

Android POS Program Manual

V3.5.8

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

1. Instruction

This manual describes how to implement ESC/POS printing. Constant variables are defined in POSConst class.

2. POSPrinter

2.1. POSPrinter

Constructor to create print objects.

POSPrinter(IDeviceConnection connection)

[Parameter]

➤ Connection

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

2.2. printString

This function is used for text-printing.

POSPrinter printString(String data)

[Parameter]

➤ data

Printed text string

[Return]

POSPrinter Instance

2.3. printText

This function is used for format-specific text printing.

POSPrinter printText(String data, int alignment, int attribute, int textSize)

POSPrinter printTextSize(String data, int textSize)

POSPrinter printTextAttribute(String data, int attribute)

POSPrinter printTextAlignment(String data, int alignment)

[Parameter]

➤ data

Printed text string

➤ alignment

The alignment of the text, and the default is ALIGNMENT_LEFT

Note: When using alignment, data needs to end with "\n", otherwise it may become invalid.

Variable	Description
ALIGNMENT_LEFT	Align left
ALIGNMENT_CENTER	Align center
ALIGNMENT_RIGHT	Align right

➤ attribute

This value is text attributes. It sets text attributes to print. default is FNT_DEFAULT

Variable	Description
FNT_DEFAULT	FontA, Set up as a standard
FNT_FONTB	Set up as FontB
FNT_BOLD	bold font
FNT_REVERSE	Set up as reverse print attribute
FNT_UNDERLINE	Set up as Underline attribute
FNT_UNDERLINE2	Set up as Bold Underline attribute

➤ textSize

The font size of the printed text font,default is TXT_1WIDTH|TXT_1HEIGHT

Variable(Set up width ratio)	Description
TXT_1WIDTH	Set up width ratio as x1
TXT_2WIDTH	Set up width ratio as x2
TXT_3WIDTH	Set up width ratio as x3
TXT_4WIDTH	Set up width ratio as x4
TXT_5WIDTH	Set up width ratio as x5
TXT_6WIDTH	Set up width ratio as x6
TXT_7WIDTH	Set up width ratio as x7
TXT_8WIDTH	Set up width ratio as x8

Variable(Set up height ratio)	Description
TXT_1HEIGHT	Set up height ratio as x1
TXT_2HEIGHT	Set up height ratio as x2
TXT_3HEIGHT	Set up height ratio as x3
TXT_4HEIGHT	Set up height ratio as x4
TXT_5HEIGHT	Set up height ratio as x5
TXT_6HEIGHT	Set up height ratio as x6
TXT_7HEIGHT	Set up height ratio as x7
TXT_8HEIGHT	Set up height ratio as x8

[Return]

POSPrinter Instance

2.4. printBitmap

POSPrinter printBitmap(String bitmapPath, int alignment, int width)

POSPrinter printBitmap(String bitmapPath, int alignment, int width, int model)

POSPrinter printBitmap(String bitmapPath, int alignment, int width, int model, AlgorithmType bmpType)

POSPrinter printBitmap(Bitmap bmp, int alignment, int width)

POSPrinter printBitmap(Bitmap bmp, int alignment, int width, int model)

POSPrinter printBitmap(Bitmap bmp, int alignment, int width, int model, AlgorithmType bmpType)

Compress and send printed images, supporting some models.

POSPrinter printBitmapCompress(Bitmap bmp, int alignment, int width, int model)

POSPrinter printBitmapCompress(Bitmap bmp, int alignment, int width)

[Parameter]

➤ bitmapPath

with full path of bitmap file.

➤ bmp

Android Bitmap Object.

➤ alignment

The alignment mode of the pictures.

Variable	Description
ALIGNMENT_LEFT	Align left
ALIGNMENT_CENTER	Align center
ALIGNMENT_RIGHT	Align right

➤ width

Print the picture width.

➤ model

Print mode

Variable	Description
BMP_NORMAL	Original(Normal) size
BMP_WIDTH_DOUBLE	Double width
BMP_HEIGHT_DOUBLE	Double height
BMP_WIDTH_HEIGHT_DOUBLE	Double size

➤ bmpType

Image algorithm

Variable	Description
Threshold , None	Binary algorithm (default)
Dithering, Diffusion	Diffusion algorithm
Halftone	Halftone algorithm

[Return]

2.5. printBarcode

This function is used for supporting barcode printing.

