**Android-SDK 打印机开发文档**

(V1.6)

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，孙超 | | | |
| 文件名称 | | Android-SDK打印机开发文档.doc | | | |
| 摘要 | | |  | | --- | |  |   本文档属于Android-SDK5.6配套文档，该版本在SDK5.5版本基础上进行修改。 | | | |
| 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 |  | 1、修改SDK中PDF打印  2、优化在非root设备上打开串口崩溃问题  3、添加了打印多国语言说明  4、优化打印图片方法  5、SDK中添加了打印直线，虚线方法  6、优化其他已知bug  7、文档中添加了快速阅读指南小节 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1.3 | 2017-12-14 | 杨学志 |  | |  | | --- | | 1、优化了查询打印机状态的方法。调整查询逻辑顺序，保证查询的稳定性，同时细化异常返回为读异常和写异常。  2.优化了查询打印机打印是否完成的方法。去掉显示查询结果信息的参数。  3.重载esc/pos指令打印图片的方法。该方法可直接传递彩色Bitmap对象，并且可以修改图片二值化的阀值。  4.修改TSPL指令下载位图到打印机的方法。 使能正常下载位图到打印机缓存。  5.修改USB通信方式读写数据的方法，提升打印机通信速率。修改发送的单包数据大小和单包数据间的延时。同时修改读数据的返回值保证读取到的返回值是确保已发送成功的字节数。  6.修改TSPL指令打印位图的参数有限制的bug。保证位图可在任意合法位置打印。7.规范了修改过的接口的调制日志。 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1.4 | 2018-12-07 | 孙超  杨学志 |  | 1、添加页模式接口打印功能，[详情请查看3.8](#CanvasDrawPrint页模式接口打印类)  2、优化在设备上用USB超时丢数据的问题。  3、添加打印LOG保存到SD卡，[详情请查看3.9](#_3.9、记录Log保存到SD卡)  4.添加Y33采集打印功能，[详情请查看3.10](#_3.10、Y33采集打印)  5.修改串口的实例化方式，添加一个新的参数ctsrts选择流控方式。  6.添加设置Y33打印服务是否可配置的功能、获取Y33打印服务是否可配置状态的功能，[详情请查看3.1.2](#isSetAble) |
| 1.5 | 2019-9-5 | 孙超  杨学志 |  | 1、添加软流控配置，详情请查看[获得串口打印机实体类对象](#获取串口打印机实体类对象) |
| 1.6 | 2020-04-16 | 孙超  杨学志 |  | 1. zxing包名冲突，多个用户反馈冲突，更换为com.printer.sdk.  zxing包名来解决相关问题。  2. 打印图片转单色图的问题，缺少 或者 多出问题。  3.新增[查询打印机走纸距离的接口](#getFeedDistance)和[重置打印机走纸距离的接口](#resetFeedDistance)。 |

目录

[1、简介 1](#_Toc2673724)

[2、名词解释 1](#_Toc2673725)

[2.1名词解释 1](#_Toc2673726)

[2.2相关知识 1](#_Toc2673727)

[2.3本文档快速阅读指南 2](#_Toc2673728)

[3、相关类说明 3](#_Toc2673729)

[3.1、PrinterInstance 打印核心类 3](#_Toc2673730)

[3.1.1 获得打印机实体类相关方法 3](#_Toc2673731)

[3.1.2 打印机通信相关方法 5](#_Toc2673732)

[3.1.3 打印机设置相关方法 7](#_Toc2673733)

[3.1.4 ESC/POS指令打印相关方法 9](#_Toc2673734)

[3.1.5 CPCL指令标签打印相关方法 13](#_Toc2673735)

[3.1.6 TSPL指令标签打印相关方法 19](#_Toc2673736)

[3.1.7 控制打印机硬件方法 33](#_Toc2673737)

[3.1.8 查询打印机状态相关方法 35](#_Toc2673738)

[3.1.9 连接打印调用示例 37](#_Toc2673739)

[3.2、Barcode条码打印类 38](#_Toc2673740)

[**3.2.1 构造函数** 38](#_Toc2673741)

[**3.2.2 条码打印调用示例** 39](#_Toc2673742)

[3.3、Table表格打印类 39](#_Toc2673743)

[**3.3.1 创建Table类实例** 40](#_Toc2673744)

[**3.3.2 添加一行数据** 40](#_Toc2673745)

[**3.3.3 设置Table中列对齐方式** 40](#_Toc2673746)

[**3.3.4 表格打印调用示例** 41](#_Toc2673747)

[3.4、CanvasPrint画布打印类 41](#_Toc2673748)

[**3.4.1 初始化画布** 41](#_Toc2673749)

[**3.4.2 设置字体属性** 42](#_Toc2673750)

[**3.4.3 画布内容相关方法** 43](#_Toc2673751)

[**3.4.4 画布打印调用示例** 44](#_Toc2673752)

[3.5、BitmapConvertor类 45](#_Toc2673753)

[**3.5.1彩色图转换为黑白二值bitmap** 45](#_Toc2673754)

[3.6、PDF文件打印相关类 46](#_Toc2673755)

[**3.6.1 PDF打印调用示例** 46](#_Toc2673756)

[3.7、CodePagePrinter多国语言代码页打印类 46](#_Toc2673757)

[3.8、CanvasDrawPrint页模式接口打印类 48](#_Toc2673758)

