

# PRT Android TSPL SDK 说明文档

PRT Android TSPL SDK 说明文档.....	1
一 SDK 指令函数.....	2
1 页标签开始指令.....	2
2 页标签结束指令（行模式除外）.....	3
3 打印打印机信息.....	4
4 设置标签纸的间距.....	5
5 走纸.....	6
6 清除打印缓冲区内容.....	7
7 文本打印，包含一些变体.....	7
8 打印方向.....	9
9 定义标签的参考起点.....	9
10 定义国际字符集代码.....	9
11 条码.....	10
12 打印二维码.....	11
13 标签进纸指定长度.....	12
14 将标签定位到下个标签起始处.....	12
15 打印线条.....	12
16 将标签定位到内部传感器确定的起点.....	13
17 打印图片.....	13
18 打印浓度.....	14
19 打印速度.....	14
20 切纸.....	15
21 打印矩形框.....	15
22 设置蜂鸣器鸣叫时间.....	15
23 发数据函数.....	16
24 读数据函数.....	16
25 反白框.....	16
26 文字加粗.....	18
27 获取打印机状态.....	18
28 文本段落打印.....	19
二 通讯部分函数.....	20
1.1 蓝牙连接函数.....	20
1.2 蓝牙断开函数.....	20
1.3 蓝牙是否连接.....	20
2.1 USB 连接函数.....	21
2.2 USB 断开函数.....	21
2.3 USB 是否连接.....	21
3.1 WIFI 连接函数.....	22
3.2 WIFI 断开函数.....	22

## 一 SDK 指令函数

### 1 页标签开始指令

`int printAreaSize(String width, String height)`

参数:

`width`: 打印范围的宽 (单位 mm) 。

`height`: 打印范围的高 (单位 mm) 。

返回:

大于 0: 正常, 否则异常。

例子:

```
HPRTPrinterHelper.printAreaSize("100", "100")
```

```
HPRTPrinterHelper.printText("0","0","9","0","1","1","TEXT")//打印文字 TEXT
```

```
HPRTPrinterHelper.Print("1","1")
```

## 2 页标签结束指令（行模式除外）

`int Print(String strnum, String strcopies)`

参数:

`strnum`: 打印次数。

`strcopies`: 计数器（默认 1）。

返回:

大于 0: 正常，否则异常。

例子:

```
HPRTPrinterHelper.printAreaSize("100", "100")
```

```
HPRTPrinterHelper.printText("0","0","9","0","1","1","TEXT")//打印文字 TEXT
```

```
HPRTPrinterHelper.Print("1","1")
```

### 3 打印打印机信息

`int SelfTest()`//打印自检页

返回:

大于 0: 正常, 否则异常。

例子:

`HPRTPrinterHelper.SelfTest()`

#### 4 设置标签纸的间距

`int Gap(String distance, String offset)`

参数:

Distance: 两个标签的间距 (单位: mm)

Offset: 标签里的内容与标签底部的间距 (单位: mm)

返回:

大于 0: 正常, 否则异常。

例子:

`HPRTPrinterHelper.Gap("5","5")`

## 5 走纸

`int Offset(String distance)`

参数:

`distance`: 走纸距离(单位 mm)。

返回:

大于 0: 正常, 否则异常。

例子:

`HPRTPrinterHelper.Offset("5")`

## 6 清除打印缓冲区内容

`int CLS()`

返回:

大于 0: 正常, 否则异常。

例子:

`HPRTPrinterHelper.CLS()`

## 7 文本打印, 包含一些变体

(1) `Int printText(String x_pos,String y_pos,String font,String rotation,String x_multiplication,String y_multiplication,int alignment,String code_data)`

参数:

`x_pos`: 文字起始 x 坐标:

`y_pos`: 文字起始 y 坐标:

`font`:

