

# **iMin Printer Developer Documentation**

## Documentation Update Description

Version	Update Date	Update Content	Written By
v1.0.0	2024/1/31	Initial Version	Xie Huayen
v1.0.1	2024/4/18	<ol style="list-style-type: none"><li>1. Add instructions for 2.2 Obtaining printer status</li><li>2. Add note for 3.5t</li></ol>	Xie Huayen

## Table of Content

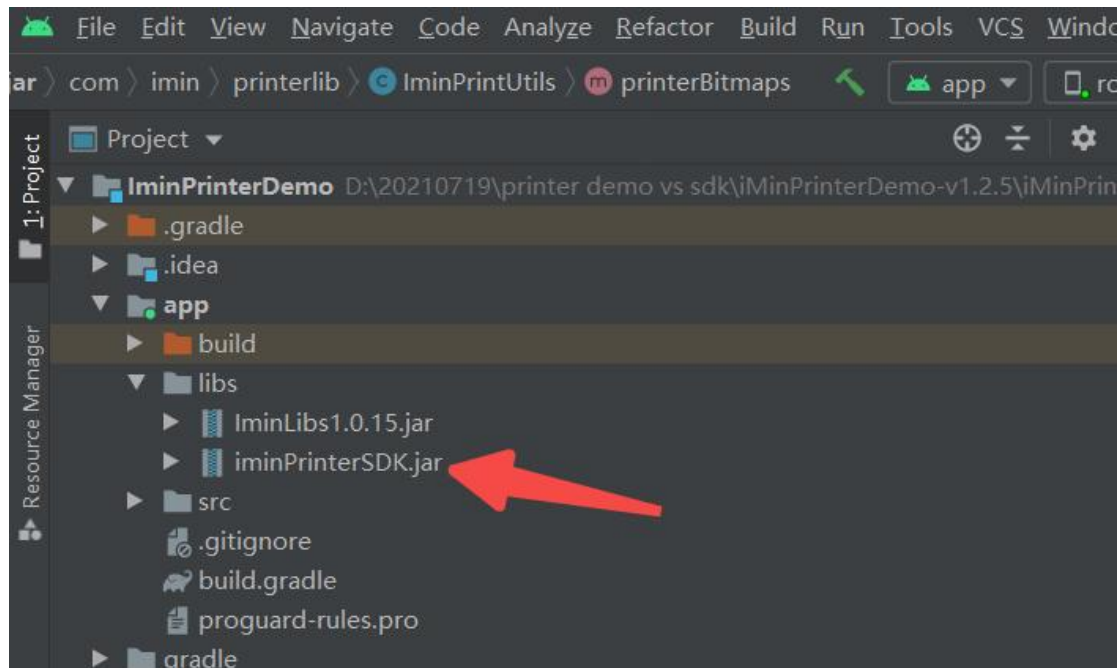
iMinPrinter Developer Documentation .....	1
Documentation Update Description .....	2
1.Printer integration through SDK.jar method .....	5
<b>1.1 Integration Method</b> .....	5
2.1.1. Initialize SPI printer: .....	7
2.1.3. Initialize Bluetooth printing (Supports all models mentioned above) .....	7
2.2.1 SPI Type Printer Method: .....	7
2.2.2.USB Type printer method: .....	8
2.3.1. Print and feed one line .....	8
2.3.2.Print and feed n number of 0.125mm .....	8
2.5.1. Configure alignment method .....	9
2.5.2. Configure text size .....	9
2.5.3 Configure font type .....	10
2.5.4 Configure font style .....	10
2.5.5. Configure the printing text line spacing .....	11
2.5.6. Configure the content width .....	11
2.5.7 Print text image .....	11
2.5.8 Alignment method of adding individual text image .....	11
2.5.9 Print text and picture with Reverse Printing effect .....	12
2.7.1. Set the width of the 1D code .....	12
2.7.2. Set the height of the 1D code .....	13
2.7.3. Set the position of the text of the 1D code .....	13
2.7.4.Print 1D Code .....	13
2.7.5. Print 1D code and set the alignment .....	14
2.8.1. Configure QR code size .....	14
2.8.2.Configure QR code error correction .....	14
2.8.4. Print QR Code .....	15
2.8.5.Print QR Code and add alignment .....	15
2.9.1. Configure paper format .....	15
3.0.1. Print image .....	15
3.0.2. Print image and configure alignment .....	16
3.0.3. Print multiple bitmap .....	16
3.0.4. Print multiple bitmap and configure alignment method .....	16
3.0.5. Bitmap printing with image processing function .....	16
3.0.6. Print 19 bit 1D Code with bitmap printing .....	16
3.1.1. Configure the size of dual QR code .....	17
3.1.2. Configure dual QR code offset value .....	17
3.1.3. Set the left margin of the first QR code .....	18

