

# **Android CPCL Program Manual**

**v3.5.0**

**(Note: Please use the PDF left navigation bar when browsing)**

# 1. Instruction

This manual describes how to print labels with CPCL instructions. Constant variable are defined in CPCLConst class.

## 2. CPCLPrinter

### 2.1. CPCLPrinter

Constructor to create print objects.

CPCLPrinter(IDeviceConnection connection)

[Parameter]

➤ connection

Connected object, available via POSConnect.createDevice(deviceType).

### 2.2. initializePrinter

Label initialization

CPCLPrinter initializePrinter(int height)

CPCLPrinter initializePrinter(int height, int qty)

CPCLPrinter initializePrinter(int offset, int height, int qty)

[Parameter]

➤ offset

The lateral offset of the label. The default value is 0.

➤ height

Maximum height of label

➤ qty

The number of labels to print, default is 1.

[Return]

CPCLPrinter Instance

## 2.3. addText

The method is used to place text on a label.

CPCLPrinter addText(int x, int y, String content)

CPCLPrinter addText(int x, int y, String font, String content)

CPCLPrinter addText(int x, int y, String rotation, String font, String content)

[Parameter]

➤ x

Horizontal starting position.

➤ y

Vertical starting position.

➤ font

Name/number of the font, Default is FNT\_0, Values are FNT\_0, FNT\_1, FNT\_2, FNT\_4, FNT\_5, FNT\_6, FNT\_7, FNT\_24, FNT\_55.

➤ rotation

Rotate the angle, Default is ROTATION\_0

Variable	Description
ROTATION_0	No rotation
ROTATION_90	Rotate 90 degrees clockwise
ROTATION_180	Rotate 180 degrees clockwise
ROTATION_270	Rotate 270 degrees clockwise

➤ content

The text to be printed.

[Return]

CPCLPrinter Instance

## 2.4. setmag

The method magnifies a resident font to the magnification factor specified.

CPCLPrinter setmag(int w, int h)

[Parameter]

➤ w

Width magnification of the font. Valid magnifications are 1 thru 16.

➤ h

Height magnification of the font. Valid magnifications are 1 thru 16.

[Return]

CPCLPrinter Instance

## 2.5. addPrint

The method terminates and prints the file.

```
void addPrint()
```

[Return]

void

## 2.6. addBarcode

The method is used to label bar codes with the same data used to create the bar code.

```
CPCLPrinter addBarcode(int x, int y, String type, int height, String data)
```

```
CPCLPrinter addBarcode(int x, int y, String type, int width, int ratio, int height, String data)
```

Horizontal 1D barcode

```
CPCLPrinter addBarcodeV(int x, int y, String type, int height, String data)
```

```
CPCLPrinter addBarcodeV(int x, int y, String type, int width, int ratio, int height, String data)
```

Vertical 1D barcode

[Parameter]

➤ x

Horizontal starting position.

➤ y

Vertical starting position.

➤ type

Barcode Type

Variable	Description
BCS_128	Code 128
BCS_UPCA	UPC-A
BCS_UPCE	UPC-E
BCS_EAN13	EAN/JAN-13
BCS_EAN8	EAN/JAN-8
BCS_39	Code 39
BCS_93	Code 93/Ext.93
BCS_CODABAR	Codabar

➤ width

Unit-width of the narrow bar.

➤ ratio

Ratio of the wide bar to the narrow bar, Default is BCS\_RATIO\_1

Variable	Description	Variable	Description
----------	-------------	----------	-------------

BCS_RATIO_0	1.5 :1	BCS_RATIO_23	2.3:1
BCS_RATIO_1	2.0 :1	BCS_RATIO_24	2.4:1
BCS_RATIO_2	2.5 :1	BCS_RATIO_25	2.5:1
BCS_RATIO_3	3.0 :1	BCS_RATIO_26	2.6:1
BCS_RATIO_4	3.5 :1	BCS_RATIO_27	2.7:1
BCS_RATIO_20	2.0:1	BCS_RATIO_28	2.8:1
BCS_RATIO_21	2.1:1	BCS_RATIO_29	2.9:1
BCS_RATIO_22	2.2:1	BCS_RATIO_30	3.0:1

➤ height

Unit-height of the bar code.

➤ data

Bar code data.

[Return]

CPCLPrinter Instance

## 2.7. addBarcodeText

The method is used to label bar codes with the same data used to create the bar code.

