

Flutter ZPL Program Manual

v1.2.0

1.Instruction

By reading this manual, developers can quickly learn how to use Flutter to implement ZPL printing functions and apply them to actual development. This manual includes the use of the `PrinterManager` class and the `ZPLCommand` class, as well as the meaning and usage of the constants in the `ZPLConst` class.

2.PrinterManager

2.1.PrinterManager

Constructor, create a printer management object

`PrinterManager()`

2.2.connectBt

This method is used for Bluetooth connection (Android uses classic Bluetooth, iOS uses BLE)

`Future<void> connectBt(String address, Function(int result) callback)`

[Parameter]

➤address

Bluetooth Address

➤callback

Connection status callback

2.3.connectUsb

This method is used for USB connection (only supports Android)

`Future<void> connectUsb(String path, Function(int result) callback)`

[Parameter]

➤path

usb path

➤callback

Connection status callback

2.4.getUsbPaths

This method is used to obtain the currently connected USB device list (only supports android)

Future<List> getUsbPaths()
[Return]
Usb List object: usb address list collection

2.5.searchCallback

Bluetooth search callback

searchCallback(Map args)
[Return]
Map dictionary object of Bluetooth information: key: address, value: name

2.6.sendZPL

This method is used to send ZPL instructions
Future<void> sendZPL(List<Map<String, dynamic>> data)

2.7.checkPermissions

This method is used to check Android Bluetooth authorization
Future<bool> checkPermissions()
[Return]
Bool object

2.8.startScan

This method is used for Bluetooth search (Android: Classic Bluetooth search, iOS: BLE search)
Future<void> startScan()

2.9.disconnect

Disconnect
Future<void> disconnect()

2.10.PrinterLabelStatus

Get label printer status
Future<int> printerLabelStatus(int timeout)

[Parameter]
➤ timeout
Receive timeout, Unit is ms,Default is 5000ms
[Return]

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
-1	Receive timeout

2.11.checkIsConnect

Query connection status

Future<int> checkIsConnect()

[Return]

status	Description
STS_CONNECT	connect
STS_DISCONNECT	disconnect

3.ZPLCommand

3.1.ZPLCommand

Constructor to create print objects.

ZPLCommand()

[Parameter]

➤connection

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

3.2.addStart

This method is used at the beginning of the label

ZPLCommand addStart()

[Return]

ZPLCommand Instance

3.3.addEnd

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

ZPLCommand addEnd()

[Return]

ZPLCommand Instance

3.4.addText

text printing

ZPLCommand addText(int x, int y,String content,{String fontName = ZPLConst.FNT_F, String rotation = ZPLConst.ROTATION_0, int sizeW = 1, int sizeH = 1})

[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_0	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]

ZPLCommand Instance

3.5.setCustomFont

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

ZPLCommand 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

CharacterEncoding

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]

ZPLCommand Instance

3.6.setPrinterWidth

Set Printer Width

ZPLCommand setPrinterWidth(int width)

[Parameter]

➤width

label width (in dots)

[Return]

ZPLCommand Instance

3.7.setLabelLength

The method defines the length of the label.

ZPLCommand setLabelLength(int length)

[Parameter]

➤length

label length(in dots)

[Return]

ZPLCommand Instance

3.8.addReverse

Area Reverse

ZPLCommand addReverse(int x, int y, int width, int height, {int radius = 0})

[Parameter]

➤x

Start x value of the area

➤y

Start y value of the area

➤width Area

width

➤height Area

height

➤radius

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

[Return]

ZPLCommand Instance

3.9.addBox

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

ZPLCommand addBox(int x, int y, int width, int height, int thickness, {int radius = 0})

[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 cornerrounding, Range: 0(no rounding) to 8 (heaviest rounding), Default is 0.

[Return]

ZPLCommand Instance

3.10.addGraphicDiagonalLine

The method is used to draw diagonals.

ZPLCommand addGraphicDiagonalLine(int x, int y, char orientation, int width, int height, int thickness)

[Parameter]

➤x

Horizontal starting position

➤y

Vertical starting position

➤orientation

The direction of the diagonal.