3.1.4. Configure the left margin of the second QR code .....	18
3.1.5. Configure the version of the first QR code .....	18
3.1.6. Configure the version of the second QR code .....	18
3.1.7. Print QR code .....	18
3.2.1. Obtain the number of cutting .....	18
3.2.2. Obtain printing length of the printer .....	19
3.2.3. Obtain serial number .....	19

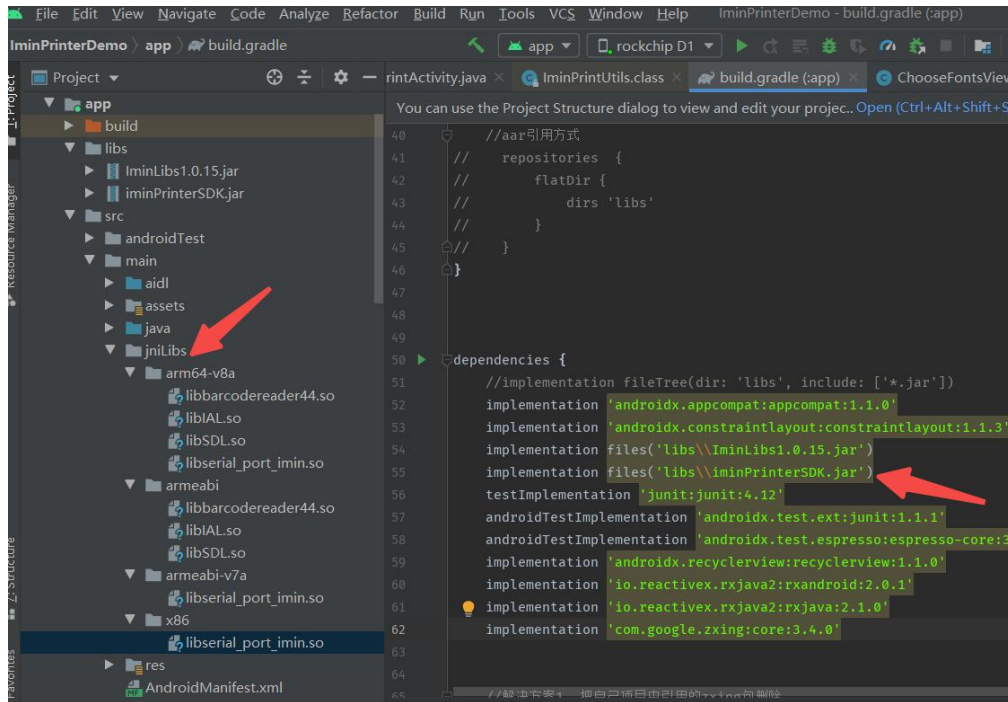
# 1. Printer integration through SDK.jar method

## 1.1 Integration Method

Download the iminPrintSDK.zip package from the developer documentation link: <https://oss-sg.imin.sg/docs/en/Printer.html>. Extract the zip folder and retrieve the iminPrintSDK.jar file, then place the .jar file to the app-libs menu directory, as shown in the screenshot below:



**Additional note: For SPI printing, you need to reference the SO library. The SO library files are located in the jniLibs folder within the demo source code in the ZIP package, as shown in the screenshot below**



## Initialize Printer Utility Class

IminPrintUtils mIminPrintUtils = IminPrintUtils.getInstance(TestPrintActivity.this);

### Utility Description

IminPrintUtils : iMin printer interface management class, including initialization of the printer and execution of various printing methods.

## 2. Interface Definition Description

After obtaining the IminPrintUtils object through the above method, call the following interface to perform your own printing task.

