Printer interface introduction

# 1 Overview

1 NETPrinting and BLEPrinting are from IO。

They have their own way to open and close function, and call back interface.

Meanwhile re-write 3 important function of IO.

- (bool) IsOpened;
- (int) Write:(Byte * ) buffer offset:(int) offset count:(int) count;
- (int) Read:(Byte *)buffer offset:(int)offset count:(int)count timeout:(int)timeout;

2 POSPrinting and LabelPrinting encapsulate print command and have an IO interface.

POSPrinting encapsulates ESC/POS command

LabelPrinting encapsulates label print command

They organize data according to command set, then call IO to Write function to write data and call Read function to read data.

3 MemoryIO

MemoryIO can save data to memory and use GetBuffer to retrive it. So you can use this data for other purpose.

# 2 Usage

Instantiate an IO subclass before use. Make an example as Bluetooth printer:

1 Instantiate an POSPrinting (Set variable name as pos to describe easily), call POS_XXX series function directly this time, you will realize it cannot print as POSPrinting doesn't have a readable IO at the moment. You need to continue the steps as below.
2 Instantiate a BLEPrinting (Set variable name as ble to describe easily), then call Open function to connect to Bluetooth printer.
3 Call SetIO of pos when it connects successfully, it let pos have ble. So that the subsequent POS_XXX series function will communicate with printer via Write and Read of ble.

# 3 Differences of each categories

## The differences of BLEPrinting, NETPrinting and IO

A) IO is father type, which offers uniform interface to use for POSPrinting and LabelPrinting.

B) BLEPrinting is for Bluetooth communication, except for Open Close IsOpened Read Write, and

scan scans Bluetooth printer. It will call back interface BLEPrintingDiscoverDelegate if it scans successfully

stopScan stopping scan

BLEPrintingOpenDelegate Open Call back the interface after succeed

BLEPrintingDiscoverDelegate Call back the interface after scanning printer

BLEPrintingReceiveDelegate Call back the interface after receiving data

BLEPrintingDisconnectDelegate Call back the interface after interrupting Bluetooth (It won't call if call Close many times)

C) NETPrinting is for internet communication, except for the basic Open Close IsOpened Read Write and

NETPrintingOpenDelegate Open Call back the interface after succeed

NETPrintingDisconnectDelegate Call back the interface after interrupting connection( it wont call if call Close many times)

## Differences for POSPrinting and LabelPrinting

A) POSPrinting packs portable command set,

B) LabelPrinting packs label command set.

Normal thermal printer( only support ESC/POS command set)

It only uses POSPrinting to control printer to print.

Label printer( support ESC/POS command set, label command set）

It uses POSPrinting to control printer to print, and also uses LabelPrinting to control printer to print.

Two sets of command can not intermixed use, which means:

When it uses LabelPrinting to control print, between PageBegin and PagePrint, cannot intermix POSPrinting function.

# 4 Interface introduction

## POSPrinting

Normal printing

### POS_PrintText

Description: printing text

**Syntax**

- (bool) POS_PrintText:(char *)pszString x:(int)x nWidthTimes:(int)nWidthTimes nHeightTimes:(int)nHeightTimes nFontType:(int)nFontType nFontStyle:(int)nFontStyle

**Parameters**

pszString

content need to print。utf8 code strings

x

specify number of dots from starting area of horizontal direction to left margin of printing area. （horizontal axis）

support left center right alignment

incoming x see table

| x | implication |
|---|---|
| -1 | Left alignment |
| -2 | Center alignment |
| -3 | Right alignment |
| Great than or equal to 0 | Horizontal direction |

nWidthScale

specified width magnification [0,7]

nHeightScale

specified height magnification [0,7]

nFontType

font type

0 standard font

1 compressed font

nFontStyle

specify font styles, can be one or more of following table,(add together)

| Value | Meaning |
|---|---|
| 0x00 | normal |
| 0x08 | Bold |
| 0x80 | 1 dot bold underline |
| 0x100 | 2 dot bold underline |
| 0x200 | invert（only valid at the head of first line） |
| 0x400 | reverse（black and white） |
| 0x1000 | rotate each character 90 degrees clockwise |

**Return value**

Return true if write success, else write failed.