[**3.8.1 初始化页面** 49](#_Toc2673759)

[**3.8.2页面内容相关方法** 49](#_Toc2673760)

[**3.8.3页模式接口相关例子实现** 55](#_Toc2673761)

[3.9、记录Log保存到SD卡 59](#_Toc2673762)

[**3.9.1例子实现** 60](#_Toc2673763)

[3.10、Y33采集打印 60](#_Toc2673764)

[4、附录 64](#_Toc2673765)

[**4.1、CODE128码综述** 64](#_Toc2673766)

[**4.2、字符集** 65](#_Toc2673767)

1、简介

Android-SDK5.5是基于我司打印机开发的一套 Android 函数接口集合。此版本在 SDK5.4基础上添加软流控配置。 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 以上版本。 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](#printerInstance) | 打印机核心类 |
| [Barcode](#Barcode打印类) | 条码打印类 |
| [Table](#Table) | 表格打印类 |
| [CanvasPrint](#CanvasPrint) | 画布打印类 |
| [BitmapConvertor](#BitmapConvertor) | 彩色Bitmap转换类 |
| [PdfContext,CodecDocument,CodecPage](#PDF) | Pdf文件打印相关类 |
| PrinterConstants | 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](#_3.1.1_获得打印机实体类相关方法)以及[3.1.2](#_3.1.2_打印机通信相关方法)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](#_3.1.9_连接打印调用示例)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](#_3.1.4_ESC/POS指令打印相关方法)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](#_3.1.5_CPCL指令标签打印相关方法)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](#_3.1.6_TSPL指令标签打印相关方法)小节。此小节提供了与[3.1.5](#_3.1.5_CPCL指令标签打印相关方法)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](#_3.1.8_查询打印机状态相关方法)小节。另外[3.6](#_3.6、PDF文件打印相关类)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](#_3.4、CanvasPrint画布打印类)节；切纸，控制蜂鸣器，寻找黑标，请阅读[3.1.7](#_3.1.7_控制打印机硬件方法)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。

|  |
| --- |
| [获得打印机实体类相关方法](#获得打印机实体类相关方法) |
| [打印机通信相关方法](#打印机通信相关方法) |
| [打印机设置相关方法](#打印机设置相关方法) |
| [打印内容相关方法](#打印内容相关方法) |
| [CPCL指令标签打印相关方法](#CPCL指令标签打印相关方法) |
| [TSPL指令标签打印相关方法](#TSPL指令标签打印相关方法) |
| [控制打印机硬件相关方法](#控制打印机硬件方法) |
| [获取打印机状态相关方法](#获取打印机状态相关方法) |
| [连接打印调用示例](#连接打印调用示例) |

### 3.1.1 获得打印机实体类相关方法

|  |  |
| --- | --- |
| 方法名 | 描述 |
| [getPrinterInstance](#获取蓝牙打印机实体类对象) | 获得蓝牙打印机实体类对象 |
| [getPrinterInstance](#获取USB打印机实体类对象) | 获得USB打印机实体类对象 |
| [getPrinterInstance](#获取wifi打印机实体类对象) | 获得WIFI打印机实体类对象 |
| [getPrinterInstance](#获取串口打印机实体类对象) | 获得串口打印机实体类对象 |
| [getPrinterInstance](#获得AIDL实体类对象) | 获得AIDL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, int ctsrts,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

ctsrts 串口的流控配置，0代表软流控，1代表硬流控，其他数字代表无流控

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\_sp.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

* 获得AIDL实体类对象

PrinterInstance static synchronized getPrinterInstance()

参数：无

返回值：PrinterInstance 实体类对象

### 3.1.2 打印机通信相关方法

|  |  |
| --- | --- |
| 方法名 | 描述 |
| [OpenConnection](#连接打印机) | 连接打印机 |
| [closeConnection](#关闭与打印机的连接) | 关闭与打印机的连接 |
| [sendBytesData](#向打印机发送十六进制数据) | 向打印机发送十六进制数据 |
| [read](#读取打印机返回的数据) | 读取打印机返回的数据 |
| [isSetAble](#设置远程打印服务是否可配置) | 设置远程打印服务是否可配置 |
| [getSetable](#获取远程打印服务是否可配置的状态) | 获取远程打印服务是否可配置的状态 |

* 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

* 设置远程打印服务是否可配置

方法名：public void isSetAble(boolean isSetable)

方法描述：设置远程打印服务是否可配置**（该方法只能配合Y33打印服务使用，其他应用场景没有意义。）**

参数：true表示设置远程打印服务可以配置，false表示设置远程打印服务不可以配置。

返回值：无

* 获取远程打印服务是否可配置的状态

方法名：public boolean getSetable()

方法描述：获取远程打印服务是否可配置的状态**（该方法只能配合Y33打印服务使用，其他应用场景没有意义。）**

参数：无

返回值：true表示远程打印服务当前处于可以配置的状态，false表示远程打印服务当前处于不可以配置的状态。

### 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 |
| [initPrinter](#初始化打印机) | 初始化打印机 |
| [setFont](#设置打印机字体) | 设置打印机字体 |
| [setPrinter](#设置打印机打印) | 设置打印机打印 |
| [setLeftMargin](#设置距离左边距点数) | 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](#打印图片)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](#打印图片)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 |
| [printText](#打印文字) | 打印文字 |
| [printImage](#打印图片) | 打印黑白二值图片 |
| [printImage](#彩色图二值化打印) | 打印彩色图片 |
| [printColorImg2Gray](#printColorImg2Gray) | 打印灰度化的彩色图片 |
| [printTable](#打印表格) | 打印表格 |
| [printBarCode](#打印条码) | 打印条码 |
| [printhorizontalLine](#打印水平直线，虚线) | 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(“helloworld\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

