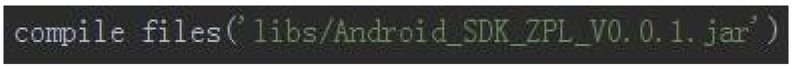
Android ZPL SDK Manual

1. SDK Introduction

1.1 SDK jar Kit

The adding method is as shown below:



```
compile files('libs/Android_SDK_ZPL_V0.0.1.jar')
```

This jar kit contains all the connecting functions (including Bluetooth connection, WIFI connection, USB connection) and all the command connector functions. All of these connectors are in the category of ZPLPrinterHelper.

1.2 How to use ZPLPrinterHelper

ZPLPrinterHelper is singleton pattern, through referencing ZPLPrinterHelper.getZPL(mContext) can call connecting functions and command functions.

2. Connecting Method

2.1 Bluetooth Connection

ZPLPrinterHelper mZPL=ZPLPrinterHelper.getZPL(mContext)

mZPL.PortOpen(portSetting)

Parameter:

mContext: context object.

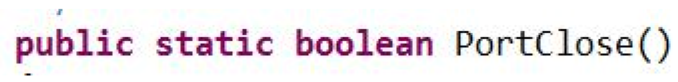
portSetting:”Bluetooth,”+MAC.(MAC: Bluetooth address)

Return:

0: connection success.

-1: connection failure.

Disconnect Bluetooth:



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

Example:

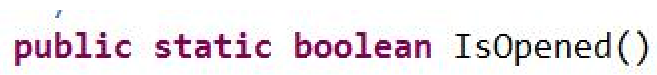
mZPL.PorClose()

Return:

true: disconnection success.

false: disconnection failure.

Whether Bluetooth is connected:



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

Example:

ZPLPrinterHelper.IsOpened()

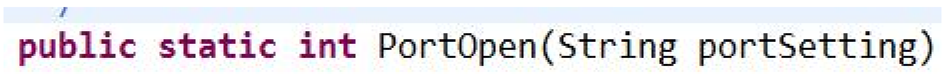
Return:

true: Bluetooth connected.

false: Bluetooth unconnected.

2.2 WIFI Connection

Connect WIFI:



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

Example:

ZPLPrinterHelper mZPL=ZPLPrinterHelper.getZPL(mContext)

mZPL.PortOpen(“WiFi,”+IP+”,”+PortNumber)

IP:Printer IP address.

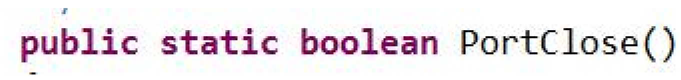
PortNumber: port. Default: 9100

Return:

0: connection success.

-1: connection failure.

Disconnect WiFi:



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

Example:

mZPL.PorClose()

Return:

true: disconnection success.

false: disconnection failure.

Whether WiFi is connected:

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

Example:

ZPLPrinterHelper.IsOpened()

Return:

true: connected.

false: unconnected.

2.3 USB Connection

Connect USB:

```
 */
public static int PortOpen(UsbDevice usbdevice)
```

Example:

ZPLPrinterHelper mZPL=ZPLPrinterHelper.getZPL(mContext)

mZPL.PortOpen(usbdevice)

usbdevice:UsbDevice object

Return:

0: connection success.

-1: connection failure.

Disconnect USB:

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

Example:

mZPL.PorClose()

Return:

true: disconnection success.

false: disconnection failure.

Whether USB is connected:

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

Example:

mZPL.IsOpened()

Return:

true:connected.

false:unconnected.

3.Print Command

3.1 Label start

Function:

int start();

Return:

Greater than 0:Normal, otherwise abnormal.

Example:

mZPL.start();

mZPL.printText(“0”,”0”,5,”N”,3,”TEXT”);

mZPL.end();

3.2 Set XY

Function:

int setXY(String x,String y);

Parameter:

X: x-axis

Y: y-axis

Return:

Greater than 0:Normal, otherwise abnormal.

3.3 Label end

Function:

```
int end();
```

Return:

Greater than 0:Normal, otherwise abnormal.

Example:

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

3.4 Field data start

Function:

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

Parameter:

A: Text

Return:

Greater than 0:Normal, otherwise abnormal.

Example:

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

3.5 Field data end

Function:

```
int FD();
```

Return:

Greater than 0:Normal, otherwise abnormal.

