

Android CPCL SDK 说明文档

一 SDK 加载及使用.....	3
二 SDK 连接方法.....	3
2.1 蓝牙连接函数.....	3
2.2 WIFI 连接函数.....	3
2.3 断开函数.....	4
2.4 蓝牙是否连接.....	4
三 SDK 打印方法.....	4
3.1 页标签开始指令.....	4
3.2 页标签结束指令.....	5
3.3 编码指令.....	6
3.4 走纸到下一张标签.....	7
3.5 注释.....	8
3.6 终止指令.....	9
3.7 文本打印.....	10
3.8 计数.....	13
3.9 设置字符宽高放大倍数.....	14
3.10 对齐方式.....	15
3.11 条码.....	16
3.12 打印二维码.....	18
3.13 打印 PDF417 码.....	19
3.14 画矩形框.....	20
3.15:画直线.....	21
3.16 反白框.....	22
3.17 打印图片.....	23
3.18 打印浓度.....	24
3.19 打印速度.....	25
3.20 设置字间距.....	26
3.21 走纸再打印.....	27
3.22 打印后走纸一定距离.....	28
3.23 设置蜂鸣器鸣叫时间.....	29
3.24 下划线.....	30
3.25 打印一页标签后延时.....	31
3.26 打印宽度.....	32
3.27 行模式下设置行间距.....	33
3.28 设置字体与字符大小及行间距.....	34
3.29 发数据函数.....	35
3.30 读数据函数.....	35
3.31 字体加粗.....	36
3.32 获取打印机状态.....	37
3.33 文字自动换行.....	38
3.34 文字在文本框内居中显示.....	40
3.35 设置打印机纸张类型.....	41

3.36 自检页.....	42
3.37 旋转 180 度打印.....	43
3.38 获取打印完成时状态开关.....	44
3.39 获取打印完成时状态.....	45
3.40 打印机回退.....	46
3.41 水印.....	47
3.42 获取打印机 SN.....	48
3.43 设置 Codepage.....	49
3.44 设置 Qrcode 版本.....	50
3.45 读取 Qrcode 版本.....	51
3.46 关闭高棉语输入指令.....	52
3.47 获取文字自动换行打印的高度.....	53
四 行打印模式.....	54
4.1 行模式设置字体样式.....	54
4.2 行打印模式打印文本.....	54
4.3 行打印模式字体加粗.....	55
4.4 设置行模式的 X 坐标.....	55

一 SDK 加载及使用

- 1.1 在 Android studio 中加载我们打印机的 SDK jar 包。
compile files('libs/CPCL_V1.01.01PRO.jar')
- 1.2 我们所用的接口都在 PrinterHelper 这个类中所有的接口都是静态方法，可以直接调用。

二 SDK 连接方法

2.1 蓝牙连接函数

```
public static int PortOpenBT(String mac)
```

参数:

mac: 蓝牙地址。

返回:

0: 连接成功, -1: 连接异常, -2: 蓝牙地址错误,
-3: 打印机与 SDK 不匹配（握手指令错误）。

例子:

```
if(PrinterHelper.PortOpenBT(蓝牙地址)==0){  
    //连接成功  
}
```

2.2 WIFI 连接函数

```
public static int PortOpenWIFI(String printIP)
```

参数:

printIP: 打印机 IP 地址（查看自检页）。

返回:

0: 连接成功, -1: 连接异常, -2: 地址格式错误,
-3: 打印机与 SDK 不匹配（握手指令错误）。

例子:

```
if(PrinterHelper.PortOpenWIFI(IP)==0){  
    //连接成功
```

2.3 断开函数

int **PortClose()**

参数:

无

返回:

等于 0: 连接成功, -1 连接失败。

例子:

PrinterHelper .PortClose()

2.4 蓝牙是否连接

boolean **IsOpened()**

注意: 该接口只显示 PortOpen 和 PortClose 的结果, 并不是监测实时的蓝牙状态 (监测蓝牙的实时连接状态需要监听系统蓝牙广播)。

参数:

无

返回:

等于 true: 连接, false: 断开。

例子:

PrinterHelper .IsOpened()

三 SDK 打印方法

3.1 页标签开始指令

int **printAreaSize**(String **offset**,String **Horizontal**, String **Vertical**,String **height**,String **qty**)

参数:

offset: 此值使所有字段将水平偏移指定的单位数量。

Horizontal: 水平方向的 dpi。（根据打印机的 dpi 设置，200dpi 打印机：8px=1mm）

Vertical: 垂直方向的 dpi。（同上）

Height: 整个标签的高度。（单位：px）

Qty: 打印的次数。

返回:

大于 0: 正常，否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","0","TEXT")
PrinterHelper .Form()//打印标签起定位作用（连续纸不可用）
PrinterHelper .Print()
```

3.2 页标签结束指令

注意: 行模式不能用。

int **Print()**

返回:

大于 0: 正常，否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","0","TEXT")
PrinterHelper .Form()//打印标签定位作用（连续纸不可用）
PrinterHelper .Print()
```

3.3 编码指令

int **Encoding**(String **code**):

参数:

Code: 编码。

返回:

大于 0: 正常, 否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
PrinterHelper .Encoding(GB18030)
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","0","中文")
PrinterHelper .Form()
PrinterHelper .Print()
```

3.4 走纸到下一张标签

注意：（必须跟 PRINT 在一起才有效，指令只对标签有效）

int **Form()**

返回：

大于 0：正常，否则异常。

例子：

```
PrinterHelper .printAreaSize("0","200","200","100","1")
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","0","TEXT")
PrinterHelper .Form()//打印标签定位作用（连续纸不可用）
PrinterHelper .Print()
```

3.5 注释

int **Note**(String **note**)

参数:

note: 注释的内容。

返回:

大于 0: 正常, 否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
```

```
PrinterHelper .Note("注释: ")
```

```
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","0","TEXT")
```

```
PrinterHelper .Form()
```

```
PrinterHelper .Print()
```


3.6 终止指令

int **Abort()**

返回:

大于 0: 正常, 否则异常。

3.7 文本打印

注意：文本打印有三个接口 **PrintTextCPCL** 和 **Text** 还有 **PrintCodepageTextCPCL**

PrintTextCPCL 用于中文固件。

PrintCodepageTextCPCL 用于英文固件。

Text 两种固件都能用。

(1) **int PrintTextCPCL(String command,int font ,String x,String y,String data,int n,boolean Iscenter,int width)**

参数：

Command: 文字的方向，总的有两种：

PrinterHelper.TEXT: 水平。

PrinterHelper.TEXT270: 垂直。

Font: 字体点阵大小（单位：px）

1: 打印繁体字（24x24 或者 12x24，视中英文而定。）

16: 16x16 或 8x16，视中英文而定。

24: 24x24 或 12x24，视中英文而定。

32: 32x32 或 16x32，由 ID3 字体宽高各放大 2 倍。

X: 起始点的横坐标。（单位：px）

Y: 起始点的纵坐标。（单位：px）

Data: 文本数据。

N: 字体的特效：

N 的低 4 位	3	2	1	0
加粗	-	-	-	1
反白	-	-	1	-
倍宽	-	1	-	-
倍高	1	-	-	-

Iscenter: 是否居中。

True: 是。

False: 否。

Width: 要居中的范围。（Iscenter=true 时才生效）单位：px。

返回：

大于 0: 正常，否则异常。

例子：

```
PrinterHelper .printAreaSize("0","200","200","500","1")
```

```
//15 表示所有的特效都有
```

```
PrinterHelper .PrintTextCPCL(PrinterHelper.TEXT,24,"10","10","TEXT",15,false,0)
```

```
PrinterHelper .Form()
```

```
PrinterHelper .Print()
```

(2) int **Text**(String **command**,String **font**,String **size** ,String **x**,String **y**,String **data**)

参数:

Command: 文字的方向，总的有四种：

PrinterHelper.TEXT: 水平。

PrinterHelper.TEXT90: 逆时针旋转 90 度。

PrinterHelper.TEXT180: 逆时针旋转 180 度。

PrinterHelper.TEXT270: 逆时针旋转 270 度。

font: 字体点阵大小：（单位：px）

注意：英文固件只支持（0 和 1）。

0: 12x24。

1: 12x24（中文模式下打印繁体），英文模式下字体变成（9x17）大小

2: 8x16。

3: 20x20。

4: 32x32 或者 16x32，由 ID3 字体宽高各放大两倍。

7: 24x24 或者 12x24，视中英文而定。

8: 24x24 或者 12x24，视中英文而定。

20: 16x16 或者 8x16，视中英文而定。

24: 24x24 或者 12x24，视中英文而定。

55: 16x16 或者 8x16，视中英文而定。

其它默认 24x24 或者 12x24，视中英文而定。

size : 字体大小。（该功能被屏蔽统一参数传 0）

X: 起始点的横坐标。（单位：px）

Y: 起始点的纵坐标。（单位：px）

Data: 文本数据。

例子:

```
PrinterHelper .printAreaSize("0","200","200","500","1")
```

```
PrinterHelper .SetBold("1")//对下面的字体进行加粗（如不需要加粗不用添加）
```

```
PrinterHelper .SetMag("2","2")//对下面的字体进行放大（如不需要不用添加）
```

```
PrinterHelper .Text(PrinterHelper.TEXT,"7","0","10","10","TEXT")
```

```
PrinterHelper .SetMag("1","1")//关闭放大
```

```
PrinterHelper .SetBold("0")//关闭加粗
```

```
PrinterHelper .Form()
```

```
PrinterHelper .Print()
```