0: Monotype CG Triumvirate Bold Condensed, font width and height is stretchable

1: 8 x 12 fixed pitch dot font

2: 12 x 20 fixed pitch dot font

3: 16 x 24 fixed pitch dot font

4: 24 x 32 fixed pitch dot font

5: 32 x 48 dot fixed pitch font

6: 14 x 19 dot fixed pitch font OCR-B

7: 21 x 27 dot fixed pitch font OCR-B

8: 14 x 25 dot fixed pitch font OCR-A

9: 只有这个字体能够打印中文。

`Rotation`: 打印方向。

0 : No rotation

90 : degrees, in clockwise direction

180 : degrees, in clockwise direction

270 : degrees, in clockwise direction

`x_multiplication`: x 轴方向文字拉伸的倍数 (0 号字体和 9 号字体代表字体的真实大小, 单位 px) 。

`y_multiplication`: y 轴方向文字拉伸的倍数 (0 号字体和 9 号字体代表字体的真实大小, 单位 px) 。

`Alignment`: 对齐方式 (部分旧机型不支持该功能, 可以去掉该参数)

1: 左对齐

2: 居中

3: 右对齐

**code\_data:** 文本数据。

返回:

大于 0: 正常, 否则异常。

例子:

```
HPRTPrinterHelper.printAreaSize("100","200")
HPRTPrinterHelper.CLS();
HPRTPrinterHelper.printText("0","0","1","0","0","0","TEXT")
HPRTPrinterHelper.Print("1","1")
```

(2) `int printText(String x_pos,String y_pos,String font,String rotation,int size, int alignment,String code_data)`

参数:

**x\_pos:** 文字起始 x 坐标。

**y\_pos:** 文字起始 y 坐标。

**font:** 字体:

同上

**Rotation:** 打印方向。

0 : No rotation

90 : degrees, in clockwise direction

180 : degrees, in clockwise direction

270 : degrees, in clockwise direction

**Size:** 字体大小 1~7。

**Alignment:** 对齐方式 (部分旧机型不支持该功能, 可以去掉该参数)

1: 左对齐

2: 居中

3: 右对齐

**code\_data:** 文本数据。

返回:

大于 0: 正常, 否则异常。

例子:

```
HPRTPrinterHelper.printAreaSize("100","200")
HPRTPrinterHelper.CLS();
HPRTPrinterHelper.printText("0","0","0","0","0","0","TEXT")
HPRTPrinterHelper.Print("1","1")
```



## 8 打印方向

int Direction(String direction)

参数:

direction:

0: 垂直。

1: 水平。

返回:

大于 0: 正常, 否则异常。

例子:

```
HPRTPrinterHelper.printAreaSize("100","200")
```

```
HPRTPrinterHelper.Direction("0");
```

```
HPRTPrinterHelper.printText("0","0","0","0","0","TEXT")
```

```
HPRTPrinterHelper.Print("1","1")
```

## 9 定义标签的参考起点

int Reference(String x\_pos,String y\_pos)

参数:

x: 起点 x 轴坐标。

y: 起点 y 轴坐标。

返回:

大于 0: 正常, 否则异常。

例子:

```
HPRTPrinterHelper.Reference("0","0")
```

## 10 定义国际字符集代码

int Codepage(String codepage)

参数:

codepage: 代码编号

USA: USA 、 BRI: British 、 GER: German 、 FRE: French

DAN: Danish 、 ITA: Italian 、 SPA: Spanish 、 SWE: Swedish

SWI: Swiss 、 437: United States 、 850: Multilingual 、 852: Slavic

860: Portuguese 、 863: Canadian/French 、 865: Nordic 、

857: Turkish (TSPL2 printers only)

1250: Central Europe (TSPL2 printers only)

1252: Latin I (TSPL2 printers only) 、 1253: Greek (TSPL2 printers only)  
1254: Turkish (TSPL2 printers only)

返回:

大于 0: 正常, 否则异常。

例子:

```
HPRTPrinterHelper.Codepage("437")
```

## 11 条码

```
int printBarcode(String x_pos, String y_pos, String code_type, String height, String  
readable, String rotation, String narrow, String wide, String code_data)
```

