

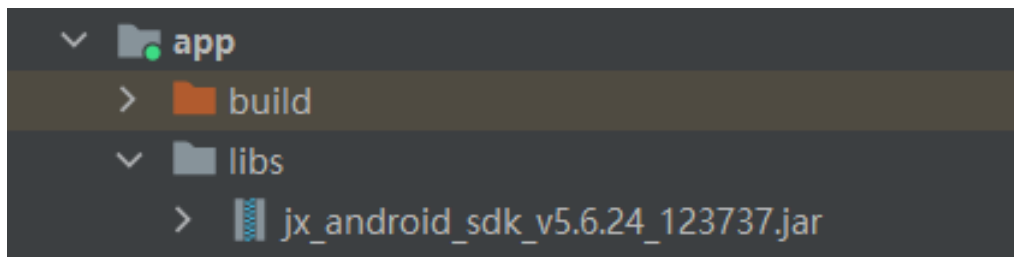
Android M20 SDK Standard Development...

1. SDK project integration configuration

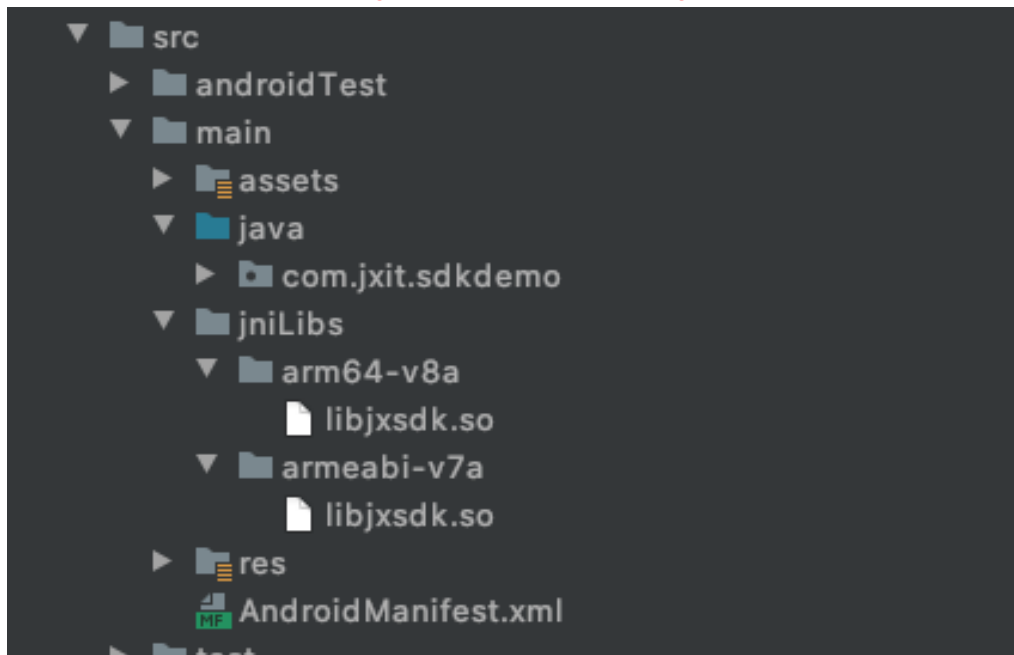
You can use the jixin Android SDK in your application in the following steps:

1.1. Integration SDK

1.1.1. jxit_printer_android_v3.0.jar,copy it to the project directory under the Libs Folder, and synchronize the project to compile it into the project.



1.1.2. Copy the JNILIBS folder to the SRC/main/directory, the same level as the Java folder. jinLibs provides support so for two mainstream CPU architectures: armeabi-v7a and arm64-v8a.(No need to integrate JNI without unlocking and flow control functions)



1.2. Configure the android manifest permission

1.2.1 Integrated USB printer added below in <manifest>

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

Note: Using USB communication requires using an OTG cable and keeping the phone in USB Host mode.

The main activity is in the ANDROIDMANIFESTO. The XML needs to be configured as follows:

```
<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>
```

Add to the <activity> Tab

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

Add to the <intent-filter> Tab

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

Create the XML folder under the RES folder and copy sdkdemo to QWM. XML files, see the SDK demo or refer to the Android USB Host mode development guide.