(3) int **PrintCodepageTextCPCL**(String **command**,int **font** ,String **x**,String **y**,String **data**,int **n**)

参数:

Command: 文字的方向, 总的有两种:

PrinterHelper.TEXT: 水平。

PrinterHelper.TEXT270: 垂直。

Font: 字体点阵大小 (单位: px)

0: 12x24。

1: 9x17。

X: 起始点的横坐标。

Y: 起始点的纵坐标。

Data: 文本数据。

N: 字体的特效:

N的低4位	3	2	1	0
加粗	-	-	-	1
反白	-	-	1	-
倍宽	-	1	-	-
倍高	1	-	-	-

返回:

大于 0: 正常, 否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","500","1")
```

```
PrinterHelper.Country("ISO8859-1");//设置打印机编码
```

```
PrinterHelper.LanguageEncode="iso8859-1";//设置 SDK 编码
```

```
//15 表示所有的特效都有
```

```
PrinterHelper .PrintCodepageTextCPCL(PrinterHelper.TEXT,0,"10","10","TEXT",15)
```

```
PrinterHelper .Form()
```

```
PrinterHelper .Print()
```

3.8 计数

int **Count**(String ml)

参数:

ml : 下次加减的数值。

返回:

大于 0: 正常, 否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","2")
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","0","10086")
PrinterHelper .Count("10")
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","0","10000")
PrinterHelper .Count("-10")
PrinterHelper .Form()
PrinterHelper .Print()
```

3.9 设置字符宽高放大倍数

注意：使用后记得关闭。

int SetMag(String width,String height)

参数：

Width：宽的放大倍数。

Height：高的放大倍数。

返回：

大于 0：正常，否则异常。

例子：

```
PrinterHelper .printAreaSize("0","200","200","500","1")
PrinterHelper .SetBold("1")//对下面的字体进行加粗（如不需要加粗不用添加）
PrinterHelper .SetMag("2","2")//对下面的字体进行放大（如不需要不用添加）
PrinterHelper .Text(PrinterHelper.TEXT,"7","0","10","10","TEXT")
PrinterHelper .SetMag("1","1")//关闭放大
PrinterHelper .SetBold("0")//关闭加粗
PrinterHelper .Form()
PrinterHelper .Print()
```

3.10 对齐方式

int **Align**(String align)

参数:

Align: 对齐方式总的有三种:

PrinterHelper.CENTER: 居中。

PrinterHelper.LEFT: 左对齐。

PrinterHelper.RIGHT: 右对齐。

返回:

大于 0: 正常, 否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
PrinterHelper .Align(PrinterHelper.CENTER)
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","0","TEXT")
PrinterHelper .Form()
PrinterHelper .Print()
```

3.11 条码

int **Barcode**(String **command**,String **type**,String **width**, String **ratio**,String **height**,String **x**,String **y**,boolean **undertext**,String **number**,String **size**,String **offset**, String **data**)

参数:

Command: 条码方向:

PrinterHelper.BARCODE: 水平方向

PrinterHelper.VBARCODE: 垂直方向

Type: 条码类型:

PrinterHelper.UPCA,PrinterHelper.UPCA2,

PrinterHelper.UPCA5,PrinterHelper.UPCE,

PrinterHelper.UPCE2,PrinterHelper.UPCE5 ,

PrinterHelper.EAN13,PrinterHelper.EAN132,

PrinterHelper.EAN135,PrinterHelper.EAN8,

PrinterHelper.EAN82,PrinterHelper.EAN85,

PrinterHelper.code39, PrinterHelper.code39C,

PrinterHelper.F39,PrinterHelper.F39C,

PrinterHelper.code93,PrinterHelper.I2OF5,

PrinterHelper.I2OF5C,PrinterHelper.I2OF5G,

PrinterHelper.code128,PrinterHelper.UCC EAN128,

PrinterHelper.CODABAR,PrinterHelper.CODABAR16,

PrinterHelper.MSI, PrinterHelper.MSI10,

PrinterHelper.MSI1010,PrinterHelper.MSI1110,

PrinterHelper.POSTNET,PrinterHelper.FIM

Width: 窄条的单位宽度。

Ratio: 宽条窄条的比例。比例如下:

0 = 1.5 : 1	20 = 2.0:1	26 = 2.6:1
1 = 2.0 : 1	21 = 2.1:1	27 = 2.7:1
2 = 2.5 : 1	22 = 2.2:1	28 = 2.8:1
3 = 3.0 : 1	23 = 2.3:1	29 = 2.9:1
4 = 3.5 : 1	24 = 2.4:1	30 = 3.0:1
	25 = 2.5:1	

Height: 条码高度。

X: 条码的起始横坐标。(单位: px)

Y: 条码的起始纵坐标。(单位: px)

Undertext: 条码下方的数据是否可见。ture: 可见, false: 不可见。

Number: 字体的类型。

Size: 字体的大小。

Offset: 条码与文字间的距离。

Data: 条码数据。

返回:

大于 0: 正常, 否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
PrinterHelper .Barcode(PrinterHelper.BARCODE,PrinterHelper.128,"1","1","50","0","0",
true,"7","0","5","123456789")
PrinterHelper .Form()
PrinterHelper .Print()
```