Variable	Description
R (or /)	right slanted diagonal
L (or \)	left slanted diagonal

➤width

The width of the box (range: 1-32000, unit: dot).

➤height

The height of the box (range: 1-32000, unit: dot).

➤thickness

Boundary thickness (range: 1-32000, unit: dot).

[Return]

ZPLCommand Instance

3.11.addGraphicEllipse

The method is used to draw a graphical ellipse.

ZPLCommand addGraphicEllipse(int x, int y, int width, int height, int thickness)

[Parameter]

➤x

Horizontal starting position

➤y

Vertical starting position

➤width

Ellipse width (range: 3-4095, unit: dot).

➤height

Ellipse height (range: 3-4095, unit: dot).

➤thickness

Boundary thickness (range: 2-4095, unit: dot).

[Return]

ZPLCommand Instance

3.12.addGraphicCircle

The method is used to command produces a circle on the printed label.

ZPLCommand addGraphicCircle(int x, int y, int diameter, int thickness)

[Parameter]

➤x

Horizontal starting position

➤y

Vertical starting position

➤diameter

Round diameter(range:3-4095,unit:dot).

➤thickness

Boundary thickness(range:1-4095,unit:dot).

[Return]

ZPLCommand Instance

3.13.addBarcode

The method is used to prints 1D barcodes.

```
ZPLCommand addBarcode(int x, int y, String codeType,String data, {String ratio = ZPLConst.ROTATION_0,  
    int textPosition = ZPLConst.HRI_TEXT_BELOW, int width = 2, int height = 50})
```

[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]

ZPLCommand Instance

3.14.addQRCode

Add 2D barcode

ZPLCommand addQRCode(int x, int y, String data,{int size = 3})

[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]

ZPLCommand Instance

3.15.printBitmap

Print pictures

ZPLCommand printBitmap(int x, int y, Uint8List bitmap, int width)

By transmitting images through compression, you can save transmission time.

ZPLCommand printBmpCompress(int x, int y, Uint8List bitmap, int width)

[Parameter]

➤ x

Horizontal starting position

➤ y

Vertical starting position

➤ bmp

Print width of picture

➤ width

Print width of picture

[Return]

ZPLCommand Instance

3.16.downloadBitmap

The method is used to download a graphic image.

ZPLCommand downloadBitmap(int width, String bmpName, Uint8List bitmap)

[Parameter]

➤width

Print width of picture

➤bmpName

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

➤bmp Bitmap

object

➤algorithmType

Algorithm type. Default is AlgorithmType.Threshold.

AlgorithmType.Dithering

AlgorithmType.Threshold

[Return]

ZPLCommand Instance

3.17.useDownloadBitmap

The method is used to print bitmap

ZPLCommand useDownloadBitmap(int x, int y, String bmpName, {int mx = 1, int my = 1})

[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]

ZPLCommand Instance

3.18.addPrintCount

The method controls the number of labels to print

ZPLCommand addPrintCount(int count)

[Parameter]

➤count

total quantity of labels to print

[Return]

ZPLCommand Instance

3.19.setPrintSpeed

This method is used to set the printing speed.

ZPLCommand setPrintSpeed(int speed)

[Parameter]

➤speed

print speed. Unit is inches/sec

[Return]

ZPLCommand Instance

3.20.setPrintOrientation

The method inverts the label format 180 degrees.The label appears to be printed upside down.

ZPLCommand setPrintOrientation(String orientation)

[Parameter]

➤orientation Print

Orientation

Variable	Description
ROTATION_0	normal
ROTATION_180	invert

[Return]

ZPLCommand Instance

3.21.setPrintDensity

The method is used to set the darkness of printing.

ZPLCommand setPrintDensity(int density)

[Parameter]

➤ density

desired darkness(range: 0-30)

[Return]

ZPLCommand Instance

3.22.setCharSet

Set character encoding

ZPLCommand setCharSet(String charSet)

[Parameter]

➤ charSet

Character set name.

3.23.sendData

The method is used to send data to the printer.

ZPLCommand sendData(UInt8List data)

[Parameter]

➤ data

Byte array to be sent

[Return]

ZPLCommand Instance