**Remarks**

POS_PrintText will not print rightnow(if content not fill full line).

callPOS_FeedXXX series function will print buffer right now.

don’t support muti alignment at a line.

# POS_PrintBarcode

Print barcode

**Syntax**

- (bool) POS_PrintBarcode:(char *)pszString x:(int)x nType:(int)nType nUnitWidth:(int)nUnitWidth nHeight:(int)nHeight nHriFontType:(int)nHriFontType nHriFontPosition:(int)nHriFontPosition

**Parameters**

pszString

barcode content

It needs the barcode character sets

Some barcodes has format requirements, pls print barcodes as rules.

x

specify number of dots from starting area of horizontal direction to left margin of printing area. （horizontal axis）

support left center right alignment

incoming x see table

| x | implication |
|---|---|
| -1 | Left alignment |
| -2 | Center alignment |
| -3 | Right alignment |
| Great than or equal to 0 | Horizontal direction |

nType

Assign barcode type

It can be any value as below chart.

| Value | Meaning |
|---|---|
| 0x41 | UPC-A |
| 0x42 | UPC-C |
| 0x43 | JAN13(EAN13) |
| 0x44 | JAN8(EAN8) |
| 0x45 | CODE39 |
| 0x46 | ITF |
| 0x47 | CODEBAR |
| 0x48 | CODE93 |
| 0x49 | CODE 128 |

nUnitWidth

It assigns the basic element width.

It can be any value(n) in the chart as below

| n | Single basic module width(succe ssive type) | Double basic module width(discrete type) | |
|---|---|---|---|
| | | Narrow element width | Wide element width |
| 2 | 0．25mm | 0．25mm | 0．625mm |
| 3 | 0．375mm | 0．375mm | 1．0mm |
| 4 | 0．5mm | 0．5mm | 1．25mm |
| 5 | 0．625mm | 0．625mm | 1．625mm |

nHeight

Barcode height

nHriFontType

It assigns HRI（Human Readable Interpretation）character font types.

It can be any value (n) of the chart as below.

| Value | Meaning |
|---|---|
| 0x00 | standard ASCII |
| 0x01 | compressed ASCII |

nHriFontPosition

It assigns HRI（Human Readable Interpretation）character position

It can be any value (n) of the chart as below.

| Value | Meaning |
|---|---|
| 0x00 | Doesn’t print |
| 0x01 | Only print in upward side of barcode |
| 0x02 | Only print in downward side of barcode |
| 0x03 | Print both in upward side and downward |

**Return value**

Return true if write success, else write failed.

**Remarks**

Some models do not support the specified starting position, please use the left, center alignment, right-aligned typesetting layout

# POS_PrintQRCode

Print QR Code

**Syntax**

- (bool) POS_PrintQRcode:(char *)pszString x:(int)x nUnitWidth:(int)nUnitWidth nVersion:(int)nVersion nECCLevel:(int)nECCLevel

**Parameters**

pszString

QR Code Content

x

specify number of dots from starting area of horizontal direction to left margin of printing area. （horizontal axis）

support left center right alignment

incoming x see table

| x | implication |
|---|---|
| -1 | Left alignment |
| -2 | Center alignment |
| -3 | Right alignment |
| Great than or equal to 0 | Horizontal direction |

nUnitWidth

QR code unit width, the range is [1,16].

QR code unit is wider, QR code is bigger.

nVersion

QR code version. 0 means to calculate version automatically.

QR code version is bigger, the edited characters are more, and the QR code is bigger.

nECCLevel

QR code error correction level.[1,4]

**Return value**

Return true if write success, else write failed.

**Remarks**

Some models do not support the specified starting position, please use the left, center alignment, right-aligned typesetting layout

# POS_PrintPicture

Print picture

**Syntax**

- (bool) POS_PrintPicture:(UIImage *)mImage x:(int)x nWidth:(int)nWidth nHeight:(int)nHeight nBinaryAlgorithm:(int)nBinaryAlgorithm nCompressMethod:(int)nCompressMethod

**Parameters**

mImage

the image to be printed

x

specify number of dots from starting area of horizontal direction to left margin of printing area. （horizontal axis）

support left center right alignment

incoming x see table

| x | implication |
|---|---|
| -1 | Left alignment |
| -2 | Center alignment |
| -3 | Right alignment |
| Great than or equal to 0 | Horizontal direction |

nWidth

The printed width

nHeight

The printed height

nBinaryAlgorithm

algorithm used to convert a color bitmap to binary bitmap

0 Use dithering algorithm for color images have better effect

1 Using the average threshold algorithm, and has better effect on text class picture

nCompressMethod

compress method

0 don’t compress data

1 compress data

**Return value**

Return true if write success, else write failed.

