize(0); //当前获得第 1 页 PDF 文件的宽高
```

```
float pageW = pageSize.x;
```

```
float pageH = pageSize.y;
```

//4、PDF 文件转 `bitmap`

```
Bitmap bitmap = Bitmap.createBitmap((int)pageW, (int)pageH, Bitmap.Config.ARGB_8888);
```

```
core.drawPage(0, bitmap, (int) pageW, (int) pageH, 0, 0, (int)pageW, (int)pageH);
```

//5、缩放上述 PDF 文件

```
Bitmap zoomImage = Utils.zoomImage(bitmap, 500); //按照宽度 500 像素比例缩放
```

//6、`bitmap` 转单色 `bitmap`，当 `bitmap` 比较大时此操作处理稍慢，可在线程中调用处理

//由于缩放和转单色位图原因，会有一定程度的失真。

```
Bitmap monoChromeBitmap = convertor.convertBitmap(zoomImage);
```

//7、打印 `Bitmap`

```
mPrinter.printImage(monoChromeBitmap, PAlign.START, 0, false);
```

注：更详尽的 PDF 打印调用请参见 `PdfActivity.java` 调用

### 3.7、CodePagePrinter 多国语言代码页打印类

此类中定义了多国语言文字打印示例，支持了世界上的部分语言，需要提前知道将要打印语种的代码页，然后在调用指定代码页的方法即可打印此语言文字。部分打印机可能不支持想要的代码页，需联系我司售前和销售定制。

注：繁体中文，可以直接调用 `printText` 方法打印，如：`mPrinter.printText("打印测试!\n");`

方法名	描述
<code>CodePagePrinter(PrinterInstance mprinter)</code>	此类的构造方法
<code>printTextInCP437</code>	打印 CP437 代码页文字(美国，欧洲标准)
<code>printTextInCP850</code>	打印多语言（Multilingual）
<code>printTextInCP932</code>	打印 CP932 代码页文字（Katakana）文字
<code>printTextInCP860</code>	打印 CP860 代码页葡萄牙语文字
<code>printTextInCP863</code>	打印 CP863 代码页加拿大法语文字
<code>printTextInCP865</code>	打印 CP863 代码页北欧文字
<code>printTextInWCP1251</code>	打印 WCP1251 代码页斯拉夫语文字
<code>printTextInMIK</code>	打印 MIK 代码页斯拉夫语/保加利亚语文字
<code>printTextInCP862</code>	打印 CP862 代码页希伯来语文字
<code>printTextInWCP1252</code>	打印 WCP1252 代码页，拉丁语 1 文字
<code>printTextInWCP1253</code>	打印 WCP1252 代码页希腊语文字
<code>printTextInCP852</code>	打印 CP852 代码页，拉丁语 2 文字
<code>printTextInCP858</code>	打印 CP858 多种语言拉丁语 1+欧元符文字
<code>printTextInCP720</code>	打印 CP720 代码页阿拉伯语文字
<code>printTextInCP864</code>	打印 CP864 代码页阿拉伯语文字
<code>printTextInISO_8859_1</code>	打印 ISO-8859-1 代码页西欧语文字
<code>printTextInCP737</code>	打印 CP737 代码页希腊语文字
<code>printTextInWCP1257</code>	打印 CP737 代码页波罗的海语文字

printTextInCP855	打印 CP855 代码页斯拉夫语文字
printTextInCP857	打印 CP857 土耳其语文字
printTextInWCP1250	打印 WCP1250 代码页中欧语文字
printTextInCP775	打印 CP775 代码页文字
printTextInWCP1254	打印 WCP1254 代码页土耳其语文字
printTextInWCP1255	打印 WCP1255 代码页希伯来语文字
printTextInWCP1256	打印 WCP1256 代码页阿拉伯语文字
printTextInWCP1258	打印 WCP1258 代码页越南语文字
printTextInISO_8859_2	打印 ISO_8859_2 代码页拉丁语 2 文字
printTextInISO_8859_3	打印 ISO_8859_2 代码页拉丁语 3 文字
printTextInISO_8859_4	打印 ISO_8859_4 代码页波罗的语文字
printTextInISO_8859_5	打印 ISO_8859_5 代码页斯拉夫语文字
printTextInISO_8859_6	打印 ISO_8859_6 代码页阿拉伯语文字
printTextInISO_8859_7	打印 ISO_8859_7 代码页希腊语文字
printTextInISO_8859_8	打印 ISO_8859_8 代码页希伯来语文字
printTextInISO_8859_9	打印 ISO_8859_9 代码页土耳其语文字
printTextInISO_8859_15	打印 ISO_8859_15 代码页拉丁语 9 文字
printTextInCP874	打印 CP874 代码页文字

## 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			