POSPrinter printBarcode(String data, int codeType)

POSPrinter printBarcode(String data, int codeType, int width, int height, int alignment)

POSPrinter printBarcode(String data, int codeType, int width, int height, int alignment, int textPosition)

[Parameter]

➤ data

Barcode string to be printed

➤ codeType

Barcode type

Variable	Description
BCS_UPCA	UPC A
BCS_UPCE	UPCE
BCS_EAN8	EAN-8
BCS_EAN13	EAN-13
BCS_JAN8	JAN-8
BCS_JAN13	JAN-13
BCS_ITF	ITF
BCS_Codabar	Codabar
BCS_Code39	Code 39
BCS_Code93	Code 93
BCS_Code128	Code 128, For this type, the data must be added with {A, {B, {C, etc

➤ height

Barcode height, range [1,255].Default is 162

➤ width

This values barcode width in Dot Units, range [2, 6], Default is 3

➤ alignment

It sets barcode alignment, Default is ALIGNMENT_CENTER

Variable	Description
ALIGNMENT_LEFT	Align left
ALIGNMENT_CENTER	Align center
ALIGNMENT_RIGHT	Align right

➤ textPosition

This value is printing position of barcode HRI letters(barcode data).Default is HRI_TEXT_BELOW.

Variable	Description
HRI_TEXT_NONE	Do not print barcode data
HRI_TEXT_ABOVE	Print barcode data above the barcode
HRI_TEXT_BELOW	Print barcode data below the barcode
HRI_TEXT_BOTH	Print barcode data top and bottom

[Return]

POSPrinter Instance

2.6. feed

This function is used for sending feeding command to printer.

POSPrinter feedLine(int lineCount)

POSPrinter feedLine()

[Parameter]

➤ lineCount

This value is the number of lines for line feeding. Default is 1

[Return]

POSPrinter Instance

2.7. printQRCode

This function is used for supporting QRCode barcode printing.

POSPrinter printQRCode(String data)

POSPrinter printQRCode(String data, int alignment)

POSPrinter printQRCode(String data, int moduleSize, int alignment)

POSPrinter printQRCode(String data, int moduleSize, int ecLevel, int alignment)

[Parameter]

➤ data

QRCode data to print

➤ moduleSize

Module size. Range[1, 16], Default is 8.

➤ ecLevel

Error Correction Level, Default is QRCODE_EC_LEVEL_L

Variable	Description
QRCODE_EC_LEVEL_L	Error correction Level L (7%)
QRCODE_EC_LEVEL_M	Error correction Level M (15%)
QRCODE_EC_LEVEL_Q	Error correction Level Q (25%)

QRCODE_EC_LEVEL_H	Error correction Level H (30%)
-------------------	--------------------------------

➤ alignment

It sets QRCode alignment, Default is ALIGNMENT_CENTER

Variable	Description
ALIGNMENT_LEFT	Align left
ALIGNMENT_CENTER	Align center
ALIGNMENT_RIGHT	Align right

[Return]

POSPrinter Instance

2.8. cutPaper

This method is used for cutting the paper

POSPrinter cutPaper()

POSPrinter cutPaper(int model)

POSPrinter cutHalfAndFeed(int distance)

Feed paper distance, and half cut paper.

[Parameter]

➤ model

Cut paper mode, Default is CUT_HALF.

Variable	Description
CUT_ALL	Full cut
CUT_HALF	Half cut

➤ distance

Feed distance

[Return]

POSPrinter Instance

2.9. openCashBox

Open a cash drawer.

POSPrinter openCashBox(int pinNum)

POSPrinter openCashBox(int pinNum, int onTime, int offTime)

[Parameter]

➤ pinNum

Pin number to generate pulse.

Variable	Description
PIN_TWO	PIN 2
PIN_FIVE	PIN 5

➤ onTime

Start tiime to generate pulse. onTime*2ms, range [0,255], Default is 30

➤ offTime

Stop time to generate pulse. offTime*2ms, range [0,255], Default is 255

[Return]

POSPrinter Instance

2.10. setCharSet

Set character encoding,Default is “gbk”

void setCharSet(String charSet)

[Parameter]

➤ charSet

Character set name.

2.11. setTextStyle

This function is used for set the font style.

POSPrinter setTextStyle(int attribute, int textSize)

[Parameter]

➤ attribute

This value is text attributes. It sets text attributes to print. default is FNT_DEFAULT

Variable	Description
FNT_DEFAULT	FontA, Set up as a standard
FNT_FONTB	Set up as FontB
FNT_BOLD	bold font
FNT_REVERSE	Set up as reverse print attribute
FNT_UNDERLINE	Set up as Underline attribute
FNT_UNDERLINE2	Set up as Bold Underline attribute

➤ textSize

The font size of the printed text font,default is TXT_1WIDTH|TXT_1HEIGHT

Variable(Set up width ratio)	Description
TXT_1WIDTH	Set up width ratio as x1

TXT_2WIDTH	Set up width ratio as x2
TXT_3WIDTH	Set up width ratio as x3
TXT_4WIDTH	Set up width ratio as x4
TXT_5WIDTH	Set up width ratio as x5
TXT_6WIDTH	Set up width ratio as x6
TXT_7WIDTH	Set up width ratio as x7
TXT_8WIDTH	Set up width ratio as x8