1.2.2 Integrated Bluetooth printers are added below in <manifest>

```
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" />
```

Note: the use of partial permissions since Android 6.0 requires developers to apply dynamically in the code. Android 12 BLUETOOTH_SCAN, BLUETOOTH_ADVERTISE, and BLUETOOTH_CONNECT runtime permissions were introduced

1.2.3 Integrated WiFi printer is added below in <manifest>

```
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. Initialization SDK

Create a new custom Application and complete the SDK initialization in its onCreate method. The example code is as follows:

```

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

```

In the Androidmanifesto St. The Application is declared in the. XML file

```

<application
    android:name=".DemoApplication"

```

1.4. Obfuscation SDK

If your application uses code obfuscation, add the following configuration to avoid the SDK being obfuscated and the SDK being unavailable

```

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

```

At this point, the SDK integration configuration is complete.

2. Bluetooth connected printers

2.1. General usage flow

(1) searching for a Bluetooth printer (2) a printer Bluetooth pairing (3) establishing a Bluetooth connection (4) sending a printer instruction

2.2. Get the JXBluetoothAPI Global Singleton

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

2.3. Get Connection status

Overview

Gets the current Bluetooth connection status

Name

isConnected()

Response

True is returned when connected, false when not connected.

2.4. Enable/disable Bluetooth search, search for Bluetooth devices

Overview

Enable/disable Bluetooth search, search for Bluetooth devices

Name

startDiscovery()

cancelDiscovery()

Response

Searching for Bluetooth devices requires listening to system Bluetooth broadcasts, action for Bluetooth device. Action, see sdkdemo # Bluetooth fragment or refer to the Android Bluetooth development guide.

2.5. Get a list of paired Bluetooth devices

Overview

Gets a list of paired Bluetooth devices

Name

getBondedDevices()

Response

set<BluetoothDevice>

2.6. Bluetooth pair/unpairings

Overview

Bluetooth pairing/unpairing

Name

createBTBond(BluetoothDevice device)

removeBTBond(BluetoothDevice device)

Response

The default matching password for the printer is 0000, Bluetooth device. Action, see sdkdemo # Bluetooth fragment or refer to the Android Bluetooth development guide.

2.7. Bluetooth connection/disconnect

Overview

Bluetooth connected/disconnected

Name

openConnection(String macAddress)

closeConnection()

Response

Return true successfully, otherwise return false.

3. USB connected printer

3.1. General usage flow

(1) the mobile phone connects with the OTG line in host mode and the printer (2) establishes the USB connection (3) sends the printer instructions

3.2. Get the JXUSBAPI Global Singleton

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

3.3. Get Connection status

Overview

Gets the current Bluetooth connection status

Name

isConnected()

Response

True is returned when connected, false when not connected.

3.4. Get Connected USB devices

Overview

Get the connected USB device

Name

getDevices()

Response

List<UsbDevice>

3.5. USB connectiondisconnect

Overview

USB connection/disconnection

Name

openConnection(UsbDevice device)

closeConnection()

Response

Return true successfully, otherwise return false.

Note: Using USB communication requires using an OTG cable and keeping the phone in USB Host mode.

The main activity is in the ANDROIDMANIFESTO. The XML needs to be configured as follows:

```
<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>
```

Add to the <activity> Tab

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

Add to the <intent-filter> Tab

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

Create the XML folder under the RES folder and copy sdkdemo to QWM. XML files, see the SDK demo or refer to the Android USB Host mode development guide.

4. Get the printer instance

4.1. Get an instance of a Bluetooth printer

```
1 JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2 //Get other command printer instance
3 JXPrinter printer= new JXPrinter(mAPI);
```

4.2. Get a Usb printer instance

```
1 JXInterfaceAPI mApi = JXUsbAPI.getDefault(context);
2 //Get other command printer instance
3 JXPrinter printer= new JXPrinter(mAPI);
```

5.ESC Directive

5.1. Initialize printer settings

Overview

Returns all settings to their default values when the printer was powered on

Name

`esc_reset()`

Response

Successful true, Failure returns false.

5.2.Set print text font properties

Overview

To set the printing font, it needs to be set before printing the text (used in conjunction with 5.12 Printing text)

Name

`esc_bold(boolean isBold)` (set bold mode)

`esc_underline(int n)` (select underline mode)

`esc_default_line_height()` (set the default line height)

`esc_line_height(int n)` (set line height)

`esc_right_spacing(int n)` (set character spacing)

`esc_font(int n)` (select print font)

`esc_rotate(int n)` (select/unrotate clockwise)

`esc_absolute_print_position(int position)` (set the absolute print position)

`esc_relative_print_position(int n)` (move the print position from the current position to
 $n \times$ (horizontal motion unit))

Parameter

Method	Parameter	Name	Extra
esc_bold	isBold	Whether to set to bold mode	true: set to bold mode; false: cancel bold mode;
esc_underline	n	Whether to set to underline mode	n=1 or n=49: set to underline mode and set the underline height to 1 point; n=2 or n=50: set to underline mode and set the underline height to 2 points; n=other: cancel underline mode;
esc_line_height	n	row height value	Set line height to [n × vertical or horizontal shift units] inches
esc_right_spacing	n	spacing	n<0: Set the right character spacing to 0; n≥0 or n≤255: Set the right character spacing to n* (horizontal or vertical movement unit); n>255: Set the right character spacing to 255* (horizontal or vertical movement unit);
esc_font	n	font type	Font B is selected when n=1 or 49; Font C is selected when n=2 or 50; When n=3 or 51, select font D; when n is other values, select font A;
esc_rotate	n	character set type	n=1 or n=49: set 90° clockwise rotation mode; n=2 or n=50: set 180° clockwise rotation mode; n=3 or n=51: set 270° clockwise rotation mode; Cancel the rotation mode when n takes other values;
esc_absolute_print_position	position	Indicates the low-order bits of the absolute print position	Set the current position to position × (horizontal or vertical movement unit) from the beginning of the line
esc_relative_print_position	n	x coordinate offset	Move the print position from the current position to n × (horizontal movement unit)

Examples