**Remarks**

Some models do not support the specified starting position, please use the left, center

alignment, right-aligned typesetting layout

# POS_FeedLine

**Syntax**

- (bool) POS_FeedLine

**Parameters**

**Return value**

Return true if write success, else write failed.

**Remarks**

```
Feed paper one line
```

# POS_Reset

Reset printer(software resets)

**Syntax**

- (bool) POS_Reset

**Parameters**

**Return value**

Return true if write success, else write failed.

**Remarks**

# POS_SetRightSpacing

Set the characters right blank

**Syntax**

- (bool) POS_SetRightSpacing:(int)nDistance

**Parameters**

nDistance

characters right blank

**Return value**

Return true if write success, else write failed.

**Remarks**

# POS_SetLineHeight

Set line height

**Syntax**

- (bool) POS_SetLineHeight:(int)nHeight

**Parameters**

nHeight

line height

**Return value**

Return true if write success, else write failed.

**Remarks**

# POS_SetPrintSpeed

Set the print speed note: if the print speed is greater than the speed, print will fill caton

**Syntax**

- (bool) POS_SetPrintSpeed:(int)nSpeed

**Parameters**

`nSpeed`

```
print speed (mm/s)
```

**Return value**

Return true if write success, else write failed.

**Remarks**

Set the print speed for data sending speed, can be to attain the best printing effect.

# POS_CutPaper

cut paper

**Syntax**

- (bool) POS_CutPaper

**Parameters**

**Return value**

Return true if write success, else write failed.

**Remarks**

only for cutter printer

# POS_Beep

```
Beep
```

**Syntax**

- (bool) POS_Beep:(int)nBeepCount nBeepMillis:(int)nBeepMillis

**Parameters**

`nBeepCount`

```
    Beep count
```

`nBeepMillis`

```
    beep ms = 100 * nBeemMillis ms
```

**Return value**

Return true if write success, else write failed.

**Remarks**

# POS_KickDrawer

Kick Drawer

**Syntax**

- (bool) POS_KickDrawer:(int)nDrawerIndex nPulseTime:(int)nPulseTime

**Parameters**

nDrawerIndex

    0 meaning: Pulse is sent to the output pin 2
    1 meaning: Pulse is sent to the output pin 5

nPulseTime

    Pulse Time
    High ms: nPulseTime*2ms
    Low ms:  nPulseTime*2ms

**Return value**

Return true if write success, else write failed.

**Remarks**

# POS_QueryStatus

Query status
This command is blocked all the time when printer is busy.
The return status stores in status

**Syntax**

- (bool) POS_QueryStatus:(int)type status:(Byte *)status timeout:(int)timeout MaxRetry:(int)MaxRetry

**Parameters**

type
    fixed value: 1

status
    status currently no meaning

timeout
    timeout ms per query

MaxRetry
    retry count if failed

**Return value**

It returns TRUE to show printer status is OK. Or printer is not connected or printer is busy.

**Remarks**

# POS_RTQueryStatus

Real-time query status
No matter under any status of the printer, it returns the status when it receives this command.
The return status stores in status

**Syntax**

- (bool) POS_RTQueryStatus:(int)type status:(Byte *)status timeout:(int)timeout MaxRetry:(int)MaxRetry

**Parameters**

type

type can be [1,4]

1：Printer status

| Bit | 0/1 | Hex. | Dec. | Function |
|---|---|---|---|---|
| 0 | 0 | 00 | 0 | Fixed is 0 |
| 1 | 1 | 02 | 2 | Fixed is 1 |
| 2 | 0 | 00 | 0 | One or two cashboxes are open(Fixed is 0 if this machine is without cashbox) |
| | 1 | 04 | 4 | Two cashboxes are closed |
| 3 | 0 | 00 | 0 | Online |
| | 1 | 08 | 8 | Offline |
| 4 | 1 | 10 | 16 | Fixed is 1 |
| 5, 6 | | -- | -- | Undefined |
| 7 | 0 | 00 | 00 | Paper has been tore off |
| | 1 | 80 | 96 | Paper hasn’t been tore off |

