CoiNel Bluetooth Printer
Android SDK User Manual

Version: 1.2.0

CoiNel Bluetooth Printer - Android SDK User Manual

Table of Contents

Introduction....3
Adding SDK to your android project in Android Studio....3
Using COINEL SDK in your android project....4
Printer SDK Commands....5
Printer Selection....5
Reset Printer....5
Test Print....5
Set Print Intensity....5
Print Line Feed....5
Print Reverse Line Feed....5
Print (Ascii) Text....5
Print (Unicode) Text....6
Select Font Size....6
Select Justification....6
Select Font....6
Set Bold Font....6
Print Binarized (Bitonal) Image....7
Save Binarized Image....7
Save Gray Scale Image....8
Print Barcode....9
Paper Status....10
Battery Status....10
Platten Status....10
Developer Notes....11
Sample Code to print a bill....11
Printing an Image....12
Saving an Image....12
Image Printing....13
Printing Barcodes....13
Printing QR codes....14
Printing in (unicode) various language....14
Showing List of available printers....15

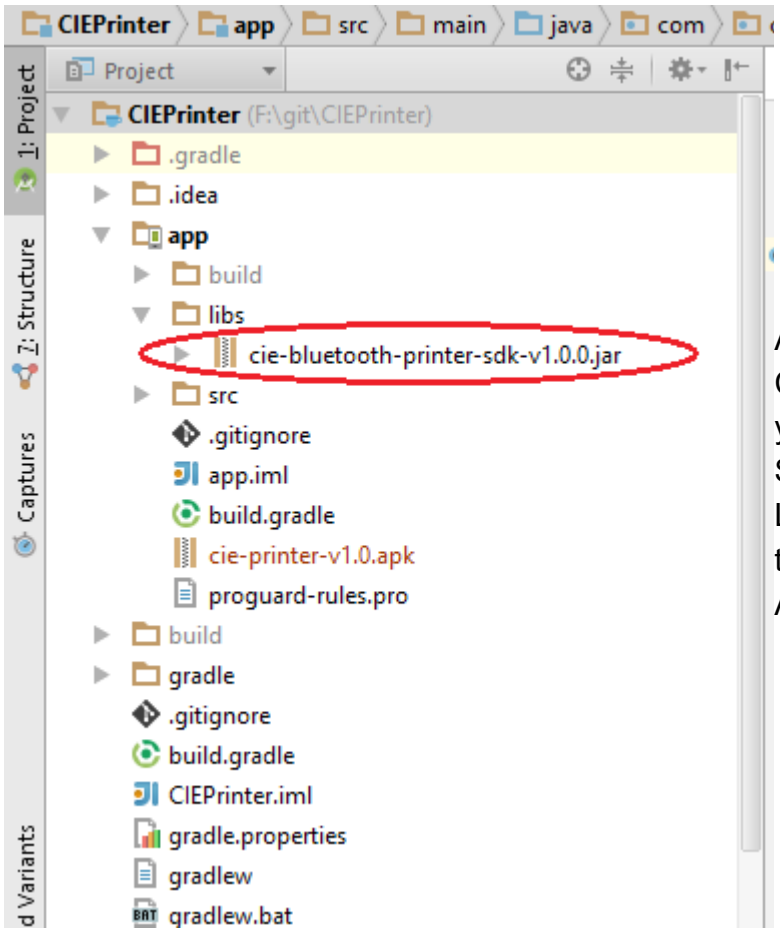
CoiNel Bluetooth Printer - Android SDK User Manual

Introduction

This document describes the android SDK for the COINEL Bluetooth printer. This SDK supports android version 14 (ICS – Icecream Sandwich) and above. The COINEL Bluetooth printer SDK is a jar library that you have to add to your android project. The accompanying android demo project code shows how to use all the functionality of the COINEL Bluetooth printer.

Adding SDK to your android project in Android Studio

After creating your android project in android studio. Create a Libs folder, is not already present and copy the SDK Jar into that folder.



As shown in the image insert COINEL Bluetooth printer SDK in your android project. After inserting SDK right click and select “Add As Library”. Now your project is ready to use the COINEL Bluetooth printer Android SDK.

Using COINEL SDK in your android project

Using the Coinel Bluetooth SDK is very simple. The SDK exposes the Printer as a Singleton Class.

Here are steps described in brief for your easy understanding. please refer to the sample demo project in github for sample code implementation.

