

Android ZPL SDK 说明文档

Android ZPL SDK 说明文档.....	1
一、SDK 的组成.....	3
1.1 SDK jar 包.....	3
1.2 如何使用 ZPLPrinterHelper.....	3
二、 连接方式.....	4
2.1 蓝牙连接.....	4
2.2 WIFI 连接.....	5
2.3 USB 连接.....	6
三、打印指令.....	8
3.1 标签开始.....	8
3.2 设置坐标.....	8
3.3 标签结束.....	9
3.4 字段开头指令.....	9
3.5 字段结束指令.....	10
3.6 打印宽度.....	11
3.7 打印方向和对齐方式.....	12
3.8 打印直线.....	13
3.9 打印圆.....	14
3.10 打印斜线.....	15
3.11 设置标签原点位置.....	16

3.12 整体向左偏移.....	16
3.13 设置文本框（可自动换行）	17
3.14 二维码.....	18
3.15 条码.....	19
3.16 打印图片.....	20
3.17 发数据函数.....	21
3.18 读数据函数.....	21
3.19 自检页.....	22
3.20 打印文本.....	23

一、SDK 的组成

1.1 SDK jar 包

添加方式如图

```
compile files('libs/Android_SDK_ZPL_V0.0.1.jar')
```

这个 jar 包里面包含了我们的所有的连接函数,包括了蓝牙连接、WIFI 连接、USB 连接。以及我们的所有的指令接口函数。这些接口都在 ZPLPrinterHelper 这个类中。

1.2 如何使用 ZPLPrinterHelper

这个类用的是单例模式,通过 ZPLPrinterHelper.getZPL(mContext)获得它的引用。有了它的引用你就可以调用连接函数和指令函数。

二、连接方式

2.1 蓝牙连接

```
ZPLPrinterHelper mZPL=ZPLPrinterHelper.getZPL(mContext)  
mZPL.PortOpen(portSetting)
```

参数：

mContext：上下文对象。

portSetting："Bluetooth,"+MAC。（MAC：蓝牙地址）

返回：

0：连接成功。

-1：连接失败。

断开蓝牙

```
public static boolean PortClose()
```

例子：

```
mZPL.PorClose()
```

返回：

true：断开成功。

false：断开失败。

蓝牙是否连接

```
public static boolean IsOpened()
```

注意：

这个不是实时监测蓝牙连接状态，监测蓝牙状态需要通过系统的蓝牙广播。

例子：

```
ZPLPrinterHelper.IsOpened()
```

返回：

true：蓝牙已连接。

false：蓝牙未连接

2.2 WIFI 连接

连接 WIFI：

```
public static int PortOpen(String portSetting)
```

例子：

```
ZPLPrinterHelper mZPL=ZPLPrinterHelper.getZPL(mContext)
```

```
mZPL.PortOpen("WiFi,"+IP+", "+PortNumber)
```

IP:打印机的 IP 地址。

PortNumber：端口。 默认：9100

返回：

0：连接成功。

-1：连接失败。

断开 WiFi：

```
public static boolean PortClose()
```

例子：

```
mZPL.PorClose()
```

返回：

true：断开成功。

false：断开失败。

WiFi 是否连接：

```
public static boolean IsOpened()
```

例子:

```
ZPLPrinterHelper.IsOpened()
```

返回:

true: 已连接。

false: 未连接。

2.3 USB 连接

连接 USB:

```
public static int PortOpen(UsbDevice usbdevice) {
```

例子:

```
ZPLPrinterHelper mZPL=ZPLPrinterHelper.getZPL(mContext)
```

```
mZPL.PortOpen(usbdevice)
```

usbdevice: UsbDevice 的对象

返回:

0: 连接成功。

-1: 连接失败。

断开 USB:

```
public static boolean PortClose()
```

例子:

```
mZPL.PorClose()
```

返回:

true: 断开成功。

false: 断开失败。

USB 是否连接:

```
public static boolean IsOpened()
```

例子:

```
mZPL.IsOpened()
```

返回:

true: 已连接。

false: 未连接。

三、打印指令

3.1 标签开始

函数：

```
int start();
```

返回：

大于 0：正常，否则异常。

例子：

```
mZPL.start();
```

```
mZPL.printText("0","0",5,"N",3,"TEXT");
```

```
mZPL.end();
```