### 2.1 Printer initialization and configuration

Method: initPrinter(int printType)

#### Printer initialization,

- int printType (1-3) ->
- 1.IminPrintUtils.PrintConnectType.USB -> USB
  - 2.IminPrintUtils.PrintConnectType.SPI -> SPI
  - 3.IminPrintUtils.PrintConnectType.Bluetooth -> Bluetooth

## Example:

### 2.1.1. Initialize SPI printer:

(M2 203, M2 202, M2 Pro)

```
IminPrintUtils.getInstance(TestPrintActivity.this).initPrinter(IminPrintUtils.PrintConnectType.SPI);
```

### 2.1.2. Initialize USB printer:

(D4 series, D1w, D1,D1 Pro,Falcon 1, Swift 1, M2 Max,S1 series)

```
IminPrintUtils.getInstance(TestPrintActivity.this).initPrinter(IminPrintUtils.PrintConnectType.USB);
```

### 2.1.3. Initialize Bluetooth printing (Supports all models mentioned above)

```
IminPrintUtils.getInstance(TestPrintActivity.this).initPrinter(IminPrintUtils.PrintConnectType.BLUETOOTH, BluetoothDevice device);
```

**Note:** Once Bluetooth is enabled, you can obtain the corresponding BluetoothPrinter from the list of the scanned nearby Bluetooth devices

## 2.2 Obtain printer latest status

**Note:** If you need to obtain the printer status after initializing the printer, it is recommended to wait for 1 second before obtaining it. Because the printer needs to be initialized when connecting to the printer, if it is obtained directly, it may not respond in time, resulting in obtaining the wrong value.

### 2.2.1 SPI Type Printer Method:

```
mIminPrintUtils.getPrinterStatus(IminPrintUtils.PrintConnectType.SPI, new Callback() {  
    @Override  
    public void callback(int status) {  
        Log.d("TAG", " print SPI status:" +  
status +  
        " PrintUtils.getPrintStatus==" +  
PrintUtils.getPrintStatus());  
    }  
});
```

**iminPrinterSDK-12\_V1.2.0\_2401251422.jar version and later are supported. Olde version are not supported.**

```
int status = mIminPrintUtils.getPrinterStatus(IminPrintUtils.PrintConnectType.SPI );
```

Return value description:

-1 : Printer is not connected or powered

0 : Printer is operating normally

99 : Other errors (Printer door is opened, out of paper, overheat etc.)

### 2.2.2.USB Type printer method:

```
int status = mIminPrintUtils.getPrinterStatus(IminPrintUtils.PrintConnectType.USB );
```

Return value description:

D4 series:

0 Printer is normal, 1 Printer is not connected or powered, 2 Printer is not compatible with the referenced library, 3 Printer door is opened, 4 Cutter position is not restored, 5 Printer head overheat, 6 Black mark error, 7 Out of paper, -1 Printer initialization failed

S1 series:

0 Printer is normal, 1 Printer is disconnected or not powered, -1 Printer initialization failed, 8 Paper is running out, 7 Out of paper/Printer door is opened

D1 /D1 Pro /M2 Max:

0 Printer is normal, -1 Printer initialization failed, 1 Printer is disconnected or not powered, 7 Out of paper/Printer door is opened

## 2.3 Feed paper related

### 2.3.1. Print and feed one line

Function: void printAndLineFeed()

Description: Feed paper by 10 vertical dots 10\*0.125 by default

```
mIminPrintUtils.printAndLineFeed();
```

### 2.3.2. Print and feed n number of 0.125mm

Function: void printAndFeedPaper(int value)

Parameters: Range 0≤value≤255

To print the data in the print buffer and feed the paper by n vertical dot distances. After printing, set the start of the next line to the print start position (where one vertical dot distance is 0.125mm, the same follows)

Example:

```
mIminPrintUtils.printAndFeedPaper(100);
```



## 2.4 Cutter (Cut paper) related (Only support Desktop D4 series, S1 series, Falcon 1)

### 2.4.1 Cut paper

Function: void partialCut ( ) (Half cut)

Parameters:

Note: Using this method, once the command is sent, the printer will cut the paper without waiting for the printing to finish, which may easily lead to issues such as the receipt content being cut off, or the paper being cut at the wrong position.