```

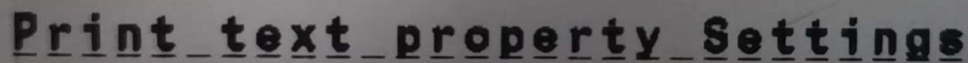
1 JXInterfaceAPI mAPI = PrinterManager.getInstance().getApi();
2 JXPrinter printer = new JXPrinter(mAPI);
3 new Thread(new Runnable() {

```

```

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

```



5.3.Set the print left margin

Overview

set print left margin

Name

esc_left_margin(int n)

Parameter

Parameter	Name	Extra
n	left margin	set the left margin to n

Response

Successful true, Failure returns false.

Examples

```

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. Select the alignment mode

Overview

Sets the alignment mode of the printed content.

Name

esc_align(int n)

Parameter

Parameter	Name	Extra
n	x coordinate offset	n=1 or n=49: select center alignment n=2 or n=50: select right alignment n=other: select left alignment

Response

Successful true, Failure returns false.

Examples

```

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. Horizontal Tabs

Overview

Move the print position to the next horizontal tab position (used with method 5.6)

Name

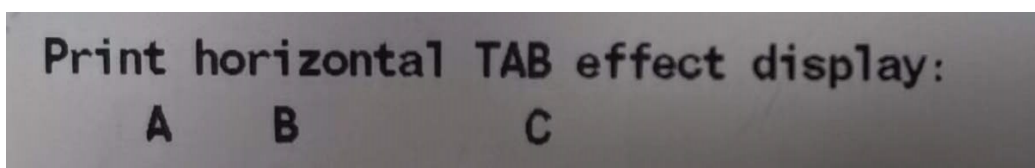
`esc_next_horizontal_tab()`

Response

Successful true, Failure returns false.

Examples

```
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();
```



5.6. Set the horizontal tab position

Overview

Set the horizontal tab position (only works on the same line, used in conjunction with 5.5 horizontal tabs)

Name

esc_horizontal_tab_position(int[]n)

Parameter

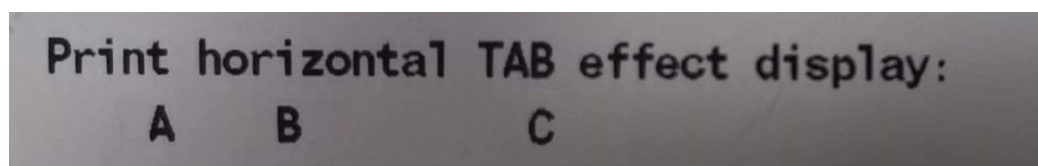
Parameter	Name	Extra
n	offset array	The length of n represents the number of horizontal tabs, and n[k] represents the value of the kth tab position. When the length of n is greater than 32, only the first 32 values are taken; when n[k] is greater than or equal to n[k-1], this command is ignored. This command is ignored when n[k]≤0 or n[k]≥255.

Response

Successful true, Failure returns false.

Examples