参数:

x\_pos: 条码的起始横坐标。

y\_pos: 条码的起始纵坐标。

code\_type: 条码类型:

128, 128M, EAN128 , 39 , 93, UPCA , MSI , ITF14 , EAN13

Height: 条码高度 (单位: PX) 。

Readable: 条码数据是否可见

0: not readable

1: human readable

rotation: 条码方向:

0 : No rotation

90 : degrees, in clockwise direction

180 : degrees, in clockwise direction

270 : degrees, in clockwise direction

narrow : 窄条的单位宽度(默认 1)。

Wide: 宽条码的单位宽度(默认 1)。

code\_data: 条码数据。

返回:

大于 0: 正常, 否则异常。

例子:

```
HPRTPrinterHelper.printAreaSize("100","200")
```

```
HPRTPrinterHelper.CLS();
```

```
HPRTPrinterHelper.printBarcode("0","0","128","100","1","0","1","1","12345678  
90")
```

```
HPRTPrinterHelper.Print("1","1")
```

## 12 打印二维码

```
printQRcode(String x_pos,String y_pos,String ecc_level,String width,String  
mode,String rotation,String code_data)
```

参数:

rotation: 条码方向:

0 : No rotation

90 : degrees, in clockwise direction

180 : degrees, in clockwise direction

270 : degrees, in clockwise direction

X: 二维码的起始横坐标。

Y: 二维码的起始纵坐标。

ecc\_level: 纠错等级

L : 7%

M : 15%

Q : 25%

H : 30%

width: 宽度: 1~10

Mode: 模式

A: Auto

M: Manual

Data: 二维码的数据。

返回:

大于 0: 正常, 否则异常。

例子:

```
HPRTPrinterHelper.printAreaSize("100","200")
```

```
HPRTPrinterHelper.CLS();
```

```
HPRTPrinterHelper.printQRcode("0","0","M","6","A","0","1234567890")
```

```
HPRTPrinterHelper.Print("1","1")
```

### 13 标签进纸指定长度

`int Feed(String len)`

参数:

`len`: 进纸长度 (单位 mm)

返回:

大于 0: 正常, 否则异常。

例子:

```
HPRTPrinterHelper.Feed("5")
```

### 14 将标签定位到下个标签起始处

`int Formfeed()`

返回:

大于 0: 正常, 否则异常。

例子:

```
HPRTPrinterHelper.Formfeed()
```

### 15 打印线条

`int Bar(String x_pos,String y_pos,String width, String height)`

参数:

`x_pos`: 起始的 X 坐标。

`y_pos`: 起始的 Y 坐标。

`width`: 线条的宽度(单位: PX)。

`height`: 线条的高度 (单位: PX) 。

返回:

大于 0: 正常, 否则异常。

例子:

```
HPRTPrinterHelper.printAreaSize("100","200")
HPRTPrinterHelper.Bar("10","10","100","2")//横线
HPRTPrinterHelper.Bar("10","10","2","100")//竖线
HPRTPrinterHelper.Print("1","1")
```

## 16 将标签定位到内部传感器确定的起点

`int Home()`

返回:

大于 0: 正常, 否则异常。

例子:

```
HPRTPrinterHelper.Home()
```

## 17 打印图片

1) `printImage(String x_pos,String y_pos,String url ,boolean isNegate,boolean isLZO)`

参数:

`x_pos`: 图片起始的 x 坐标。

`y_pos`: 图片起始的 y 坐标。

`Url`: 图片的路径(这个路径只能是本地路径)。

`isNegate`: 图片反白。

`True`: 正常显示。

`False`: 取反显示。

`isLZO`: 是否压缩 (必须打印机支持压缩功能)。

返回:

大于 0 正常; 等于 -1 图片的宽度或者高度超出打印机的范围。

例子:

```
HPRTPrinterHelper.printAreaSize("100","200");//标签高度要设置比图片的高度长
```

```
HPRTPrinterHelper.CLS()
```

```
HPRTPrinterHelper.printImage("10","10",url,true,false)
```

```
HPRTPrinterHelper.Print("1","1")
```