⮚彩色图二值化打印  
方法名： public void printImage(Bitmap bitmap, PAlign alignType,int left， intlevel)  
方法描述：距离左边距 left 点开始打印图片，注意： 此方法是打印黑白二值  
图片的重载方法，该方法可直接传入彩色位图后进行二值化转换并打印。  
并且开放图片二值化的阀值。  
参数： bitmap 要打印的位图。  
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。  
level： 图片二值化的阀值。取值范围是 0~255，默认是 128。注意：虽然理  
论上二值化图片可以取到阀值 255，但是当阀值取到 255 或者比较接近 255  
时，图片可能转换成全黑色位图，打印会损害打印机。所以建议取值 128，  
160， 192。  
例 **1**：以下描述设置打印图片居中  
Android-SDK 打印机开发文档 V1.3  
11  
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, 128);  
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, 128);  
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](#表格打印调用示例)

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](#条码打印调用示例)

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 |
| [pageSetup](#设置要打印区域的页宽和页高) | 设置要打印区域的页宽和页高 |
| [drawLine](#打印直线) | 打印直线 |
| [drawBorder](#打印框体) | 打印框体 |
| [drawText](#打印文本text) | 打印文本 |
| [drawText](#区域中打印文字) | 区域中打印文字 |
| [drawBarCode](#打印一维条码) | 打印一维条码 |
| [drawBarCode](#区域中打印一维条码) | 区域中打印一维条码 |
| [drawQrCode](#打印二维码) | 打印二维码 |
| [drawGraphic](#打印位图) | 打印位图 |
| [drawGraphic](#区域内打印bitmap位图) | 区域中打印bitmap位图 |
| [print](#打印) | 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bitmp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bitmp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S

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 |
| [pageSetupTSPL](#设置页宽和页高TSPL) | 设置页宽和页高 |
| [drawLineTSPL](#画线TSPL) | 画线 |
| [drawBorderTSPL](#打印矩形框体TSPL) | 打印矩形框体 |
| [drawTextTSPL](#打印文字TSPL) | 打印文字 |
| [drawTextTSPL](#区域中打印文字TSPL) | 区域中打印文字 |
| [drawBarCodeTSPL](#打印一维码TSPL) | 打印一维码 |
| [draw2DBarCodeTSPL](#打印二维码TSPL) | 打印二维码（TL21机型不支持 DMATRIX指令） |
| [drawBitmapTSPL](#打印位图TSPL) | 打印位图 |
| [drawBitmapTSPL](#区域内打印图片TSPL) | 区域中打印图片 |
| [printTSPL](#打印影像缓冲区内容TSPL) | 打印影像缓冲去内容 |
| [getPrinterStatusTSPL](#查询打印机状态TSPL) | 查询打印机状态 |
| [setCharsetNameTSPL](#设置字符集编码格式TSPL) | 设置字符集编码格式 |
| [setPaperbackOrPaperFeedTSPL](#控制进纸或退纸TSPL) | 控制进纸或退纸 |
| [reverseAreaTSPL](#指定区域反相打印TSPL) | 指定区域内反相打印 |
| [eraseAreaTSPL](#清除影像缓冲区数据TSPL) | 清除影像缓冲区数据 |
| [setPrinterTSPL](#设置打印机相关功能TSPL) | 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） |
| [openCashBoxTSPL](#开钱箱TSPL) | 开钱箱 |
| [getPrinterNameTSPL](#查询打印机型号TSPL) | 查询打印机型号 |
| [setGAPTSPL](#设置标签纸间垂直间距TSPL) | 设置标签间垂直间距 |
| [selectCodePageTSPL](#选择字符代码页TSPL) | 选择字符代码页 |
| [selectCountryTSPL](#选择国际字符集TSPL) | 选择国际字符集（TL21机型不支持该方法） |
| [beepTSPL](#控制蜂鸣器响一声TSPL) | 控制蜂鸣器响一声 |
| [downloadBitmap2PrinterTSPL](#下载位图到打印机TSPL) | 下载位图到打印机（TL21机型不支持该方法） |
| [putBitmapTSPL](#将已下载位图放置到打印缓冲区中TSPL) | 将已下载位图放置的打印缓冲区中（TL21机型不支持该方法） |
| [printSelfTestTSPL](#打印机打印自检页TSPL) | 打印机打印自检页（TL21机型不支持该方法） |
| [setLabelReferenceTSPL](#设置标签内容的参考坐标原点TSPL) | 设置标签内容的参考坐标原点 |
| [sendStrToPrinterTSPL](#像打印机发送指令格式的字符串TSPL) | 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\_90 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<=wideWidth/(narrowWidth+1)<5或

20<= wideWidth/(narrowWidth+1)<=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，顺时方向线旋转180度；270，顺时方向线旋转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≤value≤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=<value<=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≦m≤1(inch）， 0≦m≤25.4(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 |
| [cutPaper](#切纸) | 切纸 |
| [ringBuzzer](#控制蜂鸣器响) | 控制蜂鸣器响 |
| [blackLableFind](#黑标定位功能) | 黑标定位 |
| [resetFeedDistance](#重置打印机走纸距离) | 重置打印机走纸距离 |