Variable(Set up height ratio)	Description
TXT_1HEIGHT	Set up height ratio as x1
TXT_2HEIGHT	Set up height ratio as x2
TXT_3HEIGHT	Set up height ratio as x3
TXT_4HEIGHT	Set up height ratio as x4
TXT_5HEIGHT	Set up height ratio as x5
TXT_6HEIGHT	Set up height ratio as x6
TXT_7HEIGHT	Set up height ratio as x7
TXT_8HEIGHT	Set up height ratio as x8

[Return]

POSPrinter Instance

2.12. setAlignment

This method is used for set up the alignment of the text

POSPrinter setAlignment(int alignment)

[Parameter]

➤ alignment

The alignment of the text, and the default is ALIGNMENT_LEFT

Variable	Description
ALIGNMENT_LEFT	Align left
ALIGNMENT_CENTER	Align center
ALIGNMENT_RIGHT	Align right

[Return]

POSPrinter Instance

2.13. printerCheck

This function is used for query all of the printer states.Do not use simultaneously with

listenPrinterMessage.

```
void printerCheck(int type, int timeout, IDataCallback callback)
```

[Parameter]

➤ type

Variable	Description
STS_TYPE_PRINT	Print state
STS_TYPE_OFFLINE	off-line state
STS_TYPE_ERR	Error state
STS_TYPE_PAPER	Transfer paper state

➤ timeout

Receive timeout, Unit is ms

➤ callback

Read the data callback, the callback content is the corresponding printer state, if the data is not received in the timeout time, then the empty byte is returned.

```
public interface IDataCallback {  
    void receive(byte[] data);  
}
```

Please refer to the table below for the data [0] status returned by the corresponding type.

type	bit	function	value	decimal value
STS_T YPE_P RINT	0	fixed to 0	0	0
	1	fixed to 1	1	2
	2	One or two cash boxes open	0	0
		Both cash boxes are closed	1	4
	3	online	0	0
		Offline	1	8
	4	fixed to 1	1	16
	5,6	fixed to 0	0	0
STS_T YPE_ OFFLI NE	7	fixed to 0	0	0
	0	fixed to 0	0	0
	1	fixed to 1	1	2
	2	The upper cover is closed	0	0
		The upper cover is open	1	4
	3	Paper feed key not pressed	0	0
		Press the paper feed key	1	8
	4	fixed to 1	1	16
	5	Printer has paper	0	0
		Printer is out of paper	1	32
	6	No error conditions	0	0
		error condition	1	64
	7	fixed to 0	0	0

STS_T YPE_E RR	0	fixed to 0	0	0
	1	fixed to 1	1	2
	2	undefined	0	0
	3	Cutter without error	0	0
		Cutter error	1	8
	4	fixed to 1	1	16
	5	No unrecoverable errors	0	0
		There is an unrecoverable error	1	32
	6	Print head temperature and voltage are normal	0	0
		Print head temperature or voltage is out of range	1	64
	7	fixed to 0	0	0
STS_T YPE_P APER	0	fixed to 0	0	0
	1	fixed to 1	1	2
	2,3	enough paper	0	0
		Paper is about to run out	1	12
	4	fixed to 1	1	16
	5,6	Printer has paper	0	0
		printer is out of paper	1	96
	7	fixed to 0	0	0

2.14. printerStatusII

This method is used to query the common state of the printer, Timeout time is 3000ms, Do not use simultaneously with listenPrinterMessage.

```
void printerStatusII(IStatusCallback callback)
```

[Parameter]

➤ callback

Read the status callback.

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

When the status is less than 0, CONNECT_ERROR: Connection disconnected READ_TIMEOUT: Receiving data timed out.

When status is equal to 0, it indicates that the printer status is normal.

When the status is greater than 0, The status-values are shown in the table below.

bit	Description	value
0	Printing	1
1	The upper cover is open	2
2	Printer is out of paper	4
3	Paper near end	8

4	Cash drawer is open	16
5	Other err	32
6	Cutter err	64
7	Print head overheating	128

2.15. getPrinterInfo

Retrieve identifying information about the printer. Only supports some models. Do not use simultaneously with listenPrinterMessage.

```
void getPrinterInfo(IInfoCallback callback)
```

[Parameter]

➤ callback

Printer information callback.

```
public interface IInfoCallback {
    void receive(PrinterInfo info);
}
```

Field	Description
printerUniqueId	printer unique identifier
modelName	printer model
firmwareVersion	printer firmware version
markerName	printer manufacture name
connectionType	printer connection current mode, BLE\NET\USB
modelId	printer model id
serialNumber	printer serial number

[Return]

void

2.16. cashBoxCheck

This method is used to query the cash drawer status. Do not use simultaneously with listenPrinterMessage.

```
void cashBoxCheck(IStatusCallback callback)
```

[Parameter]

➤ callback

Read the status callback.

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

}

The status-values are shown in the table below.

STS_UNKNOWN	Unknown state, read data timeout or received data length is not 1.
STS_CASH_OPEN	Cash drawer is open.
STS_CASH_CLOSE	Cash drawer is close.