```
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
```

5.7. Select International Character Sets

Overview

Select international character set

Name

esc_national_character_set(int n)

Parameter

Parameter	Name	Extra
n	character set type	n≤0 or n>13: select the America character set; n=1: select the French character set; n=2: select German character set; n=3: select UK character set; n=4: select Denmark character set; n=5: select Swedish character set; n=6: select Italy character set; n=7: select Spain character set; n=8: select Japan character set; n=9: select the Norwegian character set; n=10: select Denmark character set; n=11: select SpainII character set; n=12: select Latin character set; n=13: select Korea character set;

Response

Successful true, Failure returns false.

5.8. Select character code page

Overview

Select character code page

Name

esc_character_code_page(int n)

Parameter

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

Response

Successful true, Failure returns false.

5.9. Set Chinese character mode

Overview

Set Chinese character mode

Name

esc_chinese_mode(boolean b)

Parameter

Parameter	Name	Extra
b	Whether to set to Chinese character mode	true: set to Chinese character mode; false: Cancel Chinese character mode;

Response

Successful true, Failure returns false.

5.10. Set character size

Overview

Set the width and height of the characters

Name

esc_character_size(int n)

Parameter

Parameter	Name	Extra
n	character size	2 times higher when n=2; 3 times higher when n=3; 4 times higher when n=4; 2 times the width when n=20; 3 times the width when n=30; 4 times the width when n=40; 2 times the width and height when n=22; 3 times the width and height when n=33; 4 times the width and height when n=44; When n takes other values, it is 1 times the width and height;

Response

Successful true, Failure returns false.

5.11. Set the inverse color

Overview

set inverse color

Name

esc_black_white_reverse(boolean b)

Parameter

Parameter	Name	Extra
b	Whether to select black and white reverse print mode	true: select the black and white reverse display printing mode; false: cancel the black and white reverse printing mode;

Response

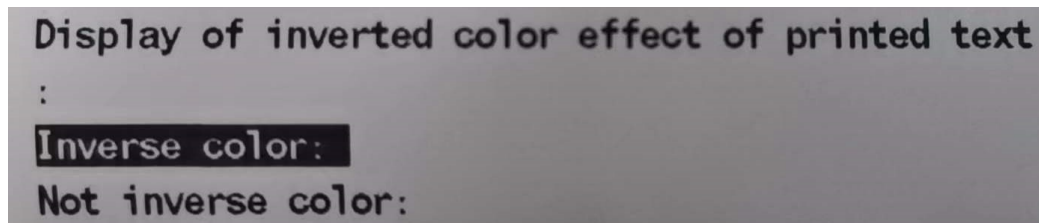
Successful true, Failure returns false.

Examples

```

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. Print Text

Overview

Send text information to the printer device and perform printing operations

Name

esc_print_text(String text)

Parameter

Parameter	Name	Extra
text	text content	the text content to be printed

Response

Successful true, Failure returns false.

Examples

```

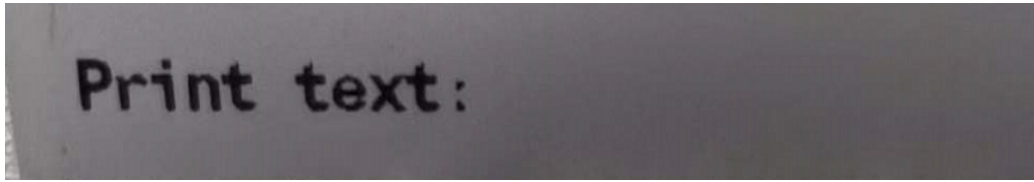
1  JXInterfaceAPI mApi = JXBluetoothAPI.getDefault(context);
2  JXPrinter printer= new JXPrinter(mAPI);
3  new Thread(new Runnable() {
4      @Override
5      public void run() {

```

```

6         printer.esc_reset();
7         printer.esc_print_text("Print text: \n");
8     }
9 }).start();
10

```



5.13. Set parameters and print barcode

Overview

Set parameters and print barcode

Name

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

Parameter

Parameter	Name	Extra
HRI_position	HRI character print position	HRI characters are displayed above the barcode when HRI_position=1 or HRI_position=49; HRI characters are displayed below the barcode when HRI_position=2 or HRI_position=50; HRI characters are not displayed when HRI_position takes other values;
HRI_font	HRI character font	Select font B when HRI_font=1 or HRI_font=49; Select font A when HRI_font takes other values;
width	barcode width	When width=2, set the barcode width to 2; When width=3, set the barcode width to 3; When width takes other values, set the barcode width to 1;
height	Bar code height	When $1 \leq \text{height} \leq 255$, set the barcode height to height; When height takes other values, set the barcode height to 162;
type	Barcode Type	When type=0 or type=65, select the barcode type as UPC-A; When type=1 or type=66, select the barcode type as UPC-E; When type=2 or type=67, select the barcode type as EAN13; When type=3 or type=68, select the barcode type as EAN8;