* 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

* 重置打印机走纸距离

方法说明：public void resetFeedDistance()

方法描述：重置打印机走纸距离（**特定机型支持**）

参数说明：无

返回值：0 重置成功

-1 重置失败

### 3.1.8 查询打印机状态相关方法

|  |  |
| --- | --- |
| 方法 | 描述 |
| [getCurrentStatus](#获取打印机当前状态) | 获取打印机当前状态 |
| [getPrintingStatus](#打印机打印是否完成) | 打印机打印是否完成 |
| [isPaperOut](#判断打印机是否缺纸) | 判断打印机是否缺纸 |
| [isPaperWillOut](#判断打印机是否纸将尽) | 判断打印机是否纸将尽 |
| [getFeedDistance](#获取打印机总计走纸距离) | 获取打印机总计走纸距离 |

* 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(int timeout)  
  方法描述：判断打印机打印是否完成。（注意：该功能在 T7 等部分机型上  
  不适用。要使用此功能请先联系售前技术确认打印机是否支持该功能。）  
  参数说明： timeout 此超时时间应长于预估的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!");  
  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(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

* 获取打印机总计走纸距离

方法名：public int getFeedDistance()

方法描述：查询当前打印机总计走纸距离（**特定打印机支持**）

参数说明： 无

返回值：当前打印机总计走纸距离（单位：毫米）,返回值<0表示查询异常。

### 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

BluetoothDevicemDevice = BluetoothAdapter.getDefaultAdapter().getRemoteDevice(devicesAddress);//devicesAddress为蓝牙的mac地址

PrinterInstancemPrinter = 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

|  |
| --- |
| [构造函数](#barcode构造函数) |
| [条码打印调用示例](#条码打印调用示例) |

**3.2.1 构造函数**

|  |  |
| --- | --- |
| 方法 | 描述 |
| [Barcode](#创建Barcode实例) | 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<=n<=6，默认为2

param2：条码高度 1<=n<=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<=n<=30。

param2：表示纠错等级，0<=n<=8。

param3：表示纵向放大倍数。

2. DATAMATRIX

param1：表示图形高，0<=n<=144(0:自动选择)。

param2：表示图形宽，8<=n<=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<=n<=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](#创建Table类实例) |
| [添加一行数据](#添加一行数据) |
| [设置Table中列对齐方式](#设置Table中列对齐方式) |
| [表格打印调用示例](#表格打印调用示例) |

**3.3.1 创建Table类实例**

|  |  |
| --- | --- |
| 方法 | 描述 |
| [Table](#Table方法) | 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 |
| [addRow](#addRow) | 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 |
| [setColumnAlignLeft](#setColumnAlignLeft) | 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：

|  |
| --- |
| [初始化画布](#初始化画布) |
| [设置字体属性](#设置字体属性) |
| [画布内容相关方法](#画布内容相关方法) |
| [画布打印调用示例](#画布打印调用示例) |

**3.4.1 初始化画布**

|  |  |
| --- | --- |
| 方法 | 描述 |
| [init](#init方法) | 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 |
| [setFontProperty](#setFontProperty) | 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 |
| [drawText](#画布画字符串) | 画字符串 |
| [drawLine](#画布绘制直线) | 绘制直线 |
| [drawRectangle](#画布绘制矩形) | 绘制矩形 |
| [drawEllips](#画布绘制椭圆) | 绘制椭圆 |
| [drawImage](#画布绘制图片) | 绘制图片 |
| [getCanvasImage](#画布获得画布上绘制的图像) | 获得画布上绘制的图像，发送给打印机打印 |
| [setTextAlignRight](#画布设置文本是否靠右) | 设置文本是否靠右 |
| [setTextExceedNewLine](#画布文本超出是否换行) | 文本超出是否换行 |
| [setUseSplit](#画布文本超出后换行是否使用分隔字符串) | 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。 |
| [setUseSplitAndString](#画布文本超出后换行是否使用分隔字符串) | 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：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=**new**FontProperty();

//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](#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 |
| [convertBitmap](#convertBitmap方法) | 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

MuPDFCorecore = 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("打印測試!\n");

|  |  |
| --- | --- |
| 方法名 | 描述 |
| CodePagePrinter(PrinterInstance mprinter) | 此类的构造方法 |
| printTextInCP437 | 打印CP437代码页文字(美国，欧洲标准) |
| printTextInCP850 | 打印多语言（Multilingual） |
| printTextInCP932 | 打印CP932代码页文字（Katakana）文字 |
| printTextInCP860 | 打印CP860代码页葡萄牙语文字 |
| printTextInCP863 | 打印CP863代码页加拿大法语文字 |
| printTextInCP865 | 打印CP863代码页北欧文字 |
| printTextInWCP1251 | 打印WCP1251代码页斯拉夫语文字 |
| printTextInMIK | 打印MIK代码页斯拉夫语/保加利亚语文字 |
| printTextInCP862 | 打印CP862代码页希伯来语文字 |
| printTextInWCP1252 | 打印WCP1252代码页，拉丁语1文字 |
| printTextInWCP1253 | 打印WCP1252代码页希腊语文字 |
| printTextInCP852 | 打印CP852代码页，拉丁语2文字 |
| printTextInCP858 | 打印CP858多种语言拉丁语1+欧元符文字 |
| printTextInCP720 | 打印CP720代码页阿拉伯语文字 |
| printTextInCP864 | 打印CP864代码页阿拉伯语文字 |
| printTextInISO\_8859\_1 | 打印ISO-8859-1代码页西欧语文字 |
| printTextInCP737 | 打印CP737代码页希腊语文字 |
| printTextInWCP1257 | 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 |