Example:

```
mIminPrintUtils.partialCut();
```

Supported by `iminPrinterSDK-12_V1.2.0_2401251422` or later .jar, not supported by older version

### 2.4.2 Feed and cut paper (Recommended)

Function: void partialCutPaper ( )

Parameters:

Note: If we use this method, it will wait for the receipt to finish printing, feed one empty line, then only cut the paper to avoid content being cut off.

Example:

```
mIminPrintUtils.partialCutPaper();
```

## 2.5 Print Text Related

### 2.5.1. Configure alignment method

Function: void setAlignment(int alignment)

Parameters: alignment-> Alignment method: 0-> Left , 1-> Centre, 2-> Right

Note: Global method, this will affect subsequent text printing, relevant settings need to be cancelled after implementing this method.

Example:

```
mIminPrintUtils.setAlignment(0);
```

### 2.5.2. Configure text size

Function: void setTextSize(int size)

Parameters: 0<size

Initialize default font size as:28px

Note: This global method affects subsequent text printing. This needs to be terminated after implementing.

Adjusting the font size will affect the character width, and the number of characters per line will also change accordingly.

Example:

```
mIminPrintUtils.setTextSize(28);
```

### 2.5.3 Configure font type

Function: `void setTextTypeface(Typeface typeface)`

Parameters: `typeface`→ Default font type `Typeface.DEFAULT`→ Monospace font type `Typeface.MONOSPACE`,

> Bold font type `Typeface.DEFAULT_BOLD` → sans serif font type `Typeface.SANS_SERIF` → serif font type `Typeface.SERIF`

**Note:** Global method, will affect the subsequent printing. This needs to be terminated after implementing

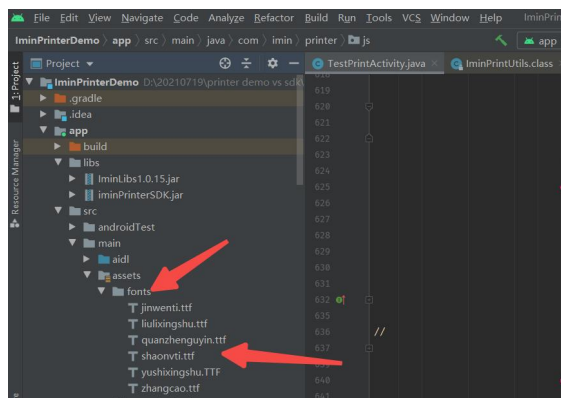
Example:

```
mIminPrintUtils.setTextTypeface(Typeface.DEFAULT);
```

Support customized font type, you need to place the font library under the assets directory.

To obtain the typeface object via `Typeface typeface = Typeface.createFromAsset(getAssets(), "fonts/zhangcao.ttf")`

Configure font type example:



Example:

```
Typeface typeface = Typeface.createFromAsset(getAssets(), "fonts/zhangcao.ttf");  
mIminPrintUtils.setTextTypeface(typeface );  
mIminPrintUtils.setTextTypeface(Typeface.DEFAULT);
```

### 2.5.4 Configure font style

Function: `void setTextStyle(int style)`

Parameters: `style`→Bold `Typeface.BOLD`→ Bold Italic `Typeface.BOLD_ITALIC`→ Italic

Typeface. ITALIC->General Typeface.NORMAL

Default initialization configuration Typeface.NORMAL

**Note:** Global method, will affect the subsequent text printing. This needs to be terminated after implementing

Example:

```
mIminPrintUtils.setTextStyle(Typeface.NORMAL);
```

#### 2.5.5. Configure the printing text line spacing

Function: void setTextLineSpacing(float space)

Parameters:  $0 \leq \text{space} \leq 255$

Default initialization 1.0f

**Note:** Global method, will affect the subsequent text printing. This needs to be terminated after implementing

Example:

```
mIminPrintUtils.setTextLineSpacing(1.0f);
```

#### 2.5.6. Configure the content width

Function: void setTextWidth(int width)

Parameters:  $0 \leq \text{width} \leq 576$

The most suitable width for a 58mm paper is 384, and the optimal width value for an 80mm paper is 576.