		barcode type as EAN8; When type=4 or type=69, select the barcode type as CODE39; When type=5 or type=70, select the barcode type as ITF; When type=6 or type=71, select the barcode type as CODABAR; When type=7 or type=72, select the barcode type as CODE93; When type=8 or type=73, select the barcode type as CODE128;
content	barcode content	UPC-A (lengths 11, 12), UPC-E (lengths 7, 8, 11, 12), EAN13 (lengths 12, 13), EAN8 (lengths 7, 8), ITF (lengths greater than 2 even number) only supports numbers; CODE39 (length greater than 1 and less than 255, supports numbers, English, spaces, '\$', '%', '*', '+', '-', '.', '/', '); CODE93 (length greater than 1 and less than 255, supports numbers, English, spaces, '\$', '%', '+', '-', '.', '/', '); CODABAR (length greater than 2 and less than 255, supports numbers, English ABCDabcd, spaces, '\$', '+', '-', '.', '/', ':'); CODE128 (length greater than 2 and less than 255, supports all English)

Response

Successful true, Failure returns false.

Examples

```

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. Print QR code

Overview

print QR code

Name

esc_print_barcode_2d(int size,String content)

esc_print_barcode_2d(int size, int qrVersion, int level, String content)

Parameter

Parameter	Name	Extra
size	QR code magnification	Value 1–4
content	QR code content	QR code content
qrVersion	qr version	qr version, 1–10, the larger the value, the larger the width of the printed barcode, 3 is recommended
level	qr error correction level	qr error correction level, 0–3, the higher the value, the easier it is to identify, recommended 1

Response

Successful true, Failure returns false.

Examples

```
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. Print Raster Bitmaps

Overview

print raster bitmap

Name

esc_raster_image(Bitmap bitmap)

Parameter

Parameter	Name	Extra
bitmap	picture	The information of the picture to be printed should not be too large due to the limited width of the printing paper.

Response

Successful true, Failure returns false.

Examples

```
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 }
```

```

17     }
18     }).start();

```



5.16. Select Bitmap Mode to Print Pictures

Overview

Set the mode of the bitmap, and print the image

Name

esc_bitmap_mode(int m, Bitmap bitmap)

Parameter

Parameter	Name	Extra
m	bitmap mode	m=1: Bitmap mode is 8-point double density; m=32: The bitmap mode is 24-point single density; m=33: Bitmap mode is 24-point double density; Except for m=1, 32, and 33, the bitmap mode is 8-point single density;
bitmap	picture	The information of the picture to be printed should not be too large due to the limited width of the printing paper.

Response

Successful true, Failure returns false.

Examples

```

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. Print a line in parallel

Overview

Based on the current line spacing, print the data in the buffer and feed one line

Name

esc_print_formfeed()

Response

Successful true, Failure returns false.

5.18. Print and advance paper n lines

Overview

Based on the current line spacing, print the data in the buffer and feed n lines

Name

esc_print_formfeed_row(int n)

Parameter

Parameter	Name	Extra
n	Lines of paper	n<0: feed 0 lines; n>255: feed 255 lines 0≤n≤255: feed n lines;

Response

Successful true, Failure returns false.

5.19. Print and feed the paper to the right black mark

Overview

Print all buffer data and feed the paper to the right black mark

Name

`esc_print_to_right_black_label()`

Response

Successful true, Failure returns false.

Examples

5.20. Print and feed the paper to the left black mark

Overview

Print out all the data in the print buffer and feed the paper to the left black mark

Name

`esc_left_black_label()`

Response

Successful true, Failure returns false.

5.21. Print and feeding paper to label

Overview

Print all buffer data and feed paper to label

Name

`esc_print_to_label()`

Response

Successful true, Failure returns false.

5.22.Send a stream of data bytes

Overview

Send a stream of bytes to the printer

Name

`esc_write_bytes(byte[] bytes)`

Parameter

Parameter	Name	Extra
bytes	byte stream	byte stream to send

Response

Successful true, Failure returns false.

5.23.Read data byte stream

Overview

Read the byte stream returned by the printer

Name

`esc_read_bytes(byte[] bytes)`

Parameter

Parameter	Name	Extra
bytes	byte stream	Read the returned byte stream

Response

Successful true, Failure returns false.

5.24.Clears the byte stream returned by the printer

Overview

Clears the byte stream returned by the printer

Name

esc_clear()

Response

Successful true, Failure returns false.

5.25.Unlock the printer

Overview

For custom printers, the printer is unlocked when connecting, otherwise it cannot be used

(Using this feature requires integration with JNI)

Name

set_print_unlock()

Parameter: None

Response

Successful true, Failure returns false.