

# Android-SDK Development Document

## Introduction

The SDK contains Bluetooth, USB and WiFi.

1. Software package name: BpPrinter.mylibrary
2. Classes name:

Class Name	Discription
BluetoothConnectivity	A class to perform various operations using Bluetooth.
WiFi	A class to perform various operations using WiFi.
USB	A class to perform various operations using USB.
Printer	A class to perform various text operations
QR Code	A class to generate QR Code.

Class “BluetoothConnectivity” provides the following method:

1. *Common Methods:*

- **startDiscover**
  - a) **Description:** Starts a live scan to find both paired and new nearby Bluetooth printers.  
**Syntax:** BpScrybeDevice.startDiscover(this);
- **onDiscoveryComplete (Callback)**
  - a) **Description:** This method is automatically called when the scan finds devices.  
It provides a list (ArrayList<String>) of all printers found during the search.
- **pairPrinter**
  - a) **Description:** Returns a list of Bluetooth printers that are already saved (paired) in your phone's system settings.  
**Syntax:** public Object pairPrinter(String printerName)

- **connectToPrinter**
  - a) **Description:** Connects the app to the specific printer you selected from the list.
  - Syntax:** BpScribeDevice.connectToPrinter(printerName);
- **disconnectPrinter**
  - a) **Description:** Breaks the connection between the phone and the printer.
  - Syntax:** BpScribeDevice.disconnectPrinter();

### Class “WiFi” provides the following method:

- 1) Common Method:
  - a) **BPTcpClient**

```
public BPTcpClient(OnMessageReceived listener, String host, int port)
```

Connect Printer using IP Address.  
Ip: String of IP address is to be provided for the connection.
  - b) **connect()**

```
public void connect()
```

Connect Printer using the provided IP Address.  
Ip: String of IP address is to be provided for the connection.  
Syntax: BPprinter.connect();
  - c) **disconnect()**

```
public synchronized void disconnect()
```

Disconnect The printer connected with the IP Address.  
Syntax: BPprinter.disconnect();

### Class “Printer” provides the following method:

- 1) Common Method:
  - a) **POS\_Set\_Char\_Mode**

```
public void POS_Set_Char_Mode (byte mode) throws IOException
```

Selects or cancels different printer modes, for different modes following bytes is to be used.

Normal Font = 0x00  
Tahoma Font = 0x01  
Calibri Font = 0x02  
Verdana Font = 0x03  
Double Height = 0x10  
Double Width = 0x20  
Underline = 0x80  
Bold = 0x08

Syntax:

    - For selecting Text Font (Normal/Tahoma/Calibri/Verdana):  
BpPrinter.POS\_Set\_Char\_Mode((byte) 0x00);

- And for selecting Double Width/Double Height/Underline/Bold:  
BpPrinter.POS\_Set\_Char\_Mode((byte) 0x10);

These two particular sets of commands can be implemented on a single String of data.

#### **b) POS\_Set\_Text\_alingment**

public void POS\_Set\_Text\_alingment (byte mode) throws IOException  
Aligns all the data in one line to a specified position, using bytes as follows  
Left Justification = 0x00  
Center Justification = 0x01  
Right Justification =  
0x02Syntax:  
BpPrinter.POS\_Set\_Text\_  
alignment((byte) 0x00);

Example Layout:

Left Alignment      Center Alignment      Right Alignment

#### **c) POS\_set\_text\_Underline**

public void POS\_set\_text\_Unerline(byte mode) throws IOException  
For turning on the Underline mode.  
The value of bytes as follows:  
For turning on Underline mode = 0x01  
For turning off Underline mode = 0x00

Syntax:  
BpPrinter.POS\_set\_text\_Underline((byte) 0x00);  
Example Layout:

Underline

#### **d) POS\_text\_Character\_Spacing**

public void POS\_text\_Character\_Spacing (byte mode) throws IOException  
For changing the size of the character.  
The value of bytes as follows:  
For turning Character Spacing On = 0x10  
For turning Character Spacing Off = 0x00

Syntax:  
BpPrinter.POS\_text\_Character\_Spacing((byte) 0x00);

Example Layout:

Character Spacing  
Character Spacing

#### **e) POS\_text\_Reverse\_Printing**

`public void POS_text_Reverse_Printing (byte mode) throws IOException`

The white/ black reverse printing mode is effective for all characters (except for HRI characters).

The value of bytes as follows:

For turning reverse printing mode On = 0x01

For turning reverse printing mode Off = 0x00

Syntax:

`BpPrinter.POS_text_Reverse_Printing((byte) 0x00);`

Example Layout:

**Reverse Printing**

#### **f) Initialize\_Printer**

`public void Initialize_Printer() throws IOException`

This command is used to initialize printer.