CPCLPrinter addBarcodeText()

[Return]

CPCLPrinter Instance

## 2.8. addBarcodeTextOff

This method is used to turn off barcode comments

CPCLPrinter addBarcodeTextOff()

[Return]

CPCLPrinter Instance

## 2.9. addQRCode

This method is used to draw a QR code.

CPCLPrinter addQRCode(int x, int y, String data)

CPCLPrinter addQRCode(int x, int y, int codeModel, int cellWidth, String data)

[Parameter]

➤ x

Horizontal starting position.

➤ y

Vertical starting position.

➤ codeModel

QR code model number. Default is QRCODE\_MODE\_ENHANCE.

Variable	Description
QRCODE_MODE_ORG	the original specification
QRCODE_MODE_ENHANCE	Enhanced form of the symbology.

➤ cellWidth

Unit-width/Unit-height of the module.Range is 1 to 32. Default is 6.

➤ data

Describes information required for generating a QR code.

[Return]

CPCLPrinter Instance

## 2.10. addBox

The method provides the user with the ability to produce rectangular shapes of specified line thickness.

CPCLPrinter addBox(int x, int y, int width, int height, int thickness)

[Parameter]

➤ x

Horizontal starting position.

➤ y

Vertical starting position.

➤ width

rectangle width(in dots)

➤ height

rectangle height(in dots)

➤ thickness

Unit-width (or thickness) of the line.

[Return]

CPCLPrinter Instance

## 2.11. addLine

Lines of any length, thickness, and angular orientation can be drawn using the method.

CPCLPrinter addLine(int x, int y, int xend, int yend, int thickness)

[Parameter]

➤ x

X-coordinate of the top-left corner.

➤ y

Y-coordinate of the top-left corner.

➤ xend

abscissa of the end point of the line

➤ yend

The vertical coordinate of the end point of the line

➤ thickness

Unit-width (or thickness) of the line

[Return]

CPCLPrinter Instance

## 2.12. addInverseLine

Previously created objects that lie within the area defined by the method will have their black areas re-drawn white, and white areas re-drawn black.

CPCLPrinter addInverseLine(int x, int y, int xend, int yend, int width)

[Parameter]

➤ x

X-coordinate of the top-left corner.

➤ y

Y-coordinate of the top-left corner.

➤ xend

abscissa of the end point of the line

➤ yend

The vertical coordinate of the end point of the line

➤ width

Unit-width (or thickness) of the line

[Return]

CPCLPrinter Instance

## 2.13. addGraphics

Image printing, it is recommended to use addCGraphics.

CPCLPrinter addCGraphics(int x, int y, int width, Bitmap bmp)

CPCLPrinter addCGraphics(int x, int y, int width, Bitmap bmp, AlgorithmType algorithmType)

Use byte type to transfer image data

CPCLPrinter addEGraphics(int x, int y, int width, Bitmap bmp)

CPCLPrinter addEGraphics(int x, int y, int width, Bitmap bmp, AlgorithmType algorithmType)

Use hexadecimal character type to transmit image data

[Parameter]

➤ x

Horizontal starting position.

➤ y

Vertical starting position.

➤ width

Print width of picture.

➤ bmp

Bitmap object.

➤ algorithmType

Algorithm type. Default is AlgorithmType.Threshold.

AlgorithmType.Dithering

AlgorithmType.Threshold

[Return]

CPCLPrinter Instance

## 2.14. printPdf

This method is used to print PDF documents

CPCLPrinter printPdf(int x, int y, String path, int pageWidth, int labelHeight)

CPCLPrinter printPdf(int x, int y, String path, int pageWidth, int labelHeight, AlgorithmType algorithmType)

[Parameter]

➤ x

Horizontal starting position.

➤ y

Vertical starting position.

➤ path

PDF document path

➤ pageWidth

Print width of pdf.



➤ labelHeight

Maximum height of label

➤ algorithmType

Algorithm type. Default is AlgorithmType.Threshold.

AlgorithmType.Dithering

AlgorithmType.Threshold

[Return]

CPCLPrinter Instance

## 2.15. addAlign

Alignment of fields can be controlled by using the method.

CPCLPrinter addAlign(int align)

CPCLPrinter addAlign(int align, int end)

[Parameter]

➤ align

Alignment

Variable	Description
ALIGNMENT_LEFT	Left justifies all subsequent fields.
ALIGNMENT_CENTER	Center justifies all subsequent fields.
ALIGNMENT_RIGHT	Right justifies all subsequent fields.