3.12 打印二维码

int **PrintQR**(String **command**, String **x**, String **y**, String **M** , String **U**, String **data**)

参数:

Command: 打印方向:

PrinterHelper.BARCODE: 水平方向。

PrinterHelper.VBARCODE: 垂直方向。

X: 二维码的起始横坐标。

Y: 二维码的起始纵坐标。

M: QR 的类型: 类型 1 和类型 2; 类型 2 是增加了个别的符号, 提供了而外的功能。

U: 单位宽度/模块的单元高度。

范围是 1 到 32 默认为 6。

Data: 二维码的数据。

返回:

大于 0: 正常, 否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
```

```
PrinterHelper .PrintQR(PrinterHelper.BARCODE, "0", "0", "2", "6", "123ABC" )
```

```
PrinterHelper .Form()
```

```
PrinterHelper .Print()
```

3.13 打印 PDF417 码

int **PrintPDF417**(String **command**, String **x**, String **y**, String **XD** , String **YD**, String **C**, String **S** ,
String **data**)

参数:

Command: 打印方向:

PrinterHelper.BARCODE: 水平方向。

PrinterHelper.VBARCODE: 垂直方向。

X: 二维码的起始横坐标。

Y: 二维码的起始纵坐标。

XD: 最窄元素的单位宽度。范围是 1 到 32，默认为 2。

YD: 最窄元素的单位高度。范围是 1 到 32，默认值是 6。

C: 使用的列数。数据列不包括启动/停止字符和左/右指标。范围为 1 到 30;默认值是 3。

S: 安全级别表示要检测到的错误的最大金额和/或校正。范围为 0 到 8;默认值是 1。

Data: PDF417 码的数据。

返回:

大于 0: 正常，否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
PrinterHelper .PrintPDF417(PrinterHelper.BARCODE, "0", "0", "2" , "6",
"3","1","123ABC" )
PrinterHelper .Form()
PrinterHelper .Print()
```

3.14 画矩形框

int **Box**(String **X0**,String **Y0**,String **X1**,String **Y1**,String **width**)

参数:

X0: 左上角的 X 坐标。（单位：px）

Y0: 左上角的 Y 坐标。（单位：px）

X1: 右下角的 X 坐标。（单位：px）

Y1: 右下角的 Y 坐标。（单位：px）

Width: 线条的单位宽度。

返回:

大于 0: 正常，否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","200","1")
```

```
PrinterHelper .Box("0","0","150","150","1")
```

```
PrinterHelper .Form()
```

```
PrinterHelper .Print()
```

3.15:画直线

int **Line**(String **X0**,String **Y0**,String **X1**,String **Y1**,String **width**)

参数:

X0: 起始的 X 坐标。（单位： px）

Y0: 起始的 Y 坐标。（单位： px）

X1: 结尾的 X 坐标。（单位： px）

Y1: 结尾的 Y 坐标。（单位： px）

Width: 线条的单位宽度。

返回:

大于 0: 正常，否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","200","1")
```

```
PrinterHelper .Line("10","10","150","10","1")
```

```
PrinterHelper .Form()
```

```
PrinterHelper .Print()
```

3.16 反白框

int **InverseLine**(String **X0**,String **Y0**,String **X1**,String **Y1**,String **width**)

参数:

X0: 起始的 X 坐标。（单位：px）

Y0: 起始的 Y 坐标。（单位：px）

X1: 结尾的 X 坐标。（单位：px）

Y1: 结尾的 Y 坐标。（单位：px）

Width: 反白框的宽度。

返回:

大于 0: 正常，否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","200","1")
```

```
PrinterHelper .InverseLine("10","10","150","10","1")
```

```
PrinterHelper .Form()
```

```
PrinterHelper .Print()
```

3.17 打印图片

注意：打印图片有两个接口。

1) `int Expanded(String x, String y, String url)`

参数：

X: 图片起始的 x 坐标。（单位：px）

Y: 图片起始的 y 坐标。（单位：px）

Url: 图片的路径。

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。

2) `int Expanded(String x, String y, Bitmap bmap,int type,int light)`

参数：

X: 图片起始的 x 坐标。（单位：px）

Y: 图片起始的 y 坐标。（单位：px）

bmap: 图片的 Bitmap 的对象。

type: 图片算法。0: 二值算法；1: 半色调算法。

light: 亮度（默认 0）。

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：