Default: 58mm paper is 384, 80mm paper is 576

**Note:** Global method, will affect the subsequent text/bitmap printing, restore the default settings after implementing. It might affect the printing result if different value is being used.

Example:

```
mIminPrintUtils.setTextWidth(576);
```

#### 2.5.7 Print text image

Function: void printText(String text)

Parameters: text (the content to print)

**Note:** If '\n' is added to the end of the text, it indicates immediate printing. If '\n' is not added, it will enter buffer mode and it only be printed when the buffer is full or before the next printing content begins.

Example:

```
mIminPrintUtils.printText("PrinterTestContent \n");
```

#### 2.5.8 Alignment method of adding individual text image

**Function:** void printText(String text, int alignment)

**Parameters:** text (content to print), need to add“\n”at the end of text

**Alignment->** Alignment method: 0->Left , 1->Center, 2-> right, only for a single print text image content

**Note:** If you add “\n” to the end of the text, it means instant printing, if “\n” is not added, it will enter buffer mode, and it will be printed only when the buffer is full or before the next content begins to print

**Example:**

```
mIminPrintUtils.printText("PrinterTestContent \n",0);
```

### 2.5.9 Print text and picture with Reverse Printing effect

**Function :** void printAntiWhiteText(String text)

**Parameter:** text prints, ending with “\n”.

**Note:** If you add “\n” to the end of the text, it means instant printing, if “\n” is not added, it will not execute reverse printing effect, the content will enter buffer mode and proceed with normal printing

**Example:**

```
mIminPrintUtils.printAntiWhiteText("ORDER NOTE:no\n");
```

## 2.6 Table Printing

**Function:** :mIminPrintUtils.printColumnsText(String[] colTextArr, int[] colWidthArr, int[] colAlign, int[] size);

**Parameter:**

colTextArr -> array of text strings for each column

colWidthArr-> The width weight of each column, i.e., the proportion of each column."

colAlign-> Column alignment: 0 on the left, 1 in the center, and 2 on the right

Size—> The font size of the text strings in each column

**Example:**

```
mIminPrintUtils.printColumnsText(new String[]{"1","iMin","iMin"},new int[]{1,2,1},new int[]{1,0,2} ,  
new int[]{26,26,26} );
```

## 2.7 Printing 1D Code

### 2.7.1. Set the width of the 1D code

**Function:** void setBarCodeWidth(int width)

**Parameter:** width -> barcode width, Width level 2<=width<=6, if the width is not set, the default barcode width level is 3

**Example:**

```
mIminPrintUtils.setBarCodeWidth(3);
```

### 2.7.2. Set the height of the 1D code

Function: void setBarCodeHeight(int height)

Parameters: height -> barcode height 1<=height<=255, 1mm for every 8 dots, if the height is not configured, default barcode is set to 100

Example:

```
mIminPrintUtils.setBarCodeHeight(100);
```

### 2.7.3. Set the position of the text of the 1D code

Function: void setBarCodeContentPrintPos(int position)

Parameter: position-> text position (0 - 3): If not set, 0 is default

0 -> Does not print text

1 -> Text is printed above of the barcode

2 -> Text is printed below of the barcode

3 -> Texts are printed on top and below the barcode

Example:

```
mIminPrintUtils.setBarCodeContentPrintPos(2);
```

### 2.7.4. Print 1D Code

Function: void printBarCode(int barCodeType, String barCodeContent) throws UnsupportedOperationException

Parameters: barCodeType -> Barcode type ; barCodeContent-> Barcode content

Barcode type (0-6,73,8)	Supported content length of the barcode	Supported ASCII code range
0 --> UPC-A	Barcode Content Length = 11,12	48 ≤ range ≤ 57
1 --> UPC-E	Barcode Content Length = 11,12	48 ≤ range ≤ 57
2 --> JAN13 / EAN13	Barcode content length = 12,13	48 ≤ range ≤ 57
3 --> JAN8 / EAN8	Barcode Content Length = 7	48 ≤ range ≤ 57
4 --> CODE39	Barcode content length >=1	48 ≤ range ≤ 57, 65 ≤ range ≤ 90, range= 32, 36, 37, 42, 43, 45, 46, 47
5 --> ITF	Barcode content length >=2	48 ≤ range ≤ 57
6 --> CODABAR	Barcode content length >=2	48 ≤ range ≤ 57, 65 ≤ range ≤ 68, 97 ≤ range ≤ 100, range = 36, 43, 45, 46, 47, 58
73 --> CODE128, 8 --> CODE128	Barcode content length >=2	0 ≤ range ≤ 127