2) `int printImage(String x_pos,String y_pos,Bitmap bmp ,boolean isNegate,boolean isLZO)`

参数:

`X`: 图片起始的 x 坐标。

`Y`: 图片起始的 y 坐标。

`bmap`: 图片的 `Bitmap` 的对象。

`isNegate`: 图片反白。

True: 正常显示。

False: 取反显示。

isLZO:是否压缩（必须打印机支持压缩功能）。

返回:

大于 0 正常；等于-1 图片的宽度或者高度超出打印机的范围。

例子:

```
HPRTPrinterHelper.printAreaSize("100","200");//标签高度要设置比图片的高度长
HPRTPrinterHelper.CLS()
HPRTPrinterHelper.printImage("10","10",bitmap,true,false)
HPRTPrinterHelper.Print("1","1")
```

## 18 打印浓度

int Density(String contrast )

参数:

Contrast: 0~15

0: specifies the lightest level

15: specifies the darkest level

返回:

大于 0: 正常，否则异常。

例子:

```
HPRTPrinterHelper.Density("5")
```

## 19 打印速度

int Speed(String speed )

参数:

1    1.5   2    2.5   3    3.5   4    5    6    8    10   12

speed

返回:

大于 0: 正常，否则异常。

例子:

HPRTPrinterHelper.Speed("2")

## 20 切纸

int Cut()

返回:

大于 0: 正常, 否则异常。

例子:

HPRTPrinterHelper.Cut()

## 21 打印矩形框

int Box(String x\_start,String y\_start,String x\_end, String y\_end,String thickness)

参数:

x\_start: 左上角 x 坐标。

y\_start: 左上角 y 坐标。

x\_end: 右下角 x 坐标。

y\_end: 右下角 y 坐标。

Thickness: 线条宽度。

返回:

大于 0: 正常, 否则异常。

例子:

HPRTPrinterHelper.printAreaSize("100","200")

HPRTPrinterHelper.CLS()

HPRTPrinterHelper.Box("10","10","100","100","2")

HPRTPrinterHelper.Print("1","1")

## 22 设置蜂鸣器鸣叫时间

int Sound(String level, String interval)

参数:

level: 蜂鸣声的持续时间, (1/8) 秒为单位指定。

Interval: 间隔的时间

返回:

大于 0: 正常, 否则异常。

例子:

```
HPRTPrinterHelper.Sound("8","8")
```

## 23 发数据函数

```
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:

```
HPRTPrinterHelper.WriteData(bData)//向打印机发送数据
```

## 24 读数据函数

```
Byte[] ReadData(int timeout)
```

参数:

timeout: 超时时间 (单位毫秒) 。

返回:

Data: 从打印机读取到的数据

例子:

```
bData=HPRTPrinterHelper.ReadData(2000)//bData 是读取出来的数据
```

## 25 反白框

```
int Reverse(String x_start,String y_start,String x_width,String y_height)
```

参数:

x\_start: X 坐标。

y\_start: Y 坐标。



x\_width: 宽度。

y\_height: 高度。

注意:

***200 DPI: 1 mm = 8 dots***

***300 DPI: 1 mm = 12 dots***

返回:

大于 0: 正常, 否则异常。

例子:

```
HPRTPrinterHelper.printAreaSize("100","200")
```

```
HPRTPrinterHelper.CLS()
```

```
HPRTPrinterHelper.printText("100","100","9","0","32","32","REVERSE" )
```

```
HPRTPrinterHelper.Reverse("90","90","128","40")
```

```
HPRTPrinterHelper.Print("1","1")
```

**REVERSE**

## 26 文字加粗

`int Bold(int bold)`

参数:

`bold`: 0: 不加粗。  
1: 加粗。

返回:

大于 0: 正常, 否则异常。

例子:

`HPRTPrinterHelper.Bold(1)`

## 27 获取打印机状态

`int getPrinterStatus()`

参数:

无

返回:

<code>HPRTPrinterHelper.STATUS_DISCONNECT:</code>	断开连接
<code>HPRTPrinterHelper.STATUS_TIMEOUT:</code>	查询超时
<code>HPRTPrinterHelper.STATUS_OK:</code>	打印机正常
<code>HPRTPrinterHelper.STATUS_COVER_OPENED:</code>	开盖
<code>HPRTPrinterHelper.STATUS_NOPAPER:</code>	缺纸
<code>HPRTPrinterHelper.STATUS_OVER_HEATING:</code>	过热
<code>HPRTPrinterHelper.STATUS_PRINTING:</code>	打印中

例子:

`HPRTPrinterHelper.getPrinterStatus()`

## 28 文本段落打印

```
int printBlock(int startX,int startY,int width,int height,int font,int rotation,  
int multiplicationX,int multiplicationY,int space,int alignment,String content)
```

参数:

startX: 文本左上角 x 坐标。

startY: 文本左上角 y 坐标。

Width: 文本的宽度。

Height: 文本的高度。

Font: 字体大小 (0: 16\*16, 1: 24\*24)

Rotation: 旋转方向。 (0, 90, 180, 270)

multiplicationX: 字体 X 轴方向放大倍数。

multiplicationY: 字体 Y 轴方向放大倍数。

Space: 行间距。

Alignment: 对齐方式。1: 左对齐, 2: 居中, 3: 右对齐

Content: 打印内容。

返回:

大于 0: 正常, 否则异常。

例子:

```
HPRTPrinterHelper.printAreaSize("100","200")
```

```
HPRTPrinterHelper.CLS()
```

```
HPRTPrinterHelper.printBlock(0,0,100,100,0,0,2,2,16,2,"Test,Test,Test,Test")
```

```
HPRTPrinterHelper.Print("1","1")
```

## 二 通讯部分函数

### 1.1 蓝牙连接函数

`int PortOpen(String portSetting)`

参数:

portSetting: “Bluetooth,蓝牙地址”

返回:

等于 0: 连接成功, -1 连接失败。

例子:

`HPRTPrinterHelper .PortOpen(“Bluetooth,蓝牙地址”)`

### 1.2 蓝牙断开函数

`int PortClose()`

参数:

无

返回:

等于 0: 断开成功, -1 断开失败。

例子:

`HPRTPrinterHelper .PortClose()`

### 1.3 蓝牙是否连接

`boolean IsOpened()`

参数:

无

返回:

等于 true: 连接, false: 断开。

例子:

`HPRTPrinterHelper .IsOpened()`

## 2.1 USB 连接函数

`int PortOpen(Context context, UsbDevice usbdevice)`

参数:

usbdevice: USB 设备对象。

返回:

等于 0: 连接成功, -1 连接失败。

例子:

`HPRTPrinterHelper .PortOpen(usbdevice)`

## 2.2 USB 断开函数

`int PortClose()`

参数:

无

返回:

等于 0: 断开成功, -1 断开失败。

例子:

`HPRTPrinterHelper .PortClose()`

## 2.3 USB 是否连接

`boolean IsOpened()`

参数:

无

返回:

等于 true: 连接, false: 断开。

例子:

`HPRTPrinterHelper .IsOpened()`

### 3.1 WIFI 连接函数

`int PortOpen(String portSetting)`

参数:

portSetting: "WiFi,"+strIP+", "+strPort  
strIP: IP 地址,  
strPort: 端口号 (默认 9100)

返回:

等于 0: 连接成功, -1 连接失败。

例子:

`HPRTPrinterHelper .PortOpen("WiFi,"+strIP+", "+strPort)`

### 3.2 WIFI 断开函数

`int PortClose()`

参数:

无

返回:

等于 0: 断开成功, -1 断开失败。

例子:

`HPRTPrinterHelper .PortClose()`