

# Android X30 SDK 标准 开发指南-V5.0

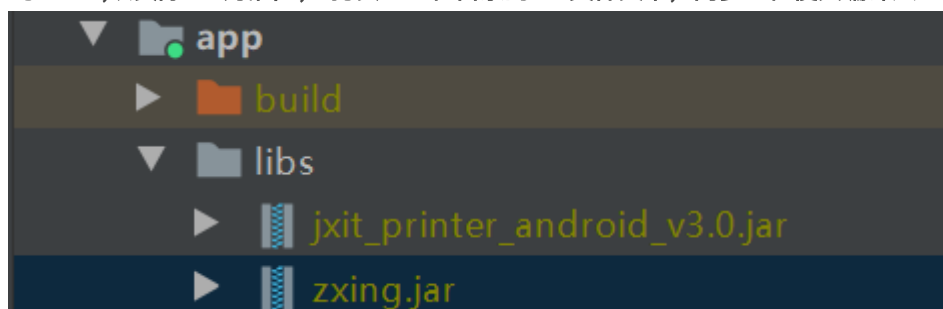
## 1. SDK项目集成配置

通过以下几步操作，即可在您的应用中使用捷昕android SDK：

### 1.1. 集成sdk

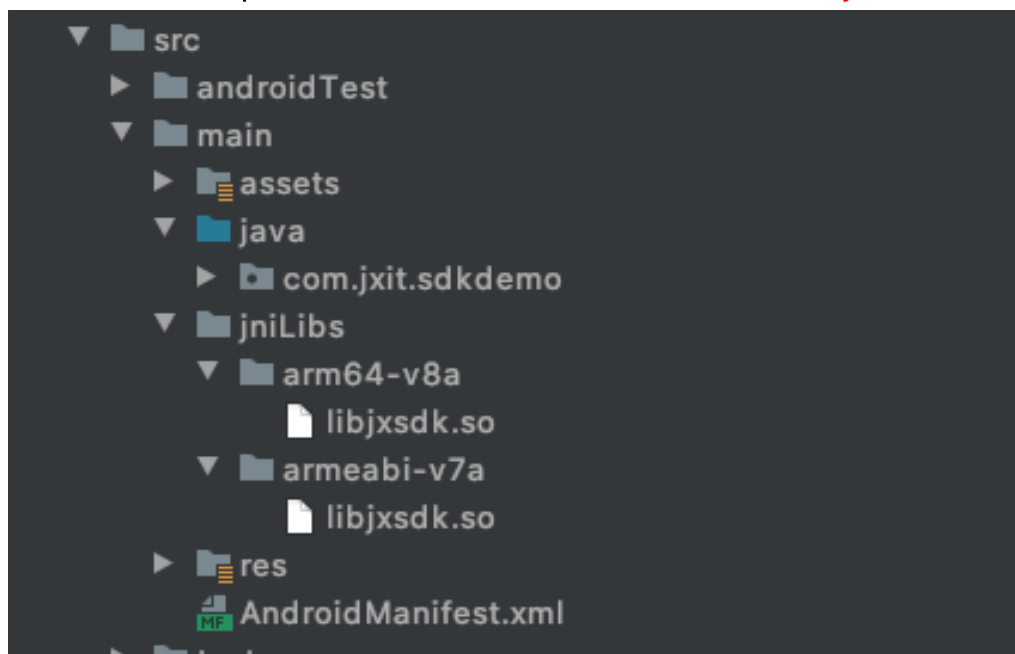
#### 1.1.1. 添加sdk

将jxit\_printer\_android\_v3.0.jar， **zxing.jar**（未使用global sdk理论上不需要集成zxing）（参考demo,以实际sdk为准）， 拷贝至工程目录的libs文件夹下，同步工程使其编译入工程。



#### 1.1.2. 添加 Jni

将jniLibs文件夹拷贝到src/main/目录下，和java文件夹同级。jniLibs提供了armeabi-v7a和arm64-v8a两种主流cpu架构的支持so（未使用解锁、流控相关功能无需集成jni）



## 1.2. 配置AndroidManifest权限

### 1.2.1 集成usb打印机在<manifest>中添加如下

```
1 <!-- usb permissions -->
2 <uses-feature android:name="android.hardware.usb.host" />
3 <uses-permission android:name="android.hardware.usb.host" />
```

注：使用Usb通信需要使用OTG线，并使手机处于Usb Host模式。

MainActivity在AndroidManifest.xml需要配置如下：

```
<activity
    android:name=".MainActivity"
    android:screenOrientation="portrait">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
        <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
    </intent-filter>
    <meta-data
        android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
        android:resource="@xml/qwm_usb_xml" />
    </activity>
```

<activity>标签中添加

```
1 <meta-data
2     android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
3     android:resource="@xml/qwm_usb_xml" />
```

<intent-filter>标签中添加

```
1 <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
```

res文件夹下创建xml文件夹，并copy sdkdemo对应qwm\_usb\_xml.xml文件，详见sdk demo或参考Android Usb Host模式开发指南。

## 1.2.2 集成蓝牙打印机在<manifest>中添加如下

```
1 <!-- bluetooth permissions -->
2 <uses-permission android:name="android.permission.BLUETOOTH" />
3 <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
4 <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
5 <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
6 <!-- Android 12 bluetooth -->
7 <uses-permission android:name="android.permission.BLUETOOTH_SCAN" />
8 <uses-permission android:name="android.permission.BLUETOOTH_ADVERTISE" />
9 <uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
```

注：自Android6.0起部分权限的使用需要开发者在代码中动态申请。Android 12起蓝牙权限系统引入了新的运行时权限 BLUETOOTH\_SCAN、BLUETOOTH\_ADVERTISE 和 BLUETOOTH\_CONNECT权限

## 1.2.3 集成WiFi打印机在<manifest>中添加如下

```
1 <!-- Tcp permissions-->
2 <uses-permission android:name="android.permission.INTERNET" />
3 <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
4 <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

## 1.3. 初始化sdk

新建一个自定义的Application，在其onCreate方法中完成SDK的初始化。示例代码如下：

```
public class DemoApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        //printer init
        JXPrinter.init(context: this, BuildConfig.DEBUG);
        //global printer init
        JXGlobalPrinter.init(context: this, BuildConfig.DEBUG);
    }
}
```

在AndroidManifest.xml文件中声明该Application

```
<application
    android:name=".DemoApplication"
```

## 1.4. 混淆sdk

如果您的应用使用了代码混淆，请添加如下配置，以避免SDK被错误混淆导致SDK不可用

```
1 -keep class com.jx.it.printer.* {*;}
```

至此，sdk集成配置已经完成。

## 2. 蓝牙连接打印机

### 2.1. 一般使用流程

(1) 搜索蓝牙打印机 (2) 打印机蓝牙配对 (3) 建立蓝牙连接 (4) 发送打印机指令

### 2.2. 获取JXBluetoothAPI全局单例

```
1 JXBluetoothAPI mApi = JXBluetoothAPI.getDefault(context)
```

### 2.3. 获取连接状态

功能说明

获取当前蓝牙连接状态

方法名称

isConnected()

返回参数

已连接返回true,未连接则返回false。

### 2.4. 开启/关闭蓝牙搜索，搜索蓝牙设备

功能说明

开启/关闭蓝牙搜索，搜索蓝牙设备

方法名称

startDiscovery()

cancelDiscovery()

返回参数

搜索蓝牙设备，需要监听系统蓝牙广播，action为BluetoothDevice.ACTION\_FOUND，详见 sdkdemo#BluetoothFragment或参考Android蓝牙开发指南。

### 2.5. 获取已配对蓝牙设备列表

功能说明

获取已配对蓝牙设备列表

**方法名称**

getBondedDevices()

**返回参数**

set<BluetoothDevice>

## 2.6. 蓝牙配对/取消配对

**功能说明**

蓝牙配对/取消配对

**方法名称**

createBTBond(BluetoothDevice device)

removeBTBond(BluetoothDevice device)

**返回参数**

打印机默认配对密码为 0000，BluetoothDevice.ACTION\_BOND\_STATE\_CHANGED，详见 sdkdemo#BluetoothFragment或参考Android蓝牙开发指南。

## 2.7. 蓝牙连接/断开

**功能说明**

蓝牙连接/断开

**方法名称**

openConnection(String macAddress)

closeConnection()

**返回参数**

成功返回true,否则返回false。

## 3. Usb连接打印机

### 3.1. 一般使用流程

(1)手机用OTG线以主机模式连接和打印机 (2) 建立Usb连接 (3) 发送打印机指令

### 3.2. 获取JXUsbAPI全局单例

```
1 JXUsbAPI mApi = JXUsbAPI.getDefault(getContext());
```

### 3.3. 获取连接状态

**功能说明**

获取当前蓝牙连接状态

**方法名称**

isConnected()

**返回参数**

已连接返回true,未连接则返回false。

### 3.4. 获取已连接的Usb设备

**功能说明**

获取已连接的Usb设备

**方法名称**

getDevices()

**返回参数**

List<UsbDevice>

### 3.5. Usb连接/断开

#### 功能说明

Usb连接/断开

#### 方法名称

openConnection(UsbDevice device)

closeConnection()

#### 返回参数

成功返回true,否则返回false。

注：使用Usb通信需要使用OTG线，并使手机处于Usb Host模式。

MainActivity在AndroidManifest.xml需要配置如下：

```
<activity
    android:name=".MainActivity"
    android:screenOrientation="portrait">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
        <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
    </intent-filter>
    <meta-data
        android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
        android:resource="@xml/qwm_usb_xml" />
</activity>
```

<activity>标签中添加

```
1 <meta-data
2     android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
3     android:resource="@xml/qwm_usb_xml" />
```

<intent-filter>标签中添加

```
1 <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
```

res文件夹下创建xml文件夹，并copy sdkdemo对应qwm\_usb\_xml.xml文件，详见sdk demo或参考Android Usb Host模式开发指南。

## 4. 获取打印机实例

### 4.1. 获取蓝牙打印机实例

```
1 JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2 //获取行模式打印机实例
3 JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMode.Line);
4 //获取页模式打印机实例
5 JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMode.Page);
6 //获取其他指令打印机实例
7 JXPrinter printer= new JXPrinter(mAPI);
```

### 4.2. 获取Usb打印机实例