Example:

```
mIminPrintUtils.printBarCode(4, "123456");  
mIminPrintUtils.printBarCode(73, "{A1456AAA}");//CODE128 A  
mIminPrintUtils.printBarCode(73, "{B12CAa--}");//CODE128 B  
mIminPrintUtils.printBarCode(73, "{C009999789101}");//CODE128 C
```

### 2.7.5. Print 1D code and set the alignment

Function: void printBarcode(int barcodeType, String barcodeContent, int alignmentMode) throws  
UnsupportedEncodingException

Parameters: barcodeType (as above), barcodeContent (as above), alignmentMode → (0-2)  
0 → Left, 1 → Centre, 2 → Right

Example:

```
mIminPrintUtils.printBarcode(4, "123456",1);  
mIminPrintUtils.printBarcode(73, "{A1456AAA",2);//CODE128 A  
mIminPrintUtils.printBarcode(73, "{B12CAa--", 1);//CODE128 B  
mIminPrintUtils.printBarcode(73, "{C009999789101",1);//CODE128 C
```

Note: There are the different types of barcodes, as followed

Encoding	Description
code39	Print up to 13 digits
UPC-E	Print up to 12 digits
UPC-A	Print up to 12 digits
JAN13 / EAN13	Print up to 13 digits
JAN8 / EAN8	8 digits are required (the last digit is check digit) and the valid length is 8 digits
ITF	Must enter a number, and the valid number is less than 14 digits, and must be an even number
CODABAR	Required number from 0-9 and 6 special characters are required, print up to 18 digits
CODE128	There are the types of Code128: Category A: Contains uppercase letters, numbers, punctuation, etc.; Category B: Uppercase and lowercase letters, numbers; Category C: Pure numbers, complex digits, if it is a single digit, the last one will be ignored; By default is Category B encoding type, If you want to use Class A and Class C encoding, you need to prefix "{A", "{C" in front of the content, for example: "{A2344A", "{C123123", "{A1A{B13B{C12".

## 2.8 QR Code Printing

### 2.8.1. Configure QR code size

Function: void setQrCodeSize(int level)

Parameters: level → QR code size, Unit: Dot, 1 ≤ level ≤ 13, By default is 9

Example:

```
mIminPrintUtils.setQrCodeSize(2);
```

### 2.8.2. Configure QR code error correction

Function: void setQrCodeErrorCorrectionLev(int level)

Parameters: level → 48 ≤ level ≤ 51, By default is 51

Example:

```
mIminPrintUtils.setQrCodeErrorCorrectionLev(48);
```

### 2.8.3. Set the left margin of the barcode and QR code

Function: void setLeftMargin(int marginValue)

Parameters: marginValue → 0 < marginValue < 576, Default value is 0

Example:

```
mIminPrintUtils.setLeftMargin(0);
```

**Note: Global method, valid for all subsequent content after configured**

### 2.8.4. Print QR Code

Function: void printQrCode(String qrStr)

Parameters: qrStr → QR code content

Example:

```
mIminPrintUtils.printQrCode("123456");
```

### 2.8.5. Print QR Code and add alignment

Function: void printQrCode(String qrStr, int alignmentMode)

Parameters: qrStr → QR Code content; alignmentMode → (0-2)

0 → Left , 1 → Centre , 2 → Right

Example:

```
mIminPrintUtils.printQrCode("123456", 0);
```

## 2.9 Configure Paper Format

### 2.9.1. Configure paper format

Function: void setPageFormat(int style)

Parameters: style → (0-1)

0 → 80mm, (D4 series, S1, Falcon 1 (Set 80mm) ,D1w)

1 → 58mm (M2 series, Swift 1, D1, D1 Pro)

Example:

```
mIminPrintUtils.setPageFormat(type);
```

**Note: Global method, you just need to initialize the printer settings once**

## 3.0 Print Image

### 3.0.1. Print image

Function: void printSingleBitmap(Bitmap bitmap)

