BluPrints Thermal Receipt Printer
Developer Guide (3 Inch-80/mm)

Aadharshila Mobility Solutions Pvt. Ltd.

Web: http://www.bluprints.in/

VERSION CONTROL: 6.2 / March 2021

PLEASE NOTE THAT THIS SDK VERSION, NAMELY, BluPrints_SDK6.1 IS APPLICABLE FOR BluPrints PRINTER FIRMWARE VERSIONS 8.0 AND ABOVE.
DOCUMENT NAME: ANDROID DEVELOPER GUIDE RELEASE DATE: 23-March-2021
SDK SUPPORTED: BluPrints _ SDK6.1
BluPrints FIRMWARE SUPPORTED: 8.0 AND ABOVE

TABLE OF CONTENTS

BluPrints SDK……02
BluPrintsPrinter Class……02
BluPrintsScrybeDevice……..04
QR Code Generation……...8

SDK (Software Development Kit) for ANDROID

This document describes the use of BluPrints SCRYBE Thermal Printer SDK for Bluetooth, USB and Wi-Fi Operations. This SDK provide an Interface between an Android Application and the BluPrints Thermal Printer. The SDK is android based, and requires at least 10 level of android SDK and 1.6 java compiler. It comprises of the following Interfaces, Classes and functions.
Com. BluPrints.api consists of Classes – BluPrintsPrinter, BluPrintsScrybeDevice, BluPrintsCardScanner and
BluPrintsWifiPrinter.

BluPrintsPrinter: BluPrintsPrinter class handles all the Bluetooth related functions. These include:

SetLineFeed: public void setLineFeed (int noOfFeeds) throws IOException
In order to print specific number of blank lines or line feeds you need to call this function. You need to call this function and pass desired number.
LINE_FEED=0X0A

Print: public void printThreeInch (String text) throws IOException
This function is used to print a specified text (in the form of a string) to the printer.

PrintFormatedText: public void POS_S_TextOutThreeInch (String pszString, String encoding, 0,int nWidthTimes,int nHeightTimes,int nFontTypes,int nFontStyle) throws IOException

In order to set the font size to Double width or double height of a particular line in the printer, you need to call this function and pass the desired parameter of Double Width or Double Height.
The desired parameter DOUBLE_WIDTH or DOUBLE_HEIGHT and UnderLine as defined below.
DOUBLE_WIDTH =nScaleTimesWidth=1 DOUBLE_HEIGHT= nScaleTimesHeight=1
UNDERLINE=nFontStyle=0X100 encoding="US-ASCII"
pszString=String to be Printed

SendByteArray: public void sendByteArrayBT (byte [] byteArr) throws IOException
In order to send any of specific byte array (sequence) to the printer via Bluetooth, you need to call this function and pass the desired Byte Array.

printImage: public void printImageThreeInch(Bitmap originalBitmap) throws IOException
This function is used to print an Image in the form of a Raster Image, based on the standard ESC/POS Raster Image command set of GS v protocol. You need to pass the bitmap of the desired image to be printed. Please note that this function can be used to print QR Codes as well, by first generating the QR code from a given string and thereafter, sending its bitmap to the PrintImage Function. The source code for generating the QR Code has been provided at the end of this document.

printTextAsImage: public void printTextAsImageThreeInch(String TextToConvert)throws IOException This function is used to print any text (multilingual, Unicode type characters, etc) in the form of a Raster Image. You need to pass the String that you need to be printed as an image. The main utility of this function is that the user can easily input any multilingual string, that can be printed as is on the printer, so as to enable printing in any language irrespective of the fonts. (Printing characters of Urdu, Gujrati, Arabic, Oriya, Tamil, Tamil, Telugu, etc.). The benefit of
this function is that the print is so clear and fast that there is no perceptible difference to an end user in understanding whether this text has been printed in the form of text or in the form of an image.

printBitImage: public void printBitImage(Bitmap originalBitmap, Context context, byte image_alignment)throws IOException IMAGE_LEFT_ALIGNMENT = 0x6C; IMAGE_CENTER_ALIGNMENT = 0x63; IMAGE_RIGHT_ALIGNMENT = 0x72;
This is a deprecated function that has only been kept for backward compatibility. PrintBitImage fiunction uses the standard ESC * algorithm for printing of a bitmap The maximum image size should be in the following range: - 355 X 500 (WxH) pixels.
For image printing, you need to half the pixel size of height of an image to get the desired width and height.
E.g.: - Suppose you need to print a logo of dimensions 355 X 300 (WXH) pixels, then half the size of the height of an image i.e. 150 pixel for one time in your code by scaling function (as explained below). Width will remain same.
Bitmapscaled_bitmap = bitmap.createScaledBitmap(bitmap, 355, 150, false); 355 pixel= Width (Original Width)
150 pixel= Height (Height will get double i.e. 300 pixels as original height of an image while printing).
Likewise, you can print the logo of desired dimensions. Note: -
Maximum Width size is 355 pixels Maximum Height size is 500 pixels

