

Android ZPL Program Manual

v2.2.0

1. Instruction

This manual describes how to print labels with ZPL instructions. Constant variables are defined in ZPLConst class.

2. ZPLPrinter

2.1. ZPLPrinter

Constructor to create print objects.

ZPLPrinter(IDeviceConnection connection)

[Parameter]

➤ connection

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

2.2. addStart

This method is used at the beginning of the label

ZPLPrinter addStart()

[Return]

ZPLPrinter Instance

2.3. addEnd

End of label format. After calling this method, the label is printed.

ZPLPrinter addEnd()

[Return]

ZPLPrinter Instance

2.4. addText

text printing

ZPLPrinter addText(int x, int y, String fontName, String rotation, int sizeW, int sizeH, String

content)

ZPLPrinter addText(int x, int y, char fontName, int sizeW, int sizeH, String content)

ZPLPrinter addText(int x, int y, char fontName, String content)

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

[Parameter]

➤ x

the starting x value of the text

➤ y

the starting y value of the text

➤ font

The font type of the text, the default is FNT_F.

| Variable | Description |
|-------------|-------------|
| FNT_A | 9 x 5 |
| FNT_B | 11x7 |
| FNT_C、FNT_D | 18x10 |
| FNT_E | 28x15 |
| FNT_F | 26x13 |
| FNT_G | 60x40 |
| FNT_O | 15*12 |

For other fonts, please use custom names.

➤ sizeW

The effective width of the text, the default is the base size. Please use an integer multiple of the base size.

➤ sizeH

The effective height of the text. Default is base size. Please use an integer multiple of the base size.

➤ rotation

Clockwise rotation angle, default 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

text content

[Return]

ZPLPrinter Instance

2.5. setCustomFont

Set custom font. After the machine is powered off, the settings will be invalid.

ZPLPrinter setCustomFont(String font, char alias, int codePage)

[Parameter]

➤ font

Font name and suffix of font library, for example: LZHONGHEI.TTF

➤ alias

Font alias, corresponding to fontName in addText. Range: A to Z and 0 to 9.

➤ codePage

Character Encoding

| Variable | Description |
|-------------------|--|
| CODE_PAGE_UTF8 | Unicode (UTF-8 encoding) - Unicode Character Set |
| CODE_PAGE_UTF16 | Unicode (UTF-16 Big-Endian encoding) - Unicode Character Set |
| CODE_PAGE_UTF16_2 | Unicode (UTF-16 Little-Endian encoding) - Unicode Character Set |
| CODE_PAGE_USA1 | Single Byte Encoding - U.S.A. 1 Character Set |
| CODE_PAGE_USA2 | Single Byte Encoding - U.S.A. 2 Character Set |
| CODE_PAGE_UK | Single Byte Encoding - U.K. Character Set |
| CODE_PAGE_NL | Single Byte Encoding - Holland Character Set |
| CODE_PAGE_DK | Single Byte Encoding - Denmark/Norway Character Set |
| CODE_PAGE_SWEDE | Single Byte Encoding - Sweden/Finland Character Set |
| CODE_PAGE_GER | Single Byte Encoding - Germany Character Set |
| CODE_PAGE_FR1 | Single Byte Encoding - France 1 Character Set |
| CODE_PAGE_FR2 | Single Byte Encoding - France 2 Character Set |
| CODE_PAGE_ITA | Single Byte Encoding - Italy Character Set |
| CODE_PAGE_ES | Single Byte Encoding - Spain Character Set |
| CODE_PAGE_JA | Single Byte Encoding - Japan (ASCII with Yen symbol) Character Set |

[Return]

ZPLPrinter Instance

2.6. setPrinterWidth

Set Print Width

ZPLPrinter setPrinterWidth(int width)

[Parameter]

➤ width

label width (in dots)

[Return]

ZPLPrinter Instance

2.7. addReverse

Area Reverse

ZPLPrinter addReverse(int x, int y, int width, int height)

ZPLPrinter addReverse(int x, int y, int width, int height, int radius)

[Parameter]

➤ x

Start x value of the area

➤ y

Start y value of the area

➤ width

Area width

➤ height

Area height

➤ radius

degree of corner rounding. Range: 0 (no rounding) to 8 (heaviest rounding). Default is 0.

[Return]

ZPLPrinter Instance

2.8. addBox

The method is used to draw boxes and lines as part of a label format.