3.2 设置坐标

函数： `int setXY(String x,String y);`

参数：

X：横坐标。

Y：纵坐标。

返回：

大于 0：正常，否则异常。

3.3 标签结束

函数：

```
int end();
```

返回：

大于 0：正常，否则异常。

例子：

```
mZPL.start();
```

```
mZPL.printText("0","0",5,"N",3,"TEXT");
```

```
mZPL.end();
```

3.4 字段开头指令

函数：

```
int FD(String a)
```

参数：

A: 文本

返回：

大于 0：正常，否则异常。

例子：

```
mZPL.start();
```

```
mZPL.setXY("0","0");
```

```
mZPL.FD("TEXT");
```

```
mZPL.FS();
```

```
mZPL.end();
```

3.5 字段结束指令

函数：

```
int FS();
```

返回：

大于 0：正常，否则异常。

例子：

```
mZPL.start();
```

```
mZPL.setXY("0","0");
```

```
mZPL.FD("TEXT");
```

```
mZPL.FS();
```

```
mZPL.end();
```

3.6 打印宽度

函数：

```
int PW(String a);
```

参数：

A: 宽度 (in dots) 。

返回：

大于 0: 正常，否则异常。

例子：

```
mZPL.start();
```

```
mZPL.PW("100")
```

```
mZPL.setXY("0","0");
```

```
mZPL.FD("TEXT");
```

```
mZPL.FS();
```

```
mZPL.end();
```

3.7 打印方向和对齐方式

函数：

`int FW(String rotate,String justification)`

参数：

rotate：打印方向，取值如下：

N = normal

R = rotated 90 degrees

I = inverted 180 degrees

B = bottom-up 270 degrees, read from bottom up

Justification：对其方式，取值如下：

0 = left justification

1 = right justification

2 = auto justification (script dependent)

返回：

大于 0：正常，否则异常。

例子：

```
mZPL.start();
```

```
mZPL.FW("N","0")
```

```
mZPL.setXY("0","0");
```

```
mZPL.FD("TEXT");
```

```
mZPL.FS();
```

```
mZPL.end();
```

3.8 打印直线

函数：

```
int printLine(String w,String h,String t,String c,String r);
```

参数：

w: 直线的宽度。（1~32000 单位 点）

H: 直线的高度。（1~32000 单位 点）

T: 线条的宽度。（1~32000 单位 点）

C: 线条的颜色：

B=black; （默认： B）

W=white;

R: 圆角的弧度（0~8）

返回：

大于 0: 正常，否则异常。

例子：

```
mZPL.start();
```

```
mZPL.setXY("0","0");
```

```
mZPL.printLine("100","2","2","B","0")//打印 100 点宽，2 点高的  
横线
```

```
mZPL.end();
```

3.9 打印圆

函数：

```
int printCircle(String d,String t,String c)
```

参数：

d: 圆的直径（3~4095）。

t: 边框厚度（1~4095）。

C: 线条颜色：

B=black; （默认： B）

W=white;

返回：

大于 0: 正常，否则异常。

例子：

```
mZPL.start();
```

```
mZPL.setXY("0","0");
```

```
mZPL.printCircle("100","2","B")
```

```
mZPL.end();
```

3.10 打印斜线

函数：

```
int printSlashLine(String w,String h,String t,String c,String o)
```

参数：

w: 斜线的宽度（3~32000）。

H: 斜线的高度（3~32000）。

T: 线条宽度（1~32000）。

C: 线条颜色：

B=black; （默认： B）

W=white;

O: 斜线的方向：

R (or /) = right-leaning diagonal

L (or \) = left-leaning diagonal

（默认： R）

返回：

大于 0： 正常， 否则异常。

例子：

```
mZPL.start();
```

```
mZPL.setXY("0","0");
```

```
mZPL.printSlashLine("100","100","2","B","R")
```

```
mZPL.end();
```