The data in the printer buffer is cleared and the printer mode(s) is reset to the mode that was in effect when the power was turned on.

Syntax:

`BpPrinter.Initialize_Printer();`

#### **g) print\_column\_formated\_text**

`public String print_column_formated_text(String inputData, int Size) throws IOException{`

this method is used to arrange the data in proper alignment and in a column format in which data and sizes are present for 2 or 3 inch printing

in which

[L] is for left alinment.

[R] is for right alignment

[C] is for center alignment

And also

0 for 2inch printing and

1 is for 3inch printing

For example :

String data = "[L] left Alignment [C]Center alignment [R] Right alignment

`BpPrinter.print_columnnformated(data,0);`

#### **h) setlinefeed**

`public void setLineFeed(int noOfFeeds) throws IOException`

Prints the data in the print buffer and feeds one line. The amount of paper fed per line is based on the value set using the line spacing command. After printing, the printing position moves the beginning of the line.

noOfFeeds:

it is basically the number of feed lines that is to be generated as an integer which is 1,2,3,4,5....

Syntax:

`BpPrinter.setlinefeed(1);`

Example Layout:

Line Feed  
Line Feed  
Line Feed

Line Feed

Line Feed

Line Feed

### i) **setCarriageReturn**

`public void setCarriageReturn() throws IOException`

Print and carriage return. Print all data in printing buffer area and paper feed one line forward with the line space set.

Syntax:

`BpPrinter.setCarriageReturn();`

### j) **print**

`public void print(String text) throws IOException`

Used to print a String of Data.

Syntax:

`BpPrinter.print(text);`

Example Layout:

BluPrints

### k) **printBarcode**

`public void printBarcode(String barcodeData, BARCODE_TYPE barcodetype, BARCODE_HEIGHT barcodeheight, CHAR_POSITION HRIchar ) throws IOException`

Print bar code.

For this user have to provide the following details:

barcodeData: The string of data to be provided for which the barcode is to be generated.

barcodetype: The barcode type is to be selected as per the requirement.

The user has to choose in between

- UPCA (for 11 characters),
- UPCE (for 11 characters),
- EAN13 (for 12 characters),
- EAN8 (for 7 characters),
- CODE39 (for 1-255 characters)

and

barcodeheight: The is used to select the height of the barcode

HT\_SMALL - For Small Height

HT\_MEDIUM - For Medium Height

HT\_LARGE - For Large Height

HRIchar: Select printing position of HRI characters

POS\_NONE- For not Printing of the HRI Characters.

POS\_ABOVE- Printing of the HRI Characters above the barcode.

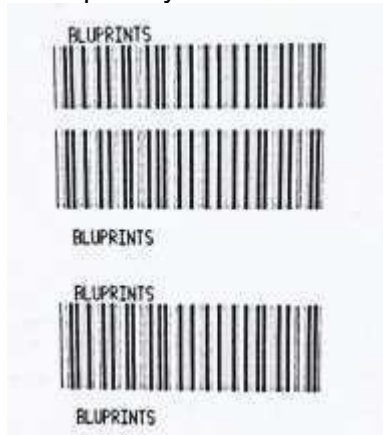
POS\_BELOW- Printing of the HRI Characters below the barcode.

POS\_BOTH - Printing of the HRI Characters both above and below of the barcode.

Syntax:

```
m_AemPrinter.printBarcode(text, BpPrinter.BARCODE_TYPE.CODE39,  
BpPrinter.BARCODE_HEIGHT.HT_MEDIUM, BpPrinter.CHAR_POSITION.POS_BOTH);
```

Example Layout:



## 1) printTextAsImage

```
public void printTextAsImage(String TextToConvert, float textSize, int  
Alignment, int PaperSize) throws IOException {
```

Used to print Text as Image.

TextToConvert : It is the String of Data that is to be converted as an Bitmap Image.

textSize: It is used to select the textSize of the String of data provided to it.

Alignment: It is used for different alignment of the text for which

- 0 is for Left Alignment
- 1 is for Center Alignment
- 2 is for Right Alignment

PaperSize: It is used to select the width size of the Printing Paper.

- 0 is for 2-inch paper size
- 1 is for 3-inch paper size

Syntax:

```
m_AemPrinter.printTextAsImage(text,40,0,0);
```

Example Layout:

ब्लूप्रिंट प्रिंटर

ब्लूप्रिंट प्रिंटर

ब्लूप्रिंट प्रिंटर

### **m) printImage**

```
public void printImage(Bitmap resizedBitmap, int Size, Boolean FitToPaper)
```

Used to print a Bit Image by selecting the width Size of the paper.

resizedBitmap:

It is the Bitmap image that is to be printed.

Size: It is the Width Size of the Paper which can be

- 0 is for 2-inch paper size
- 1 is for 3-inch paper size

FitToPaper: Set this to true to scale the image to the full width of the paper. Set it to false to maintain the image's original size.

