

Android ZPL SDK 說明文檔

Android ZPL SDK 说明文档	錯誤! 尚未定義書籤。
一、SDK 的组成	錯誤! 尚未定義書籤。
1.1 SDK jar 包	錯誤! 尚未定義書籤。
1.2 如何使用 ZPLPrinterHelper.....	錯誤! 尚未定義書籤。
二、 连接方式	錯誤! 尚未定義書籤。
2.1 蓝牙连接	錯誤! 尚未定義書籤。
2.2 WIFI 连接	錯誤! 尚未定義書籤。
2.3 USB 连接	錯誤! 尚未定義書籤。
三、打印指令	錯誤! 尚未定義書籤。
3.1 标签开始	錯誤! 尚未定義書籤。
3.2 设置坐标	錯誤! 尚未定義書籤。
3.3 标签结束	錯誤! 尚未定義書籤。
3.4 字段开头指令.....	錯誤! 尚未定義書籤。
3.5 字段结束指令.....	錯誤! 尚未定義書籤。
3.6 打印宽度	錯誤! 尚未定義書籤。
3.7 打印方向和对齐方式	錯誤! 尚未定義書籤。
3.8 打印直线	錯誤! 尚未定義書籤。
3.9 打印圆	錯誤! 尚未定義書籤。
3.10 打印斜线.....	錯誤! 尚未定義書籤。

3.11 设置标签原点位置	錯誤!	尚未定義書籤。
3.12 整体向左偏移	錯誤!	尚未定義書籤。
3.13 设置文本框（可自动换行）	錯誤!	尚未定義書籤。
3.14 二维码	錯誤!	尚未定義書籤。
3.15 条码	錯誤!	尚未定義書籤。
3.16 打印图片	錯誤!	尚未定義書籤。
3.17 发数据函数	錯誤!	尚未定義書籤。
3.18 读数据函数	錯誤!	尚未定義書籤。
3.19 自检页	錯誤!	尚未定義書籤。
3.20 打印文本	錯誤!	尚未定義書籤。
3.21 打印数量和切刀	錯誤!	尚未定義書籤。
3.22 打印模式	錯誤!	尚未定義書籤。
3.23 写入 RFID	錯誤!	尚未定義書籤。
3.24 读取 RFID	錯誤!	尚未定義書籤。

一、SDK 的組成

1.1 SDK jar 包

添加方式如圖 

這個 jar 包裡面包含了我們的所有的連接函數，包括了藍牙連接、WIFI 連接、USB 連接。以及我們的所有的指令介面函數。這些介面都在 ZPLPrinterHelper 這個類中。