➤ end

End point of justification. If no parameter is entered, justification commands use the printhead's width for horizontal printing or zero (top of form) for vertical printing.

[Return]

CPCLPrinter Instance

## 2.16. printerStatus

Get printer status

void printerStatus(IStatusCallback callback)

void printerStatus(int timeout, IStatusCallback callback)

[Parameter]

➤ callback

The callback content is the corresponding printer state

public interface IStatusCallback {

```

    void receive(int status);
}

```

status(HEX)	Description
00	Normal
01	Head opened
02	Paper Jam
03	Paper Jam and head opened
04	Out of paper
05	Out of paper and head opened
08	Out of ribbon
09	Out of ribbon and head opened
0A	Out of ribbon and paper jam
0B	Out of ribbon, paper jam and head opened
0C	Out of ribbon and out of paper
0D	Out of ribbon, out of paper and head opened
10	Pause
20	Printing
80	Other error
READ_TIMEOUT	Receive timeout
CONNECT_ERROR	Connection error

#### ➤ timeout

Receive timeout, Unit is ms,Default is 5000ms

[Return]

CPCLPrinter Instance

## 2.17. addSpeed

This method is used to set the highest motor speed level.

CPCLPrinter addSpeed(int level)

[Parameter]

#### ➤ level

A number between 0 and 5, 0 being the slowest speed.

[Return]

CPCLPrinter Instance

## 2.18. addPageWidth

The printer assumes that the page width is the full width of the printer. The maximum height of a print session is determined by the page width and the available print memory. If the page width is less than the full width of the printer, the user can increase the maximum page height by specifying the page width.

CPCLPrinter addPageWidth(int width)

[Parameter]

➤ width

Unit-width of the page.

[Return]

CPCLPrinter Instance

## 2.19. addBeep

This method instructs the printer to sound the beeper for a given time length. Printers not equipped with a beeper will ignore this method.

CPCLPrinter addBeep(int length)

[Parameter]

➤ length

Duration of beep, specified in (1/8th) second increments. This example instructs the printer to beep for two seconds (16 x .125 seconds = 2 seconds).

[Return]

CPCLPrinter Instance

## 2.20. getBtMac

Get Bluetooth mac address

void getBtMac(int timeout, IStrCallback callback)

[Parameter]

➤ timeout

Receive timeout, Unit is ms.

➤ callback

mac address callback.

public interface IStrCallback {

```
        void receive(String info);  
    }
```

info is the returned mac address, if it is an empty string, it means the acquisition failed.

[Return]  
void

## 2.21. setCharSet

Set character encoding, Default is "gbk"  
void setCharSet(String charSet)

[Parameter]  
➤ charSet  
Character set name.

## 2.22. sendData

This function is used to send data to the printer.  
CPCLPrinter sendData(byte[] data)  
CPCLPrinter sendData(List<byte[]> datas)

[Parameter]  
➤ data  
Byte array to be sent  
➤ datas  
Byte array collection to be sent

[Return]  
CPCLPrinter Instance

## 2.23. waitSendResultSync

Check if the data has been sent successfully.

void waitSendResultSync(int timeout)

[Parameter]  
➤ timeout  
Maximum detection time, Unit in milliseconds

[Return]

void

## ImageUtils

### handleImageEffect

This method is used to adjust the contrast and brightness of the image.

static Bitmap handleImageEffect(Bitmap bmp, float contrast, float brightness)

[Parameter]

- bmp  
Original image
- contrast  
Contrast, The range is 0~2
- brightness  
Brightness, The range is -255~255

[Return]

processed image object

## 3. LabelUdpNet

Label Printer UDP Message Receive Class, which enables the discovery of printer devices within the local network.

### 3.1. searchNetDevice

Search for printing devices in the local area network

void searchNetDevice(UdpCallback callback)

[Parameter]

- callback

Returns the found device information by way of callback.

```
public interface UdpCallback {  
    void receive(UdpDevice device);  
}
```

[Return]

void

## 4. UdpDevice

Found network printer information object. The label printer only has the MAC address and IP address fields as valid.

### 4.1. getMacAddress

The MAC address of the printer

byte[] getMacAddress()

String getMacStr()

[Return]

Mac Address

### 4.2. getIpStr

IP address.

String getIpStr() throws UnknownHostException

[Return]

Ip address