```
PrinterHelper .printAreaSize("0","200","200","500","1")
```

```
PrinterHelper .Expanded("0","0",url)
```

```
PrinterHelper .Form()
```

```
PrinterHelper .Print()
```

3.18 打印浓度

int **Contrast**(String **contrast**)

参数:

Contrast: 浓度类型，总的有四种：

默认 =0

中 =1

黑暗 =2

非常深 =3

返回:

大于 0: 正常，否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
PrinterHelper .Contrast("1")
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","0","TEXT")
PrinterHelper .Form()
PrinterHelper .Print()
```


3.19 打印速度

int Speed(String speed)

参数:

speed : 速度类型，总的有 5 种：从 0 到 5 越来越快；5 是理想状态的最快速度。

返回:

大于 0: 正常，否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
PrinterHelper .Speed("4")
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","0","TEXT")
PrinterHelper .Form()
PrinterHelper .Print()
```

3.20 设置字间距

int **SetSp**(String **setsp**)

参数:

Setsp: 间距。

返回:

大于 0: 正常, 否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","0","TEXT")
PrinterHelper .SetSp(1)
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","50","TEXT")
PrinterHelper .SetSp(5)
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","90","TEXT")
PrinterHelper .Form()
PrinterHelper .Print()
```

3.21 走纸再打印

int **Prefeed**(String **prefeed**)

参数:

Prefeed: 走纸的距离。（单位：px）。

返回:

大于 0: 正常，否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
PrinterHelper .Prefeed("40")
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","0","TEXT")
PrinterHelper .Form()
PrinterHelper .Print()
```

3.22 打印后走纸一定距离

int **Postfeed**(String **posfeed**)

参数:

Posfeed: 走纸的距离。（单位: px）

返回:

大于 0: 正常, 否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","0","TEXT")
PrinterHelper .Form()
PrinterHelper .Postfeed("40")
PrinterHelper .Print()
```

注意: 要在 FORM 指令之后

3.23 设置蜂鸣器鸣叫时间

int **Beep**(String **beep**)

参数:

beep : 蜂鸣声的持续时间, (1/8) 秒为单位指定。

返回:

大于 0: 正常, 否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","0","TEXT")
PrinterHelper .Beep("16")
PrinterHelper .Form()
PrinterHelper .Print()
```

3.24 下划线

int **Underline**(boolean **UL**)

参数:

UL : true: 添加下划线, false: 取消下划线。

返回:

大于 0: 正常, 否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
PrinterHelper .Underline(true)
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","0","TEXT")
PrinterHelper .Form()
PrinterHelper .Print()
```

3.25 打印一页标签后延时

int **Wait**(String **wait**)

参数:

Wait: 延时的单位是: 1/8 秒。

返回:

大于 0: 正常, 否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
PrinterHelper .Wait("80")
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","0","TEXT")
PrinterHelper .Form()
PrinterHelper .Print()
```

3.26 打印宽度

int **PageWidth**(String pw)

参数:

Pw: 指定页面宽度。（单位: px）

返回:

大于 0: 正常, 否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
```

```
PrinterHelper .PageWidth("100")
```

```
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","0","TEXT")
```

```
PrinterHelper .Form()
```

```
PrinterHelper .Print()
```


3.27 行模式下设置行间距

int **Setlf**(String **SF**)

参数:

SF: 间距

返回:

大于 0: 正常, 否则异常。

注意: 与 SETLP 效果相同, 若 2 个同时存在, 则以最后设置的参数值为准

3.28 设置字体与字符大小及行间距

int **Setlp**(String **font**,String **size**,String **spacing**)

参数:

Font: 字体。

Size: 大小。

Spacing: 字体的高度。

返回:

大于 0: 正常, 否则异常。

3.29 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。

3.30 读数据函数

byte[] **ReadData**(int **second**)

参数:

second: 读取时间 (单位: 秒)。

返回:

读取到数据。

3.31 字体加粗

int **SetBold**(String **bold**)

参数:

bold: 加粗系数（范围：1-5）。

返回:

大于 0: 正常，否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","500","1")
PrinterHelper .SetBold("1")//对下面的字体进行加粗（如不需要加粗不用添加）
PrinterHelper .SetMag("2","2")//对下面的字体进行放大（如不需要不用添加）
PrinterHelper .Text(PrinterHelper.TEXT,"7","0","10","10","TEXT")
PrinterHelper .SetMag("1","1")//关闭放大
PrinterHelper .SetBold("0")//关闭加粗
PrinterHelper .Form()
PrinterHelper .Print()
```

注意: 使用完后记得关闭加粗。

3.32 获取打印机状态

`int getstatus()`

参数:

无

返回:

0: 打印机准备就绪。

1: 打印机打印中。

2: 打印机缺纸。

6: 打印机开盖。

其他: 出错。

3.33 文字自动换行

注意：该功能有两个接口分别是 **AutLine** 和 **AutLine2**。前者不能使用泰语。后者打印机固件必须在 V1.01.40.01 以上且文本字节数不得超过 1024，超过部分自动忽略（不支持 4 寸机型）。

1) int **AutLine**(String x,String y,int width,int size,boolean isbole,boolean isdouble,String str)

参数：

X: 文字的起始的 x 坐标。（单位：PX）

Y: 文字的起始的 y 坐标。（单位：PX）

Width: 一行打印的宽度。（单位：PX）

Size: 字体。

3: 20x20 或 10x20，视中英文而定。

4: 32x32 或 16x32，由 ID3 字体宽高各放大 2 倍。

8: 24x24 或 12x24，视中英文而定。

55: 16x16 或 8x16，视中英文而定。

isbole: 加粗。

true: 加粗。

false: 不加粗。

isdouble: 放大两倍字体。

true: 放大。

false: 不放大。

Str: 要打印的文本。

返回：

大于 0: 正常，否则异常。

例子：

```
PrinterHelper.printAreaSize("0","200","200","100","1")
```

```
PrinterHelper.AutLine("0","0",100,4,true,true"Text")
```

```
PrinterHelper.Form()
```

```
PrinterHelper.Print()
```

2) int **AutLine2**(String x,String y,int width,int size,boolean isbole,boolean isdouble,String str)

参数：

X: 文字的起始的 x 坐标。（单位：PX）

Y: 文字的起始的 y 坐标。（单位：PX）

Width: 一行打印的宽度（单位：PX）

Size: 字体。

0: 24x24 或 12x24，视中英文而定。（泰语：24x48）

1: 7x19（英文），24x24（繁体）。

3: 20x20 或 10x20，视中英文而定。

4: 32x32 或 16x32，由 ID3 字体宽高各放大 2 倍。

8: 24x24 或 12x24，视中英文而定。

55: 16x16 或 8x16, 视中英文而定。

isbole: 加粗。

true: 加粗。

false: 不加粗。

isdouble: 放大两倍字体。

true: 放大。

false: 不放大。

Str: 要打印的文本。

返回:

大于 0: 正常, 否则异常。

例子:

```
PrinterHelper.printAreaSize("0","200","200","100","1")
```

```
PrinterHelper.AutLine2("0","0",100,4,true,true"Text")
```

```
PrinterHelper.Form()
```

```
PrinterHelper.Print()
```

3.34 文字在文本框内居中显示

int **AutCenter**(String **command**, String **x**,String **y**,int **width**,int **size**,String **str**)

参数:

Command: 文字的方向, 总的有两种:

PrinterHelper.TEXT: 水平。

PrinterHelper.TEXT270: 垂直。

X: 文本框起始的 x 坐标。

Y: 文本框起始的 y 坐标。

Width: 文本框的宽度 (单位: px)。

Size: 字体。

3: 16x16 或 8x16, 视中英文而定。

4: 32x32 或 16x32, 由 ID3 字体宽高各放大 2 倍。

8: 24x24 或 12x24, 视中英文而定。

55: 16x16 或 8x16, 视中英文而定。

Str: 要打印的文本。

返回:

大于 0: 正常, 否则异常。

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
PrinterHelper .AutCenter(PrinterHelper.TEXT,"0","0",100,4,"Text")
PrinterHelper .Form()
PrinterHelper .Print()
```


3.35 设置打印机纸张类型

1) void papertype_CPCL(int page)

page	0	1	2	3	4	5
A300	连续纸	标签纸	后黑标	前黑标	三寸黑标	两寸黑标
T300	连续纸	三寸黑标	两寸黑标	后黑标	标签纸	前黑标

2) void papertype_CPCL_TWO(int page)

注意：该接口适用于 A300 V1.01.38.01 版本和 T300 V1.0.11 版本及以上版本。

参数：page

PrinterHelper .PAGE_STYPE_RECEIPT：连续纸。

PrinterHelper .PAGE_STYPE_LABEL：标签纸。

PrinterHelper .PAGE_STYPE_LEFT_TOP_BM：3 寸左上黑标。

PrinterHelper .PAGE_STYPE_LEFT_BEL_BM：3 寸左下黑标。

PrinterHelper .PAGE_STYPE_RIGHT_TOP_BM：右上黑标。

PrinterHelper .PAGE_STYPE_RIGHT_BEL_BM：右下黑标。

PrinterHelper .PAGE_STYPE_CENTRAL_TOP_BM：中上黑标。

PrinterHelper .PAGE_STYPE_CENTRAL_BEL_BM：中下黑标。

PrinterHelper .PAGE_STYPE_2INCH_LEFT_TOP_BM：2 寸左上黑标。

PrinterHelper .PAGE_STYPE_2INCH_LEFT_BEL_BM：2 寸左下黑标。

3) setPaperFourInch(int type)