2.17. printerStatus

This method is used to query the common state of the printer, which will be called back in `listenPrinterMessage`

```
void printerStatus()
```

[Return]

void

2.18. listenPrinterMessage

Monitor printer feedback messages, including task start([RevTaskStartMo](#)), task end([RevTaskEndMo](#)), printer status changes([RevStatusMo](#)), and Printer firmware version([RevFwmVersionMo](#))

```
void listenPrinterMessage(IReceiveCallback callback)
```

[Parameter]

➤ callback

Callbacks for related state objects

```
public interface IReceiveCallback {  
    void callback(IReceive receive);  
}
```

[Return]

void

2.19. stopListenPrinterMessage

```
void stopListenPrinterMessage()
```

[Return]

void

2.20. taskStart

The function is used to inform the printer of the start of the current task

POSPrinter taskStart(long taskID)

[Parameter]

➤ taskID

Task id

[Return]

POSPrinter Instance

2.21. taskEnd

The function is used to inform the printer that the current task has ended

POSPrinter taskEnd(long taskID)

[Parameter]

➤ taskID

Task id

[Return]

POSPrinter Instance

2.22. checkFirmwareVersion

The function is used to query the firmware version of the printer, which will be called back in listenPrinterMessage

void checkFirmwareVersion()

[Return]

void

2.23. setPrintArea

Set up the print area in page mode.

POSPrinter setPrintArea(int x, int y, int width, int height)

POSPrinter setPrintArea(int width, int height)

[Parameter]

➤ x

The x-coordinate of the starting position, Default is 0.

➤ y

The y-coordinate of the starting position, Default is 0.

➤ width

Width of printing area.

➤ height

Height of printing area.

[Return]

POSPrinter Instance

2.24. setPageModel

Change to page mode or standard mode.

POSPrinter setPageModel(boolean isOpen)

[Parameter]

➤ isOpen

Enable or Disable page mode. (TRUE, FALSE)

[Return]

POSPrinter Instance

2.25. printPageModelData

Print and return to standard mode in page mode.

POSPrinter printPageModelData()

[Return]

POSPrinter Instance

2.26. setPrintDirection

Select print direction in page mode.

POSPrinter setPrintDirection(int direction)

[Parameter]

➤ direction

Print direction

Variable	Description
DIRECTION_LEFT_TOP	From top left to right
DIRECTION_LEFT_BOTTOM	From bottom left to top
DIRECTION_RIGHT_BOTTOM	From bottom right to top
DIRECTION_RIGHT_TOP	From top right to bottom

[Return]

POSPrinter Instance

2.27. setAbsoluteHorizontal

Set absolute horizontal print position . (X axis)

POSPrinter setAbsoluteHorizontal(int position)

[Parameter]

➤ position

Starting position.

[Return]

POSPrinter Instance

2.28. setRelativeHorizontal

Set relative horizontal print position. (X axis)

POSPrinter setRelativeHorizontal(int position)

[Parameter]

➤ position

Starting position.

[Return]

POSPrinter Instance

2.29. setAbsoluteVertical

Set absolute vertical print position in page mode. (Y axis)

POSPrinter setAbsoluteVertical(int position)

[Parameter]

➤ position

Starting position.

[Return]

POSPrinter Instance

2.30. setRelativeVertical

Set relative vertical print position in page mode. (Y axis)

POSPrinter setRelativeVertical(int position)

[Parameter]

➤ position

Starting position.

[Return]

POSPrinter Instance

2.31. downloadNVImage

This function is used for save the NV images in flash.

POSPrinter downloadNVImage(String imagePaths, int imageWidth)

POSPrinter downloadNVImage(String imagePaths, int imageWidth, AlgorithmType bmpType)

POSPrinter downloadNVImage(List<Bitmap> bitmaps, int imageWidth)

POSPrinter downloadNVImage(List<Bitmap> bitmaps, int imageWidth, AlgorithmType bmpType)

[Parameter]

➤ imagePaths

It sets the absolute path of the image files.

',' = separator

(Example: "/storage/emulated/0/tmp/logo1.bmp,/storage/emulated/0/tmp/logo2.bmp")

➤ bitmaps

The bitmap list

➤ imageWidth

This value is image width.

➤ bmpType

Image algorithm

Variable	Description
Threshold , None	Binary algorithm (default)
Dithering, Diffusion	Diffusion algorithm
Halftone	Halftone algorithm

[Return]

POSPrinter Instance

2.32. printNVImage

This function is used to support the Bitmap Image printing stored in Flash Memory.

POSPrinter printNVImage(int index, int model)

[Parameter]

➤ index

It sets the index image stored in Flash Memory to print,range[1,255]

➤ model

Print model

Variable	Description
BMP_NORMAL	Normal size
BMP_WIDTH_DOUBLE	Double width
BMP_HEIGHT_DOUBLE	Double height
BMP_WIDTH_HEIGHT_DOUBLE	Double size

[Return]

POSPrinter Instance

2.33. initializePrinter

Initialize Printer, This function clears the print buffer data.