```
1 JXInterfaceAPI mApi = JXUsbAPI.getDefault(context);
2 //获取行模式打印机实例
```

```
3 JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMode.Line);
4 //获取页模式打印机实例
5 JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMode.Page);
6 //获取其他指令打印机实例
7 JXPrinter printer= new JXPrinter(mAPI);
```

## 5. 打印机设置

### 5.1. 获取打印黑度

功能说明

获取打印机的打印黑度

方法名称

get\_print\_darkness()

参数说明

无

返回参数

参数	名字	说明
value	打印机黑度值	打印黑度 0~4

### 5.2. 设置打印机黑度

功能说明

设置打印机的打印黑度

方法名称

set\_print\_darkness(String value)

参数说明

参数	名字	说明
value	打印机黑度值	打印黑度 0~4

返回参数

成功则返回true,失败则返回false。

### 5.3. 获取打印质量

功能说明

获取打印机的打印质量

方法名称

get\_print\_quality()

参数说明

无

返回参数

参数	名字	说明
value	打印机质量值	打印质量 0~2

## 5.4. 设置打印机质量

功能说明

设置打印机的打印黑度

方法名称

set\_print\_quality( String value)

参数说明

参数	名字	说明
value	打印机质量	打印质量 0~2

返回参数

成功则返回true,失败则返回false。

## 5.5. 获取打印机状态

功能说明

获取打印机当前状态

方法名称

get\_printer\_status()

参数说明

无

返回参数

参数	名字	说明
JXPrinterStatus	打印机状态	JXPrinterStatus.isReady//空闲 JXPrinterStatus.isPrinting//忙碌（打印中） JXPrinterStatus.isCoverOpened//开盖 JXPrinterStatus.isNoPaper//缺纸 JXPrinterStatus.isOverHeat//过热 JXPrinterStatus.isBatteryLow//电量低 JXPrinterStatus.isUnknownError//未定义错误

## 5.6. 打印自检页

功能说明

让打印机打印自检页

方法名称

print\_check\_page( )

参数说明：无

返回参数

成功则返回true,失败则返回false。

## 5.7. 解锁打印机（定制功能）

功能说明

对于定制打印机在连接时进行打印机解锁，否则不能使用（使用此功能需要集成 jni）

方法名称

set\_print\_unlock( )

参数说明：无

返回参数

成功则返回true,失败则返回false。

## 6. Global LINE指令

Global LINE指令支持多国语言文字打印。使用Global LINE指令需要额外依赖zxing.jar。指令方法均以line\_开头。

```
1 一般用于连续纸的打印，步骤：  
2 1、获取打印服务单例对象  
3 2、调用设置打印机最大打印宽度方法设置打印宽度  
4 3、调用打印文字或图片等方法设置打印内容，如果内容宽度超出一行的宽度就会打印出来，不足一行的会缓存起来，和后面要打印的内容拼接到一起，也可以调用强制打印缓存区的方法，把缓存内容强制打印出来。
```

### 6.1. 设置最大打印宽度

**功能说明**

根据打印机的实际打印宽度设置，默认384。创建JXGlobalPrinter()实例后需要优先设置，设置一次即可。**(打印前必须设置)**

**方法名称**

line\_setMaxPrintWidth(int maxWidth)

**参数说明**

参数	名字	说明
maxWidth	打印机最大宽度点数	页面宽度点数（8点/mm）

**返回参数**

无。

### 6.2. 清除缓存区数据

**功能说明**

清除缓存区数据所有数据恢复默认值。

**方法名称**

line\_reset()

**返回参数**

成功则返回true,失败则返回false。

### 6.3. 走纸

**功能说明**

向前走纸。

**方法名称**

line\_feed(int n)

**参数说明**

参数	名字	说明
n	向前走纸点	点数（8点/mm）

**返回参数**

成功则返回true,失败则返回false。



## 6.4. 打印控制内容指令

### 功能说明

打印并清空缓存区内容。

### 方法名称

line\_print()

### 返回参数

成功则返回true,失败则返回false。

## 6.5. 设置左边距

### 功能说明

设置打印的左边距，设置后除非主动取消，否则一直生效

### 方法名称

line\_setLeftMargin(int margin)

### 参数说明

参数	名字	说明
margin	边距	点数（8点/mm） 0-65535 需要考虑左右边距，如果同时设置左右边距当可打印区域<100时设置失败

### 返回参数

成功则返回true,失败则返回false。

## 6.6. 设置右边距

### 功能说明

设置打印的右边距，设置后除非主动取消，否则一直生效

### 方法名称

line\_setRightMargin(int margin)

### 参数说明

参数	名字	说明
margin	边距	点数（8点/mm） 0-65535 需要考虑左右边距，如果同时设置左右边距当可打印区域<100时设置失败

### 返回参数

成功则返回true,失败则返回false。

## 6.7. 设置相对打印位置

### 功能说明

设置相对打印位置，设置后除非主动取消，否则一直生效

### 方法名称

line\_setAlign(int align)

参数说明

参数	名字	说明
align	位置	1居中，2居右，其它居左

返回参数

成功则返回true,失败则返回false。

6.8.设置打印文字字体属性

功能说明

对打印字体进行设置，设置后除非主动取消，否则一直生效，需要在打印文字前设置

方法名称

line\_setFontTypeface(Typeface typeface)（设置打印字体）

line\_setFontUnderline(boolean isUnderline)（设置文字是否有下划线）

line\_setFontBold(boolean isBold)（设置文字是否加粗）

line\_setFontItalic(boolean isItalic)（设置文字是否斜体）

line\_setFontSize(int fontSize)（设置字体大小）

line\_setFontScale(int widthScale, int heightScale)（设置字体放大倍数）

line\_setLineSpacing(int lineSpacing)（设置行距）

line\_setLetterSpacing(int letterSpacing)（设置字符间距）

参数说明

方法名称	参数	名字	说明
line_setFontTypeface()	typeface	字体	设置字体样式，若为null，则清除字体样式
line_setFontUnderline()	isUnderline	下划线模式	true：带下划线； false：不带下划线；
line_setFontBold()	isBold	粗体模式	false：取消粗体模式； true：设置为粗体模式；
line_setFontItalic()	isItalic	斜体模式	false：取消斜体模式； true：设置为斜体模式；
line_setFontSize()	fontSize	字体大小	打印字体的高度，1mm = 8 dot
line_setFontScale()	widthScale	宽度放大倍数	字体放大宽度倍数1-10
line_setFontScale()	heightScale	高度放大倍数	字体放大高度倍数1-10
line_setFontScale()	lineSpacing	行距大小	行距大小，1mm = 8 dot
line_setLetterSpacing()	letterSpacing	字符间距	字符间距，1mm = 8 dot

返回参数

无

## 6.9. 设置反色

### 功能说明

设置反色，设置后除非主动取消，否则一直生效

### 方法名称

line\_setFontReverse(boolean isReverse)

### 参数说明

参数	名字	说明
isReverse	黑白反显模式	true: 设置黑白反显模式; false: 取消黑白反显模式;

### 返回参数

无

### 示例

```
1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2  JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMode.Line);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          printer.line_setMaxPrintWidth(576);
7          printer.line_reset();
8          printer.line_setFontReverse(true); //设置反色
9          printer.line_drawText("你好hello123456\n");
10     }
11 }).start();
```



## 6.10. 绘制文字

### 功能说明

绘制文字，满一行自动打印，可以使用\n控制换行，可调用line\_print可强制打印缓冲区文字

### 方法名称

line\_drawText(String content)

### 参数说明

参数	名字	说明
content	内容	满一行自动打印 可以使用\n换行打印 调用{line_print()}可强制打印

### 返回参数

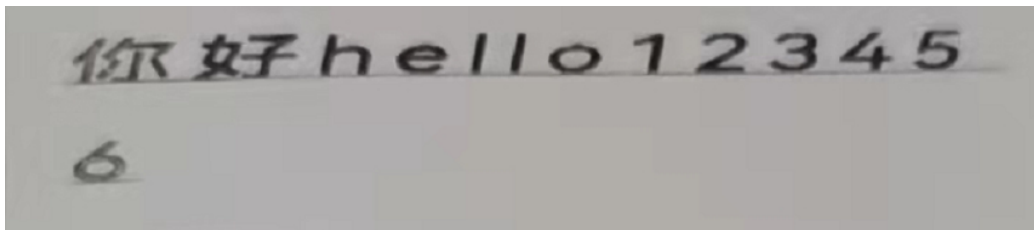
无

### 示例

```

1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2  JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMode.Line);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          printer.line_setMaxPrintWidth(576);
7          printer.line_reset();//清除缓存
8          printer.line_setFontBold(true);//设置文字是否加粗
9          printer.line_setFontUnderline(true);//设置下划线
10         printer.line_setFontItalic(true);//设置斜体
11         printer.line_setFontSize(32);//设置字体大小
12         printer.line_setFontScale(2, 1);//设置字体倍宽
13         printer.line_setLineSpacing(8);//设置行距
14         printer.line_setLetterSpacing(8);//设置字体间距
15         printer.line_drawText("你好hello123456");
16         printer.line_print();
17     }
18 }).start();

```



## 6.11. 线段指令

### 功能说明

添加指令画一条线段

### 方法名称

line\_drawLine(float lineLength, int lineWidth, boolean fullLine)

### 参数说明

参数	名字	说明
lineLength	线段长度	线条长度，如果超出最大打印长度，只打印可打印部分，可调用{line_print()}可强制打印缓冲区
lineWidth	线段的线宽	线段的线宽
fullLine	线段类型	true：实线 false：虚线

### 返回参数

成功则返回true,失败则返回false。

### 示例

```

1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);

```

```

2  JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMo
   de.Line);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          printer.line_setMaxPrintWidth(576);
7          printer.line_reset();//清除缓存
8          printer.line_drawLine(80, 2, true);
9          printer.line_print();
10     }
11 }).start();

```



## 6.12. 一维条码指令

### 功能说明

添加一维条码 默认不显示文字。如果需要打印文字，需要先调用line\_setBarcode1DText  
可调用line\_print可强制打印缓冲区

### 方法名称

line\_drawBarcode1D(int width, int height, String content, int rotate, BarcodeFormat  
format)

### 参数说明

参数	名字	说明
width	条码宽度	条码宽度点数 (8点/mm)
height	条码高度	条码高度点数 (8点/mm)
content	条码数据	条码数据
rotate	顺时针旋转角度	90: 顺时针旋转90度; 180: 顺时针旋转180度;
format	条码类型	BarcodeFormat.CODABAR, BarcodeFormat.CODE_39, BarcodeFormat.CODE_93, BarcodeFormat.CODE_128, BarcodeFormat.EAN_8, BarcodeFormat.EAN_13, BarcodeFormat.ITF, BarcodeFormat.UPC_A, BarcodeFormat.UPC_E;

### 返回参数

成功则返回true,失败则返回false。

### 示例