1.2 如何使用 ZPLPrinterHelper

這個類用的是單例模式，通 ZPLPrinterHelper.getZPL(mContext)獲得它的引用。有了它的引用你就可以調用連接函數和指令函數。

二、連接方式

2.1 藍牙連接

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

參數：

mContext：上下文對象。

portSetting：“Bluetooth,”+MAC。（MAC：藍牙地址）

返回：

0：連接成功。

-1：連接失敗。

斷開藍牙

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

例子：

```
mZPL.PorClose()
```

返回：

true：斷開成功。

false：斷開失敗。

藍牙是否連接

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

注意：

這個不是即時監測藍牙連接狀態，監測藍牙狀態需要通過系統的藍牙廣播。

例子：

```
ZPLPrinterHelper.IsOpened()
```

返回：

true：藍牙已連接。

false：藍牙未連接

2.2 WIFI 連接

連接 WIFI：

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

例子：

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

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

IP:印表機的 IP 地址。

PortNumber：埠。 默認：9100

返回：

0：連接成功。

-1：連接失敗。

斷開 WiFi：

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

例子：

```
mZPL.PorClose()
```

返回：

true：斷開成功。

false：斷開失敗。

WiFi 是否連接：

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

例子：

```
ZPLPrinterHelper.IsOpened()
```

返回：

true：已連接。

false：未連接。

2.3 USB 連接

連接 USB：

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

例子：

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

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

usbdevice：UsbDevice 的對象

返回：

0：連接成功。

-1：連接失敗。

斷開 USB：

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

例子：

```
mZPL.PorClose()
```

返回：

true：斷開成功。

false：斷開失敗。

USB 是否連接：

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

例子：

```
mZPL.IsOpened()
```

返回：

true：已連接。

false：未連接。

三、列印指令

3.1 標籤開始

函數：

```
int start();
```

返回：

大於 0：正常，否則異常。

例子：

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

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

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

3.2 設置座標

函數： `int setXY(String x,String y);`

參數：

X：橫坐標。

Y：縱坐標。

返回：

大於 0：正常，否則異常。

3.3 標籤結束

函數：

```
int end();
```

返回：

大於 0：正常，否則異常。

例子：

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

3.4 欄位開頭指令

函數：

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

參數：

A：文本

返回：

大於 0：正常，否則異常。

例子：

```
mZPL.start();  
  
mZPL.setXY("0","0");  
  
mZPL.FD("TEXT");  
  
mZPL.FS();  
  
mZPL.end();
```

3.5 欄位結束指令

函數：

```
int FS();
```

返回：

大於 0：正常，否則異常。

例子：

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

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

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

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

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

3.6 列印寬度

函數：

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

參數：

A：寬度（in dots）。

返回：

大於 0：正常，否則異常。

例子：

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

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

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

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

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

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

3.7 列印方向和對齊方式

函數：

```
int FW(String rotate,String justification)
```

參數：

rotate：列印方向，取值如下：

N = normal

R = rotated 90 degrees

I = inverted 180 degrees

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

Justification：對其方式，取值如下：

0 = left justification

1 = right justification

2 = auto justification (script dependent)

返回：

大於 0：正常，否則異常。

例子：

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

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

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

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

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

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

3.8 列印直線

函數：

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

參數：

w：直線的寬度。（1~32000 單位 點）

H：直線的高度。（1~32000 單位 點）

T：線條的寬度。（1~32000 單位 點）

C：線條的顏色：

B=black；（默認：B）

W=white；

R：圓角的弧度（0~8）

返回：

大於 0：正常，否則異常。

例子：

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

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

```
mZPL.printLine("100","2","2","B","0")//列印 100 點寬，2 點高的橫線
```

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

3.9 列印圓

函數：

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

參數：

d：圓的直徑（3~4095）。

t：邊框厚度（1~4095）。

C：線條顏色：

B=black；（默認：B）

W=white；

返回：

大於 0：正常，否則異常。

例子：

```
mZPL.start();  
mZPL.setXY("0","0");  
mZPL.printCircle("100","2","B")  
mZPL.end();
```

3.10 列印斜線

函數：

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

參數：

w：斜線的寬度（3~32000）。

H：斜線的高度（3~32000）。

T：線條寬度（1~32000）。

C：線條顏色：

B=black；（默認：B）

W=white；

O：斜線的方向：

R (or /) = right-leaning diagonal

L (or \) = left-leaning diagonal

（默認：R）

返回：

大於 0：正常，否則異常。

例子：

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

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

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

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

3.11 設置標籤原點位置

函數：

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

參數：

X：原點的橫坐標。

Y：原點的從座標。

返回：

大於 0：正常，否則異常。

例子：

略

3.12 整體向左偏移

函數：

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

參數：

a：偏移量（-9999~9999）負數代表向右偏移，（默認 0）。

返回：

大於 0：正常，否則異常。

例子：

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

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

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

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


3.13 設置文字方塊（可自動換行）

函數

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

參數：

a：文字方向：

N = normal

R = rotate 90 degrees clockwise

I = invert 180 degrees

B = read from bottom up-270 degrees

b：文字方塊的寬度。

c：文字方塊的高度。

返回：

大於 0：正常，否則異常。

例子：

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

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

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

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

3.14 二維碼

函數：

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

參數：

x：橫坐標。

y：縱坐標。

orientation：方向（N）。

magnification：模式：

1=普通模式。

2=加強模式。

默認（2）

size：尺寸（1~10）。

data:二維碼的內容

返回：

大於 0：正常，否則異常。

例子：

```
mZPL.start();  
mZPL.printData("^CI14\r\n");  
mZPL.printQRcode("10","10","N","2","5","abc123");  
mZPL.end();
```