## 3.8、CanvasDrawPrint页模式接口打印类

简要说明：主要定义页模式的打印接口，由上位机平台软件来实现，方式是在内存中构建一张位图，由用户通过接口来完成页的布局，最终调用打印接口的时候，将在内存中构建好的位图数据，通过驱动打印的方式把数据发送给打印机来完成页面的打印，构建位图失败会在setCanvasDrawBack实现的接口返回中提示，详情请看[setCanvasDrawBack](#监听绘制失败)，画布绘制计量单位为dot（点）。

调用2.1.4中printImage完成画布的打印。主要方法如下：

|  |
| --- |
| [初始化页面](#初始化页面) |
| [页面内容相关方法](#页面内容相关方法) |
| [页模式接口相关例子实现](#页模式接口相关例子实现) |

**3.8.1** **初始化页面**

|  |  |
| --- | --- |
| 方法 | 描述 |
| [SPRTSetPageMode](#SPRTSetPageMode) | 初始化画布 |

方法名：public void SPRTSetPageMode(int width, int height)

方法描述：初始化页面

参数说明：width页面的宽

Height页面的高

返回值：无

异常：页模式的建立受制于测试设备内存大小限制，过大绘制失败需自行捕获该异常。

**3.8.2****页面内容相关方法**

|  |  |
| --- | --- |
| 方法 | 描述 |
| [SPRTPrintLine](#页面上绘制直线) | 绘制直线 |
| [SPRTPrintBox](#页面上绘制矩形) | 绘制矩形 |
| [SPRTPrintText](#页面上绘制文字) | 绘制文字 |
| [SPRTPrintPDFCode](#页面上绘制专业版PDF417) | 绘制专业版PDF417二维码 |
| [SPRTPrintPDFCode](#页面上绘制普通版PDF417二维码) | 绘制普通版PDF417二维码 |
| [SPRTPrintQRCode](#页面上绘制专业版QRCode二维码) | 绘制专业版QRCode二维码 |
| [SPRTPrintQRCode](#页面上绘制普通版QRCode二维码) | 绘制普通版QRCode二维码 |
| [SPRTPrintCode](#页面上绘制一维码) | 绘制一维码 |
| [SPRTPrintPicture](#页面上绘制Bitmap) | 绘制Bitmap |
| [getCanvasImage](#获得页面上绘制的图像) | 获得页面上绘制的图像，发送给打印机打印 |
| [setPaintStyle](#设置画笔空心还是实心) | 设置画笔空心还是实心 默认：实心 |
| [setTextExceedNewLine](#文本超出是否换行) | 文本超出是否换行 |
| [setTextAlignRight](#设置页面上文本是否靠右) | 设置文本是否靠右 |
| [setUseSplit](#页面文本超出后换行是否使用分隔字符串) | 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。按空格拆分（分隔字符 = 空格）。 |
| [setCanvasDrawBack](#监听绘制失败) | 监听绘制失败 |
| [getLength](#返回当前页面绘制的剩余高度) | 返回当前页面绘制的剩余高度，方便计算下一元素坐标点。  详情请看 3.83页模式接口代码例子 |

* 页面上绘制直线。方法名：SPRTPrintLine(int X\_Start, int Y\_Start, int X\_End, int Y\_End, int L\_Width)

参数说明：

1. X\_Start X打印线的起始点的坐标值
2. Y\_Start Y打印线的起始点的坐标值
3. X\_End X打印线的终止点的坐标值
4. Y\_End Y打印线的终止点的坐标值
5. L\_Width 线条的宽度

返回值：无

异常： 无

* 页面上绘制矩形。方法名：public void SPRTPrintBox(int X\_Start, int Y\_Start, int X\_End, int Y\_End, int L\_Width);

参数说明：

1. X\_Start X打印线的起始点的坐标值
2. Y\_Start Y打印线的起始点的坐标值
3. X\_End X打印线的终止点的坐标值
4. Y\_End Y打印线的终止点的坐标值
5. L\_Width 线条的宽度

返回值：无

异常： 无

* 页面上绘制文字。方法名：public void SPRTPrintText(int xPos, int yPos, String font, int font\_size, boolean bold, boolean underline, float italic, String content, Context mContext)

参数说明：

1. xPos 字符串的X坐标值
2. yPos 字符串的Y坐标值
3. font 使用的字体 例如："heiti.ttf"（字体文件必须放在assets目录下） 默认：传null为默认字体
4. font\_size 字体的大小
5. bold 是否粗体 boolean
6. underline 是否下划线 boolean
7. italic 是否斜体（正数表示向左倾斜，负数表示向右倾斜，正常-1~1 之间取值）
8. content 字符串内容
9. mContext 上下文

返回值：无

异常： 无

* 页面上绘制专业版PDF417。方法名：public void SPRTPrintPDFCode(int xPos, int yPos, int unit\_width, int unit\_height, int rows, int columns, int err\_level, String content)

