

## PRT Android TSPL SDK Manual

PRT Android TSPL SDK Manual.....	1
1. SDK Command Function.....	2
1.1 Page label starting command:.....	2
1.2 Page label ending command (except for line mode):.....	2
1.3 Print printer information:.....	3
1.4 Set the distance of label paper :.....	3
1.5 Paper feed:.....	3
1.6 Clear the data in the buffer area.....	4
1.7 Text print, including some variant:.....	4
1.8 Print direction.....	5
1.9 Define the reference starting point of label.....	6
1.10 Define international character set code.....	6
1.11 Barcode:.....	7
1.12 Print QR Code.....	8
1.13 Specified length of label paper feed.....	9
1.14 Position the label to the starting point of next label.....	9
1.15 Print line.....	9
1.16 Position the label to the starting point specified by the internal sensor.....	10
1.17 Print image:.....	10
1.18 Print density.....	11
1.19 Print speed.....	11
1.20 Cut paper.....	11
1.21 Print rectangle box.....	12
1.22 Set beep time of buzzer.....	12
1.23 Send data.....	12
1.24 Read data.....	13
1.25 Anti-white box.....	13
1.26 Text Bold.....	14
1.27 Get Printer Status.....	14
1.28 Print Block.....	14
2.1 Bluetooth connection.....	16
Bluetooth disconnection.....	16
Whether Bluetooth is connected.....	16
2.2 USB connection.....	17
USB disconnection.....	17
Whether USB is connected.....	17
2.3 WIFI connection.....	18
WIFI disconnection.....	18

## 1. SDK Command Function

### 1.1 Page label starting command:

`int printAreaSize(String width, String height)`

Parameter:

width:width within the print area (unit: Millimeter).

height:height within the print area (unit: Millimeter).

Return:

Greater than 0: normal, otherwise abnormal.

Example:

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

```
HPRTPrinterHelper.printText("0","0","9","0","1","1","TEXT");//Print TEXT
```

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

### 1.2 Page label ending command (except for line mode):

`int Print(String strnum, String strcopies)`

Parameter:

strnum: times of printing

strcopies: counter (default 1).

Return:

Greater than 0: normal, otherwise abnormal.

Example:

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

```
HPRTPrinterHelper.printText("0","0","9","0","1","1","TEXT");//Print  
TEXT
```

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

### 1.3 Print printer information:

```
int SelfTest()//Print self test page
```

Return:

Greater than 0: normal, otherwise abnormal.

Example:

```
HPRTPrinterHelper.SelfTest()
```

### 1.4 Set the distance of label paper :

```
int Gap(String distance, String offset)
```

Parameter:

Distance: distance between two labels (unit:mm).

Offset: distance between label content and label bottom (unit:mm)

Return:

Greater than 0: normal, otherwise abnormal.

Example:

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

### 1.5 Paper feed:

```
int Offset (String distance)
```

Parameter:

distance: paper feeding distance (unit:mm).

Return:

Greater than 0: normal, otherwise abnormal.

Example:

```
HPRTPrinterHelper.Offset("5")
```

## 1.6 Clear the data in the buffer area

int CLS()

Return:

Greater than 0: normal, otherwise abnormal.

Example:

HPRTPrinterHelper.CLS()

## 1.7 Text print, including some variant:

(1) Int printText(String x\_pos, String y\_pos, String font, String rotation, String x\_multiplication, String y\_multiplication, int alignment, String code\_data)

Parameter:

x\_pos: starting x-coordinate of text

y\_pos: starting y-coordinate of text

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: Only this font can print Chinese.

Rotation: print direction.

0 : No rotation

90 : degrees, in clockwise direction

180 : degrees, in clockwise direction

270 : degrees, in clockwise direction

x\_multiplication: text stretch multiplication in x-coordinate direction.

y\_multiplication: text stretch multiplication in y-coordinate direction.

Alignment: alignment mode (some old models do not support this function, this parameter can be removed)

1: Left aligned

2: Centered

3: Right aligned

code\_data: text data.

Return:

Greater than 0: normal, otherwise abnormal.