Your Projects Android Manifest file should contain permissions for **Bluetooth & Bluetooth Admin**.

In your base activity which will interact with the printer do the following

- override the following methods `onActivityResult, onResume, onPause & onDestroy.`
- Implement a message handler to receive messages from the printer.
- In the `onCreate` method, initialize the printer.

Once the printer is initialized, now you are ready to issue commands to the printer. You can call any of the printer methods that can send commands to the printer. The List of printer commands are given below

Printer SDK Commands

Printer Selection

To select the printer pass the command according to the size of printer.

```
setPrinterWidth(PrinterWidth.PRINTER_WIDTH_48MM); // 2 Inch Printer

setPrinterWidth(PrinterWidth.PRINTER_WIDTH_72MM); // 3 Inch Printer

setPrinterWidth(PrinterWidth.PRINTER_WIDTH_104MM);// 4 Inch Printer
```

Reset Printer

```
resetPrinter()
```

It will reset all commands to default state

Test Print

```
printerTest()
```

Executes the test print command to print printer details

Set Print Intensity

```
setLowIntensity()

setNormalIntensity()

setHighIntensity()
```

Intensity of font by using this commands we can change the Intensity of font as light, medium and bold.

Print Line Feed

```
printLineFeed()
```

Makes the paper to move forward.

Print Reverse Line Feed

```
printReverseLineFeed()
```

Makes the paper to move backward.

Print (Ascii) Text

```
printTextLine(String txt)
```

Input: String text

Prints the given ascii text, please note you can print a maximum of 42 / 48 / 72 characters on a single line for 2 / 3 / 4 Inch Printer respectively. Please ensure that the string is not more than 100 bytes in length.

Print (Unicode) Text

```
printUnicodeText(txt);

printUnicodeText(txt, Layout.Alignment a, TextPaint tp);
```

Input:

String text : the Unicode text to print

Alignment : Left / Center / right Alignment.

TextPaint : Text Paint Object, which defines the text properties.

Select Font Size

`setFontSizeSmall()`

`setFontSizeMedium()`

`setFontSizeLarge()`

`setFontSizeXLarge()`

By this commands we can change the size of the fonts.

Select Justification

`setAlignmentCenter()`

`setAlignmentLeft()`

`setAlignmentRight()`

Aligns the data in a line centre, left and right by using this commands.

Select Font

`setStyleFixedsys()`

`setStyleCourier()`

`setStyleHindi()`

Selects the particular font type for text printing.

Set Bold Font

`SetBold()`

`SetRegular()`

Set font style for printing.

Horizontal Tab

`setTab()`

Moves the print position to the next horizontal tab position.

Print Binarized (Bitonal) Image

```
boolean printBinarizedImage (String path, boolean invert, int threshold,
                                    int align)
```

Input:

String path : image file path

boolean invert : to invert the image or not,

int threshold : Image threshold, pass 0 to auto calculate

int align : image alignment

0 – Left align , 1 – Center align, 2 Right align.

Returns : true when the image is printed successfully

Print Binarized (Bitonal) Image

```
boolean printBinarizedImage (Bitmap bitmap, boolean invert, int threshold,
                                    int align)
```

Input:

Bitmap bitmap : Image Bitmap

boolean invert : to invert the image or not,

int threshold : Image threshold, pass 0 to auto calulate

int align : image alignment

0 – Left align , 1 – Center align, 2 Right align.

Returns : true when the image is printed successfully

Save Binarized Image

```
boolean saveBinarizedImage (String file, boolean bInvertBitmap, int threshold,
            int imageAlign, int imageId)
```

Input:

String file : image file path,

boolean invert : To invert the image or not,

int threshold : Image threshold, pass 0 to auto calulate

int imageAlign : image alignment

0 – Left align , 1 – Center align, 2 Right align.

Int imageId : Image Id, can be 1 – 9 (used when printing saved images.)

Returns : true when the image is saved successfully

Print Gray scale Image

```
boolean printGrayScaleImage (String path, int align)
```

Input:

String path : image file path

int align : image alignment

0 – Left align , 1 – Center align, 2 Right align.

Returns : true when the image is printed successfully

Print Gray scale Image

```
boolean printGrayScaleImage (Bitmap bitmap, int align)
```

Input:

Bitmap bitmap : Image Bitmap

int align :image alignment

0 – Left align , 1 – Center align, 2 Right align.