参数说明：

1. xPos 二维码的起始x坐标
2. yPos 二维码的起始y坐标
3. unit\_width 宽度 （二维码的宽度，跟rows，columns，err\_level 相关参数可影响数值）
4. unit\_height 高度 （二维码的高度，跟rows，columns，err\_level 相关参数可影响数值）
5. rows 条码的行数，范围1-90
6. columns 条码的列数,范围1-30，如果rows指定了，那columns也必须指定
7. err\_level 纠错等级，范围1-8
8. content 条码的内容

返回值：无

异常： 无

* 页面上绘制普通版PDF417二维码。方法名：public void SPRTPrintPDFCode(int xPos, int yPos, int width, int height, String content)

参数说明：

1. xPos二维码的起始x坐标
2. yPos二维码的起始y坐标
3. width 宽
4. height 高 （大部分PDF417是长方形，为了符合宽高比例采用的是等比例缩放）
5. content 条码内容

返回值：无

异常： 无

* 页面上绘制专业版QRCode二维码。方法名：public void SPRTPrintQRCode(int xPos, int yPos, int module\_width, int module\_height, int white\_space, int version, int err\_level, String content)

参数说明：

1. xPos 二维码的起始x坐标
2. yPos 二维码的起始y坐标
3. module\_width 宽度 （二维码的宽度，跟version ，err\_level 相关参数可影响数值）
4. module\_height 高度 （二维码的高度，跟version ，err\_level 相关参数可影响数值）
5. white\_space 二维码四周的空白区域的宽度
6. version QR版本，范围1-40
7. err\_level 纠错等级，范围0-3
8. content 条码的内容

返回值：无

异常： 无

* 页面上绘制普通版QRCode二维码。方法名：public void SPRTPrintQRCode(int xPos, int yPos, int width, int height, String content)

参数说明：

1. xPos二维码的起始x坐标
2. yPos二维码的起始y坐标
3. width 宽
4. height 高
5. content 条码内容

返回值：无

异常： 无

* 页面上绘制一维码。方法名：public void SPRTPrintCode(int codeType, int xPos, int yPos, int width, int height, boolean show\_text, String content)

参数说明：

1. codeType 一维码的类型,目前也就是支持如下条码类（传参数如：BarcodeCreater.CODE39）
2. ITF
3. 2、CODE39的数据的长度最大是75
4. CODE128的数据长度最大是108
5. EAN13
6. EAN8
7. UPC-A
8. UPC-E
9. CODABAR
10. CODE\_93
11. xPos 一维码的起始x坐标
12. yPos 一维码的起始y坐标
13. width 宽
14. height 高
15. show\_text 是否显示文字
16. content 一维码的数据

返回值：无

异常： 无

* 页面上绘制Bitmap。方法名：public void SPRTPrintPicture(int xPos, int yPos, Bitmap bitmap)

参数说明：

1. xPos起始x坐标
2. yPos起始y坐标
3. Bitmap图片内容

返回值：无

异常： 无

* 监听绘制失败。方法名：public void setCanvasDrawBack(CanvasDrawBack mCanvasDrawBack)

返回参数说明CanvasDrawPrint：

int LINE = 1; 直线绘制失败

int BOX = 2; 矩形绘制失败

int TEXT = 3; 文字绘制失败

int PDF417\_PROFESSIONAL = 4; PDF417专业版绘制失败

int PDF417\_ORDINARY = 5; PDF417普通版绘制失败

int QRCODE\_PROFESSIONAL = 6; QRCODE专业版绘制失败

int QRCODE\_ORDINARY = 7;QRCODE普通版绘制失败

int ONE\_DIMENSION\_CODE = 8; 一维码绘制失败

int IMG = 9; 图片绘制失败

详细错误信息请查看相关LOG

返回值：无

* 设置画笔空心还是实心 默认：实心

setPaintStyle(Paint.Style style)

异常： 如果没有初始化页面会导致空指针异常（初始化过可不用理会），需要自己捕获异常。

* 获得页面上绘制的图像，发送给打印机打印。

getCanvasImage();

* 设置页面上文本是否靠右

setTextAlignRight(boolean alignRight);

* 文本超出是否换行。

setTextExceedNewLine(boolean newLine);

注：超过一行换行还是基于X轴开始绘制下一行

* 页面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。按空格拆分。（分隔字符串 = 空格）

mCanvasDrawPrint.setUseSplit(boolean useSplit)

* 返回当前页面绘制的剩余高度。

getLength();

注：该函数返回的是当前绘制的最高位的下方位置，方便层级绘制

**3.8.3页模式接口相关例子实现**

// 创建画布

CanvasDrawPrint mCanvasDrawPrint = new CanvasDrawPrint();

// 定义绘制失败监听mCanvasDrawPrint.setCanvasDrawBack(mCanvasDrawBack);

// 初始化画布宽高

mCanvasDrawPrint.SPRTSetPageMode(500, 800);

mCanvasDrawPrint.setTextAlignRight(true);//设置对其方式

mCanvasDrawPrint.setUseSplit(true); 页面上文本超出后换行是否使用分隔字符串//

// 绘制字体

mCanvasDrawPrint.SPRTPrintText(0, 0, null, 60, true, true, 0.5f, "测试字体多文字测试文本换行问题", mContext);

mCanvasDrawPrint.SPRTPrintText(0, mCanvasDrawPrint.getLength(), "kaiti.ttf", 60, false, false, 0, "楷体文字", mContext);