3.15 條碼

函數：

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

參數：

x：橫坐標。

y：縱坐標。

Type：條碼類型：

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：條碼高度（1-32000）。

F：條碼內容是否見（預設 Y）：

Y = yes

N = no

返回：

大於 0：正常，否則異常。

例子：

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

3.16 列印圖片

函數：

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

參數：

x：起始的 X 座標。

y：起始的 Y 座標。

Bmp：圖片物件。

返回：

大於 0：正常，否則異常。

例子：

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

3.17 發資料函數

函數：

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

參數：

bData：需要發給印表機的資料。

返回：

大於 0：正常，否則異常。

例子：

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

3.18 讀數據函數

函數：

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

參數：

outTime：超時時間（單位秒）。

返回：

讀取到的數據。

例子：

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

3.19 自檢頁

函數：

```
int selfTest()
```

返回：

大於 0：正常，否則異常。

例子：

```
mZPL.selfTest()
```

3.20 列印文本

注意：列印中文時需要選擇編碼，請參照例子。

函數：

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

參數：

x：橫坐標。

y：縱坐標。

type：字體（0~6：中文無效，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：字體大小。

1：10px。

2：20px。

3：30px。

4：40px。

5：50px。

6：60px。

data：文本資料。

返回：

大於 0：正常，否則異常。

例子：

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

```
mZPL.printData("^CI14\r\n")//列印中文時需要加入這句
```

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

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


3.21 列印數量和切刀

函數：

```
int PQ(String q,String p,String r,String o)
```

參數：

q:列印數量

p:暫停前或者切刀前數量

r:每個序號的副本數

o:切刀或者暫停

Y:切刀

N:暫停

返回：

大於 0：正常，否則異常。

例子：

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

```
mZPL.printData("^CI14\r\n")//列印中文時需要加入這句
```

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

```
mZPL.PQ(1,1,1,Y);//列印一張後切刀
```

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

3.22 列印模式

函數：

```
int setPrinterModel(String model)
```

參數：

model:

T = 撕扯

P = 剝離 (在 S-300 上不可用)

返回：

大於 0：正常，否則異常。

例子：

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

```
mZPL.printData("^CI14\r\n")//列印中文時需要加入這句
```

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

```
mZPL.setPrinterModel("P");//切換成剝離模式
```

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

3.23 寫入 RFID

函數：

```
int writeRFID(int address, int memory,byte[] data)
```

參數：

address：起始地址，範圍：大於 0。(EPC 區地址必須從 2 開始)

memory：寫入區域, (0=保留區, 1 是 EPC 區, 3 是 User 區)

data：需寫入的資料。(保留區, EPC 不超過 12 個位元組, User 不超過 128)

返回：

-1：發送失敗，-2：參數錯誤，0：寫入成功。

例子：

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

```
mZPL.writeRFID(0,1,"Test RFID".getBytes());
```

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

3.24 讀取 RFID

函數：

```
byte[] readRFID(int address,int length,int memory)
```

參數：

address：起始地址，範圍：大於 0。(EPC 區地址必須從 2 開始)

length：讀取的長度。(保留區，EPC 不超過 12 個位元組，User 不超過 128)

memory：寫入區域，(0=保留區，1 是 EPC 區，2 是 TID 區 3 是 User 區)

返回：

讀取到的資料，空表示讀取失敗。

例子：

```
mZPL.start();  
  
mZPL.writeRFID(2,1,"中文".getBytes("GB2312"));  
  
mZPL.readRFID(2, 4, 1);  
  
mZPL.end();  
  
byte[] bytes = mZPL.ReadData(3);  
  
if (bytes!=null&&bytes.length>0){  
    String hexStr = new String(bytes);  
    byte[] hexByte = UtilityTooth.hexToByte(hexStr);  
    Toast.makeText(thisCon,newString(hexByte,"GB2312"),  
    Toast.LENGTH_SHORT).show();  
}
```