POSPrinter initializePrinter()

[Return]

POSPrinter Instance

2.34. selectBitmapModel

Select bitmap model

POSPrinter selectBitmapModel(int model, int width, Bitmap bmp)

[Parameter]

➤ model

Bitmap model

Variable	Description
SINGLE_DENSITY_8	8-point single density
DOUBLE_DENSITY_8	8-point double density
SINGLE_DENSITY_24	24-point single density(76 impact printers does not support)
DOUBLE_DENSITY_24	24-point double density(76 impact printers does not support)

➤ width

Print the picture width.

➤ bmp

Bitmap image

[Return]

POSPrinter Instance

2.35. setLineSpacing

Set line-height

POSPrinter setLineSpacing(int space)

➤ space

Line-height,If you want to restore to the default height, use SPACE_DEFAULT.

[Return]

POSPrinter Instance

2.36. setTurnUpsideDownMode

Select / cancel the inverted printing mode.

POSPrinter setTurnUpsideDownMode(boolean on)

[Parameter]

➤ on

True indicates selection, false indicates cancel.

[Return]

POSPrinter Instance

2.37. setChineseModel

Set Chinese Mode

void setChineseModel(boolean on)

[Parameter]

➤ on

true means enable Chinese mode, false means disable Chinese mode.

Note:

1. Before setting the encoding for Chinese, Japanese, or Korean, you need to use this interface to switch to Chinese mode. For other encodings, you need to disable Chinese mode. You can check whether the printer is in Chinese mode by printing a self-test page.
2. After using this interface, the printer will restart.

2.38. selectCodePage

Select character code page

POSPrinter selectCodePage(int page)

[Parameter]

➤ page

Code page

Value	Description	Value	Description
0	PC437(Std.Europe)	56	PC861(Icelandic)
1	Katakana	57	PC863(Canadian)
2	PC850(Multilingual)	58	PC865(Nordic)
3	PC860(Portugal)	59	PC866(Russian)
4	PC863(Canadian)	60	PC855(Bulgarian)
5	PC865(Nordic)	61	PC857(Turkey)
6	West Europe	62	PC862(Hebrew)
7	Greek	63	PC864(Arabic)
8	Hebrew	64	PC737(Greek)
9	East Europe	65	PC851(Greek)

10	Iran	66	PC869(Greek)
16	WPC1252	67	PC928(Greek)
17	PC866(Cyrillic#2)	68	PC772(Lithuanian)
18	PC852(Latin2)	69	PC774(Lithuanian)
19	PC858	70	PC874(Thai)
20	IranII	71	WPC1252(Latin-1)
21	Latvian	72	WPC1250(Latin-2)
22	Arabic	73	WPC1251(Cyrillic)
23	PT1511251	74	PC3840(IBM-Russian)
24	PC747	75	PC3841(Gost)
25	WPC1257	76	PC3843(Polish)
27	Vietnam	77	PC3844(CS2)
28	PC864	78	PC3845(Hungarian)
29	PC1001	79	PC3846(Turkish)
30	Uigur	80	PC3847(Brazil-ABNT)
31	Hebrew	81	PC3848(Brazil)
32	WPC1255(Israel)	82	PC1001(Arabic)
255	Thai	83	PC2001(Lithuan)
33	WPC1256	84	PC3001(Estonian-1)
50	PC437(Std.Europe)	85	PC3002(Eston-2)
51	Katakana	86	PC3011(Latvian-1)
52	PC437(Std.Europe)	87	PC3012(Tatv-2)
53	PC858(Multilingual)	88	PC3021(Bulgarian)
54	PC852(Latin-2)	89	PC3041(Maltese)
55	PC860(Portuguese)	255	[Thai]
256	GB2312(Simplified Chinese)	267	BIG5(Traditional Chinese)
258	Korean	259	Japanese

[Return]

POSPrinter Instance

Note:

1. After setting the printer encoding, you also need to use the [setCharSet](#) interface to set the corresponding host machine encoding to ensure that the data sent can be correctly parsed by the printer.
2. This interface only temporarily modifies the printer's code page and will be invalid after a power-off.

2.39. selectDefaultCodePage

Set the Default Code Page (persistent after power-off)

POSPrinter selectDefaultCodePage(int page)

[Parameter]

➤ page

Same as in the [selectCodePage](#) method.

[Return]

POSPrinter Instance

2.40. selectCharacterFont

Select font

POSPrinter selectCharacterFont(int font)

[Parameter]

➤ font

Font type

Variable	Description
FONT_STANDARD	Standard ascii font (12 × 24)
FONT_COMPRESS	Compress ASCII font (9 × 17)

[Return]

POSPrinter Instance

2.41. setCharRightSpace

Set the right spacing of characters

POSPrinter setCharRightSpace(byte space)

[Parameter]

➤ space

Right spacing distance is space*hor_motion_unit

[Return]

POSPrinter Instance

2.42. printPDF417

This method is used for supporting PDF417 barcode printing.