//绘制直线

mCanvasDrawPrint.SPRTPrintLine(0, mCanvasDrawPrint.getLength(), 450, mCanvasDrawPrint.getLength(), 10);

//设置画笔样式

mCanvasDrawPrint.setPaintStyle(Paint.Style.STROKE);

//绘制专业PDF417

mCanvasDrawPrint.SPRTPrintPDFCode(0, mCanvasDrawPrint.getLength(), 500, 500, 23, 8, 3, "123456789");

//绘制专业QRCode

mCanvasDrawPrint.SPRTPrintQRCode(0, mCanvasDrawPrint.getLength(), 200, 200, 10, 10, 4, "123456");

// 绘制一维码

mCanvasDrawPrint.SPRTPrintCode(BarcodeCreater.CODE\_128, 0, mCanvasDrawPrint.getLength(),300,200, true, "123456");

//将画布保存成图片并进行打印

mPrinter.printImage(mCanvasDrawPrint.getCanvasImage(), PAlign.*NONE*, 0);

CanvasDrawPrint.CanvasDrawBack mCanvasDrawBack = new CanvasDrawPrint.CanvasDrawBack() {

@Override

public void resultsBack(int i) {

String str = "";

if (i == CanvasDrawPrint.LINE) {

str = ("绘制直线出错请查看相关log");

} else if (i == CanvasDrawPrint.BOX) {

str = ("绘制矩形出错请查看相关log");

} else if (i == CanvasDrawPrint.TEXT) {

str = ("绘制文字出错请查看相关log");

} else if (i == CanvasDrawPrint.PDF417\_PROFESSIONAL) {

str = ("绘制专业PDF417出错请查看相关log");

} else if (i == CanvasDrawPrint.PDF417\_ORDINARY) {

str = ("绘制普通PDF417出错请查看相关log");

} else if (i == CanvasDrawPrint.QRCODE\_PROFESSIONAL) {

str = ("绘制专业二维码出错请查看相关log");

} else if (i == CanvasDrawPrint.QRCODE\_ORDINARY) {

str = ("绘制二维码普通出错请查看相关log");

} else if (i == CanvasDrawPrint.ONE\_DIMENSION\_CODE) {

str = ("绘制一维码出错请查看相关log");

} else if (i == CanvasDrawPrint.IMG) {

str = ("绘制bitmap出错请查看相关log");

}

Toast.makeText(mContext, str, 0).show();

}

};

注：更详尽的页模式打印调用请参见PicturePrintActivity.java中btn\_page\_mode\_printingt打印页模式点击事件。

## 3.9、记录Log保存到SD卡

简要说明：主要方便调试和找bug，用户遇到问题可以把SD卡里面的log文件导出来方便我司技术人员分析，记住当时发生bug时间提供相关log可以高效率解决发生的bug，该工具需要调用ZLog.Init();来初始化，初始化完成后默认是打开状态，该方法是常量标示，每次进入需要重新设置开关，保存的每个日志文件最大为 2Mb，超过 2Mb 文件名累加 1，一共生成9个文件，9个文件写满后会依次替换之前文件。遇到问题可以直接把设定的文件夹发送给我司。

|  |  |
| --- | --- |
| 方法 | 描述 |
| [ZLog.Init()](#注册log工具。); | 注册log工具类并且传入路径 |
| [ZLog.openSaveToFile()](#打开记录log。) | 打开记录log |
| [ZLog.closeSaveToFile ()](#关闭记录log。) | 关闭记录log |

* 注册log工具。方法名：public static synchronized void Init(String logDir)

参数说明：

logDir 保存log的路径

返回值：无

异常： 无

* 打开记录log。方法名：public static synchronized void openSaveToFile()

参数说明：无

返回值：无

异常： 无

* 关闭记录log。方法名：public static synchronized void closeSaveToFile()

参数说明：无

返回值：无

异常： 无

**3.9.1例子实现**

// 初始化 ZldLog 使用前应该先调用一下次方法

ZLog.Init(Environment.getExternalStorageDirectory()

.getAbsolutePath() + "/SPRT\_LOGS/");

// 打开记录

ZLog.openSaveToFile()

// 关闭记录

ZLog. closeSaveToFile ()

## 3.10、Y33采集打印

**采集打印功能只适用于基于Y33一体机开发的Android应用。**

**采集打印逻辑示意图:**

三方应用

定制应用

打印SDK

网络

打印服务

蓝牙

RS232

usb1

usb2

POS

**采集打印功能简述**：Y33一体机提供了一个打印服务用于解析并处理POS端（Usb口或Serial口）或者第三方应用（网络端口或蓝牙端口）发送过来的数据。通过解析数据将打印数据下发至打印机同时将需要采集的数据上传至Android定制应用。基于Y33一体机开发的Android定制应用可以检测到打印动作（打印前，打印后）并决定是否追加数据或进行记录操作。Android定制应用同样可以收集到Y33一体机上传的打印数据。