```

1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);

```

```
2 JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMode.Line);
3 new Thread(new Runnable() {
4     @Override
5     public void run() {
6         printer.line_setMaxPrintWidth(576);
7         printer.line_reset();//清除缓存
8         printer.line_drawBarcode1D( 300, 100, "236563666", 0,
9             BarcodeFormat.CODE_128);
10        printer.line_print();
11    }
12 }).start();
```



### 6.13. 打印条码文字

#### 功能说明

打印条码的文字，仅生效一次,可调用line\_drawBarcode1D使用

#### 方法名称

line\_setBarcode1DText(boolean isPositionBelow, int fontSize, boolean isBold, int margin, boolean breakLine)

#### 参数说明

参数	名字	说明
isPositionBelow	文字在下方	true: 下方 false: 上方
fontSize	字体大小	打印字体的高度，1mm = 8 dot
isBold	粗体模式	false: 取消粗体模式; true: 设置为粗体模式;
margin	文字上下间距	行距，1mm = 8 dot
breakLine	设置条码最大宽度	false: 打印条码以文字和条码的最大宽度为宽度，文字和条码居中展示; true: 以条码宽度为最大宽度，当文字宽度小于条码宽度时，不展示后续文字;

#### 返回参数

无

#### 示例

```
1 JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
```

```

2  JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMo
   de.Line);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          printer.line_setMaxPrintWidth(576);
7          printer.line_reset();//清除缓存
8          printer.line_setBarcode1DText(false, 24, false, 0, true);
9          printer.line_drawBarcode1D( 300, 100, "236563666", 0,
10             BarcodeFormat.CODE_128);
11             printer.line_print();
12     }
13 }).start();

```



## 6.14. QR 条码指令

### 功能说明

添加指令打印QR条码 ,可调用line\_print可强制打印缓冲区

### 方法名称

line\_drawBarcode2D(int width, int height, String content, int rotate, BarcodeFormat format)

### 参数说明

参数	名字	说明
width	二维码宽度	条码宽度点数 (8点/mm)
height	二维码高度	条码高度点数 (8点/mm)
content	条码数据	条码数据
rotate	顺时针旋转角度	90: 顺时针旋转90度; 180: 顺时针旋转180度;
format	条码类型	BarcodeFormat.AZTEC, BarcodeFormat.PDF_417, BarcodeFormat.QR_CODE, BarcodeFormat.DATA_MATRIX,

### 返回参数

成功则返回true,失败则返回false。

### 示例

```

1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);

```

```

2  JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMode.Line);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          printer.line_setMaxPrintWidth(576);
7          printer.line_reset();//清除缓存
8          printer.line_drawBarcode2D( 80, 80, "1234567890", 0,
9              BarcodeFormat.QR_CODE);
10         printer.line_print();
11     }
12 }).start();

```



## 6.15. 制图指令

### 功能说明

位图打印指令，该指令可以打印黑白位图

### 方法名称

line\_drawGraphic(Bitmap bitmap)

### 参数说明

参数	名字	说明
bmp	待打印位图	待打印位图

### 返回参数

成功则返回true,失败则返回false。

### 示例

```

1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2  JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMode.Line);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          printer.line_setMaxPrintWidth(576);
7          printer.line_reset();//清除缓存
8          printer.line_drawGraphic( bitmap);
9          printer.line_print();
10     }
11 }).start();

```

## 6.16. 曲线

### 功能说明

打印曲线的线段（曲线是由连续的线段组成的，点可以理解为长度为1的线段）

### 方法名称



line\_drawLineSegments(int... xPos)

#### 参数说明

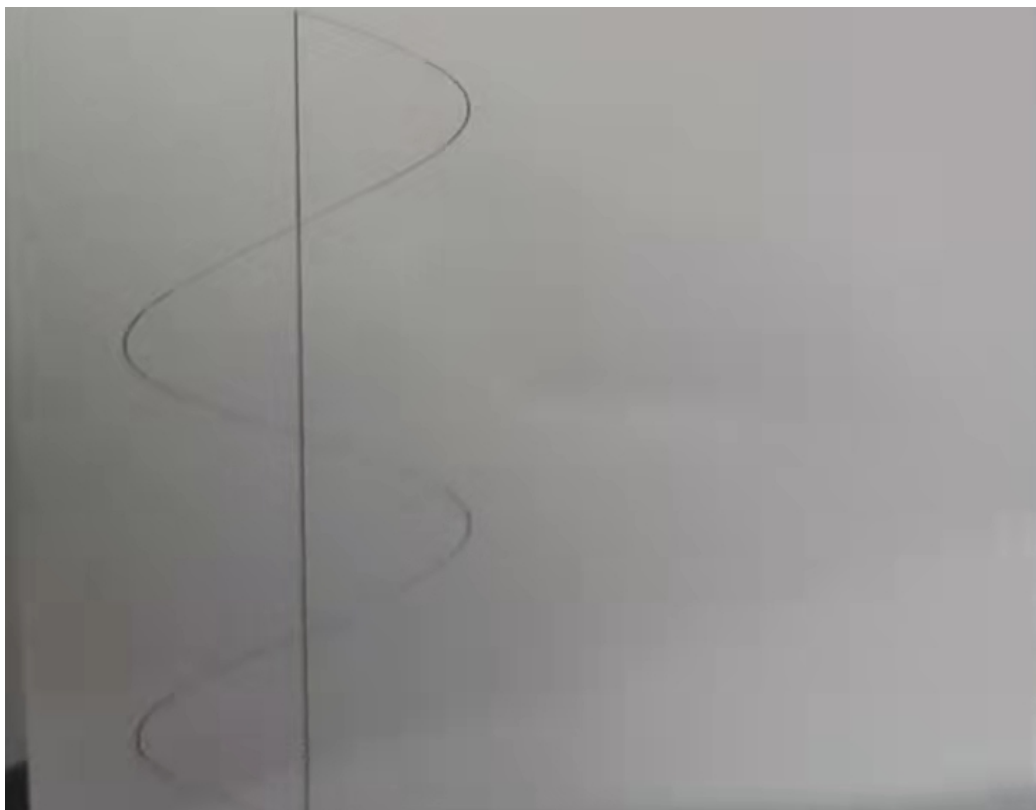
参数	名字	说明
xPos	点	x坐标数组，每2个数表示一个点或线段，x取值0-243。 如果仅打印一个点X，即参数传 X, X。 如果打印一个线段X，即传入Xs, Xe。 如果在同一水平线上打印多个线段X1..Xn，即传入Xs1,Xe1,Xs2,Xe2...Xsn,Xen。

#### 返回参数

成功则返回true,失败则返回false。

#### 示例

```
1 JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2 JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMode
3 new Thread(new Runnable() {
4     @Override
5     public void run() {
6         printer.line_setMaxPrintWidth(576);
7         printer.line_reset();//清除缓存
8         int x;
9         for (int y = 1; y <= 512; y++) {
10             x = (int) (sin(Math.PI * y/ 128) * 100 + 130);
11             printer.line_drawLineSegments(130, 130, x, x);
12         }
13     }
14 }).start();
```



## 7. Global PAGE指令

Global PAGE指令支持多国语言文字打印，使用Global PAGE指令需要额外依赖zxing.jar。  
PAGE 指令方法均以page\_开头。

- 1 一般用于标签纸的打印，步骤：
- 2 1、获取打印服务单例对象
- 3 2、调用设置打印纸张大小方法设置打印区域
- 4 3、调用打印文字或图片等方法设置打印内容
- 5 4、调用执行打印方法进行打印

### 7.1. 进入控制指令

#### 功能说明

让打印机进入命令控制状态，并设置打印宽度，高度，配合page\_print使用(打印前必须设置)

#### 方法名称

page\_setup(int width, int height, int paperType)

#### 参数说明

参数	名字	说明
width	页面宽度点数	页面宽度点数（8点/mm）
height	页面高度点数	页面最大高度点数（8点/mm）
paperType	纸张类型	0 间隙纸 1 连续纸 2黑标纸

#### 返回参数

成功则返回true,失败则返回false。

### 7.2. 打印控制内容指令

#### 功能说明

使打印机控制状态结束，并打印出控制部分的内容。（发送完内容后必须调用此方法，否则不会打印发送的指令）

#### 方法名称

page\_print(boolean isUpsideDown, int times)

page\_print()

#### 参数说明

参数	名字	说明
isUpsideDown	打印顺序	true: 上下180颠倒打印; false: 正向打印; 默认 false
times	打印份数	1-255 默认 1

#### 返回参数

成功则返回true,失败则返回false。

### 7.3. 清除文字设置

**功能说明**

清除之前的文字设置

**方法名称**

page\_resetFontSettings()

**返回参数**

无

### 7.4. 设置打印文字字体属性

**功能说明**

对打印字体进行设置，设置后除非主动取消，否则一直生效，需要在打印文字前设置

**方法名称**

page\_setFontTypeface(Typeface typeface)（设置打印字体）  
page\_setFontUnderline(boolean isUnderline)（设置文字是否有下划线）  
page\_setFontBold(boolean isBold)（设置文字是否加粗）  
page\_setFontItalic(boolean isItalic)（设置文字是否斜体）  
page\_setFontSize(int fontSize)（设置字体大小）  
page\_setFontScale(int widthScale, int heightScale)（设置字体放大倍数）  
page\_setLineSpacing(int lineSpacing)（设置行距）  
page\_setLetterSpacing(int letterSpacing)（设置字符间距）

**参数说明**

参数	名字	说明
typeface	字体	设置字体样式，若为null，则清除字体样式
isUnderline	下划线模式	true：带下划线； false：不带下划线；
isBold	粗体模式	false：取消粗体模式； true：设置为粗体模式；
isItalic	斜体模式	false：取消斜体模式； true：设置为斜体模式；
fontSize	字体大小	打印字体的高度，1mm = 8 dot
widthScale	宽度放大倍数	字体放大宽度倍数1-10
heightScale	高度放大倍数	字体放大高度倍数1-10
lineSpacing	行距大小	行距大小，1mm = 8 dot
letterSpacing	字符间距	字符间距，1mm = 8 dot

**返回参数**

无

### 7.5. 设置反色

**功能说明**

设置反色，设置后除非主动取消，否则一直生效

**方法名称**

page\_setFontReverse(boolean isReverse)

参数说明

参数	名字	说明
isReverse	黑白反显模式	true： 设置黑白反显模式; false： 取消黑白反显模式;

返回参数

无

示例