POSPrinter printPDF417(String pdfData)

POSPrinter printPDF417(String pdfData, int cellWidth, int cellHeightRatio, int numberOfColumns,

int numberOfRows, int eclType, int eclValue, int alignment)

➤ pdfData

Barcode data to print.

➤ cellWidth

Cell width.

Range[2 – 8].

➤ cellHeightRatio

Cell height ratio. [Cell height = [cellHeightRatio × cellWidth].

Range[2 – 8].

cell height = cellHeightRatio x cellWidth

➤ numberOfColumns

Set the number of columns.

Range : [0 – 30].

➤ numberOfRows

Set the number of columns.

Range : [0 – 30].

➤ eclType

Set the error correction level.

Range[0 – 1].

0= The error correction level is set by "level"

1= The error correction level is set by "ratio."The ratio is [eclValue × 10%].

➤ eclValue

- Set the error correction level.

eclType = 0 : Range[0 – 8].

eclType = 1 : Range[1 – 40].

➤ alignment

This value is alignment. It sets barcode alignment.

Variable	Description
ALIGNMENT_LEFT	Align left
ALIGNMENT_CENTER	Align center
ALIGNMENT_RIGHT	Align right

[Return]

POSPrinter Instance

2.43. sendData

This function is used to send data to the printer.

POSPrinter sendData(byte[] data);

POSPrinter sendData(List<byte[]> datas);

void sendData(byte[] datas, IStatusCallback callback);

[Parameter]

➤ data

Byte array to be sent

➤ datas

Byte array collection to be sent

➤ callback

The callback will call back the send status, 1 indicates successful sending, 0 indicates failed sending

[Return]

POSPrinter Instance

2.44. setTransaction

The function's purpose is to set the transaction status. When a transaction is opened, it stores the data being sent until the transactionPrint function is called, at which point all cached data will be sent to the printer. When the transaction is closed, if there is any cached transaction data, it will be sent to the printer. Only asynchronous communication is supported.

`void setTransaction(boolean isOpen);`

[Parameter]

➤ isOpen

Whether to open a transaction

[Return]

void

2.45. transactionPrint

Send data from the transaction cache. Do not use simultaneously with listenPrinterMessage.

`void transactionPrint(IStatusCallback callback);`

[Parameter]

➤ callback

The callback will call back the send status, 1 indicates successful sending, 0 indicates failed sending

[Return]

void

2.46. clearTransaction

Clear the data from the transaction cache.

```
void clearTransaction();
```

[Return]

void

2.47. printTable

Print table

```
POSPrinter printTable(PTable table)
```

[Parameter]

➤ table

Table objects to be printed, See [PTable](#)

[Return]

POSPrinter Instance

2.48. wifiConfig

Set up printer Wi-Fi

```
void wifiConfig(byte[] ip, byte[] mask, byte[] gateway, String ssid, String password, byte encrypt)
```

[Parameter]

➤ ip

ip address, a byte array of length 4.eg:new byte[] {(byte)192,(byte)168,1,100}

➤ mask

Subnet mask, byte array of length 4.

➤ gateway

default gateway, byte array of length 4.

➤ ssid

Wi-Fi name, cannot be empty.

➤ password

Wi-Fi password.

➤ encrypt

encryption type.

Variable	Description
ENCRYPT_NULL	NULL

ENCRYPT_WEP64	WEP64
ENCRYPT_WEP128	WEP128
ENCRYPT_WPA_AES_PSK	WPA_AES_PSK
ENCRYPT_WPA_TKIP_PSK	WPA_TKIP_PSK
ENCRYPT_WPA_TKIP_AES_PSK	WPA_TKIP_AES_PSK
ENCRYPT_WPA2_AES_PSK	WPA2_AES_PSK
ENCRYPT_WPA2_TKIP	WPA2_TKIP
ENCRYPT_WPA2_TKIP_AES_PSK	WPA2_TKIP_AES_PSK
ENCRYPT_WPA_WPA2_MixedMode	WPA_WPA2_MixedMode

[Return]

void

2.49. setIp

set network ip address.

void setIp(byte[] ip)

[Parameter]

➤ ip

ip address, a byte array of length 4.eg:new byte[] {(byte)192,(byte)168,1,100}

[Return]

void

2.50. setMask

set subnet mask.

void setMask(byte[] mask)

[Parameter]

➤ mask

Subnet mask, byte array of length 4.

[Return]

void

2.51. setGateway

set default gateway

`void setGateway(byte[] gateway)`

[Parameter]

➤ gateway

Default gateway, a byte array of length 4.

[Return]

void

2.52. setNetAll

Set up network configuration

`void setNetAll(byte[] ip, byte[] mask, byte[] gateway, boolean dhcpIsOpen)`

[Parameter]

➤ ip

ip address, a byte array of length 4. eg: new byte[] {(byte)192, (byte)168, 1, 100}

➤ mask

Subnet mask, byte array of length 4.