Example:

```
mZPL.start();

mZPL.setXY(“0”,”0”);

mZPL.FD(“TEXT”);

mZPL.FS();

mZPL.end();
```

3.6 Print width

Function:

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

Parameter:

A: width (in dots).

Return:

Greater than 0:Normal, otherwise abnormal.

Example:

```
mZPL.start();

mZPL.PW(“100”)
mZPL.setXY(“0”,”0”);

mZPL.FD(“TEXT”);

mZPL.FS();

mZPL.end();
```

3.7 Print direction and justification method

Function:

int FW(String rotate,String justification)

Parameter:

rotate: print direction, value is as below:

N = normal

R = rotated 90 degrees

I = inverted 180 degrees

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

Justification: justification method, value is as below:

0 = left justification

1 = right justification

2 = auto justification (script dependent)

Return:

Greater than 0:Normal, otherwise abnormal.

Example:

```
mZPL.start();

mZPL.FW(“N”,”0”)
```

mZPL.setXY(“0”,”0”);

mZPL.FD(“TEXT”);

mZPL.FS();

mZPL.end();

3.8 Print line

Function:

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

Parameter:

w: width of line (1~32000 unit: dot)

H: height of line (1~32000 unit: dot)

T: thickness of line (1~32000 unit: dot)

C: color of line:

B=black; (Default:B)

W=white;

R: radian of rounded corner(0~8)

Return:

Greater than 0:Normal, otherwise abnormal.

Example:

mZPL.start();

mZPL.setXY(“0”,”0”);

mZPL.printLine(“100”,”2”,”2”,”B”,”0”)//Print the horizontal line with 100-dot width and 2-dot height

mZPL.end();

3.9 Print circle

Function:

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

Parameter:

d: diameter of circle (3~4095).

t: thickness of circle border (1~4095).

C: color of circle:

B=black； (Default:B)

W=white；

Return:

Greater than 0: Normal, otherwise abnormal.

Example:

```
mZPL.start();

mZPL.setXY(“0”,”0”);

mZPL.printCircle(“100”,”2”,”B”)

mZPL.end();
```

3.10 Print slash line

Function:

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

Parameter:

w: width of slash line (3~32000).

H: height of slash line (3~32000).

T: thickness of slash line (1~32000).

C: color of slash line:

B=black; (Default:B)

W=white；

O: direction of slash line:

R (or /) = right-leaning diagonal

L (or \) = left-leaning diagonal

(Default:R)

Return:

Greater than 0:Normal, otherwise abnormal.

Example:

```
mZPL.start();

mZPL.setXY(“0”,”0”);

mZPL.printSlashLine(“100”,”100”,”2”,”B”,”R”)

mZPL.end();
```

3.11 Set the label home position

Function:

int LH(String x,String y)

Parameter:

X: x-axis of home position

Y: y-axis of home position

Return:

Greater than 0:Normal, otherwise abnormal.

3.12 Label shift

Function:

int LS(String a)

Parameter:

a: shift value (-9999~9999) negative indicates right shift, (Default:0).

Return:

Greater than 0:Normal, otherwise abnormal.

Example:

```
mZPL.start();
mZPL.LS(“20”)
mZPL.printText(“100”,”100”,5,”N”,3,”TEXT”);
mZPL.end();
```

3.13 Text blocks (has an automatic word-wrap function)

Function:

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

Parameter:

a: block rotation:

N = normal

R = rotate 90 degrees clockwise

I = invert 180 degrees

B = read from bottom up-270 degrees

b: block width in dots

c: block height in dots

Return:

Greater than 0:Normal, otherwise abnormal.

Example:

```
mZPL.start();

mZPL.TB(“N”,”300”,”300”)

mZPL.printText(“100”,”100”,5,”N”,3,”TEXT”);

mZPL.end();
```

3.14 QR code

Function:

int printQRcode(String x,String

y,Stringorientation,Stringmagnification, String size,String data)

Parameter:

x: x-axis

y: y-axis

orientation: (N).

magnification:

1= Normal mode

2= Enhanced mode

Default(2)

size: size (1~10).

data: content of 2D code

Return:

Greater than 0:Normal, otherwise abnormal.