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

ZPLPrinter addBox(int x, int y, int width, int height, int thickness, int radius)

[Parameter]

➤ x

Start x value of the box

➤ y

Start y value of the box

➤ width

box width (in dots)

➤ height

box height (in dots)

➤ thickness

border thickness (in dots)

➤ radius

degree of corner rounding, Range: 0(no rounding) to 8 (heaviest rounding), Default is 0.

[Return]

ZPLPrinter Instance

2.9. addBarcode

This method prints 1D barcodes.

ZPLPrinter addBarcode(int x, int y, String codeType, String ratio, byte textPosition, String data, int width, int height)

ZPLPrinter addBarcode(int x, int y, String codeType, String data, int height)

ZPLPrinter addBarcode(int x, int y, String codeType, String data)

[Parameter]

➤ x

Start x value of the barcode

➤ y

Start y value of the barcode

➤ codeType

Code type

| Variable | Description |
|---------------------|-----------------------------|
| BCS_CODE11 | Code 11 barcode |
| BCS_INTERLEAVED2OF5 | Interleaved 2 of 5 Bar Code |
| BCS_CODE39 | Code 39 Barcode |
| BCS_EAN8 | EAN-8 Barcode |
| BCS_UPCE | UPC-E Barcode |
| BCS_CODE93 | Code 93 Barcode |
| BCS_CODE128 | Code 128 Barcode |
| BCS_EAN13 | EAN-13 Barcode |
| BCS_CODABAR | ANSI Codabar Bar Code |
| BCS_MSI | MSI Bar Code |
| BCS_PLESSEY | Plessey Bar Code |
| BCS_UPC_EAN | UPC/EAN Extensions |
| BCS_UPCA | UPC-A Bar Code |

➤ ratio

Barcode direction, Default is ROTATION_0

➤ textPosition

Interpretation line position, Default is HRI_TEXT_BELOW.

| Variable | Description |
|----------|-------------|
|----------|-------------|

| | |
|----------------|--------------------------------------|
| HRI_TEXT_NONE | No Interpretation |
| HRI_TEXT_ABOVE | print interpretation line above code |
| HRI_TEXT_BELOW | print interpretation line below code |

➤ data

Barcode content

➤ width

module width (in dots), Default is 2.

➤ height

bar code height (in dots), Default is 50.

[Return]

ZPLPrinter Instance

2.10. addQRCode

Add 2D barcode

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

ZPLPrinter addQRCode(int x, int y, int size, String data)

[Parameter]

➤ x

Start x value of the qrcode

➤ y

Start y value of the qrcode

➤ data

QRCode content

➤ size

magnification factor. Values:1 to 10, default is 3.

[Return]

ZPLPrinter Instance

2.11. downloadBitmap

The method downloads a graphic image.

ZPLPrinter downloadBitmap(int width, String bmpName, Bitmap bmp)

[Parameter]

➤ width

image width

➤ bmpName

image name and extension, The number or character whose name is 1 to 8.

➤ bmp

Bitmap object

[Return]

ZPLPrinter Instance

2.12. addBitmap

print bitmap

ZPLPrinter addBitmap(int x, int y, String bmpName, int mx, int my)

ZPLPrinter addBitmap(int x, int y, String bmpName)

[Parameter]

➤ x

Start x value of the bitmap

➤ y

Start y value of the bitmap

➤ bmpName

Bitmap name and extension name

➤ mx

magnification factor on the x-axis,The default value is 1, and the range is 1~10.

➤ my

magnification factor on the y-axis,The default value is 1, and the range is 1~10.

[Return]

ZPLPrinter Instance

2.13. addPrintCount

The method controls the number of labels to print

ZPLPrinter addPrintCount(int count)

[Parameter]

➤ count

total quantity of labels to print

[Return]

ZPLPrinter Instance

2.14. isConnected

Query connection status

```
void isConnected(IStatusCallback callback)
```

[Parameter]

➤ callback

Status callback.

```
public interface IStatusCallback {  
    void receive(int status);  
}
```

| status | Description |
|----------------|-------------|
| STS_CONNECT | connect |
| STS_DISCONNECT | disconnect |

2.15. setCharSet

Set character encoding, Default is "UTF-8"

```
void setCharSet(String charSet)
```

[Parameter]

➤ charSet

Character set name.

2.16. sendData

This function is used to send data to the printer.

```
ZPLPrinter sendData(byte[] data);
```

```
ZPLPrinter sendData(List<byte[]> datas);
```

[Parameter]

➤ data

Byte array to be sent

➤ datas

Byte array collection to be sent

[Return]

ZPLPrinter Instance