```
1 JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2 JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMode.Page);
3 new Thread(new Runnable() {
4     @Override
5     public void run() {
6         printer.page_setup(576, 800,0);
7         printer.page_resetFontSettings();//清除之前设置的文字信息
8         printer.page_setFontReverse(true);//设置反色
9         printer.page_drawText(10, 210, 400, 700, 0, false,
10             "你好hello123456");
11         printer.page_print();
12     }
13 }).start();
```



7.6. 文字指令

功能说明

打印传参文字，支持多国语言。 配合page\_print使用

方法名称

page\_drawText(int start\_x, int start\_y, int end\_x, int end\_y, int rotate, boolean singleLine, String content)

参数说明

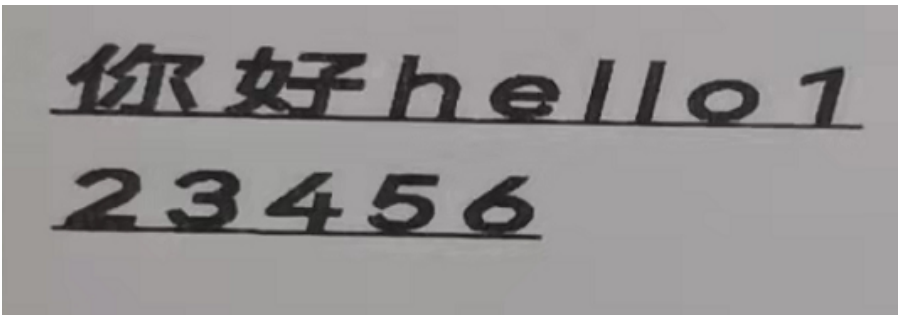
参数	名字	说明
start_x	X轴起始位置	X轴起始位置
start_y	Y轴起始位置	Y轴起始位置
end_x	X轴结束位置	文本打印区域X轴结束位置
end_y,	Y轴结束位置	文本打印区域Y轴结束位置
content	内容	打印文字内容
rotate	顺时针旋转角度	90: 顺时针旋转90度; 180: 顺时针旋转180度; 270: 顺时针旋转170度;
singleLine	单行模式	true: 只打印一行, 多余截断 false: 自动换行

### 返回参数

成功则返回true,失败则返回false。

### 示例

```
1 JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2 JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMode.Page);
3 new Thread(new Runnable() {
4     @Override
5     public void run() {
6         printer.page_setup(576, 800, 0);
7         printer.page_resetFontSettings(); //清除之前设置的文字信息
8         printer.page_setFontUnderline(true); // ( 设置文字是否有下划线 )
9         printer.page_setFontBold(true); // ( 设置文字是否加粗 )
10        printer.page_setFontItaly(true); // ( 设置文字是否斜体 )
11        printer.page_setFontSize(32); // ( 设置字体大小, 默认24 )
12        printer.page_setFontScale(2, 1); // ( 设置字体放大 )
13        printer.page_setLineSpacing(16); // ( 设置行距 )
14        printer.page_setLetterSpacing(8); // ( 设置字符间距 )
15        printer.page_drawText(10, 210, 400, 700, 0, false,
16            "你好hello123456");
17        printer.page_print();
18    }
19 }).start();
```



## 7.7. 矩形框指令

### 功能说明

添加指令画一个矩形框

### 方法名称

page\_drawBox(int start\_x, int start\_y, int width, int height, int lineWidth, boolean fullLine)

### 参数说明

参数	名字	说明
start_x	左上角X坐标	左上角X坐标
start_y	左上角Y坐标	左上角Y坐标
width	矩形宽度	点宽
height	矩形高度	点高
lineWidth	形成矩形的线宽	形成矩形的线宽
fullLine	线段类型	true: 实线 false: 虚线

### 返回参数

成功则返回true,失败则返回false。

### 示例

```
1 JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2 JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMode.Page);
3 new Thread(new Runnable() {
4     @Override
5     public void run() {
6         printer.page_setup(576, 800,0);
7         printer.page_drawBox( 0, 10, 510, 110, 2, false);
8         printer.page_print();
9     }
10 }).start();
```



## 7.8. 线段指令

### 功能说明

添加指令画一条线段

### 方法名称

page\_drawLine(int start\_x, int start\_y, int end\_x, int end\_y, int lineWidth, boolean fullLine)

### 参数说明

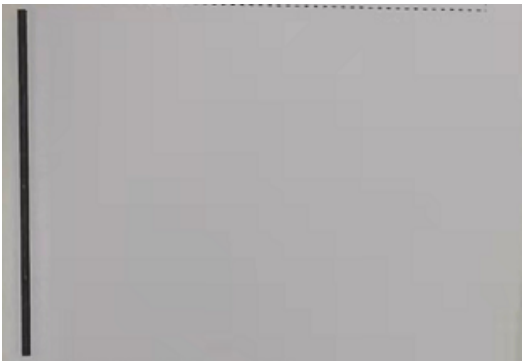
参数	名字	说明
lineWidth	线段的线宽	线段的线宽
start_x	左上角X坐标	左上角X坐标
start_y	左上角Y坐标	左上角Y坐标
end_x	右下角X坐标	右下角X坐标
end_y	右下角Y坐标	右下角Y坐标
fullLine	线段类型	true: 实线 false: 虚线

**返回参数**

成功则返回true,失败则返回false。

**示例**

```
1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2  JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMode
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          printer.page_setup(576, 800,0);
7          printer.page_drawLine(2, 0, 10, 510, 10, true);
8          printer.page_print();
9      }
10 }).start();
```



**7.9. 一维条码指令**

**功能说明**

添加一维条码 默认不显示文字。如果需要打印文字，需要先调用page\_setBarcode1DText

**方法名称**

page\_drawBarcode1D(int start\_x, int start\_y, int width, int height, String content, int rotate, BarcodeFormat format)

**参数说明**

参数	名字	说明
start_x	条码开始的X轴坐标	条码开始的X轴坐标
start_y	条码开始的Y轴坐标	条码开始的Y轴坐标
width	条码宽度	条码宽度点数 (8点/mm)
height	条码高度	条码高度点数 (8点/mm)
content	条码数据	条码数据
rotate	顺时针旋转角度	90: 顺时针旋转90度; 180: 顺时针旋转180度;
format	条码类型	BarcodeFormat.CODABAR, BarcodeFormat.CODE_39, BarcodeFormat.CODE_93, BarcodeFormat.CODE_128, BarcodeFormat.EAN_8, BarcodeFormat.EAN_13, BarcodeFormat.ITF, BarcodeFormat.UPC_A, BarcodeFormat.UPC_E;

### 返回参数

成功则返回true,失败则返回false。

### 示例

```

1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2  JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMode.Page);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          printer.page_setup(576, 800, 0);
7          printer.page_drawBarcode1D(10, 10, 300, 100, "236563666", 0,
8              BarcodeFormat.CODE_128);
9          printer.page_print();
10     }
11 }).start();

```



## 7.10. 打印条码文字



功能说明

打印条码的文字，仅生效一次，配合page\_drawBarcode1D使用

方法名称

page\_setBarcode1DText(boolean isPositionBelow, int fontSize, boolean isBold, int margin, boolean breakLine)

参数说明

参数	名字	说明
isPositionBelow	文字在下方	true: 下方 false: 上方
fontSize	字体大小	打印字体的高度，1mm = 8 dot
isBold	粗体模式	false: 取消粗体模式; true: 设置为粗体模式;
margin	文字上下间距	行距，1mm = 8 dot
breakLine	设置条码最大宽度	false: 打印条码以文字和条码的最大宽度为宽度，文字和条码居中展示; true: 以条码宽度为最大宽度，当文字宽度小于条码宽度时，不展示后续文字;

返回参数

无

示例

```
1 JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2 JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMode.Page);
3 new Thread(new Runnable() {
4     @Override
5     public void run() {
6         printer.page_setup(576, 800,0);
7         printer.page_setBarcode1DText(false, 24, false, 0, true);
8         printer.page_drawBarcode1D(10, 10, 300, 100,"236563666", 0,
9             BarcodeFormat.CODE_128);
10        printer.page_print();
11    }
12 }).start();
```



7.11. QR 条码指令

功能说明

添加指令打印QR条码

方法名称

page\_drawBarcode2D(int start\_x, int start\_y, int width, int height, String content, int rotate, BarcodeFormat format)

参数说明

参数	名字	说明
start_x	条码开始的X轴坐标	条码开始的X轴坐标
start_y	条码开始的Y轴坐标	条码开始的Y轴坐标
width	二维码宽度	条码宽度点数（8点/mm）
height	二维码高度	条码高度点数（8点/mm）
content	条码数据	条码数据
rotate	顺时针旋转角度	90：顺时针旋转90度； 180：顺时针旋转180度；
format	条码类型	BarcodeFormat.AZTEC, BarcodeFormat.PDF_417, BarcodeFormat.QR_CODE, BarcodeFormat.DATA_MATRIX,

返回参数

成功则返回true,失败则返回false。

示例

```
1 JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2 JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMode.Page);
3 new Thread(new Runnable() {
4     @Override
5     public void run() {
6         printer.page_setup(576, 800,0);
7         printer.page_drawBarcode2D(8, 8, 80, 80, "1234567890", 0,
8             BarcodeFormat.QR_CODE);
9         printer.page_print();
10    }
11 }).start();
```



## 7.12. 制图指令

### 功能说明

位图打印指令，该指令可以打印黑白位图

### 方法名称

page\_drawGraphic(int start\_x, int start\_y, Bitmap bmp)

### 参数说明

参数	名字	说明
start_x	位图开始的X轴坐标	位图开始的X轴坐标
start_y	位图开始的Y轴坐标	位图开始的Y轴坐标
bmp	待打印位图	待打印位图

### 返回参数

成功则返回true,失败则返回false。

### 示例