2：Transmit offline status

| Bit | 0/1 | Hex. | Dec. | Function |
|---|---|---|---|---|
| 0 | 0 | 00 | 0 | Fixed is 0 |
| 1 | 1 | 02 | 2 | Fixed is 1 |
| 2 | 0 | 00 | 0 | Upper cover is closed |
| | 1 | 04 | 4 | Upper cover is open |
| 3 | 0 | 00 | 0 | Paper feed key is un-pressed |
| | 1 | 08 | 8 | Paper feed key is pressed |
| 4 | 1 | 10 | 16 | Fixed is 1 |
| 5 | 0 | 00 | 0 | Paper is not out |

| | | | | |
|---|---|---|---|---|
| | 1 | 20 | 32 | Paper out |
| 6 | 0 | 00 | 00 | No error |
| | 1 | 40 | 64 | Error |
| 7 | 0 | 00 | 0 | Fixed is 0 |

3：Transmit error status

| Bit | 0/1 | Hex. | Dec. | Function |
|---|---|---|---|---|
| 0 | 0 | 00 | 0 | Fixed is 0 |
| 1 | 1 | 02 | 2 | Fixed is 1 |
| 2 | | -- | -- | Undefined |
| 3 | 0 | 00 | 0 | Cutter has no error |
| | 1 | 08 | 8 | Cutter has error |
| 4 | 1 | 10 | 16 | Fixed is 1 |
| 5 | 0 | 00 | 0 | No recoverable error |
| | 1 | 20 | 32 | Has recoverable error |
| 6 | 0 | 00 | 00 | Printer head temp. and voltage are normal |
| | 1 | 40 | 64 | Printer head temp. or voltage is out of range |
| 7 | 0 | 00 | 0 | Fixed is 0 |

4：Transmit paper sensor status

| Bit | 0/1 | Hex. | Dec | Function |
|---|---|---|---|---|
| 0 | 0 | 00 | 0 | Fixed is 0 |
| 1 | 1 | 02 | 2 | Fixed is 1 |
| 2，3 | 0 | 00 | 0 | With paper |
| | 1 | 0C | 12 | Paper will be out |
| 4 | 1 | 10 | 16 | Fixed is 1 |
| 5，6 | 0 | 00 | 0 | With paper |
| | 1 | 60 | 96 | Paper out |
| 7 | 0 | 00 | 0 | Fixed is 0 |

```
status
    status meaning see above table
timeout
    timeout ms per query
MaxRetry
    retry count if failed
```

**Return value**

It returns TRUE to show printer connection is OK. status stored in status[].

**Remarks**

# POS_TicketSucceed

Description: The printing results of the documents can be used to identify the incomplete printing of documents due to lack of paper, overheating or other errors

**Syntax**

- (int) POS_TicketSucceed:(int)dwSendIndex timeout:(int)timeout;

**Parameters**

dwSendIndex

Document index, fill 0 or increments

timeout

Timeout milliseconds, can be set according to the documents situation Normal need 3 s time to print a document, for example, so here can be up to wait 10 s, to determine whether documents printed finish parameters can fill in 10000

**Return value**

Returns 0, indicating the success of the document printing

Returns -1, indicating the failure of the document query instruction failure: the connection is disconnected or closed

Returns -2, indicating the failed failure of the document query instruction: the sending failed

Returns -3, indicating the failure of the document query instruction failure: the printer does not respond

Return -4, indicating the failure of the document printing: the printer is offline

Return -5, indicating that the document printing is not complete: the printer is interrupted to print due to lack of paper

Return - 6 to express the failure of the document printing: other reasons

**Remarks**

Some models do not support document query instructions

# LabelPrinting

Label Printing

## PageBegin

Description: assign the start of a Page, and set Page size, reference point coordinates and page rotating angle.
Parameters

**Syntax**

```
- (bool) PageBegin:(int)startx starty:(int)starty width:(int)width height:(int)height rotate:(int)rotate
```

**Parameters**

`startx`

page start point x coordinates

`starty`

page start point y coordinates

`width`

page width

startx + width range is [1,384]. The printer is 384 dots/line when editing SDK. Pls refer to printer specification if you are not sure the dots number per line.Normally there are three specification 384,576,832.