➤ gateway

default gateway, byte array of length 4.

➤ dhcpIsOpen

Set whether to enable DHCP. true to open false to close.

[Return]

void

2.53. setBluetooth

Set Bluetooth information

`void setBluetooth(String name, String pin)`

[Parameter]

➤ name

bluetooth name

➤ pin

bluetooth pin code

[Return]

void

2.54. setDensity

Set print density

`void setDensity(byte level)`

[Parameter]

➤ level

Density, Range[1, 8]

[Return]

void

2.55. getSerialNumber

Obtain the serial number of the printer, Do not use simultaneously with `listenPrinterMessage`.

`void getSerialNumber(IDataCallback callback)`

[Parameter]

➤ callback

Get the SN code queried by callback.

```
public interface IDataCallback {  
    void receive(byte[] data);  
}
```

[Return]

void

2.56. awaitSendCompletionAndDisconnect

Disconnect the connection after the data transmission is completed. Do not use simultaneously with `listenPrinterMessage`.

`void awaitSendCompletionAndDisconnect()`

[Return]

void

3. PTable

For table printing, The column width of the table is calculated by taking single byte characters as a unit, for example, the width of the letter 'a' is 1, and the width of the Chinese '印' is 2.

If a new line is needed, '\n' can be added to the text. If the text width is greater than the set column width, the line will wrap automatically.

3.1. PTable

Constructor

`PTable(String[] titles, Integer[] numberOfSingleBytesPerCol)`

`PTable(String[] titles, Integer[] numberOfSingleBytesPerCol, Integer[] align)`

[Parameter]

➤ titles

Header array collection

➤ numberOfSingleBytesPerCol

Set of single byte characters in each column

➤ align

The alignment of each column, 0 is left-aligned and 1 is right-aligned. Default is 0.

[Return]

PTable Instance

3.2. addRow

add rows

`PTable addRow(String... row)`

`PTable addRow(String title, String[] row)`

`PTable addRow(String title, String[] row, String remark)`

[Parameter]

➤ title

The title of each row. If it is empty, there will be no title. The default is empty.

➤ row

Character set of the line

➤ remark

The comments after each line, if empty, there will be no comments, and the default is empty.

Item	QTY	Price	Total
Apple Apple xxxxxxxxxxxx	1	7.99	7.99
100328	1	7.99	7.99
remarks:xxxxxxx	4	0.99	3.96
680015	1	43.99	43.99
102501102501	1	43.99	43.99
102501	1	4.99	4.99
021048	1	4.99	4.99

[Return]

PTable Instance

4. TableBarcode

Barcode auxiliary class in table row attributes.

4.1. TableBarcode

Constructor

TableBarcode(String data, int codeType)

TableBarcode(String data, int codeType, int height)

TableBarcode(String data, int codeType, int width, int height, int alignment)

TableBarcode(String data, int codeType, int width, int height, int alignment, int textPosition)

[Parameter]

➤ data

Barcode string to be printed

➤ codeType

Barcode type

Variable	Description
BCS_UPCA	UPC A
BCS_UPCE	UPCE
BCS_EAN8	EAN-8
BCS_EAN13	EAN-13
BCS_JAN8	JAN-8
BCS_JAN13	JAN-13
BCS_ITF	ITF
BCS_Codabar	Codabar

BCS_Code39	Code 39
BCS_Code93	Code 93
BCS_Code128	Code 128, For this type, the data must be added with {A, {B, {C, etc

➤ height

Barcode height, range [1,255].Default is 80

➤ width

This values barcode width in Dot Units, range [2, 6], Default is 2

➤ alignment

It sets barcode alignment, Default is ALIGNMENT_LEFT

Variable	Description
ALIGNMENT_LEFT	Align left
ALIGNMENT_CENTER	Align center
ALIGNMENT_RIGHT	Align right

➤ textPosition

This value is printing position of barcode HRI letters(barcode data).Default is HRI_TEXT_BELOW.

Variable	Description
HRI_TEXT_NONE	Do not print barcode data
HRI_TEXT_ABOVE	Print barcode data above the barcode
HRI_TEXT_BELOW	Print barcode data below the barcode
HRI_TEXT_BOTH	Print barcode data top and bottom

[Return]

POSPrinter TableBarcode

5. PosUdpNet

POS printer Udp message sending and receiving class, through which the printer device connected to the network port in the LAN can be realized and the network information can be modified.

5.1. searchNetDevice

Search for printing devices in the local area network

void searchNetDevice(UdpCallback callback)

Continuously searches for printers on the same network segment as the mobile device

void searchNetDeviceLoop(int interval, UdpCallback callback)

[Parameter]

➤ interval

Search interval time, in milliseconds (ms)

➤ callback

Returns the found device information by way of callback.