```
1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2  JXGlobalPrinter printer = new JXGlobalPrinter(api, JXGlobalPrinter.PrintMode
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          printer.page_setup(576, 800,0);
7          printer.page_drawGraphic(0, 0, bmp);
8          printer.page_print();
9      }
10 }).start();
```

## 7.13. 定位到标签

### 功能说明

定位到标签，找标签缝走纸，找不到最多走10cm

### 方法名称

page\_feed()

### 返回参数

成功则返回true,失败则返回false。

## 8.ESC指令

### 8.1. 初始化打印机设置

#### 功能说明

使所有设置恢复到打印机开机时的默认值

#### 方法名称

esc\_reset()

#### 返回参数

成功则返回true,失败则返回false。

## 8.2.设置打印文字字体属性

### 功能说明

对打印字体进行设置，需要在打印文字前设置（配合8.12打印文字使用）

### 方法名称

- esc\_bold(boolean isBold)（设置加粗模式）
- esc\_underline(int n)（选择下划线模式）
- esc\_default\_line\_height()（设置默认行高）
- esc\_line\_height(int n)（设置行高）
- esc\_right\_spacing(int n)（设定字符间距）
- esc\_font(int n)（选择打印字体）
- esc\_rotate(int n)（选择/取消顺时针旋转）
- esc\_absolute\_print\_position(int position)（设置打印绝对位置）
- esc\_relative\_print\_position(int n)（将打印位置从当前位置移至 n ×（水平运动单位）

### 参数说明

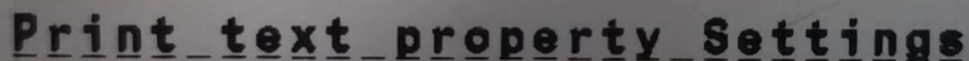
方法	参数	名字	说明
esc_bold	isBold	是否设置为加粗模式	true：设置为加粗模式； false：取消加粗模式；
esc_underline	n	是否设置为下划线模式	n=1或n=49：设置为下划线模式且设置下划线高度为1点； n=2或n=50：设置为下划线模式且设置下划线高度为2点； n=其它：取消下划线模式；
esc_line_height	n	行高值	设置行高为[n×纵向或横向移动单位]英寸
esc_right_spacing	n	间距	n<0：设定右侧字符间距为0； n≥0或n≤255：设定右侧字符间距为n*（水平或垂直移动单位）； n>255：设定右侧字符间距为255*（水平或垂直移动单位）；
esc_font	n	字体类型	n=1 或 49 时选择字体B；n=2 或 50 时选择字体C； n=3 或 51 时选择字体D；n为其他值时选择字体A；
esc_rotate	n	字符集类型	n=1或n=49：设置90°顺时针旋转模式； n=2或n=50：设置180°顺时针旋转模式； n=3或n=51：设置270°顺时针旋转模式； n取其他值时取消旋转模式；
esc_absolute_print_position	position	表示绝对打印位置的低位	将当前位置设置到距离行首 position ×（横向或纵向移动单位）处
esc_relative_print_position	n	x坐标偏移量	将打印位置从当前位置移至n ×（水平运动单位）点

### 示例

```

1  JXInterfaceAPI mAPI = PrinterManager.getInstance().getApi();
2  JXPrinter printer = new JXPrinter(mAPI);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          printer.esc_reset();
7          printer.esc_bold(true);
8          printer.esc_underline(1);
9          //printer.esc_default_line_height();
10         printer.esc_line_height(5);
11         printer.esc_right_spacing(5);
12         printer.esc_font(0);
13         printer.esc_rotate(0);
14         printer.esc_absolute_print_position(10);
15         //printer.esc_relative_print_position(int n);
16         printer.esc_print_text("Print text property Settings");
17     }
18 }).start();
19
20

```



### 8.3.设置打印左边距

#### 功能说明

设置打印左边距

#### 方法名称

esc\_left\_margin(int n)

#### 参数说明

参数	名字	说明
n	左边距	将左边距设为n

#### 返回参数

成功则返回true,失败则返回false。

#### 示例

```

1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2  JXPrinter printer = new JXPrinter(mAPI);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          printer.esc_reset();
7          try {
8              InputStream is = getResources().getAssets().open("logo.jpeg");
9              Bitmap bitmap = BitmapFactory.decodeStream(is);
10             printer.esc_print_formfeed();
11             printer.esc_left_margin(10); //打印左边距

```

```

12         printer.esc_bitmap_mode(1, bitmap);
13         printer.esc_reset();
14         printer.esc_print_formfeed();
15     } catch (IOException e) {
16         e.printStackTrace();
17     }
18 }
19 }).start();

```



## 8.4. 选择对齐模式

### 功能说明

设置打印内容的对齐模式。

### 方法名称

esc\_align(int n)

### 参数说明

参数	名字	说明
n	对齐模式	n=1或n=49: 选择居中对齐 n=2或n=50: 选择右对齐 n=其它: 选择左对齐

### 返回参数

成功则返回true,失败则返回false。

### 示例

```

1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2  JXPrinter printer = new JXPrinter(mAPI);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          printer.esc_reset();
7          try {
8              InputStream is = getResources().getAssets().open("logo.jpeg");
9              Bitmap bitmap = BitmapFactory.decodeStream(is);
10             printer.esc_print_formfeed();
11             printer.esc_align(1); //打印内容的对齐模式
12             printer.esc_bitmap_mode(1, bitmap);
13             printer.esc_reset();
14             printer.esc_print_formfeed();
15         } catch (IOException e) {

```

```

16         e.printStackTrace();
17     }
18 }
19 }).start();

```



## 8.5. 水平制表符

### 功能说明

将打印位置移动至下一水平制表符位置（配合8.6方法使用）

### 方法名称

esc\_next\_horizontal\_tab()

### 返回参数

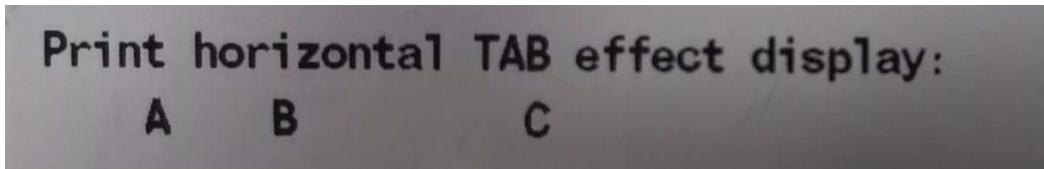
成功则返回true,失败则返回false。

### 示例

```

1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2  JXPrinter printer = new JXPrinter(mAPI);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          printer.esc_reset();
7          printer.esc_print_text("\n Print horizontal TAB effect display : \n");
8          printer.esc_horizontal_tab_position(new int[]{5,10,20});
9          printer.esc_next_horizontal_tab();
10         printer.esc_print_text("A");
11         printer.esc_next_horizontal_tab();
12         printer.esc_print_text("B");
13         printer.esc_next_horizontal_tab();
14         printer.esc_print_text("C");
15         printer.esc_print_formfeed();
16     }
17 }).start();
18

```



### 8.6. 设置水平制表符位置

功能说明

设置水平制表符位置(只在同一行有作用，结合8.5水平制表符使用)

方法名称

esc\_horizontal\_tab\_position(int[]n)

参数说明

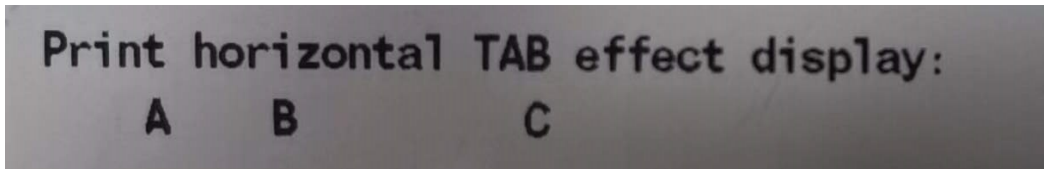
参数	名字	说明
n	偏移量数组	n 的长度表示横向跳格数，n[k]表示第 k 个跳格位置的值，当 n 的长度大于 32 时，只取前 32 个值；当 n[k]大于等于 n[k-1]时忽略该命令。当 n[k]≤0 或 n[k]≥255 时忽略该命令。

返回参数

成功则返回true,失败则返回false。

示例

```
1 JXPrinter printer = new JXPrinter(mAPI);
2 new Thread(new Runnable() {
3     @Override
4     public void run() {
5         printer.esc_reset();
6         printer.esc_print_text("\n Print horizontal TAB effect display:\n");
7         printer.esc_horizontal_tab_position(new int[]{5,10,20});
8         printer.esc_next_horizontal_tab();
9         printer.esc_print_text("A");
10        printer.esc_next_horizontal_tab();
11        printer.esc_print_text("B");
12        printer.esc_next_horizontal_tab();
13        printer.esc_print_text("C");
14        printer.esc_print_formfeed();
15    }
16 }).start();
17
```



### 8.7. 选择国际字符集

功能说明

选择国际字符集



方法名称

```
esc_national_character_set(int n)
```

参数说明

参数	名字	说明
n	字符集类型	n≤0或n>13： 选择America字符集； n=1： 选择France字符集； n=2： 选择German字符集； n=3： 选择UK字符集； n=4： 选择Denmar字符集； n=5： 选择Sweden字符集； n=6： 选择Italy字符集； n=7： 选择Spain字符集； n=8： 选择Japan字符集； n=9： 选择Norway字符集； n=10： 选择Denmar字符集； n=11： 选择SpainII字符集； n=12： 选择Latin字符集； n=13： 选择Korea字符集；

返回参数

成功则返回true,失败则返回false。

8.8. 选择字符代码页

功能说明

选择字符代码页

方法名称

```
esc_character_code_page(int n)
```

参数说明

参数	名字	说明
n	代码标识	<p>n=1：选择Page1Katakana;</p> <p>n=2：选择Page2Multilingual(Latin-1) [CP850];</p> <p>n=3：选择Page 3Portuguese[CP860];</p> <p>n=4：选择Page4Canadian-French[CP863];</p> <p>n=5：选择Page5Nordic[CP865];</p> <p>n=6：选择Page6Slavic(Latin-2)[CP852];</p> <p>n=7：选择Page7Turkish[CP857];</p> <p>n=8：选择Page8 Greek [CP737];</p> <p>n=9：选择Page9Russian(Cyrillic) [CP866];</p> <p>n=10：选择Page10 Hebrew [CP862];</p> <p>n=11：选择Page 11 Baltic [CP775];</p> <p>n=12：选择Page 12 Polish;</p> <p>n=13：选择Page 13 Latin-9 [ISO8859-15];</p> <p>n=14：选择Page 14 Latin1[Win1252];</p> <p>n=15：选择Page 15 Multilingual Latin I + Euro[CP858];</p> <p>n=16：选择Page 16 Russian(Cyrillic) [CP855];</p> <p>n=17：选择 Page 17 Russian(Cyrillic) [Win1251];</p> <p>n=18：选择Page 18 Central Europe[Win1250];</p> <p>n=19：选择Page 19 Greek[Win1253];</p> <p>n=20：选择Page20 Turkish[Win1254];</p> <p>n=21：选择Page21Hebrew[Win1255];</p> <p>n=22：选择Page22Vietnam[Win1258];</p> <p>n=23：选择Page23Baltic[Win1257];</p> <p>n=24：选择Page 24Azerbaijani;</p> <p>n=30：选择Thai[CP874];</p> <p>n=40：选择Page25Arabic[CP720];</p> <p>n=41：选择Page 26 Arabic [Win 1256];</p> <p>n=42：选择Page 27 Arabic (Farsi);</p> <p>n=43：选择Page 28 Arabic presentation formsB;</p> <p>n=50：选择Page 29 Page25 Hindi_Devanagari;</p> <p>n=252：选择Page 30Japanese[CP932];</p> <p>n=253：选择Page 31 Korean[CP949];</p> <p>n=254：选择Page 32 Traditional Chinese[CP950];</p> <p>n=255：选择 Page33SimplifiedChinese[CP936];</p> <p>n取其他值时选择 Page0USA, StandardEurope[CP437];</p>

返回参数

成功则返回true,失败则返回false。

### 8.9. 设置汉字模式

**功能说明**

设置汉字模式

**方法名称**

esc\_chinese\_mode(boolean b)

**参数说明**

参数	名字	说明
b	是否设置为汉字模式	true: 设置为汉字模式; false: 取消汉字模式;

**返回参数**

成功则返回true,失败则返回false。

### 8.10. 设置字符大小

**功能说明**

设置字符的宽度和高度

**方法名称**

esc\_character\_size(int n)

**参数说明**

参数	名字	说明
n	字符大小	n=2时2倍高; n=3时3倍高; n=4时4倍高; n=20时2倍宽; n=30时3倍宽; n=40时4倍宽; n=22时2倍宽高; n=33时3倍宽高; n=44时4倍宽高; n取其他值时1倍宽高;

**返回参数**

成功则返回true,失败则返回false。

### 8.11. 设置反色

**功能说明**

设置反色

**方法名称**

esc\_black\_white\_reverse(boolean b)

**参数说明**

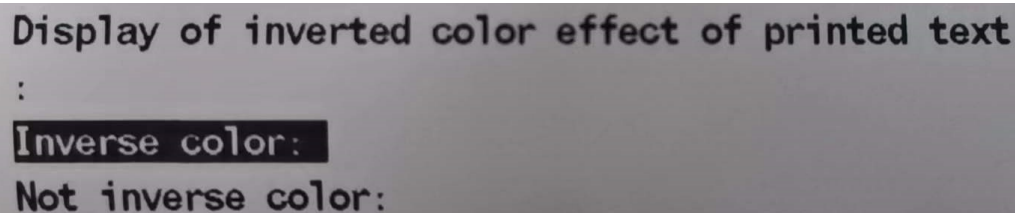
参数	名字	说明
b	是否选择黑白反显打印模式	true: 选择黑白反显打印模式; false: 取消黑白反显打印模式;

## 返回参数

成功则返回true,失败则返回false。

## 示例

