

Android M20 SDK开发指南-V5.0

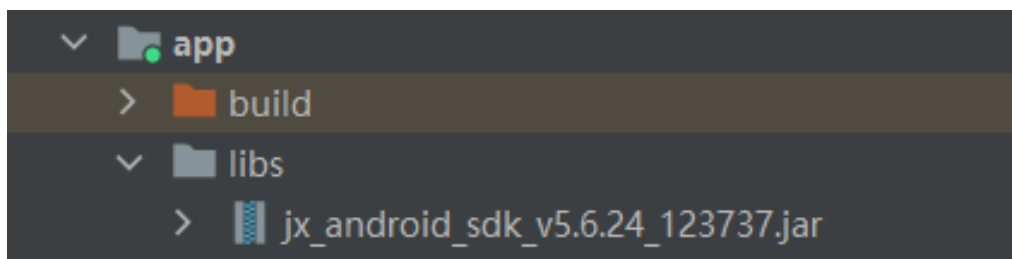
1. SDK项目集成配置

通过以下几步操作，即可在您的应用中使用 M20 android SDK：

1.1. 集成sdk

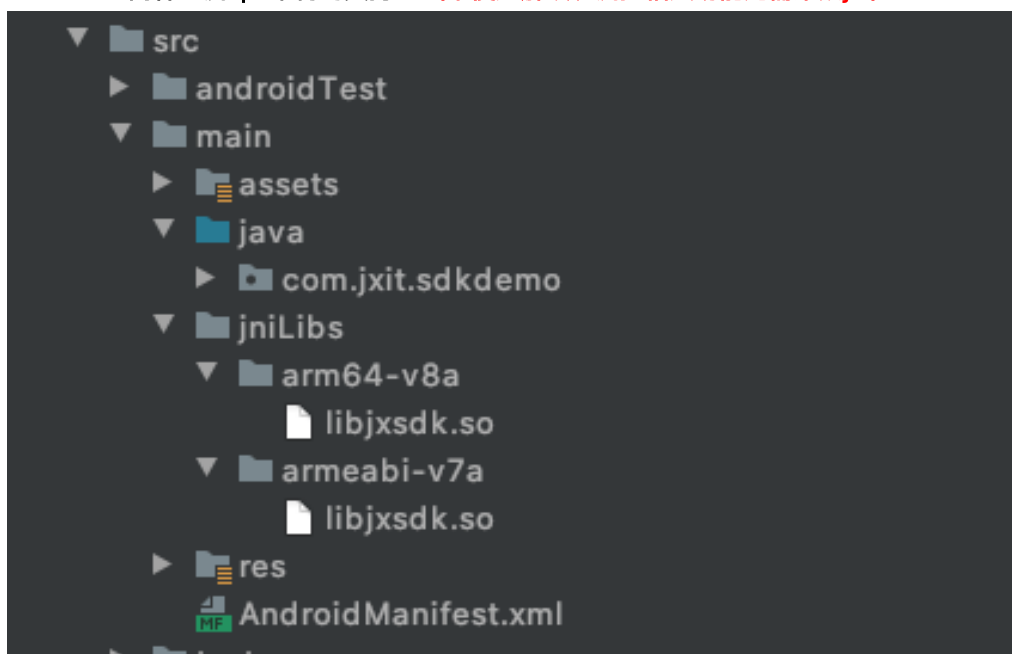
1.1.1. 添加sdk

将jxit_printer_android_v3.0.jar 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。jinLibs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);
    }
}
```

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.jxit.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 `sdksdemo#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 //获取其他指令打印机实例
3 JXPrinter printer= new JXPrinter(mAPI);
```

4.2. 获取Usb打印机实例

```
1 JXInterfaceAPI mApi = JXUsbAPI.getDefault(context);
2 //获取其他指令打印机实例
3 JXPrinter printer= new JXPrinter(mAPI);
```

5.ESC指令

5.1. 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。

5.2.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5.12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 $n \times$ （水平运动单位）

参数说明

参数	名字	是否必须	说明
isBold	是否设置为加粗模式	是	true: 设置为加粗模式; false: 取消加粗模式;
n	是否设置为下划线模式	是	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;
n	行高值	是	设置行高为[n×纵向或横向移动单位]英寸
n	间距	是	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) ;
n	字体类型	是	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;
n	字符集类型	是	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;
position	表示绝对打印位置的低位	是	将当前位置设置到距离行首 position × (横向或纵向移动单位) 处
n	x坐标偏移量	是	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     }
19 }).start();
20
```

Print text property Settings

5.3.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	是否必须	说明
n	左边距	是	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();
```



5.4. 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	是否必须	说明
n	对齐模式	是	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();
```



5.5. 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5.6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
```

```
Print horizontal TAB effect display:
      A      B      C
```

5.6. 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5.5水平制表符使用)

方法名称

esc_horizontal_tab_position(int[]n)

参数说明

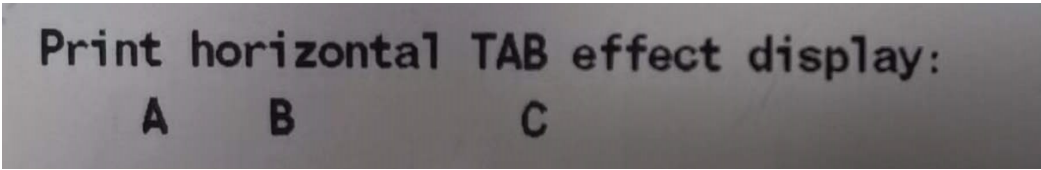
参数	名字	是否必须	说明
n	偏移量数组	是	n 的长度表示横向跳格数，n[k]表示第 k 个跳格位置的值，当 n 的长度大于 32 时，只取前 32 个值；当 n[k]大于等于 n[k-1]时忽略该命令。当 n[k]≤0 或 n[k]≥255 时忽略该命令。