**打印服务配置描述：**在打印服务的配置界面可以配置采集方式，采集端口，是否打印，是否打印前通知Android定制应用以及是否打印后通知Android定制应用等配置选项。如下所示：

采集方式：可以选择以驱动方式采集驱动数据或者选择以sdk方式采集普通数据。

采集端口：可以选择采集通过网口传送过来的数据或者通过USB口传送过来的数据或者通过蓝牙传送过来的数据。（三者一次只能采集一个，不可并发。）

是否打印：可以选择只采集数据或者采集数据并打印。

是否打印前通知：可以选择是否在打印机打印一单数据前通知Android定制应用来决定是否追加打印数据头或者做一些记录操作。

是否打印后通知：可以选择是否在打印机打印一单数据后通知Android定制应用来决定是否追加打印数据尾或者做一些记录操作。

关闭服务和启动服务：默认情况下打印服务是开机自起的。当出现异常时，可通过关闭并且重新启动服务来重新采集数据。



**基于Y33一体机的Android应用开发流程：**

1.依赖printersdkv5.4.jar

2.在配置文件AndroidManifest.xml中将应用的application 继承com.printer.sdk.application.MyApplication

3.在配置文件AndroidManifest.xml中注册广播接收者com.receiver.sprt.yxz.printservice.client.receiver.PrintServiceBroadcastReceiver。并添加以下action：

<!-- PrintService发送的广播 usbep1 -->

<actionandroid:name="com.sprt.yxz.printservice.service.PrinterService.usbep1\_start\_print"/>

<action android:name="com.sprt.yxz.printservice.service.PrinterService.usbep1\_end\_print" />

<action android:name="com.sprt.yxz.printservice.service.PrinterService.usbep1\_parser\_data" />

<!-- PrintService发送的广播 net -->

<action android:name="com.sprt.yxz.printservice.service.PrinterService.net\_start\_print" />

<action android:name="com.sprt.yxz.printservice.service.PrinterService.net\_end\_print" />

<action android:name="com.sprt.yxz.printservice.service.PrinterService.net\_parser\_data" />

<!-- PrintService发送的广播 bluetooth -->

<action android:name="com.sprt.yxz.printservice.service.PrinterService.bt\_start\_print" />

<action android:name="com.sprt.yxz.printservice.service.PrinterService.bt\_end\_print" />

<action android:name="com.sprt.yxz.printservice.service.PrinterService.bt\_parser\_data" />

4.开发演示

4.1实例化PrinterInstance对象

PrinterInstance myPrinter = PrinterInstance.getPrinterInstance();

4.2打开追加数据的端口

boolean isConnected = myPrinter.openConnection();

4.3监听打印动作

myPrinter.setOnPrintListener(new OnPrintListener() {

@Override

public void doBeforePrint() {

XLog.i(TAG, "客户端 打印开始前追加打印头。");

try {

4.4监听到打印开始，追加打印数据头

if(myPrinter！=null)

int writeLen = myPrinter.sendBytesData("\*\*\*\*\*\*\*\*\*\*\*\*\*一单数据开始\*\*\*\*\*\*\*\*\*\*\*\*\*\*\r\n".getBytes("gbk"));

} catch (UnsupportedEncodingException e) {

e.printStackTrace();

}

}

@Override

public void doAfterPrint() {

4.5监听到打印结束，追加打印数据尾

XLog.i(TAG, "客户端 打印结束后追加数据尾");

try {

if(myPrinter！=null)

int writeLen = myPrinter.sendBytesData("\*\*\*\*\*\*\*\*\*\*\*\*\*一单数据结束\*\*\*\*\*\*\*\*\*\*\*\*\*\*\r\n".getBytes("gbk"));

} catch (UnsupportedEncodingException e) {

e.printStackTrace();

}

}

@Override

public void onReceiveParserData(int type, byte[] data) {

if (data == null || data.length == 0) {

XLog.i(TAG, "客户端 上传的采集数据为空");

return;

}

if (type == 0) //客户端 接收到USB口上传的采集数据

if (type == 1) //客户端 接收到NET口上传的采集数据

if (type == 2) //客户端 接收到BT口上传的采集数据 }

});

4.6关闭追加数据的端口（注意：closeConnection()方法与openConnection()方法成对出现，请在发送完追加数据后关闭端口）。

myPrinter.closeConnection();

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  SOH  STX  ETX  EOT  ENQ  ACK  BEL  BS  HT  LF  VT  FF  CR  SO  SI  DLE  DC1  DC2  DC3  DC4  NAK  SYN  ETB  CAN  EM  SUB  ESC  FS  GS  RS  US  SP  !  "  #  $  % | 00  01  02  03  04  05  06  07  08  09  0A  0B  0C  0D  0E  0F  10  11  12  13  14  15  16  17  18  19  1A  1B  1C  1D  1E  1F  20  21  22  23  24  25 | 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | &  '  (  )  \*  +  ,  -  .  /  0  1  2  3  4  5  6  7  8  9  :  ;  <  =  >  ?  @  A  B  C  D  E  F  G  H  I  J  K | 26  27  28  29  2A  2B  2C  2D  2E  2F  30  31  32  33  34  35  36  37  38  39  3A  3B  3C  3D  3E  3F  40  41  42  43  44  45  46  47  48  49  4A  4B | 38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75 | L  M  N  O  P  Q  R  S  T  U  V  W  X  Y  Z  [  \  ]  ^  \_  FNC1  FNC2  FNC3  FNC4  SHIFT  CODEB  CODEC | 4C  4D  4E  4F  50  51  52  53  54  55  56  57  58  59  5A  5B  5C  5D  5E  5F  7B,31  7B,32  7B,33  7B,34  7B,53  7B,42  7B,43 | 76  77  78  49  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  123,49  123,50  123,51  123,52  123,83  123,66  123,67 |

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

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