```
1 JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2 JXPrinter printer = new JXPrinter(mAPI);
3 new Thread(new Runnable() {
4     @Override
5     public void run() {
6         printer.esc_reset();
7         printer.esc_print_text("Display of inverted color effect of printed
8         printer.esc_black_white_reverse(true);
9         printer.esc_print_text("Inverse color : \n");
10        printer.esc_black_white_reverse(false);
11        printer.esc_print_text("Not inverse color : \n");
12    }
13 }).start();
```



Display of inverted color effect of printed text  
:  
Inverse color:  
Not inverse color:

## 8.12. 打印文本

### 功能说明

发送文本信息到打印机设备，并进行打印操作

### 方法名称

esc\_print\_text(String text)

### 参数说明

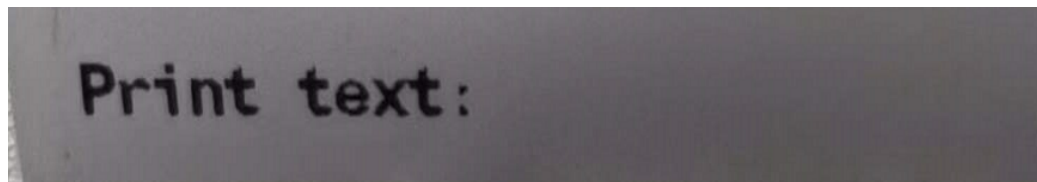
参数	名字	说明
text	文本内容	所要打印的文本内容

## 返回参数

成功则返回true,失败则返回false。

## 示例

```
1 JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2 JXPrinter printer= new JXPrinter(mAPI);
3 new Thread(new Runnable() {
4     @Override
5     public void run() {
6         printer.esc_reset();
7         printer.esc_print_text("Print text : \n");
8     }
9 }).start();
10
```



### 8.13. 设置参数并打印条码

#### 功能说明

设置参数并打印条码

#### 方法名称

`esc_barcode_1d(int HRI_position,int HRI_font,int width,int height,int type,String content)`

#### 参数说明

参数	名字	说明
HRI_position	HRI 字符打印位置	HRI_position=1或HRI_position=49时HRI字符显示在条形码上方; HRI_position=2或HRI_position=50时HRI字符显示在条形码下方; HRI_position取其他值时HRI字符不显示;
HRI_font	HRI 字符字体	HRI_font=1或HRI_font=49时选择字体B; HRI_font取其他值时选择字体A;
width	条码宽度	width=2时设置条形码宽度为2; width=3时设置条形码宽度为3; width取其他值时设置条形码宽度为1;
height	条码高度	1≤height≤255时设置条码高度为height; height取其他值时设置条码高度为162;
type	条码类型	type=0或type=65时选择条码类型为UPC-A; type=1或type=66时选择条码类型为UPC-E; type=2或type=67时选择条码类型为EAN13; type=3或type=68时选择条码类型为EAN8; type=4或type=69时选择条码类型为CODE39; type=5或type=70时选择条码类型为ITF; type=6或type=71时选择条码类型为CODABAR; type=7或type=72时选择条码类型为CODE93; type=8或type=73时选择条码类型为CODE128;
content	条码内容	UPC-A(长度为 11、12)、UPC-E(长度为 7、8、11、12)、EAN13(长度为 12、13)、EAN8(长度为 7、8)、ITF(长度为大于 2 的偶数)只支持数字; CODE39(长度大于 1 且小于 255, 支持数字、英文、空格、'\$'、'%','*','+','-','.'、'/'); CODE93(长度大于 1 且 小于 255, 支持数字、英文、空格、'\$'、'%','+','-','.'、'/'); CODABAR(长度大于 2 且小于 255, 支持数字、英文 ABCDabcd、空格、'\$'、'+','-','.'、'/',':'); CODE128(长度大于 2 且小于 255, 支持所有英文)

**返回参数**

成功则返回true,失败则返回false。

**示例**

```
1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2  JXPrinter printer = new JXPrinter(mAPI);
3  new Thread(new Runnable() {
4      @Override
```

```

5      public void run() {
6          printer.esc_reset();
7          printer.esc_print_formfeed();
8          printer.esc_barcode_1d(0, 1, 3, 80, 8, "123456789012");
9          printer.esc_print_formfeed();
10     }
11 }).start();

```



## 8.14. 打印二维码

### 功能说明

打印二维码

### 方法名称

esc\_print\_barcode\_2d(int size,String content)

esc\_print\_barcode\_2d(int size, int qrVersion, int level, String content)

### 参数说明

参数	名字	说明
size	二维码放大倍数	取值1-4
content	二维码内容	二维码内容
qrVersion	qr版本	qr版本，1-10，取值越大打印条码宽度越大，推荐3
level	qr纠错等级	qr纠错等级，0-3，值越高越容易识别，推荐1

### 返回参数

成功则返回true,失败则返回false。

### 示例

```

1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2  JXPrinter printer = new JXPrinter(mAPI);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          printer.esc_reset();
7          printer.esc_print_formfeed();
8          printer.esc_print_barcode_2d(4, "1234567890");
9          printer.esc_print_formfeed();
10         printer.esc_reset();
11         printer.esc_print_formfeed();
12         printer.esc_print_barcode_2d(4, 3,1,"1234567890");
13         printer.esc_print_formfeed();
14     }

```

```
15 }).start();
```



## 8.15. 打印光栅位图

### 功能说明

打印光栅位图

### 方法名称

esc\_raster\_image(Bitmap bitmap)

### 参数说明

参数	名字	说明
bitmap		bitmap 表示要打印的光栅位图

### 返回参数

成功则返回true,失败则返回false。

### 示例

```
1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2  JXPrinter printer = new JXPrinter(mAPI);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          try {
7              InputStream is = getResources().getAssets().open("logo.jpeg");
8              Bitmap bitmap = BitmapFactory.decodeStream(is);
9              printer.esc_print_formfeed();
10             printer.esc_align(1);
11             printer.esc_raster_image( bitmap);
12             printer.esc_reset();
13             printer.esc_print_formfeed();
14         } catch (IOException e) {
15             e.printStackTrace();
16         }
17     }
```



18 `}).start();`



## 8.16. 选择位图模式打印图片

功能说明

设置位图的模式，并打印图片

方法名称

`esc_bitmap_mode(int m, Bitmap bitmap)`

参数说明

参数	名字	说明
m	位图模式	m=1：位图模式为8点双密度； m=32：位图模式为24点单密度； m=33：位图模式为24点双密度； 除m=1、32、33之外位图模式都为8点单密度；
bitmap	图片	所要打印的图片的信息，由于打印纸宽度有限，图片不可太大。

返回参数

成功则返回true,失败则返回false。

示例

```
1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2  JXPrinter printer = new JXPrinter(mAPI);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          printer.esc_reset();
7          try {
8              InputStream is = getResources().getAssets().open("logo.jpeg");
9              Bitmap bitmap = BitmapFactory.decodeStream(is);
10             printer.esc_print_formfeed();
11             printer.esc_align(1);
12             printer.esc_bitmap_mode(1, bitmap);
13             printer.esc_reset();
14             printer.esc_print_formfeed();
```