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

[Return]

void

5.2. stopSearchLoop

Stops the continuous network printer search loop

```
void stopSearchLoop()
```

[Return]

void

5.3. udpNetConfig

Modify the network port information of the printing device through UDP

```
static void udpNetConfig(byte[] macAddress, byte[] ipAddress, byte[] mask, byte[] gateway,  
boolean dhcp)
```

[Parameter]

➤ macAddress

MAC address of the device

➤ ipAddress

Ip address

➤ mask

subnet mask

➤ gateway

default gateway

➤ dhcp

Whether to enable dhcp

[Return]

void

6. PosFirmwareUpdate

PosFirmwareUpdate is used for upgrading the firmware of the receipt printer

Note: Please do not power off the printer before the update and restart are completed

6.1. PosFirmwareUpdate

PosFirmwareUpdate(IDeviceConnection connect, String filePath, IStatusCallback progress)

[Parameter]

➤ connect

Connected object

➤ filePath

The file path of the firmware

➤ progress

Update progress callback.

Status	Description
UPDATE_FAIL	Update fail
UPDATE_SUCCESS	Update success
UPDATE_FMW_ERROR	The firmware does not match the printer
>0	Update the actual progress

[Return]

void

6.2. getGd427PrinterVersion

Get printer version number. **Note:** only supports GD427 model main control chip

void getGd427PrinterVersion(IDeviceConnection connect, IStrCallback callback)

[Parameter]

➤ callback

Callback version number, if there is an error, callback an empty string

[Return]

void

6.3. startUpdate

Start updating printer firmware

`void startUpdate()`

[Return]

void

7. UdpDevice

Found network printer information object

7.1. isIPv6

Is the device an IPv6 address printer. Otherwise, it is an IPv4 address printer

7.2. getMacAddress

The MAC address of the printer

`byte[] getMacAddress()`

`String getMacStr()`

[Return]

Mac Address

7.3. getIPv6UserStr

IPv6 user address. Please use it when isIPv6 is true.

`String getIPv6UserStr()` throws `UnknownHostException`

[Return]

IPv6 user address

7.4. getIPv6LocalStr

The local address of IPv6. Please use it when isIPv6 is true.

String getIpv6LocalStr() throws UnknownHostException

[Return]

The local address of IPv6.

7.5. **getIpV4Str**

IPv4 address. Please use it when isIpV6 is false.

String getIpV4Str()

[Return]

ipv4 address。

7.6. **getMaskStr**

Subnet mask for IPv4 printer. Please use it when isIpV6 is false.

String getMaskStr()

[Return]

Subnet mask.

7.7. **getGatewayStr**

The default gateway for IPv4 devices. Please use it when isIpV6 is false.

String getGatewayStr()

[Return]

Gateway

7.8. **isDhcp**

Is it DHCP

boolean isDhcp()

[Return]

dhcp

7.9. getIpStr

IP address. If it is IPv6, return the IPv6 Local address; otherwise, return the IPv4 address
String getIpStr() throws UnknownHostException

[Return]
Ip address

7.10. getModelName

Get printer model, only supports some models. Unsupported devices return an empty string.

String getModelName()

[Return]
printer model name

8. RevFwmVersionMo

When calling the checkFirmwareVersion function, the printer firmware version number object is returned through listenPrinterMessage.

[attribute]

- String connectInfo
Connection information

- String version

The current firmware version of the printer

9. RevStatusMo

The printer status object returns through listenPrinterMessage when the printer status changes or when the printerStatus function is called.

[attribute]

- boolean isNormal

If the value is true, the printer status is normal

➤ boolean isCashBoxOpen

If the value is true, the printer cash box is turned on

➤ boolean isCoverOpen

If the value is true, the printer cover is open

➤ boolean isCutterError

If the value is true, the printer cutter is incorrect

➤ boolean isPaperNearEmpty

If the value is true, the printer is close to running out of paper,Applicable to some models

➤ boolean isOutOfPaper

If the value is true, the printer has run out of paper

➤ boolean isOtherError

If the value is true, the printer has other errors

➤ boolean isHeadOverheat

If the value is true, the print head temperature is too high

10. RevTaskEndMo

When the printer finishes the current task, it returns through listenPrinterMessage and is used in conjunction with the taskEnd function

[attribute]

➤ long taskId

Current task ID

➤ String connectInfo

Connection information

11. RevTaskStartMo

When the printer starts the current task, it returns through listenPrinterMessage and is used in conjunction with the taskStart function

[attribute]

➤ long taskId

Current task ID

➤ String connectInfo

Connection information

12. SendFailMo

When an exception occurs during message transmission, it is returned via listenPrinterMessage

[attribute]

➤ String errMsg

Exception message

➤ String connectInfo

Connection information

13. ConnectInterruptMo

When the connection is interrupted, return through listenprinterMessage

[attribute]

➤ String errMsg

Exception message

➤ String connectInfo

Connection information