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 Method:

a) pairPrinter

```
public Object pairPrinter(String printerName)
```

Shows the list of the remote printer devices paired with the Bluetooth.

b) connectToPrinter

```
public boolean connectToPrinter(String printerName) throws IOException
```

Establishes the printer which is selected by user

Syntax:

```
BpScrybeDevice.connectToPrinter(printerName);
```

c) disConnectPrinter

public boolean disConnectPrinter() throws IOException.
Disconnects the device with the printer.

Syntax:
BpScrybeDevice.disConnectPrinter();

Class “WiFi” provides the following method:

1) Common Method:

a) validIP

public static boolean validIP(String ip)
Connect Printer using IP Address.
Ip: String of IP address is to be provided for the connection.

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);

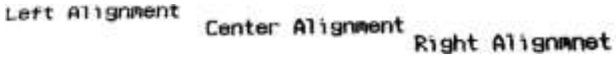
These two particular sets of commands can be implemented on a single String of data.

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 = 0X02

Syntax:
BpPrinter.POS_Set_Text_alignment((byte) 0x00);

Example Layout:



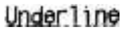
Left Alignment
Center Alignment
Right Alignmnet

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

Synatx:
BpPrinter.POS_set_text_Underline((byte) 0x00);

Example Layout:



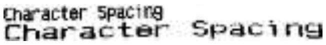
Underline

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:



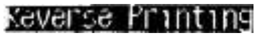
Character Spacing
Character Spacing

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) 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.

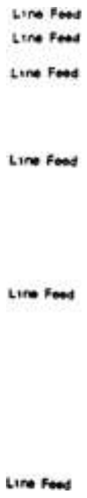
noOfFeeds:

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

h) 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();

i) print

public void print(String text) throws IOException
Used to print a String of Data.

Syntax:

BpPrinter.print(text);

Example Layout:

BluPrints

j) 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

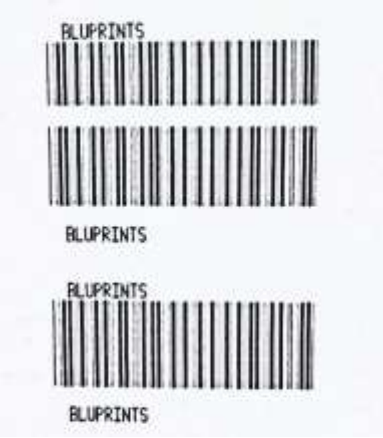
HT_SMALL - For Small Height
HT_MEDIUM - For Medium Height
HT_LARGE - For Large Height

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:



k) 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 selct 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

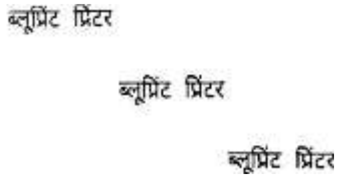
PaperSize: It is used to select the width size of the Printing Paper.

- 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

l) printImage

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

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

Syntax:

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

Example Layout:

m) sendByte

public void sendByte(byte bt) throws IOException
Send byte data.

n) sendByteArrayBT

public void sendByteArrayBT(byte[] byteArr) throws IOException
Send Byte Array using Bluetooth.

Syntax:

BpPrinter.sendByteArrayBT((byte) 0x10);

o) 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();

p) printByte

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

Syntax:

BpPrinter.printByte((byte) 0x10);

q) 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

Synatx:

BpPrinter.POS_Set_Char_Font((byte) 0x00)

r) POS_set_text_Emphasized

public void POS_set_text_Emphasized(byte mode) throws IOException

Turns On/OFF Emphasized text.

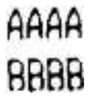
For turning OFF 0x00 is used

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();

Class “QR Code” provides the following method:

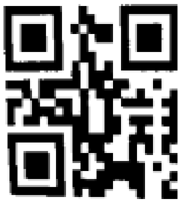
1) Common Method:

a) QRGEncoder

```
public QRGEncoder(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.

- **Integration NEXT RD SERVICE**

Function Capture Button

```
protected void requestDiscovery()  {
//         cleartext ();
try {
Intent infoIntent = new Intent (INFO_INTENT); infoIntent.setPackage(NextSelectedpkg);
startActivityForResult(infoIntent,  DISCOVERY_REQUEST_CODE);
} catch  (Exception  ee)  {  textStatus.setTextColor(Color.parseColor("#FF0000"));
textStatus.setText("RDService  Not  Installed.");  download_Link();
}
}
protected void requestCapture(boolean  auth)  {  cleartext(); try {
Intent capIntent  =  new  Intent(CAPTURE_INTENT);  pidOptXML  =  createPidOptXML();
capIntent.putExtra("PID_OPTIONS",  pidOptXML);  capIntent.setPackage(NextSelectedpkg); if
(auth) {
startActivityForResult(capIntent,  AUTH_REQUEST_CODE);
} else {
startActivityForResult(capIntent,  CAPTURE_REQUEST_CODE);
}

} catch (Exception ee) {download_Link();
}
}

private void  download_Link()  { android.app.AlertDialog.Builder  alertDialog  =  new
android.app.AlertDialog.Builder(this);  alertDialog.setTitle("Please  install  NEXT Biometrics  L0
Registered Device  Service");  alertDialog.setMessage("Please  install  NEXT Biometrics  L0
Registered  Device  Service  from  Play
Store ");
alertDialog.setIcon(R.drawable.app_icon); alertDialog.setPositiveButton("OK",  new
DialogInterface.OnClickListener() {
public void onClick(DialogInterface  dialog,  int  which)  {
/*  Intent  intent  =  new  Intent(Intent.ACTION_VIEW);  intent.setData(Uri.parse(
"https://play.google.com/store/apps/details?id=com.nextbiometrics.rdservice"));
startActivity(intent);*/
}
});
alertDialog.show();
}
```

Function Device Info button

```
protected void requestInfo()  {  cleartext();try {
Intent infoIntent  =  new  Intent(INFO_INTENT);  infoIntent.setPackage(NextSelectedpkg);
startActivityForResult(infoIntent,  INFO_REQUEST_CODE);
} catch (Exception ee) {download_Link();
}
}

private void  download_Link()  { android.app.AlertDialog.Builder  alertDialog  =  new
android.app.AlertDialog.Builder(this);  alertDialog.setTitle("Please  install  NEXT Biometrics  L0
Registered Device Service");  alertDialog.setMessage("Please  install  NEXT Biometrics  L0
Registered  Device  Service  from  Play
Store ");
alertDialog.setIcon(R.drawable.app_icon); alertDialog.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog, int which) {
/*Intent intent = new Intent(Intent.ACTION_VIEW); intent.setData(Uri.parse(
"https://play.google.com/store/apps/details?id=com.nextbiometrics.rdservice"));
startActivity(intent);*/
}
});
alertDialog.show();
}
```

Funtion Authentication Button

```
  btnAuth.setOnClickListener(new Button.OnClickListener() {@Override
public void onClick(View v) { requestDiscovery();
if (! envSel.getSelectedItem().toString().toUpperCase().equals("STAGING")) {
showMessageDialogue("This tool supports authentication in STAGING environment only",
"Select Environment->STAGING"); return;
}
String aadh_No = txtAadharNo.getText().toString().trim();
if (aadh_No != null && !aadh_No.isEmpty() && aadh_No.length() == 12) {
requestCapture(true);
} else {
showMessageDialogue("Please Enter Valid Aadhar No", "Error");
}
} });
```

If you want to know any further details please view Aadhaar_Registered_Devices_2_0_4.