```

15         } catch (IOException e) {
16             e.printStackTrace();
17         }
18     }
19 }).start();

```



## 8.17. 打印并行进一行

### 功能说明

基于当前的行间距，打印缓冲区内的数据并走纸一行

### 方法名称

esc\_print\_formfeed()

### 返回参数

成功则返回true,失败则返回false。

## 8.18. 打印并向前走纸n行

### 功能说明

基于当前的行间距，打印缓冲区内的数据并走纸n行

### 方法名称

esc\_print\_formfeed\_row(int n)

### 参数说明

参数	名字	说明
n	走纸行数	n<0: 进纸0行; n>255: 进纸 255 行 0≤n≤255: 进纸 n 行;

### 返回参数

成功则返回true,失败则返回false。

## 8.19. 打印并走纸到右黑标处

### 功能说明

打印全部缓冲区数据并走纸到右黑标处

### 方法名称

esc\_print\_to\_right\_black\_label()

返回参数

成功则返回true,失败则返回false。

示例

8.20. 打印并走纸到左黑标处

功能说明

将打印缓冲区中的数据全部打印出来并走纸到左黑标处

方法名称

esc\_left\_black\_label()

返回参数

成功则返回true,失败则返回false。

8.21. 打印并走纸到标签处

功能说明

打印全部缓冲区数据并走纸到标签处

方法名称

esc\_print\_to\_label()

返回参数

成功则返回true,失败则返回false。

8.22.发送数据字节流

功能说明

对打印机发送字节流

方法名称

esc\_write\_bytes(byte[] bytes)

参数说明

参数	名字	说明
bytes	字节流	所要发送的字节流

返回参数

成功则返回true,失败则返回false。

8.23.读取数据字节流

功能说明

读取打印机返回的字节流

方法名称

esc\_read\_bytes(byte[] bytes)

参数说明

参数	名字	说明
bytes	字节流	读取返回的字节流

返回参数

成功则返回true,失败则返回false。

8.24.清空打印机返回的字节流

功能说明

清空打印机返回的字节流

方法名称

esc\_clear()

返回参数

成功则返回true,失败则返回false。

9. CPCL指令

9.1. 进入控制指令

功能说明

让打印机进入命令控制状态，并设置打印宽度，高度，直到调用“打印控制内容”方法而结束。必须与cpcl\_print()一对一使用

方法名称

cpcl\_pageSetup(int pageWidth, int pageHeight)

参数说明

参数	名字	说明
pageWidth	页面宽度点数	页面宽度点数（8点/mm）
pageHeight	页面高度点数	页面最大高度点数（8点/mm）

返回参数

成功则返回true,失败则返回false。

9.2. 打印控制内容指令

功能说明

使打印机控制状态结束，并打印出控制部分的内容。必须与cpcl\_setup()一对一使用

方法名称

cpcl\_print(int horizontal, int skip)

参数说明

参数	名字	说明
horizontal	打印顺序	0：按照控制命令顺序正常打印； 其它：按照控制命令反序打印；
skip	打印开始位置	是否走纸到定位点后，再打印 0：先走纸到定位点在打印； 其它：不走纸直接打印；

返回参数

成功则返回true,失败则返回false。

9.3. 添加文字指令

功能说明

发送字节流到打印机设备

方法名称

cpcl\_drawText(int text\_x, int text\_y, String text, int fontSize, int rotate, int bold,  
boolean reverse, boolean underline)

参数说明

参数	名字	说明
text_x	X轴起始位置	X轴起始位置
text_y	Y轴起始位置	Y轴起始位置
text	内容	打印文字内容
fontSize	字体编号	v3.8.26之前版本： 1：对应字体编号 55（字体编号介绍见文档未附录）； 2：对应字体编号1； 3：对应字体编号4； 4：对应字体编号1，且字体宽高扩大一倍； 5：对应字体编号4，且字体宽高扩大一倍； 6：对应字体编号1，且字体宽高扩大四倍； 7：对应字体编号4，且字体宽高扩大三倍； 其余：对应字体编号1；  v3.8.26之后版本： 字号为打印字体高度的点数，1mm=8dot。 如打印字高为24，设置24即可。
rotate	旋转模式	1：逆时针旋转90度； 2：逆时针旋转180度； 3：逆时针旋转270度；
bold	粗体模式	0：取消粗体模式； 1：设置为粗体模式；
reverse	黑白反显模式	true：设置黑白反显模式； false：取消黑白反显模式；
underline	下划线模式	true：带下划线； false：不带下划线；

返回参数

成功则返回true,失败则返回false。

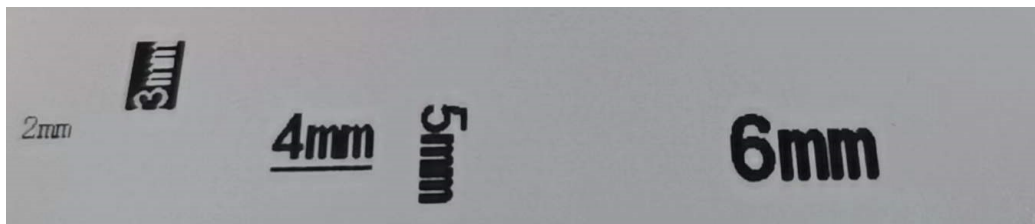
示例

```
1 JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2 JXPrinter printer= new JXPrinter(mAPI);
3 new Thread(new Runnable() {
4     @Override
5     public void run() {
6         printer.cpcl_pageSetup(576, 400, 0);
7         int y = 100;
8         printer.cpcl_drawText(8, y, "2mm", 16,0,0,false,false );
9         printer.cpcl_drawText(56, y, "3mm", 24,1,0,true,false );
10        printer.cpcl_drawText(128, y, "4mm", 32,0,0,false,true );
```

```

11     printer.cpcl_drawText(224, y, "5mm", 40,3,0,false,false );
12     printer.cpcl_drawText(344, y, "6mm", 48,0,1,false,false );
13     printer.cpcl_print(0, 0);
14 }
15 }).start();

```



## 9.4. 添加线段指令

### 功能说明

添加指令画一条线段

### 方法名称

cpcl\_drawLine(int lineWidth, int start\_x, int start\_y, int end\_x, int end\_y, boolean fullline)

### 参数说明

参数	名字	说明
lineWidth	线段的线宽	线段的线宽
start_x	左上角X坐标	左上角X坐标
start_y	左上角Y坐标	左上角Y坐标
end_x	右下角X坐标	右下角X坐标
end_y	右下角Y坐标	右下角Y坐标

### 返回参数

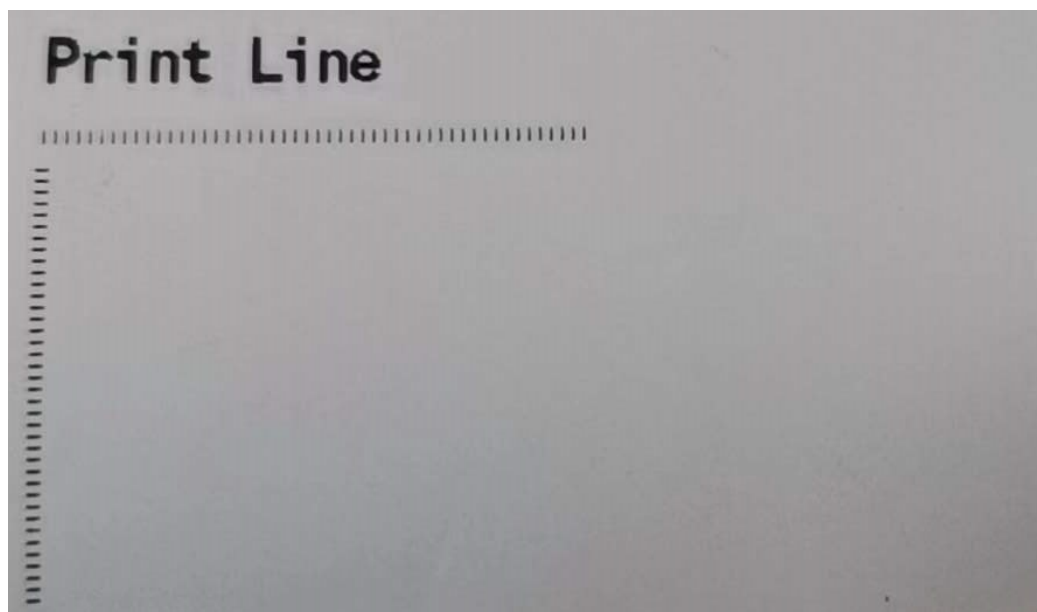
成功则返回true,失败则返回false。

### 示例

```

1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2  JXPrinter printer= new JXPrinter(mAPI);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          printer.cpcl_pageSetup(576, 400, 0);
7          int y = 0;
8          printer.cpcl_drawText(8, y, "Print Line", 24,0,0,false,false );
9          printer.cpcl_drawLine(5,8, 40, 200, 40);
10         printer.cpcl_drawLine(5,8, 50, 8, 200);
11         printer.cpcl_print(0, 0);
12     }
13 }).start();

```



## 9.5. 矩形框指令

### 功能说明

添加指令画一个矩形框

### 方法名称

cpcl\_drawBox(int lineWidth,int top\_left\_x,int top\_left\_y,int bottom\_right\_x,int bottom\_right\_y)

### 参数说明

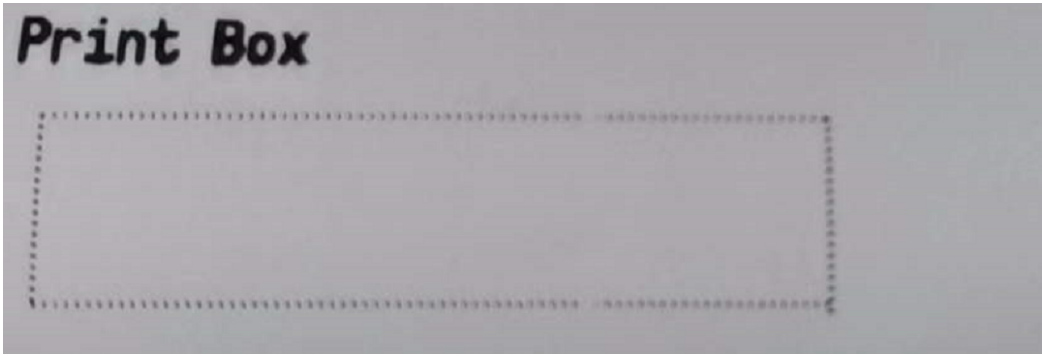
参数	名字	说明
lineWidth	形成矩形的线宽	形成矩形的线宽
top_left_x	左上角X坐标	左上角X坐标
top_left_y	左上角Y坐标	左上角Y坐标
bottom_right_x	右下角X坐标	右下角X坐标
bottom_right_y	右下角Y坐标	右下角Y坐标

### 返回参数

成功则返回true,失败则返回false。

### 示例