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
```



5.7. 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	是否必须	说明
n	字符集类型	是	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Spainll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。

5.8. 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	是否必须	说明
n	代码标识	是	n=1：选择Page1Katakana; n=2：选择Page2Multilingual(Latin-1) [CP850]; n=3：选择Page 3Portuguese[CP860]; n=4：选择Page4Canadian- French[CP863]; n=5：选择Page5Nordic[CP865]; n=6：选择Page6Slavic(Latin-2) [CP852]; n=7：选择Page7Turkish[CP857]; n=8：选择Page8 Greek [CP737]; n=9：选择Page9Russian(Cyrillic) [CP866]; n=10：选择Page10 Hebrew [CP862]; n=11：选择Page 11 Baltic [CP775]; n=12：选择Page 12 Polish; n=13：选择Page 13 Latin-9 [ISO8859-15]; n=14：选择Page 14 Latin1[Win1252]; n=15：选择Page 15 Multilingual Latin 1 + Euro[CP858];

		Latin + Euro[CP850]; n=16: 选择Page 16 Russian(Cyrillic) [CP855]; n=17: 选择 Page 17 Russian(Cyrillic) [Win1251]; n=18: 选择Page 18 Central Europe[Win1250]; n=19: 选择Page 19 Greek[Win1253]; n=20: 选择Page20 Turkish[Win1254]; n=21: 选择Page21Hebrew[Win1255]; n=22: 选择 Page22Vietnam[Win1258]; n=23: 选择Page23Baltic[Win1257]; n=24: 选择Page 24Azerbaijani; n=30: 选择Thai[CP874]; n=40: 选择Page25Arabic[CP720]; n=41: 选择Page 26 Arabic [Win 1256]; n=42: 选择Page 27 Arabic (Farsi); n=43: 选择Page 28 Arabic presentation formsB; n=50: 选择Page 29 Page25 Hindi_Devanagari; n=252: 选择Page 30Japanese[CP932]; n=253: 选择Page 31 Korean[CP949]; n=254: 选择Page 32 Traditional Chinese[CP950]; n=255: 选择 Page33SimplifiedChinese[CP936]; n取其他值时选择 Page0USA, StandardEurope[CP437];
--	--	--

返回参数

成功则返回true,失败则返回false。

5.9. 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	是否必须	说明
b	是否设置为汉字模式	是	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。

5.10. 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	是否必须	说明
n	字符大小	是	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。

5.11. 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	是否必须	说明
b	是否选择黑白反显打印模式	是	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。

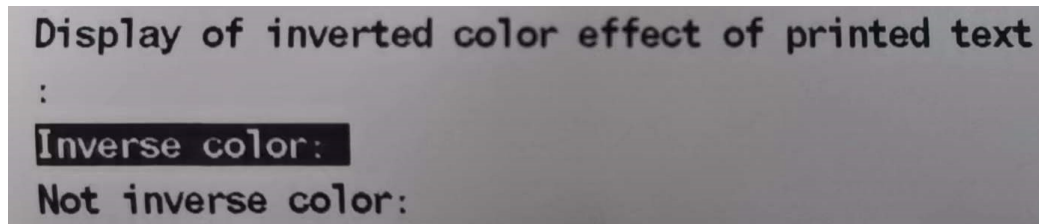
示例

```
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();

```



5.12. 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

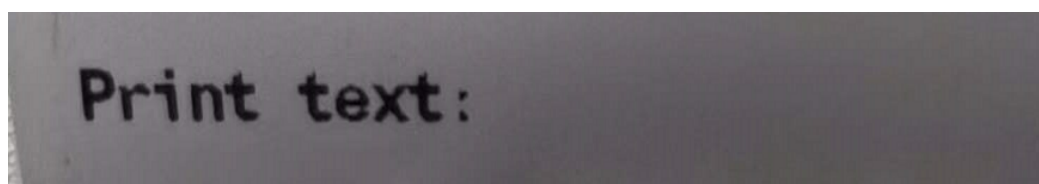
成功则返回true,失败则返回false。

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

```



5.13. 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	是否必须	说明
----	----	------	----

HRI_position	HRI 字符打印位置	是	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	是	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	是	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	是	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	是	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	是	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();
```



5.14. 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	是否必须	说明
size	二维码放大倍数	是	取值1-4
content	二维码内容	是	二维码内容
qrVersion	qr版本	是	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	是	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();

```



5.15. 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	是否必须	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();

```



5.16. 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	是否必须	说明
m	位图模式	是	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	是	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();

```



5.17. 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。

5.18. 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	是否必须	说明
n	走纸行数	是	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。

5.19. 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

5.20. 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。

5.21. 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。

5.22.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	是否必须	说明
bytes	字节流	是	所要发送的字节流

返回参数

成功则返回true,失败则返回false。

5.23.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	是否必须	说明
bytes	字节流	是	读取返回的字节流

返回参数

成功则返回true,失败则返回false。

5.24.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。

5.25. 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。