Returns : true when the image is printed successfully

Save Gray Scale Image

```
boolean saveGrayScaleImage (String file, int imageAlign, int imageId)
```

Input:

String file : image file path,

int imageAlign : image alignment

0 – Left align , 1 – Center align, 2 Right align.

Int imageId : Image Id, can be 1 – 9 (used when printing saved images.)

Returns : true when the image is saved successfully

Print Barcode

```
boolean printBarcode (String data, Barcode type,
                      int width, int height, int imageAlign);
```

Input:

String data : Barcode data,

Barcode type : Barcode format, see below for supported formats.

int Width : width of the barcode (max width should not exceed paper width in pixels),

int Height : height of the barcode.

Int imageAlign : barcode image alignment

0 – Left align , 1 – Center align, 2 Right align.

Returns : true if the bar code printed successfully

Supported Barcode Formats : AZTEC, CODABAR, CODE_39, CODE_93, CODE_128, DATA_MATRIX, EAN_8, EAN_13, ITF, MAXICODE, PDF_417, RSS_14, RSS_EXPANDED, UPC_A, UPC_E and UPC_EAN_EXTENSION

Print Barcode

```
boolean printQRcode (String data, int imageAlign);
```

Input:

String data : Barcode data,

Int imageAlign : barcode image alignment

0 – Left align , 1 – Center align, 2 Right align.

Returns : true if the bar code printed successfully

Paper Status

`getPaperStatus()`

Return the paper status from the printer while the paper is available or not.

Battery Status

`getBatteryStatus()`

Return the presentage of battery available in the printer

Platten Status

`getPlattenStatus()`

Return the Platten status if it is closed or not.

Developer Notes

Sample Code to print a bill

```
// Batch all the commands
mPrinter.setPrintMode(AppConsts.PRINT_IN_BATCH);
mPrinter.setHighIntensity();
// Bill Header Start
mPrinter.setAlignmentCenter();
mPrinter.printLineFeed("MY COMPANY BILL\n");
mPrinter.printLineFeed("~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n");
mPrinter.printLineFeed();
// Bill Header End
// Bill Details Start
mPrinter.setAlignmentLeft();
mPrinter.printTextLine("Customer Name     : John Doe  \n");
mPrinter.printTextLine("Customer Order ID : 12345     \n");
mPrinter.printTextLine("------------------------------\n");
mPrinter.printTextLine("  Item      Quantity     Price\n");
mPrinter.printTextLine("------------------------------\n");
mPrinter.printTextLine("  Item 1          1       1.00\n");
mPrinter.printTextLine("  Some big item   10   7890.00\n");
mPrinter.printTextLine("  Next Item       999 10000.00\n");
mPrinter.printLineFeed();
mPrinter.printTextLine("------------------------------\n");
mPrinter.printTextLine("  Total               17891.00\n");
mPrinter.printTextLine("------------------------------\n");
// Bill Details End
// Bill Footer Start
mPrinter.printLineFeed();
mPrinter.printTextLine("     Thank you ! Visit Again  \n");
mPrinter.printLineFeed();
mPrinter.printTextLine("******************************\n");
mPrinter.printLineFeed();
// Bill Footer End
//Clearance for Paper tear
mPrinter.printLineFeed();
mPrinter.printLineFeed();
mPrinter.resetPrinter();
//send all batched commands to printer
mPrinter.batchPrint();
```

Printing an Image

We can print directly any image of image type jpg, png, bmp. Here is the command to print image directly

```
int imageAlignment = 1; // Center Align
int threshold = 0; // Auto Calculate
boolean r;
if (bImgAlgoGrayScale) {
     r = mPrinter.printGrayScaleImage(fileUri.getPath(), imageAlignment);
}
else {
     r = mPrinter.printBinarizedImage(fileUri.getPath(), Invert, threshold,
imageAlignment);
}
if (r) {
     Toast.makeText(this, "Image Printed", Toast.LENGTH_SHORT).show();
}
else {
     Toast.makeText(this, mPrinter.getPrinterStatusMessage(), Toast.LENGTH_SHORT).show();
}
```

Saving an Image

We can save any image to Printer , the image will be retained in memory even after printer is switched off. Here is the command to print image

```
Boolean Invert;
int imageAlignment = 1; // Center Align
int threshold = 0; // Auto Calculate
Invert = !bInvertBitmap;
boolean r;

if (bImgAlgoGrayScale) {
     r = mPrinter.saveGrayScaleImage(fileUri.getPath(), imageAlignment, indexNumber);
}
else {
     r = mPrinter.saveBinarizedImage(fileUri.getPath(), Invert, threshold,
imageAlignment, indexNumber);
}
if (r) {
     Toast.makeText(this, "Image saved on sImageStoreIndex " + indexNumber,
               Toast.LENGTH_SHORT).show();
}
else {
     Toast.makeText(this, mPrinter.getPrinterStatusMessage(), Toast.LENGTH_SHORT).show();
}
```

Image Printing

For a 2 Inch Printer the max image width is 384 pixels.