3.11 设置标签原点位置

函数：

```
int LH(String x,String y)
```

参数：

X： 原点的横坐标。

Y： 原点的纵坐标。

返回：

大于 0： 正常，否则异常。

例子：

略

3.12 整体向左偏移

函数：

```
int LS(String a)
```

参数：

a： 偏移量（-9999~9999）负数代表向右偏移，（默认 0）。

返回：

大于 0： 正常，否则异常。

例子：

```
mZPL.start();
```

```
mZPL.LS("20")
```

```
mZPL.printText("100","100",5,"N",3,"TEXT");
```

```
mZPL.end();
```

3.13 设置文本框（可自动换行）

函数

```
int TB(String a,String b,String c)
```

参数：

a: 文字方向：

N = normal

R = rotate 90 degrees clockwise

I = invert 180 degrees

B = read from bottom up-270 degrees

b: 文本框的宽度。

c: 文本框的高度。

返回：

大于 0: 正常，否则异常。

例子：

```
mZPL.start();
```

```
mZPL.TB("N","300","300")
```

```
mZPL.printText("100","100",5,"N",3,"TEXT");
```

```
mZPL.end();
```

3.14 二维码

函数：

```
int printQRcode(String x,String y,String orientation,String  
magnification,String size,String data)
```

参数：

x: 横坐标。

y: 纵坐标。

orientation: 方向 (N) 。

magnification: 模式：

1=普通模式。

2=加强模式。

默认 (2)

size : 尺寸 (1~10) 。

data:二维码的内容

返回：

大于 0: 正常，否则异常。

例子：

```
mZPL.start();  
mZPL.printData("^C14\r\n");  
mZPL.printQRcode("10","10","N","2","5","abc123");  
mZPL.end();
```

3.15 条码

函数：

```
int printBarcode(String x,String y,int type,String  
orientation,String height,String f,String data)
```

参数：

x: 横坐标。

y: 纵坐标。

Type: 条码类型：

0=39

1=EAN-8

2=UPC-E

3=93

4=128

5=EAN-13

orientation: 方向：

N = normal

R = rotated 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

Height: 条码高度（1-32000）。

F: 条码内容是否见（默认 Y）：

Y = yes

N = no

返回：

大于 0：正常，否则异常。

例子：

```
mZPL.start();  
  
mZPL.printBarcode("10","10","0","N","100","Y","123456789");  
  
mZPL.end();
```

3.16 打印图片

函数：

```
printBitmap(String x,String y,Bitmap bmp)
```

参数：

x：起始的 X 坐标。

y：起始的 Y 坐标。

Bmp：图片对象。

返回：

大于 0：正常，否则异常。

例子：

```
mZPL.start();  
  
mZPL.printBitmap("10","10",bitmap);  
  
mZPL.end();
```

3.17 发数据函数

函数：

```
int WriteData(byte[] bData)
```

参数：

bData：需要发给打印机的数据。

返回：

大于 0：正常，否则异常。

例子：

```
mZPL.WriteData(byt)
```

3.18 读数据函数

函数：

```
byte[] ReadData(int outTime)
```

参数：

outTime：超时时间（单位秒）。

返回：

读取到的数据。

例子：

```
mZPL.ReadData(2)
```

3.19 自检页

函数：

```
int selfTest()
```

返回：

大于 0：正常，否则异常。

例子：

```
mZPL.selfTest()
```

3.20 打印文本

注意：打印中文时需要选择编码，请参照例子。

函数：

```
int printText(String x,String y,int type,String orientation,int  
size,String data)
```

参数：

x：横坐标。

y：纵坐标。

type：字体（0~6：中文无效，7：中文）。

orientation：方向。

N = normal

R = rotate 90 degrees clockwise

I = invert 180 degrees

B = read from bottom up-270 degrees

size：字体大小。

1：10px。

2：20px。

3：30px。

4：40px。

5：50px。

6：60px。

data：文本数据。

返回：

大于 0：正常，否则异常。

例子：

```
mZPL.start();
```

```
mZPL.printData("^CI14\r\n")//打印中文时需要加入这句
```

```
mZPL.printText("0","0",5,"N",3,"TEXT");
```

```
mZPL.end();
```