BLUPRINTSScrybeDevice:
This class is used to instantiate a BLUETOOTH SCRYBE Device, containing the Context, the Bluetooth Adapter, Bluetooth Device and Bluetooth Socket. An object of BLUPRINTSScrybeDevice once instantiated, is capable of returning the BLUPRINTSPrinter object that has been described previously. This class has the following functions:

BLUPRINTSScrybeDevice
Constructor: For Creating the Object: public BLUPRINTSScrybeDevice (IBluPrintsScrybeimpl)
startDiscover: public void startDiscover(Context iContext)

By calling this method, a list of local Bluetooth devices will be returned.
pairDevice: Public String pairDevice(String printerName)

Before pairing any printer by name, first call the method startDiscover(Context iContext) and get the result in the method public void onDiscoveryComplete(ArrayList<String>BluPrintsPrinterList) of the Interface IBluPrintsScrybe.
Example:

```
BLUPRINTSScrybeDevice m_BluPrintsScrybeDevice = new BLUPRINTSScrybeDevice
(newIBluPrintsScrybe()
{
@Override
Public void onDiscoveryComplete(ArrayList<String>BluPrintsPrinterList)
{
// TODO Auto-generated method stub
}
});
```

Now call the method public String pairDevice(String printerName) which returns a String which may be: NOT_SCANNED when you call this method before scanning. DEVICE_NOT_FOUND when the printer is not found. PAIRED when the device is successfully paired FAILED_TO_PAIREDwhen the device is failed to paired
connectToPrinter: public Boolean connectToPrinter(String printerName) throws IOException

By calling this method, printer gets connected.
disConnectPrinter: public Boolean disConnectPrinter() throws IOException

By calling this method, the connected printer gets disconnected.
Confidential Document

Function for Non –Text Printing

Print Barcode:

public void printBarCode(String Barcode, BARCODE_TYPE Btype, BARCODE_HIEGHT bHieght) throws IOException

BARCODE_TYPE_CODE39 = 0X45

This function is used to print a 2-Dimensional Barcode on the Printer, generated automatically according to the string passed to this function. The string should be of Capital English Letters or Numerals. The maximum characters in the strings can by up to

11. Under Barcode Type, the Printer supports BARCODE_TYPE_CODE39. The Barcode Height supported is DOUBLEDENSITY_FULLHEIGHT.

The following packet is generated internally within this function and sent to the printer:

barcodePacket[0] = 0x1D; 'GS' barcodePacket[1] = 0x6B; 'k'
barcodePacket[2] = BARCODE_TYPE_CODE39; 0x45 barcodePacket[3] = (byte) (barcodeBytes.length + 2); //length of barcode data barcodePacket[4]
= 0x2A;
barcodePacket[5] to barcodePacket[length of barcode string] = BarcodeBytes;
barcodePacket[Length of barcode string + 1] = 0x2A;

BLUPRINTSPrinter: public BLUPRINTSPrinter getBluPrintsPrinter ()
By calling this method, you can create the object of the class BLUPRINTSPrinter.

ArrayList: public ArrayList<String> getPairedPrinters () by calling this method, a list of paired printers is returned.

GetSDKVersion: public String getSDKVersion () by calling this method, SDK version is obtained.

BtConnStatus: public Boolean BtConnStatus ()
This method returns true if the Printer is already connected on Bluetooth (Bluetooth connection is already alive). It returns false if the connection status is False, i.e. if Bluetooth socket is disconnected

Bluetooth (Host) to printer Commands:
To send commands to printer first add [ESC] i.e. 0x1B in starting and then append the query packet. Host application to Printer command:

ESC(0x1B)	Start delimiter (0x7E)	BP*	Separator	Packet Type	Separator	Packet Data	Delimiter (0x5E)
1 Byte	1 Byte	2 Byte	1 Byte	2 Byte	1 Byte	-	1Byte

Host App to Printer Command

*BP: Bluetooth (host) to Printer:

Command Type	Packets
*Get Printer Status :	[ESC]~BP|GET| PRN_ST^
Get Printer Configurat	[ESC]~BP|GET| CONFIG ^
Get Battery Status	[ESC]~BP|GET| BAT_ST ^
Get Printer Version Sta	[ESC]~BP|GET| PRNVER ^ (Returns Ver 1.1.3.21 for Bluetooth and 1.1.3.20 for Non BT)
Printer Power off	[ESC]~BP|PRN| PWROFF ^
Test LED	[ESC]~BP|TST| TSTLED ^
Test Print	[ESC]~BP|TST| PRINT ^
Set Printer Configurati	[ESC]~BP|CON|Name,Password,FontType,EnableEncryption,MSRTimeOut,IC TimeOut^

BT to Printer

Responses from Printer:

Packet Structure:

Start Delimiter (0x7E)	PB*	Separator (1)	Packet Type	Separator (1)	Response	End delimiter (0x5E)
1 Byte	2 Byte	1 Byte	3 Byte	1 Byte	-	1 Byte

*PB: Printer to Android (Host) Response table:

Response Name	Type	Packets
No Paper	Error	~PB|PRN|NOPAPER^
Printer Head High Temp:	Error	~PB|PRN|HGHTEMP^
Printer Platen	Error	~PB|PRN|PLTNREL^
Wrong Packet:	Error	~PB|PRN|PKTEROR^
MSR Swiped	Notification	~PB|CRD|MSR_SWP^
Low Battery	Error	~PB|BAT|LOWBATT^
Battery Charging	Notification	~PB|BAT|BATCHG1^
Battery Not Charging	Notification	~PB|BAT|BATCHG0^
Battery Charge Status	Notification	~PB|BAT|BAT%3d%%^
Printer Configuration	Response	~PB|CON| Name, Password, FontType, EnableEncryption, MSRTimeOut, ICTimeOut^
Printer Power Off	Response	~PB|PRN|PWR_OFF^

Bit Map Image Print – Note that this method of Printing is only maintained for the purpose of Backward compatibility. The Faster, easier, clearer and more standard way of printing is by using PrintImage function and passing it a Bitmap object.
Select the image either from PC or from mobile's SD card.
Convert the image into monochrome bmp.
Resize the image to fit into the printer paper area if it is exceeding
Now read the processed image through fileinputstream and save it into byte array.
Make the packet of image and then send it to printer over outputstream.

ESC(0x1B)	(0x2A)	Mode m	Alignment Of Image (a)	Width Of image (w)*	Height Of image (h)*	Image Byte Array (D1 to Dn)

Bit-Map Image Structure

Mode M (In Hex)	Mode M Description	Vertical Dot	Horizontal Dot	Max Dot/Line
0x64	Singlewidthsingleheight	1	1	384 (48 bytes)
0x65	Single width double height	2	1	384

0x68	Double width single height	1	2	192 (24 bytes)
0x69	Double width double height	2	2	192
0x6B	Double width Quad. Height	4	2	192

Mode Table

Alignment (a)	Hex Value
Left align	0x6C
Centre align	0x63
Right align	0x72

Alignment

QR Code Generation Function:

```
Public void onPrintQRCodeRaster (View v) throws WriterException, IOException {

String text= editText.getText().toString(); if(text.isEmpty()){
showAlert("Write Text To Generate QR Code");
}
else {
Writer writer = new QRCodeWriter();
String finalData = Uri.encode(text, "UTF-8"); showAlert("QR " + text);

try {
BitMatrix bm = writer.encode(finalData, BarcodeFormat.QR_CODE, 300, 300); Bitmap bitmap
= Bitmap.createBitmap(300, 300, Config.ARGB_8888); for (int i = 0; i < 300; i++) {
for (int j = 0; j < 300; j++) {
bitmap.setPixel(i, j, bm.get(i, j) ? Color. BLACK: Color. WHITE);
}
}
Bitmap resizedBitmap = null; int numChars = glbPrinterWidth;
if(numChars == 32){
resizedBitmap = Bitmap.createScaledBitmap(bitmap, 384, 384, false);
m_BluPrintsPrinter.printImage(resizedBitmap); m_BluPrintsPrinter.setCarriageReturn();
```

```
m_BluPrintsPrinter.setCarriageReturn();
m_BluPrintsPrinter.setCarriageReturn();

}else {
resizedBitmap = Bitmap.createScaledBitmap(bitmap, 384, 430, false);
m_BluPrintsPrinter.printImageThreeInch(resizedBitmap); m_BluPrintsPrinter.setCarriageReturn();
m_BluPrintsPrinter.setCarriageReturn(); m_BluPrintsPrinter.setCarriageReturn();
}

} catch (Writer Exception e) { showAlert("Error WrQR: " + e.toString());
}
```