`height`

page height

starty + height range is[1,936]. The limited is 936 when editing SDK. But this value is not sure, which is about the printer resource. Even though, we don't suggest to set page width too wide. We suggest to set page width and height to be fit for label paper.

`rotate`

page rotating. The value range of rotate is {0,1}. Page doesn't rotate to print as 0, and rotate 90° to print as 1.

**Return value**

Return true if write success, else write failed.

**Remarks**

# PageEnd

```
Description: It is only the end of a Page
```

**Syntax**
- (bool) PageEnd

**Parameters**

**Return value**
Return true if write success, else write failed.
**Remarks**

# PagePrint

Description: print the Page contents to label paper

**Syntax**

- (bool) PagePrint:(int)num

**Parameters**

num

printing times,1-255。

**Return value**

Return true if write success, else write failed.

**Remarks**

## DrawPlainText

Description: draw text in assigned position of Page.
only for single line

**Syntax**

- (bool) DrawPlainText:(int)startx starty:(int)starty font:(int)font style:(int)style str:(char *)str

**Parameters**

startx

define text start position x coordinates,the value range is: [0, Page_Width-1]

starty

define text start position y coordinates,the value range is: [0, Page_Height-1]

font

Choose font, the efficient value range is {16, 24, 32, 48, 64, 80, 96},this printer only can use 24.

style

chracter style.

| Databits | define |
|---|---|
| 0 Bold flag bit: | font bold for 1,don't bold if reset zero clearing. |
| 1 underline flag bit: | underline text for 1, don't underline if rest zero clearing |
| 2 inverse flag bit: | inverse for 1(white in black), don't inverse rest zero clearing |
| 3 delete line flage bit: | for 1 text with delete line,don't delete line if reset zero clearing. |
| [5,4] rotate flag bit: | 00 rotates 0° |
| | 01 rotates 90° |
| | 10 rotates 180° |
| | 11 rotates 270° |
| [11,8] font width magnification times; | |
| [15,12] font height magnification times; | |

`str`

`character set data flow`

**Return value**

Return true if write success, else write failed.

**Remarks**

# DrawLine

Description: Draw a straightway in the assigned position of Page

**Syntax**

- (bool) DrawLine:(int)startx starty:(int)starty endx:(int)endx endy:(int)endy width:(int)width color:(int)color

**Parameters**

startx

straightway start point x coordinates,the value range is: [0, Page_Width-1]

starty

straightway start point y coordinates,the value range is: [0 , Page_Height-1]

endx

straightway end point x coordinates,the value range is: [0, Page_Width-1]

endy

straightway end point y coordinates,the value range is: [0,Page_Height-1]

width

straightway width,the value range is: [1, Page_Height-1]

color

straightway color,
the value range is: {0, 1}.
When Color is 1,straightway is black.
When Color is 0,straightway is white.

**Return value**

Return true if write success, else write failed.

**Remarks**

# DrawBox

Description: draw assigned size rectangular frame in the fixed position of Page

**Syntax**

- (bool) DrawBox:(int)left top:(int)top right:(int)right bottom:(int)bottom borderwidth:(int)borderwidth bordercolor:(int)bordercolor

**Parameters**

left

rectangular frame top left corner x coordinates, the value range is: [0, Page_Width-1]

top

rectangular frame top left corner y coordinates, the value range is: [0, Page_Height-1]

right

rectangular frame top right corner x coordinates, the value range is: [0, Page_Width-1]

bottom

rectangular frame top right corner y coordinates, the value range is: [0, Page_Height-1]

borderwidth

rectangular line width

bordercolor

rectangular line color,
range is {0, 1}.
When Color = 1 ,draw black rectangular ,
when Color = 0 ,draw white rectangular

**Return value**

Return true if write success, else write failed.