Example:

```
mZPL.start();

mZPL.printData("^CI14\r\n");

mZPL.printQRcode(“10”,”10”,”N”,”2”,”5”,”abc123”);

mZPL.end();
```

3.15 Bar code

Function:

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

Parameter:

x: x-axis

y: y-axis

Bar code type:

0=39

1=EAN-8

2=UPC-E

3=93

4=128

5=EAN-13

Orientation:

N = normal

R = rotated 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

Height: height of bar code (1-32000).

F: whether the content of bar code is visible (Default Y):

Y = yes

N = no

Return:

Greater than 0:Normal, otherwise abnormal.

Example:

```
mZPL.start();

mZPL.printBarcode(“10”,”10”,”0”,”N”,”100”,”Y”,”123456789”);

mZPL.end();
```

3.16 Print bitmap

Function:

printBitmap(String x,String y,Bitmap bmp)

Parameter:

x: starting x-axis

y: starting y-axis

Bmp: bit map image

Return:

Greater than 0:Normal, otherwise abnormal.

Example:

```
mZPL.start();

mZPL.printBitmap(“10”,”10”,bitmap);

mZPL.end();
```

3.17 Write data

Function:

int WriteData(byte[] bData)

Parameter:

bData: the data which needs to be sent to the printer

Return:

Greater than 0:Normal, otherwise abnormal.

Example:

mZPL.WriteData(byt)

3.18 Read data

Function:

byte[] ReadData(int outTime)

Parameter:

outTime: time of outTime (Unit: second)

Return:

The read data

Example:

mZPL.ReadData(2)

3.19 Self test page

Function:

int selfTest()

Return:

Greater than 0:Normal, otherwise abnormal.

Example:

mZPL.selfTest()

3.20 Print text

Note: You need to select a code when printing Chinese, please refer to the example.

Function:

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

Parameter:

x: x-coordinate

y: y-coordinate

type: font (0~6: ineffective for Chinese,7:Chinese)

orientation:

N = normal

R = rotate 90 degrees clockwise

I = invert 180 degrees

B = read from bottom up-270 degrees

size: font size

1: 10px

2: 20px

3: 30px

4: 40px

5: 50px

6: 60px

data: text data

Return:

Greater than 0:Normal, otherwise abnormal.

Example:

```
mZPL.start();

//This sentence needs to be added when printing Chinese

mZPL.printData("^CI14\r\n")

mZPL.printText(“0”,”0”,5,”N”,3,”TEXT”);

mZPL.end();
```

3.21 Print quantity and cutter

Function:

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

Parameter:

q:print quantity

p:Number before suspension or before cutting knife

r:Number of copies of each serial number

o:Cut or pause

- Y:cutter
- N:pause

Return:

Greater than 0:Normal, otherwise abnormal.

Example:

```
mZPL.start();
mZPL.printData("^CI14\r\n")
mZPL.printText(“0”,”0”,5,”N”,3,”TEXT”);
mZPL.PQ(1,1,1,Y);//Print one piece and then cut the knife
mZPL.end();
```

3.22 Printer Model

Function:

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

Parameter:

model:

T:Tear

P:Peel (not available on S-300)

Return:

Greater than 0:Normal, otherwise abnormal.

Example:

```
mZPL.start();
mZPL.printData("^CI14\r\n")
mZPL.printText(“0”,”0”,5,”N”,3,”TEXT”);
mZPL.setPrinterModel(“P”)//Switch to peel mode
mZPL.end();
```

3.23 Write RFID

Function:

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

Parameter:

Address: starting address, range: greater than 0

EPC address must start at 2.

Memory: Write area, (0= Preserved, 1=EPC, 3 = User)

Data: The data to be written. Preserved area, EPC no more than 12 bytes, User no more than 128

Return:

-1: Send failed,

-2: Parameter error,

0: Write successful.

Example:

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

3.24 Read RFID

Function:

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

Parameter:

Address: starting address, range: greater than 0

EPC address must start at 2.

Length:The length of the read. Preserved\EPC no more than

12 bytes, User no more than 128

Memory: Write area, (0= Preserved, 1=EPC,2=TID,3 = User)

Return:

Readable data, empty means read failed.

Example:

```
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();}
```

3.25 Read SN

Function:

```
String getPrinterSN()
```

Return:

SERIAL NEMBER:SN

Example:

```
mZPL.getPrinterSN();
```