Parameters: bitmap → bitmap object

Example:

```
mIminPrintUtils.printSingleBitmap(bitmap);
```

Note: This method adds the function of compressing the quality of large images, and if the bitmap needs to be reused, it is recommended to regenerate the bitmap object

### 3.0.2. Print image and configure alignment

Function: void printSingleBitmap(Bitmap bitmap, int alignmentMode)

Parameters: bitmap-> image object; alignmentMode-> (0-2)

0 -> Left , 1 -> Centre , 2 -> Right

Example:

```
mIminPrintUtils.printSingleBitmap(bitmap,1);
```

Note: This method adds the function of compressing the quality of large images, and if the bitmap needs to be reused, it is recommended to regenerate the bitmap object

### 3.0.3. Print multiple bitmap

Function: void printMultiBitmap(List<Bitmap> bitmaps)

Parameters: bitmaps -> bitmap list

Example:

```
mIminPrintUtils.printMultiBitmap(bitmaps);
```

### 3.0.4. Print multiple bitmap and configure alignment method

Function: void printMultiBitmap(List<Bitmap> bitmaps, int alignmentMode)

Parameters: bitmaps -> bitmap list; alignmentMode-> (0-2)

0 -> Left , 1 -> Centre , 2 -> Right

Example:

```
mIminPrintUtils.printMultiBitmap(bitmaps, 1);
```

### 3.0.5. Bitmap printing with image processing function

Function: void printSingleBitmapBlackWhite(Bitmap bitmap)

Parameters: bitmap-> Bitmap object

Example:

```
mIminPrintUtils.printSingleBitmapBlackWhite(bitmap);
```

Note: This method adds the function of compressing the quality of large images, and if the bitmap needs to be reused, it is recommended to regenerate the bitmap object

### 3.0.6. Print 1D Code with bitmap printing

Function: void printBarcodeToBitmapFormat(String barCodeContent,int width,int height,int codeFormat)

Parameters: barCodeContent-> ID Code content

width -> 1D code width, It is recommended to use 1300/1200 for 80mm paper, and do not exceed 1300

height-> 1D code height, Default is 120, Barcode height 1<=height<=255, 1mm for every 8 points

codeFormat-> Barcode type



CodeFormat.AZTEC -> BarcodeFormat.AZTEC  
CodeFormat.CODABAR -> BarcodeFormat.CODABAR  
CodeFormat.CODE\_39 -> BarcodeFormat.CODE\_39  
CodeFormat.CODE\_93 -> BarcodeFormat.CODE\_93  
CodeFormat.CODE\_128 -> BarcodeFormat.CODE\_128  
CodeFormat.DATA\_MATRIX -> BarcodeFormat.DATA\_MATRIX  
CodeFormat.EAN\_13 -> BarcodeFormat.EAN\_13  
CodeFormat.ITF -> BarcodeFormat.ITF  
CodeFormat.MAXICODE -> BarcodeFormat.MAXICODE  
CodeFormat.PDF\_417 -> BarcodeFormat.PDF\_417  
CodeFormat.QR\_CODE -> BarcodeFormat.QR\_CODE  
CodeFormat.RSS\_14 -> BarcodeFormat.RSS\_14  
CodeFormat.RSS\_EXPANDED -> BarcodeFormat.RSS\_EXPANDED  
CodeFormat.UPC\_A -> BarcodeFormat.UPC\_A  
CodeFormat.UPC\_E -> BarcodeFormat.UPC\_E  
CodeFormat.UPC\_EAN\_EXTENSION -> BarcodeFormat.UPC\_EAN\_EXTENSION

Example:

```
mIminPrintUtils.printBarCodeToBitmapFormat("11110AQ899015859344",1300,120,  
CodeFormat.CODE_128);
```

## 3.1 Print Dual QR code related

**Note:** Models that currently support dual QR codes (M2-203, M2 Pro,, M2 Max, D1).

### 3.1.1. Configure the size of dual QR code

Function: void setDoubleQRSize(int size)