**Remarks**

# DrawRectangel

Description: draw rectangular frame in assigned position of Page

**Syntax**

- (bool) DrawRectangel:(int)left top:(int)top right:(int)right bottom:(int)bottom color:(int)color

**Parameters**

left

rectangular block top left corner x coordinates, the value range is:[0, Page_Width-1]

top

rectangular block top left corner y coordinates, the value range is: [0, Page_Height-1]

right

rectangular block top right corner x coordinates, the value range is: [0, Page_Width-1]

bottom

rectangular block top right corner y coordinates, the value range is: [0, Page_Height-1]

color

rectangular block color,
the value range is: {0, 1}.
When color is 1, rectangular block is black.
When Color is 0, rectangular block is white.

**Return value**

Return true if write success, else write failed.

**Remarks**

# DrawBarcode

Description: Draw 1D code in the assigned position of Page

**Syntax**

- (bool) DrawBarcode:(int)startx starty:(int)starty type:(int)type height:(int)height unitwidth:(int)unitwidth rotate:(int)rotate str:(char *)str

**Parameters**

startx

barcode top left corner x coordinates, the value range is: [0, Page_Width-1]

starty

barcode top left corner y coordinates, the value range is: [0, Page_Height-1]

type

indetify barcode type, the value range is: [0, 29]. Definitions are as below:

| type | name | length | barcode value range(Dec.) |
|---|---|---|---|
| 0 | UPC-A | 11 | 48-57 |
| 1 | UPC-E | 6 | 48-57 |
| 2 | EAN13 | 12 | 48-57 |
| 3 | EAN8 | 7 | 48-57 |
| 4 | CODE39 | 1- | 48-57,65-90,32,36,37,43,45,46,47 |
| 5 | I25 | 1-even number | 48-57 |
| 6 | CODABAR | 1- | 48-57,65-68,36,43,45,46,47,58 |
| 7 | CODE93 | 1-255 | 0-127 |
| 8 | CODE128 | 2-255 | 0-127 |
| 9 | CODE11 | | |
| 10 | MSI | | |
| 11 | 128M | | |
| 12 | EAN128 | | |
| 13 | 25C | | |
| 14 | 39C | | |
| 15 | 39 | | |
| 16 | EAN13+2 | | |
| 17 | EAN13+5 | | |

| | | | |
|---|---|---|---|
| 18 | EAN8+2 | | |
| 19 | EAN8+5 | | |
| 20 | POST | | |
| 21 | UPCA+2 | | |
| 22 | UPCA+5 | | |
| 23 | UPCE+2 | | |
| 24 | UPCE+5 | | |
| 25 | CPOST | | |
| 26 | MSIC | | |
| 27 | PLESSEY | | |
| 28 | ITF14 | | |
| 29 | EAN14 | | |

height

define barcode height

unitwidth

It assigns the basic element width.

It can be any value(n) in the chart as below

| n | Single basic module width(successive type) | Double basic module width(discrete type) | |
|---|---|---|---|
| | | Narrow element width | Wide element width |
| 2 | 0. 25mm | 0. 25mm | 0. 625mm |
| 3 | 0. 375mm | 0. 375mm | 1. 0mm |
| 4 | 0. 5mm | 0. 5mm | 1. 25mm |
| 5 | 0. 625mm | 0. 625mm | 1. 625mm |

rotate

Mean barcode rotating angle,
the value range is: [0, 3].
Definitions are as below:
Rotate value define
0 barcode doesn’t rotate to draw
1 barcode rotates 90° draw.
2 barcode rotates 180° draw.
3 barcode rotates 270° draw

str

Text character data flow

**Return value**

Return true if write success, else write failed.

**Remarks**

# DrawQRCode

Description: Draw QR code in assigned position of Page

**Syntax**

- (bool) DrawQRCode:(int)startx starty:(int)starty version:(int)version ecc:(int)ecc unitwidth:(int)unitwidth rotate:(int)rotate str:(char *)str

**Parameters**

startx

QRCode code top left corner x coordinates, the value range is: [0, Page_Width-1]

starty

QRCode code top left corner y coordinates, the value range is: [0, Page_Height-1]

version

Assign charater version. The value range is:[0,20]. When version is 0, printer caculates version number according to character set automatically.

ecc

Assign error correction level.
The value range is: [1, 4].
Definitios are as below:
ECC error correction level
1 L: 7%, low error correction, much data.
2 M: 15%, medium error correction
3 Q: optimize error correction
4 H: 30%, the highest error correction, less data.

unitwidth

QRCode code block, the value range Is:[1, 4]. Definitions are the same parameter as UniWidth of 1D code command.

rotate

QRCode code rotating angle, the value range is: [0, 3]. Definitions are the smae paramter as Rotate of 1D code command.

```
str
    QRCode test character data flow
```

**Return value**

Return true if write success, else write failed.

