

Android TSPL SDK 说明文档

Android TSPL SDK 说明文档	1
一 初始化	2
1.1 初始化	2
1.2 PrinterTSPL 介绍	2
二 通讯部函数	3
2.1 蓝牙连接函数	3
2.2 USB 连接函数	4
2.3 WIFI 连接函数	5
2.4 断开连接	6
三 SDK 指令函数	7
3.1 页标签开始指令	7
3.2 页标签结束指令（行模式除外）	8
3.3 打印打印机信息	9
3.4 设置标签纸的间距	10
3.5 走纸	11
3.6 清除打印缓冲区内容	12
3.7 文本打印，包含一些变体	12
3.9 定义标签的参考起点	14
3.10 定义国际字符集代码	14
3.11 条码	15
3.12 打印二维码	16
3.13 标签进纸指定长度	17
3.14 将标签定位到下个标签起始处	17
3.15 打印线条	17
3.16 将标签定位到内部传感器确定的起点	18
3.17 打印图片	18
3.18 打印浓度	18
3.19 打印速度	19
3.20 切纸	19
3.21 打印矩形框	20
3.22 设置蜂鸣器鸣叫时间	20
3.23 发数据函数	20
3.24 读数据函数	21
3.25 反白框	21
3.26 文字加粗	22
3.27 获取打印机状态	22
3.28 文本段落打印	23
3.29 打印 PDF417	24
3.30 设置黑标位置	26
3.31 获取打印机 SN 号	27

一 初始化

1.1 初始化

```
public class App extends Application {  
    @Override  
    public void onCreate() {  
        super.onCreate();  
        PrinterHelper.init(this);  
    }  
}
```

1.2 PrinterTSPL 介绍

该类包含打印机相关的所有接口，你可以通过连接接口获取到该对象。如果需要多连接就保存多个该对象，每个对象表示一个连接。

你可以通过该对象里面的 **getPrinterSN** 接口获取打印机的序列号来识别打印机。也可以通过 **getPrinterTAG()**接口获取 SDK 内部标识, 如果是通过蓝牙连接的该标识是蓝牙地址, 如果是 wifi 连接的则是 IP 地址, 如果是 USB 则是设备节点。

二 通讯部函数

2.1 蓝牙连接函数

```
public static PrinterTSPL connectBT(String mac)
```

参数:

mac: “蓝牙地址(大写)”

返回:

printer == null 连接失败,

printer != null | 连接成功。

例子:

```
PrinterTSPL printer = PrinterHelper.connectBT("00:01:02:03:04:05");
```

2.2 USB 连接函数

```
public static PrinterTSPL connectUSB(UsbDevice usbdevice)
```

参数:

usbdevice: USB 设备对象。

返回:

printer == null 连接失败,
printer != null | 连接成功。

例子:

```
PrinterTSPL printer = PrinterHelper.connectUSB(device);
```

2.3 WIFI 连接函数

```
public static PrinterTSPL connectWifi(String address)
```

参数:

address: IP 地址

返回:

printer == null 连接失败,
printer != null | 连接成功。

例子:

```
PrinterTSPL printer = PrinterHelper.connectWifi("192.168.1.1");
```

2.4 断开连接

`public boolean portClose()`

参数:

无

返回:

`true`: 断开成功, `false` 断开失败。

例子:

```
PrinterTSPL printer = PrinterHelper.connectWifi( "192.168.1.1" );  
printer.portClose()
```

三 SDK 指令函数

3.1 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:

```
printer.printAreaSize("100", "100")
printer.printText("0","0","9","0","1","1","TEXT")//打印文字 TEXT
printer.print("1","1")
```

3.2 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:

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

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

```
printer.print("1","1")
```


3.3 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:

`printer.selfTest()`

3.4 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:

`printer.gap("5","5")`

3.5 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:

```
printer.offset("5")
```

3.6 清除打印缓冲区内容

`int cls()`

返回:

大于 0: 正常, 否则异常。

例子:

`printer.cls()`

3.7 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 x25 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:

```
printer.printAreaSize("100","200")
printer.cls();
printer.printText("0","0","1","0","0","0","TEXT")
printer.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:

```
printer.printAreaSize("100","200")
printer.cls();
printer.printText("0","0","0","0","0","0","TEXT")
printer.print("1","1")
```

3.8 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:

```
printer.printAreaSize("100","200")
printer.direction("0");
printer.printText("0","0","0","0","0","TEXT")
printer.print("1","1")
```

3.9 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:

```
printer.reference("0","0")
```

3.10 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:

```
printer.codepage("437")
```

3.11 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:

```
printer.printAreaSize("100","200")
printer.cls();
printer.printBarcode("0","0","128","100","1","0","1","1","1234567890")
printer.print("1","1")
```

3.12 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:

```
printer.printAreaSize("100","200")  
printer.cls();  
printer.printQRCode("0","0","M","6","A","0","1234567890")  
printer.print("1","1")
```


3.13 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:

```
printer.feed("5")
```

3.14 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:

```
printer.formFeed()
```

3.15 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:

```
printer.printAreaSize("100","200")
printer.bar("10","10","100","2")//横线
printer.bar("10","10","2","100")//竖线
printer.print("1","1")
```

3.16 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:

```
printer.home()
```

3.17 打印图片

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

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)。

Type: 图片算法。

0: 二值算法。

1: 半色调算法。

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:

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

3.18 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:

```
printer.density("5")
```

3.19 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:

```
printer.speed("2")
```

3.20 切纸

```
int cut()
```

返回:

大于 0: 正常, 否则异常。

例子:

```
printer.cut()
```

3.21 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:

```
printer.printAreaSize("100","200")
printer.cls()
printer.box("10","10","100","100","2")
printer.print("1","1")
```

3.22 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:

```
printer.sound("8","8")
```

3.23 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:

```
printer.writeData(bData)//向打印机发送数据
```

3.24 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=printer.readData(2000)//bData 是读取出来的数据
```

3.25 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:

```
printer.printAreaSize("100","200")
printer.cls()
printer.printText("100","100","9","0","32","32","REVERSE")
printer.reverse("90","90","128","40")
printer.print("1","1")
```



3.26 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:

`printer.bold(1)`

3.27 获取打印机状态

`int getPrinterStatus()`

参数:

无

返回:

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

例子:

`printer.getPrinterStatus()`

3.28 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:

```
printer.printAreaSize("100","200")  
printer.cls()  
printer.printBlock(0,0,100,100,0,0,2,2,16,2,"Test,Test,Test,Test")  
printer.print("1","1")
```

3.29 打印 PDF417

```
int printPDF417(int x_pos, int y_pos, int width, int height, int rotate, ArrayList<String>  
    option, String expression)
```

参数:

x_pos: x 坐标。

y_pos: y 坐标。

width: 条码的宽度。

height: 条码的高度。

rotate: 旋转方向。 (0, 90, 180, 270)

option: 条码参数。 (可以传 null)

P 数据压缩方法

0: 自动编码

1: 二进制模式

E 误差校正级别。 范围: 0~8

M 条形码区域的中心图案

0: 图案打印在左上区域

1: 图案打印在中部区域

Ux, y, c 显示文本。

X: 文本 x 坐标

Y: 文本 y 坐标

C: 一行最大字符数

W 模块宽度。 范围: 2~9

H 栏高点。 范围: 4~99

R 最大行数

C 最大列数

T 截断。

0: 未截断

1: 截断

Lm 内容长度, 1<m<2048

expression: 条码内容。

返回:

大于 0: 正常, 否则异常。

例子:

```
printer.printAreaSize("100", "100");  
printer.cls();  
ArrayList<String> option = new ArrayList<>();  
option.add("P1");
```



```

option.add("E4");
option.add("M1");
option.add("U100,500,10");
option.add("W6");
option.add("H6");
option.add("R60");
option.add("C4");
option.add("T1");
option.add("L297");
String data = "Data" +
    "compression method: P1" +
    "Error correction level: E4" +
    "Center pattern in barcode area: M1" +
    "Human Readable: Yes: U100,500,10" +
    "Module Width 6 dots: W6" +
    "Bar Height 6 dots: H6" +
    "Maximum Number of Rows: 60 Rows: R60" +
    "Maximum number of columns: 4 Cols: C4" +
    "Truncation:1: T1" +
    "Expression length:297: L297";
printer.printPDF417(50, 50, 900, 600, 0, option, data);
printer.print("1", "1");

```

3.30 设置黑标位置

`boolean setBlackPosition(int position)`

参数:

position:

- 1: 2 寸右前黑标,
- 2: 3 寸右前黑标,
- 3: 2 寸右后黑标,
- 4: 3 寸右后黑标

返回:

true: 发送成功, false: 发送失败

例子:

```
printer.setBlackPosition(1);
```

3.31 获取打印机 SN 号

String getPrinterSN()

参数:

无

返回:

打印机的序列号, (空字符串表示失败)

例子:

```
printer.getPrinterSN();
```