Example:

```
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)

Parameter:

x\_pos: starting x-coordinate of text

y\_pos: starting x-coordinate of text

font:

Same as the previous interface

Rotation: print direction

0 : No rotation

90 : degrees, in clockwise direction

180 : degrees, in clockwise direction

270 : degrees, in clockwise direction

Size: font size 1~7.

Alignment: alignment mode (some old models do not support this  
function, this parameter can be removed)

1: Left aligned

2: Centered

3: Right aligned

code\_data: text data

Return:

Greater than 0: normal, otherwise abnormal.

Example:

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

## 1.8 Print direction

int Direction(String direction)

Parameter:

direction:

0: vertical

1: horizontal

Return:

Greater than 0: normal, otherwise abnormal.

Example:

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

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

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

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

### 1.9 Define the reference starting point of label

```
int Reference(String x_pos,String y_pos)
```

Parameter:

x: x-coordinate of starting point

y: y-coordinate of starting point

Return:

Greater than 0: normal, otherwise abnormal.

Example:

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

### 1.10 Define international character set code

```
int Codepage(String codepage)
```

Parameter:

codepage: code number

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)

Return:

Greater than 0: normal, otherwise abnormal.

Example:

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

### 1.11 Barcode:

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

Parameter:

x\_pos: starting x-coordinate of bar code

y\_pos: starting y-coordinate of bar code

code\_type: bar code type

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

Height: height of bar code(Unit: pixel)

Readable: whether the bar code data is readable

0: not readable

1: human readable

rotation: bar code direction

0 : No rotation

90 : degrees, in clockwise direction

180 : degrees, in clockwise direction

270 : degrees, in clockwise direction

Narrow: unit width of narrow bar code(Default: 1).

Wide: unit width of wide bar code(Default: 1).

code\_data: bar code data

Return:

Greater than 0: normal, otherwise abnormal.

Example:

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

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

```
HPRTPrinterHelper.printBarcode("0","0","128","100","1","0","1","1","
1234567890")
```

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

## 1.12 Print QR Code

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

### Parameter:

rotation: bar code direction

0 : No rotation

90 : degrees, in clockwise direction

180 : degrees, in clockwise direction

270 : degrees, in clockwise direction

X: starting x-coordinate of QR code

Y: starting y-coordinate of QR code

ecc\_level: correction level

L : 7%

M : 15%

Q : 25%

H : 30%

width: 1~10

mode :Auto/manual encode

A: Auto

M: Manual

Data: QR code data

### Return:

Greater than 0: normal, otherwise abnormal.

### Example:

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

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

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

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

### 1.13 Specified length of label paper feed

`int Feed(String len)`

Parameter:

len: length of paper feed (unit: mm)

Return:

Greater than 0: normal, otherwise abnormal.

Example:

`HPRTPrinterHelper.Feed("5")`

### 1.14 Position the label to the starting point of next label

`int Formfeed()`

Return:

Greater than 0: normal, otherwise abnormal.

Example:

`HPRTPrinterHelper.Formfeed()`

### 1.15 Print line

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

Parameter:

x\_pos: starting x-coordinate

y\_pos: starting y-coordinate

width: width of line (pixel)

height: height of line (pixel)

Return:

Greater than 0: normal, otherwise abnormal.

Example:

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

`HPRTPrinterHelper.Bar("10","10","100","2")//horizontal line`

`HPRTPrinterHelper.Bar("10","10","2","100")//vertical line`

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

1.16 Position the label to the starting point specified by the internal sensor

```
int Home()
```

Return:

Greater than 0: normal, otherwise abnormal.

Example:

```
HPRTPrinterHelper.Home()
```

1.17 Print image:

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

Parameter:

X: starting x-coordinate of image

Y: starting y-coordinate of image

bmap: the object of image Bitmap

isNegate: image reverse

True: normal display

False: negate display

isLZO:Whether compression (the printer must support compression)

Type:Picture algorithm

0:black white algorithm

1:halftone algorithm

Return:

Greater than 0: normal,Equal to -1: width or height of image is over the printer area

Example:

```
HPRTPrinterHelper.printAreaSize("100","200");//The height of label should be greater than that of image
```

```
HPRTPrinterHelper.CLS()
```

```
HPRTPrinterHelper.printImage("10","10",bitmap,true,false,1)
```

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

### 1.18 Print density

int Density(String contrast )

Parameter:

Contrast: 0~15

0: specifies the lightest level

15: specifies the darkest level

Return:

Greater than 0: normal, otherwise abnormal.

Example:

HPRTPrinterHelper.Density("5")

### 1.19 Print speed

int Speed(String speed )

Parameter:

speed

1	1.	2	2.	3	3.	4	5	6	8	10	12
	5		5		5						

Return:

Greater than 0: normal, otherwise abnormal.

Example:

HPRTPrinterHelper.Speed("2")

### 1.20 Cut paper

int Cut()

Return:

Greater than 0: normal, otherwise abnormal.

Example:

HPRTPrinterHelper.Cut()

### 1.21 Print rectangle box

`int Box(String x_start,String y_start,String x_end, String y_end,String thickness)`

Parameter:

x\_start: x-coordinate of top left corner  
y\_start: y-coordinate of top left corner  
x\_end: x-coordinate of lower right corner  
y\_end: y-coordinate of lower right corner  
Thickness: width of the line

Return:

Greater than 0: normal, otherwise abnormal.

Example:

```
HPRTPrinterHelper.printAreaSize("100","200")
HPRTPrinterHelper.CLS()
HPRTPrinterHelper.Box("10","10","100","100","2")
HPRTPrinterHelper.Print("1","1")
```

### 1.22 Set beep time of buzzer

`int Sound(String level, String interval)`

Parameter:

level: the lasting time of beep, (1/8) second is specified unit  
Interval: the interval time

Return:

Greater than 0: normal, otherwise abnormal.

Example:

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

### 1.23 Send data

`int WriteData(byte[] bData)`

Parameter:

bData: data that needs to be sent to the printer

Return:

Greater than 0: normal, otherwise abnormal.

Example:

```
HPRTPrinterHelper.WriteData(bData)//Send data to the printer
```

## 1.24 Read data

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

Parameter:

timeout: time out(unit:millisecond)

Return:

Data:the data read

Example:

```
bData=HPRTPrinterHelper.ReadData(2000)//bData is the data read
```

## 1.25 Anti-white box

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

Parameter:

x\_start: X coordinate.

y\_start: Y coordinate.

x\_width: width.

y\_height: height.

note:

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

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

return:

Greater than 0: normal, otherwise abnormal.

Example:

```
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")
```



## 1.26 Text Bold

int Bold(int bold)

Parameter:

bold: 0: Not bold.  
1: bold.

return:

Greater than 0: normal, otherwise abnormal.

Example:

HPRTPrinterHelper.Bold(1)

## 1.27 Get Printer Status

int getPrinterStatus()

Parameter:

NULL

return:

HPRTPrinterHelper.STATUS_DISCONNECT:	Disconnection
HPRTPrinterHelper.STATUS_TIMEOUT:	Time Out
HPRTPrinterHelper.STATUS_OK:	Ready
HPRTPrinterHelper.STATUS_COVER_OPENED:	Cover Opened
HPRTPrinterHelper.STATUS_NOPAPER:	Not Paper
HPRTPrinterHelper.STATUS_OVER_HEATING:	Over Heating
HPRTPrinterHelper.STATUS_PRINTING:	Printing

Example:

HPRTPrinterHelper.getPrinterStatus()

## 1.28 Print Block

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

Parameter:

StartX: x coordinate of upper left corner of text.  
StartY: Y coordinate at upper left corner of the text.  
Width: The Width of the text.  
Height: The Height of the text.  
FONT: Font size (0:16 \*16, 1:24 \*24)  
Rotate to: (0,90,180,270)  
MultiplicationX: Multiplication X in the X-axis direction of the font.  
MultiplicationY: Multiplication Y in the Y-axis direction of the font.  
Space: Line spacing.  
Alignment: Alignment. 1: Left Align, 2: Center, 3: Right Align  
Content: Print the Content.

return:

Greater than 0: normal, otherwise abnormal.

Example:

```
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")
```

## 2. Communication Port Function

### 2.1 Bluetooth connection

`int PortOpen(String portSetting)`

Parameter:

portSetting: "Bluetooth, Bluetooth address"

Return:

0: connect success

-1: connect fail

Example:

`HPRTPrinterHelper .PortOpen("Bluetooth, bluetooth address")`

### Bluetooth disconnection

`int PortClose()`

Parameter:

None

Return:

0: connect success

-1: connect fail

Example:

`HPRTPrinterHelper .PortClose()`

### Whether Bluetooth is connected

`boolean IsOpened()`

Parameter:

None

Return:

True: connected

False: disconnected

Example:

```
HPRTPrinterHelper .IsOpened()
```

## 2.2 USB connection

```
int PortOpen(Context context,UsbDevice usbdevice)
```

Parameter:

usbdevice:USB device object

Return:

0: connect success

-1: connect fail

Example:

```
HPRTPrinterHelper .PortOpen(usbdevice)
```

## USB disconnection

```
int PortClose()
```

Parameter:

None

Return:

0: connect success

-1: connect fail

Example:

```
HPRTPrinterHelper .PortClose()
```

## Whether USB is connected

```
boolean IsOpened()
```

Parameter:

None

Return:

True: connected

False: disconnected

Example:

```
HPRTPrinterHelper .IsOpened()
```

## 2.3 WIFI connection

```
int PortOpen(String portSetting)
```

Parameter:

portSetting: "WiFi,"+strIP+", "+strPort

strIP:IP address

strPort:The port number(9100)

Return:

0: connect success

-1: connect fail

Example:

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

## WIFI disconnection

```
int PortClose()
```

Parameter:

None

Return:

0: disconnect success

-1: disconnect fail

Example:

```
HPRTPrinterHelper .PortClose()
```