注意：该接口用于 4 寸机型设置纸张类型。

参数：type:

PrinterHelper .Paper_FourInch_Receipt：连续纸。

PrinterHelper .Paper_FourInch_Label：标签纸。

PrinterHelper .Paper_FourInch_TWO_BM：2 寸黑标。

PrinterHelper .Paper_FourInch_THREE_BM：3 寸黑标。

PrinterHelper .Paper_FourInch_FOUR_BM：4 寸黑标。

返回：

空

例子：

PrinterHelper .papertype_CPCL(0)//设置为连续纸

3.36 自检页

void **setSelf()**

参数

无

返回：

空

例子：

PrinterHelper.setSelf()//调用后打印机会打印一些打印机参数（不支持 4 寸机）。

3.37 旋转 180 度打印

int **PoPrint()**

参数:

无

返回:

大于 0: 正常, 否则异常。

例子:

```
HPRTPrterHelper.PoPrint()
```

例子:

```
PrinterHelper .printAreaSize("0","200","200","100","1")
PrinterHelper .Text(PrinterHelper.TEXT,"4","0","0","0","TEXT")
PrinterHelper .Form()//打印标签定位作用（连续纸不可用）
PrinterHelper .PoPrint()
```

注意:

它不能与 Print()同时使用。

3.38 获取打印完成时状态开关

`void openEndStatic(boolean isopen)`

参数:

isopen

true: 开。

false: 关。

返回:

空。

例子:

```
PrinterHelper.openEndStatic(true);//开启
PrinterHelper.PrintData(data);//打印机打印
int endStatus = PrinterHelper.getEndStatus(16);//获取打印状态
PrinterHelper.openEndStatic(false);//关闭
```

注意:

它需要跟 `getEndStatus()` 配合使用，使用完记得关闭（打印机固件版本必须 V1.01.27.01 以上）。

3.39 获取打印完成时状态

int getEndStatus(int time)

参数:

time: 获取状态超时时间（单位 秒）。

返回:

0: 打印成功。

1: 打印失败（缺纸）。

2: 打印失败（开盖）。

-1: 获取状态超时。

例子:

```
PrinterHelper.openEndStatic(true);//开启
PrinterHelper.PrintData(data);//打印机打印
int endStatus = PrinterHelper.getEndStatus(16);//获取打印状态。
PrinterHelper.openEndStatic(false);//关闭
```

注意:

它需要跟 openEndStatic(boolean isopen)配合使用(打印机固件版本必须 V1.01.27.01 以上)。

3.40 打印机回退

int ReverseFeed(int feed)

参数:

feed: 回退距离。（单位/行，范围：1-255）。

返回:

大于 0: 正常，否则异常。

例子:

```
PrinterHelper.ReverseFeed(50);
```

3.41 水印

int **PrintBackground**(int x,int y,int size,int background,String data)

参数:

x: 横坐标 (px) 。

y: 纵坐标 (px) 。

size: 字体大小。

55: 16 (px) 。

24: 24 (px) 。

56: 32 (px) 。

其他: 24 (px) 。

background: 背景黑度 (0-255) 。

data: 数据。

返回:

大于 0: 正常, 否则异常。

例子:

```
PrinterHelper.printAreaSize("0","200","200","500","1");
PrinterHelper.SetMag("8","8");//字体放大 8 倍
PrinterHelper.PrintBackground(0,0,56,150,"A508");
PrinterHelper.SetMag("1","1");//还原字体大小
PrinterHelper.Print();
```

3.42 获取打印机 SN

String **getPrintSN()**

参数:

无。

返回:

打印机 SN。

例子:

```
PrinterHelper.getPrintSN();
```


3.43 设置 Codepage

int **Country**(String codepage)

参数:

codepage:代码页。

ISO8859-1 : 西欧语

对应的 PrinterHelper.LanguageEncode="iso8859-1"

ISO8859-2 : 拉丁语 (2)

对应的 PrinterHelper.LanguageEncode="iso8859-2"

ISO8859-3 : 拉丁语 (3)

对应的 PrinterHelper.LanguageEncode="iso8859-3"

ISO8859-4 : 波罗的语

对应的 PrinterHelper.LanguageEncode="iso8859-4"

ISO8859-5 : 西里尔语

对应的 PrinterHelper.LanguageEncode="iso8859-5"

ISO8859-6 : 阿拉伯语

对应的 PrinterHelper.LanguageEncode="iso8859-6"

ISO8859-8 : 希伯来语

对应的 PrinterHelper.LanguageEncode="iso8859-8"

ISO8859-9 : 土耳其语

对应的 PrinterHelper.LanguageEncode="iso8859-9"

ISO8859-15 : 拉丁语 (9)