For a 3 Inch Printer the max image width is 576 pixels

For a 4 Inch Printer the max image with is 832 pixels.

The with of the image should always be a multiple of 8.

If the input image is bigger in size then it will be scaled proportionately based on the aspect ratio of the image.

If there are lines in the image, ensure that they are at least 2 pixels in width to ensure visibility.

If the image contains letters and characters, please endure they are bold and legible, this will make the letters to be printed clear and understandable.

Printing Barcodes

We can Print the barcode, pass the parameters barcode data, barcode format, width, height to print bar code as shown below.

mPrinter.encodeAsBitmap(txt, BarcodeFormat.***CODE_128***, ***BARCODE_WIDTH***, ***BARCODE_HEIGHT, ImageAlign***);

the printed barcode will look like

The following barcode formats are supported AZTEC, CODABAR, CODE_39, CODE_93, CODE_128, DATA_MATRIX, EAN_8, EAN_13, ITF, MAXICODE, PDF_417, RSS_14, RSS_EXPANDED, UPC_A, UPC_E and UPC_EAN_EXTENSION

Printing QR codes

We can print QR code, pass the parameter QR code data to print QR code as shown below

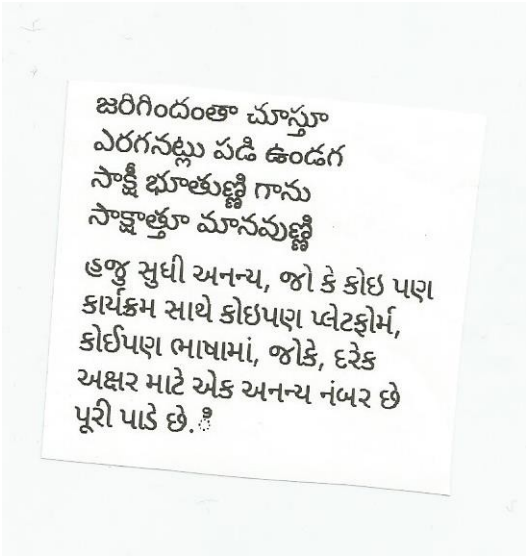
```
mPrinter.printQRcode(data, imageAlign);
```

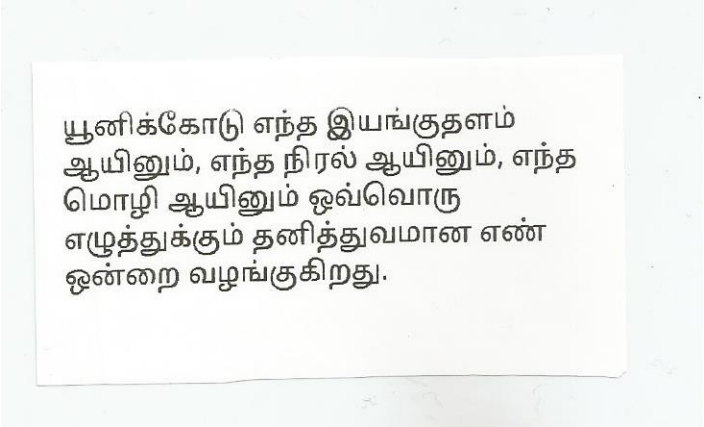
the printed barcode will look like

Printing in (unicode) various language

You can print in any language of our choice on the printer. If the language is supported by the Android OS and the respective font files are available.

```
printUnicodeText(txt);
```





Showing List of available printers

The SDK will remember the last connected Printer and attempt to connect to it automatically, when you initialize the printer. However you can scan and connect to a COINEL Bluetooth printer by calling the method

```
mPrinter.showDeviceList(this);  //it will show list of available printer
```

Here is the screen shot for the dialog that shows up and when scanning for new printers.