Parameters: size -> 1<= size <= 8

Example:

```
mIminPrintUtils.setDoubleQRSize(5);
```

### 3.1.2. Configure dual QR code offset value

Function: First QR code: void setDoubleQR2Level(int level),

Second QR code: void setDoubleQR2Level(int level)

Parameters: level -> 0-3, Default value is 2

Example:

```
mIminPrintUtils.setDoubleQR1Level(1);  
mIminPrintUtils.setDoubleQR2Level(2);
```

### 3.1.3. Set the left margin of the first QR code

Function: void setDoubleQR2MarginLeft(int marginValue)

Parameters: marginValue -> 0<= marginValue <= 200

Example:

```
mIminPrintUtils.setDoubleQR1MarginLeft(-80);
```

### 3.1.4. Configure the left margin of the second QR code

Function: void setDoubleQR2MarginLeft(int marginValue)

Parameters: marginValue -> 0<= marginValue <= 200

Example:

```
mIminPrintUtils.setDoubleQR2MarginLeft(-80);
```

### 3.1.5. Configure the version of the first QR code

Function: void setDoubleQR1Version(int version)

Parameters: version (0-40) -> 0<= version <=40 Default value is 0

Example:

```
mIminPrintUtils.setDoubleQR1Version(0);
```

### 3.1.6. Configure the version of the second QR code

Function: void setDoubleQR2Version(int version)

Parameters: version (0-40) -> 0<= version <=40 Default value is 6

Example:

```
mIminPrintUtils.setDoubleQR2Version(6);
```

### 3.1.7. Print QR code

Function: void printDoubleQR(String qrCode1, String qrCode2)

Parameters: colTextArr-> Dual QR code content

Example:

```
mIminPrintUtils.printDoubleQR("www.iMin.sg", "www.google.com");
```

## 3.2 Features supported only by TF1/Falcon 1

### 3.2.1. Obtain the number of cutting

Function: int getPrintCutterNumber()

Parameters:

Example:

```
mIminPrintUtils.getPrintCutterNumber();
```

### 3.2.2. Obtain printing length of the printer

Function: `int getPrinterPaperDistance()`

Parameters:

Example:

```
mIminPrintUtils.getPrinterPaperDistance();
```

### 3.2.3. Obtain Serial Number

Function: `int getPrinterSerialNumber()`

Parameters:

Example:

```
mIminPrintUtils.getPrinterSerialNumber();
```

## 3.3 Switch between internal and external printer

Function: `void setInitIminPrinter(boolean initIminPrinter)`

Parameters: `initIminPrinter->>true`, by default is `imin` built-in printer;

`False`, gets the default connection from the USB device list first

Example:

```
mIminPrintUtils.setInitIminPrinter(true);
```

**iminPrinterSDK-12\_V1.2.0\_2401251422.jar versions and later are supported, older versions are not supported**

## 3.4 Reset Data

**Note:** This method is called before the first initialization of the printer, or when the program is ended, and it does not need to be called during printing, if this method is called during printing, it may cause printing interruption, incomplete printing content or garbled characters

Function: `void resetDevice()`

Parameters:

Example:

```
mIminPrintUtils.resetDevice();
```

## 3.5 Disconnect SDK Connection

**Note:** This method is called when the program is closed, and does not need to be called during the printing process, if the method is called during the printing process, the entire printing will be interrupted and the connection will be closed

Note: When disconnecting the sdk connection, it is recommended to set the currently obtained mIminPrintUtils to empty, that is, mIminPrintUtils=null; to facilitate initialization the next time the sdk is used.

Parameters: void disconnectDevices()

Parameters:

Example:

```
mIminPrintUtils.disconnectDevices();
```

### 3.6 On/Off button to activate print

Note: This method is called when the program is closed, and does not need to be called during the printing process, if the method is called during the printing process, the entire printing will be interrupted and the connection will be closed

Function: void setIsOpenLog(int open) //开启日志打印 1 开启 0 Disabled , SDK is compiled and customer tested 0

参数:

示例:

```
mIminPrintUtils.setIsOpenLog(0);
```

### 3.7 NOTICE

The SDK of the iMinPrinterSDK-12\_V1.2.0\_2401251422.jar version and later versions are only fully adapted with ROMs version after iMin December 21, 2023. If the ROM version of the customer is not up to date, please apply for an upgrade via OTA.