对应的 PrinterHelper.LanguageEncode="iso8859-15"

WPC1253 : 希腊语 (windows)

对应的 PrinterHelper.LanguageEncode="iso8859-11"

KU42: 希腊语 (ISO)

对应的 PrinterHelper.LanguageEncode="iso8859-7"

TIS18: 泰语

对应的 PrinterHelper.LanguageEncode="windows-874"

Khemr:高棉语

对应的 PrinterHelper.LanguageEncode="UnicodeBigUnmarked"

返回:

大于 0: 正常, 否则异常。

例子:

```
PrinterHelper.printAreaSize("0","200","200","500","1");
PrinterHelper.Country("ISO8859-1");
PrinterHelper.LanguageEncode="iso8859-1";
PrinterHelper .PrintCodepageTextCPCL(PrinterHelper.TEXT,0,"10","10","TEXT",15)
PrinterHelper .Print()
```

3.44 设置 QRcode 版本

注意：该接口用于设置二维码的版本号，设置后不会因为二维码的内容而改变大小。
只有部分机型和版本支持（可以询问客服）。

int **setQRcodeVersion**(int version)

参数:

version: 版本号（范围 0-40）

QR 版本默认为 00，QR 版本为 00 时，QR 码效果同旧版，宽高会随数据量而改变。

设置了版本号对二维码的数据量有范围要求，超出范围二维码不打印。

具体查看结尾的表 1-1.

返回:

大于 0: 正常。

-1: 断开连接。

-2: 参数错误。

例子:

```
PrinterHelper .setQRcodeVersion(20);
```

3.45 读取 QRcode 版本

String **getQRcodeVersion()**

参数:

无。

返回:

QRcode 版本。

例子:

```
PrinterHelper.getQRcodeVersion();
```

3.46 关闭高棉语输入指令

int **setKhemrEnd()**

参数:

无。

返回:

大于 0: 正常。

-1: 断开连接。

例子:

```
PrinterHelper .setKhemrEnd();
```

3.47 获取文字自动换行打印的高度

int **getAutLineHeight**(String x,String y,int width,int size,boolean isbole,boolean isdouble,String str)

参数:

X: 文字的起始的 x 坐标。(单位: PX)

Y: 文字的起始的 y 坐标。(单位: PX)

Width: 一行打印的宽度。(单位: PX)

Size: 字体。

3: 20x20 或 10x20, 视中英文而定。

4: 32x32 或 16x32, 由 ID3 字体宽高各放大 2 倍。

8: 24x24 或 12x24, 视中英文而定。

55: 16x16 或 8x16, 视中英文而定。

isbole: 加粗。

true: 加粗。

false: 不加粗。

isdouble: 放大两倍字体。

true: 放大。

false: 不放大。

Str: 要打印的文本。

返回:

打印出的文字的总高度。(单位: PX)

例子:

```
int textHeight=PrinterHelper.getAutLineHeight("0","0",100,4,true,true"Text")
```

//textHeight: 打印出的文字的总高度。

四 行打印模式

注意：行打印模式适用于打印小票。它只能打印文字，如果需要打印图片和条码，那就需要标签模式和行模式混合编辑。

4.1 行模式设置字体样式

int **Setlp**(String **font**,String **size**,String **spacing**)

参数：

font： 字体样式（默认：5）

size： 字体大小

spacing： 行高

返回：

大于 0 发送成功。

例子：

```
PrinterHelper.Setlp("5","2","46")
```

```
PrinterHelper.PrintData("要打印的文本\r\n")
```

4.2 行打印模式打印文本

int **PrintData**(String **str**)

参数：

str： 文本内容（以\r\n结尾）

返回：

大于 0 发送成功。

例子：

```
PrinterHelper.Setlp("5","2","46")
```

```
PrinterHelper.PrintData("要打印的文本\r\n")
```

4.3 行打印模式字体加粗

int **RowSetBold**(String **bold**)

参数:

bold: 加粗倍数。

返回:

大于 0 发送成功。

例子:

```
PrinterHelper.Setlp("5","2","46")
```

```
PrinterHelper.RowSetBold("2")
```

```
PrinterHelper.PrintData("要打印的文本\r\n")
```

```
PrinterHelper.RowSetBold("1");//注意关闭加粗以免影响下面的打印内容
```

4.4 设置行模式的 x 坐标

int **RowSetX**(String **X**)

注意: 必须放在 Setlp 函数之前

参数:

X:x 坐标。

返回:

大于 0 发送成功。

例子:

```
PrinterHelper.RowSetX("200");
```

```
PrinterHelper.Setlp("5","2","32");
```

```
PrinterHelper.RowSetBold("2");
```

```
PrinterHelper.PrintData("要打印的文本\r\n");
```

```
PrinterHelper.RowSetBold("1");
```