Syntax:

```
BpPrinter.printImage(Bitmap,0);
```

Example Layout:



### **n) sendByte**

```
public void sendByte(byte bt) throws IOException
```

Send byte data.

### **o) sendByteArrayBT**

```
public void sendByteArrayBT(byte[] byteArr) throws IOException
```

Send Byte Array using Bluetooth.

Syntax:

```
BpPrinter.sendByteArrayBT((byte) 0x10);
```

### **p) AutoCut**

```
public void AutoCut() throws IOException
```

This command is used to select cut mode and cut paper.

Preferably used in Auto Cutter Printer.

This is for Utkarsh printer only.

Syntax:

BpPrinter.AutoCut();

#### **q) printByte**

`public void printBytes(byte[] printBytes) throws IOException`  
Print Data Bytes.

Syntax:

BpPrinter.printByte((byte) 0x10);

#### **r) Pos\_Set\_Char\_Font**

`public void POS_Set_Char_Font (byte mode)`

Select Different Fonts for the printer.

Normal Font = 0x00

Tahoma Font = 0x01

Calibri Font = 0x02

Verdana Font = 0x03

Syntax:

BpPrinter.POS\_Set\_Char\_Font((byte) 0x00);

POS\_set\_text\_Emphasized

`public void POS_set_text_Emphasized(byte mode) throws IOException`

Turns On/OFF Emphasized text.

For turning OFF 0x00 is used

For turning ON 0x01 is used

Syntax:

BpPrinter.POS\_set\_text\_Emphasized((byte) 0x00);

Example Layout:

AAAA  
BBBB

#### **s) FeedLine()**

`public void FeedLine() throws IOException`

Prints the Feed Lines.

Syntax:

BpPrinter.FeedLine();



#### t) **printLabel()**

```
public void printLabel(Bitmap Labelimage, int LabelWidth, int LabelHeight,int xCoordinate, int yCoodinate,String unit, int scaling, int size)
```

in which

- Bitamp Labelimage – Image that is to be printed
- LabelWidth – Width of the Label
  - “58” for 2 inch
  - “80” for 3 inch
  - “105” for 4 inch
- LabelHeight – Height of the Label
- xCoordinate – X Coordinate as per your requirement
- yCoordinate – Y Coordinate as per your requirement
- unit – Unit in which the label value is provided
  - “mm”
  - “inch”
- Scaling – Values provided in such way that
  - If “0” is provided then there’s no change in the width and height of the image
  - If “1” is provided then there’s slight change in height and width
  - If “2” is provided then there’s evident change in the height and width
- Size – values is provided as per paper size
  - “0” for 2inch
  - “1” for 3inch
  - “2” for 4inch

Syntax:

```
BpPrinter.printLabel( bitmap );
```

#### u) **get\_paper\_status()**

```
public String get_paper_status() throws InterruptedException, IOException
```

It is used to check the \*\*paper status of the printer\*\* (whether paper is available or not).

Responses:

- PaperOK` → Paper is present
- NoPaper` → Paper is not present

Syntax:

```
String data = BpPrinter.get_paper_used_status();
```

#### v) **Battery\_Status()**

```
public String Battery_Status() throws InterruptedException, IOException
```

this is used to get the battery status in percentage

Syntax:

```
String data = BpPrinter.get_paper_used_status();
```

**w) get\_paper\_used\_status()**

`public String get_paper_used_status() throws InterruptedException, IOException`  
this is used to get the length of the paper used

Syntax:

String data = BpPrinter.get\_paper\_used\_status();

**x) reset\_paper\_used\_status ()**

`public String reset_paper_used_status() throws InterruptedException, IOException`  
this is used to reset the length of the paper used.

Syntax:

String data = BpPrinter.reset\_paper\_used\_status();

**Class “QR Code” provides the following method:**

1) Common Method:

**a) QRLEncoder**

`public QRLEncoder(String data, Bundle, String type, int dimension, String header, String footer)`

Print QR Code

For this user have to provide the following details:

data: String of Data is provided

dimension if the integer is provided

header: String of header data footer:

String of footer data

Example Layout:



**Class “USB” provides the following method:**

1) Common Method:

a) connectToPrinter

`public boolean connectToPrinter( int vid, int pid)`

For the connection with the printer VID and PID is to be provided and then the connection is established.

or

`public boolean connectToPrinter()`

for the connection using this method it uses `getDev(VID, PID)`; which gets the VID and PID values of the connected device and establish the connection accordingly.

b)     disconnectPrinter()

`public boolean disconnectPrinter() throws IOException`  
Disconnects the printer connected using USB.

c)     getUsbPrinter()

`public BpPrinter getUsbPrinter()`  
it is used to get the USB printer connected to it.