```
1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2  JXPrinter printer= new JXPrinter(mAPI);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          printer.cpcl_pageSetup(576, 400, 0);
7          int y = 0;
8          printer.cpcl_drawText(8, y, "Print Box", 24,0,0,false,false );
9          y+=30;
10         printer.cpcl_drawBox( 2,20, y, 310, 110 );
11         printer.cpcl_print(0, 0);
12     }
13 }).start();
```



### 9.6. 一维条码指令

#### 功能说明

添加指令打印一维条码

#### 方法名称

`cpcl_drawBarCode(int start_x, int start_y, String text, int type, int rotate, int linewidth, int height)`

#### 参数说明

参数	名字	说明
start_x	条码开始的X轴坐标	条码开始的X轴坐标
start_y	条码开始的Y轴坐标	条码开始的Y轴坐标
text	条码数据	条码数据
type	条码类型	0: CODE_39; 1: CODE_128; 2: CODE_93; 3: CODABAR; 4: EAN_8; 5: EAN_13; 6: UPC_A; 7: UPC_E;
rotate	条码方向	0: 水平条码; 1: 竖直条码;
linewidth	窄条码的宽度点数	窄条码的宽度点数
height	条码高度点数 (8点/mm)	条码高度点数 (8点/mm)

#### 返回参数

成功则返回true,失败则返回false。

#### 示例

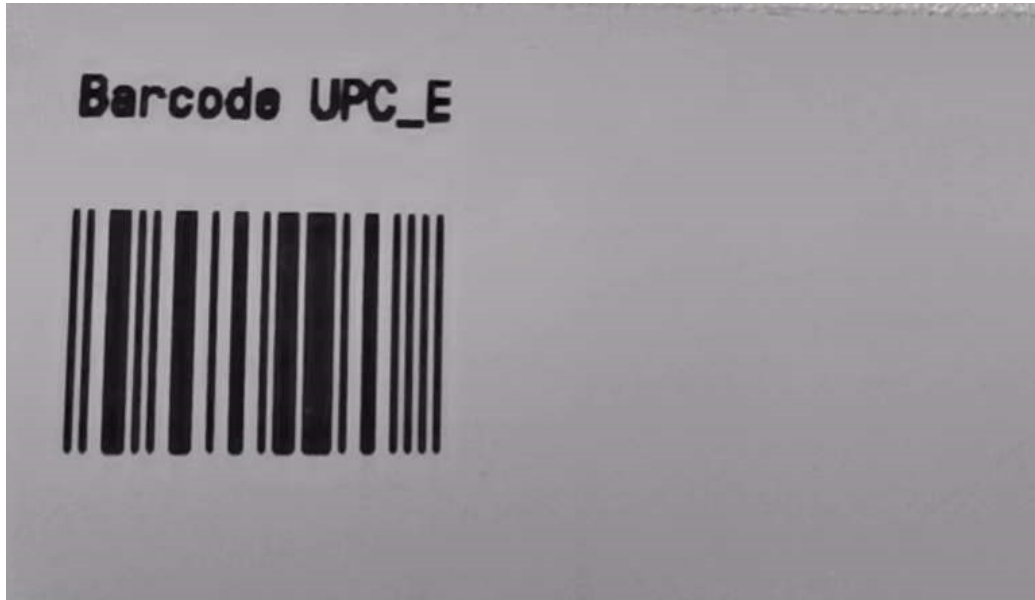
```
1 JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2 JXPrinter printer= new JXPrinter(mAPI);
3 new Thread(new Runnable() {
4     @Override
5     public void run() {
```



```

6     printer.cpcl_pageSetup(576, 400, 0);
7     int y = 0;
8     printer.cpcl_drawText(8, y, "Barcode UPC_E", 24,0,0,false,false );
9     y += 60;
10    printer.cpcl_drawBarCode(8, y, "0402931",7, 0, 3,100);
11    printer.cpcl_print(0, 0);
12 }
13 }).start();

```



## 9.7. 一维条码指令

### 功能说明

添加指令打印一维条码

### 方法名称

cpcl\_drawBarCode(int start\_x, int start\_y, int end\_x, int end\_y, String text, int type, int rotate, int linewidth, int height, int direction)

### 参数说明

参数	名字	说明
start_x	条码开始的X轴坐标	条码开始的X轴坐标
start_y	条码开始的Y轴坐标	条码开始的Y轴坐标
end_x	条码结束的X轴坐标	条码结束的X轴坐标 用来计算，让条码显示在确认区域的中心位置
end_y	条码结束的Y轴坐标	条码结束的Y轴坐标 用来计算，让条码显示在确认区域的中心位置
text	条码数据	条码数据
type	条码类型	0: CODE_39 1: CODE_128 2: CODE_93 3: CODABAR 4: EAN_8 5: EAN_13 6: UPC_A 7: UPC_E
rotate	条码方向	0: 水平条码 1: 竖直条码
linewidth	窄条码的宽度点数	窄条码的宽度点数
height	条码高度点数（8点/mm）	条码高度点数（8点/mm）
direction	对齐方式	0: 左对齐 1: 居中对齐 2: 右对齐

### 返回参数

成功则返回true,失败则返回false。

### 示例

```

1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2  JXPrinter printer= new JXPrinter(mAPI);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          printer.cpcl_pageSetup(576, 400, 0);
7          int y = 0;
8          printer.cpcl_drawText(8, y, "Barcode", 24,0,0,false,false );
9          y += 60;
10         printer.cpcl_drawBarCode(8, y, 308,y+200,"0123456789",1, 0, 3,80,2);
11         printer.cpcl_print(0, 0);
12     }
13 }).start();

```



## 9.8. 一维条码指令（带文字）

### 功能说明

添加指令打印一维条码

### 方法名称

`cpcl_drawBarCode(int start_x, int start_y, String text, int type, int rotate, int lineWidth, int height, int fontSize, int topMargin)`

### 参数说明

参数	名字	说明
start_x	条码开始的X轴坐标	条码开始的X轴坐标
start_y	条码开始的Y轴坐标	条码开始的Y轴坐标
end_x	条码结束的X轴坐标	条码结束的X轴坐标 用来计算，让条码显示在确认区域的中心位置
end_y	条码结束的Y轴坐标	条码结束的Y轴坐标 用来计算，让条码显示在确认区域的中心位置
text	条码数据	条码数据
type	条码类型	0: CODE_39 1: CODE_128 2: CODE_93 3: CODABAR 4: EAN_8 5: EAN_13 6: UPC_A 7: UPC_E
rotate	条码方向	0: 水平条码 1: 竖直条码
linewidth	窄条码的宽度点数	窄条码的宽度点数
fontSize	字体大小：点	条形码下方打印条形码内容字体大小 字体大小：点
topMargin	条形码内容距离条形码的点数	条形码内容距离条形码的点数

返回参数

成功则返回true,失败则返回false。

示例

```
1 JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2 JXPrinter printer= new JXPrinter(mAPI);
3 new Thread(new Runnable() {
4     @Override
5     public void run() {
6         printer.cpcl_pageSetup(576, 400, 0);
7         int y = 0;
8         printer.cpcl_drawText(8, y, "Barcode with text", 24,0,0,false,false
9         y += 60;
10        printer.cpcl_drawBarCode(8, y, "0123456789",1, 0, 3,80,24,0);
11        printer.cpcl_print(0, 0);
12    }
13 }).start();
```



9.9. QR 条码指令

功能说明

添加指令打印QR条码

方法名称

cpcl\_drawQrCode(int start\_x, int start\_y, String text, int rotate, int ver, int lel)

参数说明

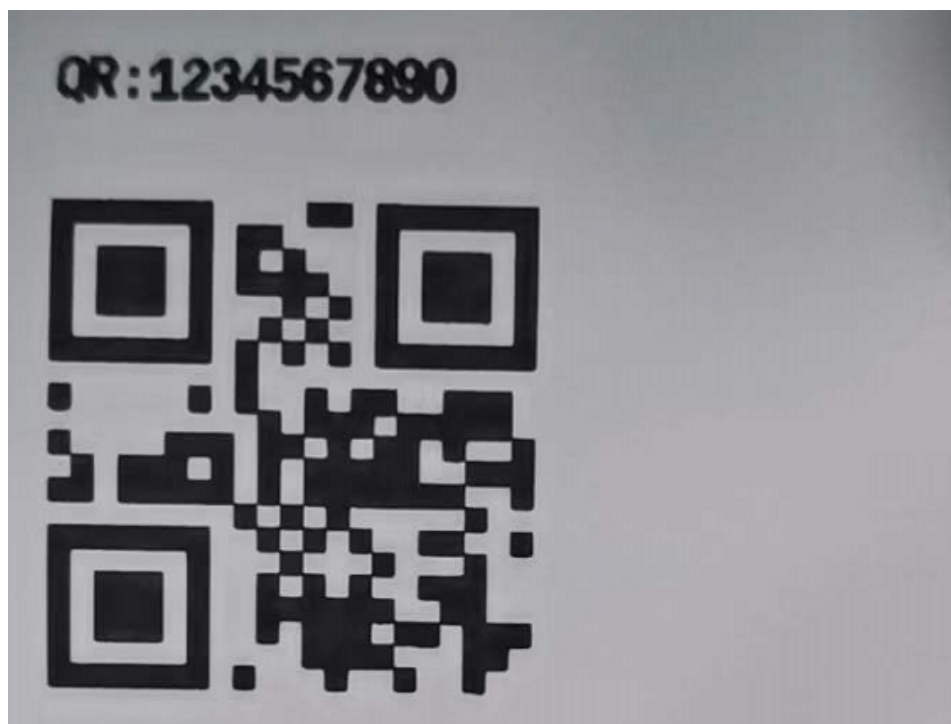
参数	名字	说明
start_x	条码开始的X轴坐标	条码开始的X轴坐标
start_y	条码开始的Y轴坐标	条码开始的Y轴坐标
text	条码数据	条码数据
rotate		无效参数
ver		
lel	高度/宽度的模块点数	高度/宽度的模块点数，值的范围是1-32，默认是6

## 返回参数

成功则返回true,失败则返回false。

## 示例

```
1 JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2 JXPrinter printer= new JXPrinter(mAPI);
3 new Thread(new Runnable() {
4     @Override
5     public void run() {
6         printer.cpcl_pageSetup(576, 400, 0);
7         int y = 0;
8         printer.cpcl_drawText(8, y, "QR:1234567890\n", 24,0,0,false,false );
9         printer.cpcl_drawQrCode(8, y, "1234567890", 1, 9);
10        printer.cpcl_print(0, 0);
11    }
12 }).start();
```



## 9.10. 制图指令

### 功能说明

位图打印指令，该指令可以打印黑白位图

### 方法名称

cpcl\_drawGraphic(int start\_x, int start\_y, int bmp\_size\_x, int bmp\_size\_y, Bitmap bmp)

### 参数说明

参数	名字	说明
start_x	位图开始的X轴坐标	位图开始的X轴坐标
start_y	位图开始的Y轴坐标	位图开始的Y轴坐标
bmp_size_x	待打印位图的宽度	待打印位图的宽度
bmp_size_y	待打印位图的高度	待打印位图的高度
bmp	待打印位图	待打印位图

### 返回参数

成功则返回true,失败则返回false

### 示例

```

1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2  JXPrinter printer= new JXPrinter(mApi);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {
6          try {
7              printer.cpcl_pageSetup(576, 400, 0);
8              InputStream is = getResources().getAssets().open("logo.jpeg");
9              Bitmap bmp = BitmapFactory.decodeStream(is);
10             printer.cpcl_drawGraphic(8, 30,bmp.getWidth(),bmp.getHeight(), k
11             printer.cpcl_print(0, 0);
12         } catch (Exception e) {
13             e.printStackTrace();
14         }
15     }
16 }).start();

```



## 9.11. 定位到标签

### 功能说明

定位到标签

### 方法名称

`cpcl_feed()`

### 返回参数

成功则返回true,失败则返回false。

## 9.12. 发送指令

### 功能说明

发送指令

### 方法名称

`cpcl_writeCmd(String cmd)`

### 返回参数

成功则返回true,失败则返回false。