**Remarks**

# DrawPDF417

Description: draw PDF417 code in the assigned position of Page

**Syntax**

- (bool) DrawPDF417:(int)startx starty:(int)starty colnum:(int)colnum lwratio:(int)lwratio ecc:(int)ecc unitwidth:(int)unitwidth rotate:(int)rotate str:(char *)str

**Parameters**

startx

PDF417 code top left corner x coordinates, the value range is: [0, Page_Width-1]

starty

PDF417 code top left corner y coordinates, the value range is: [0, Page_Height-1]

colnum

ColNum is colnum, which means how many digits in per line. A digit is 17*UnitWidth dots. Line number is produces automatically by printer, the limited range is 3~90。ColNum value range:[1,30]。

lwratio

width height ratio. The value range: [3,5]

ecc

Error correction level, the value range is: [0. 8]

Ecc value error correction number stored files number(byte)

0 2 1108

1 4 1106

2 8 1101

3 16 1092

4 32 1072

5 64 1024

6 128 957

7 256 804

8 512 496

unitwidth

PDF417 code block, the value range is: [1, 3]. Definitions are the same parameter as UniWidth of 1D code.

rotate

PDF417 code rotaing angle, the value range is: [0, 3]. Definitions are the paramter as Rotate of 1D code

str

PDF417 text character data flow.

**Return value**

Return true if write success, else write failed.

**Remarks**

# DrawBitmap

Description: Draw picture in the assigned position of Page

**Syntax**

- (bool) DrawBitmap:(int)startx starty:(int)starty width:(int)width height:(int)height style:(int)style pdata:(Byte *)pdata

**Parameters**

startx

Bitmap top left corner x coordinates, the value range is: [0, Page_Width]

starty

Bitmap top left corner y coordinates, the value range is: [0, Page_Height]

width

Bitmap pixel width

height

Bitmap pixel height

style

Bitmap print special effects, and definitions are as below:

| bit | definition |
|---|---|
| 0 | inverse flag bit, inverse 1 bitmap to print,and rest to print |
| [2:1] | rotates flag bit<br>00 Rotate 0°<br>01 rotates 90°<br>10 rotates 180°<br>11 rotates 270° |
| [7:3] | reserved |
| [11:8] | Bitmap width magnification times. |
| [15:12] | bitmap height magnification times |

pdata

Bitmap dot matrix data

**Return value**

Return true if write success, else write failed.

**Remarks**

# DrawBitmap

Description: Draw picture in the assigned position of Page

**Syntax**

- (bool) DrawBitmap:(int)startx starty:(int)starty width:(int)width height:(int)height style:(int)style img:(UIImage *)img nBinaryAlgorithm:(int)nBinaryAlgorithm

**Parameters**

startx

Bitmap top left corner x coordinates, the value range is: [0, Page_Width]

starty

Bitmap top left corner y coordinates, the value range is: [0, Page_Height]

width

the print width

height

the print height

style

Bitmap print special effects, and definitions are as below:

| bit | definition |
|---|---|
| 0 | inverse flag bit, inverse 1 bitmap to print,and rest to print |
| [2:1] | rotates flag bit<br>00 Rotate 0°<br>01 rotates 90°<br>10 rotates 180°<br>11 rotates 270° |
| [7:3] | reserved |
| [11:8] | Bitmap width magnification times. |
| [15:12] | bitmap height magnification times |

img

the image to be print

nBinaryAlgorithm

algorithm used to convert a color bitmap to binary bitmap

0 Use dithering algorithm for color images have better effect

```
    1 Using the average threshold algorithm, and has better effect on text
class picture
```

**Return value**

Return true if write success, else write failed.

**Remarks**