

打印机接口说明

一 概述.....	3
二 函数说明.....	5
IO Function.....	5
createWebSocketIO.....	5
Page Function.....	6
PAGE_PageEnter.....	6
PAGE_PagePrint.....	7
PAGE_PageExit.....	8
PAGE_SetPrintArea.....	9
PAGE_DrawString.....	11
PAGE_DrawRect.....	13
PAGE_DrawBarcode.....	14
PAGE_DrawQRCode.....	16
PAGE_DrawImageRGBA.....	17
Pos Function.....	18
POS_TextOut.....	18
POS_SetBarcode.....	20
POS_PrintQRcode.....	22
POS_PrintImageRGBA.....	23
POS_Reset.....	24
POS_PrintSelfTestPage.....	25
POS_FeedAndCutPaper.....	26
POS_FullCutPaper.....	27
POS_HalfCutPaper.....	28
POS_Beep.....	29
POS_KickOutDrawer.....	30
POS_FeedLine.....	31
POS_FeedLines.....	32
POS_FeedDots.....	33
POS_SetMotionUnit.....	34
POS_SetLineHeight.....	35
POS_SetCharRightSpacing.....	36
Label Function.....	37
LABEL_PageBegin.....	37
LABEL_PageEnd.....	38
LABEL_PagePrint.....	39
LABEL_PageFeed.....	40
LABEL_PageCalibrate.....	41
LABEL_DrawPlainText.....	42
LABEL_DrawLine.....	43

LABEL_DrawBox.....	44
LABEL_DrawRectangel.....	45
LABEL_DrawBarcode.....	46
LABEL_DrawQRCode.....	48
LABEL_DrawPDF417.....	49
LABEL_DrawImageRGBA.....	50

一 概述

1 PrinterLibs 是在跨平台 WebSocket 打印开发包，用 JavaScript 编写。

2 PrinterLibs 有以下几类函数

A `createWebSocketIO`

目前 JavaScript 打印机开发包只支持 WebSocket 打印机。

B `PAGE_XXX`

以 `PAGE` 开头的函数，封装了页模式指令，可以控制打印机以页模式的方式打印。

- ① `PAGE_PageEnter` 进入页模式
- ② `PAGE_SetPrintArea` 设置页模式打印区域
- ③ `PAGE_DrawXXX` 系列函数在指定区域打印
- ④ `PAGE_PagePrint` 打印整个页面
- ⑤ `PAGE_PageExit` 退出页模式

备注：

②③可以重复调用

仅支持页模式的机型可以使用这些函数

C `POS_XXX`

以 `POS` 开头的函数，主要是封装了 `ESC/POS` 指令，可以控制打印机打印。

- ① 进纸系列函数可以控制打印机进纸
- ② 设置系列函数可以设置打印的格式等
- ③ 打印系列函数可以打印文本，条码，QR 码，图片等
- ④ 查询系列函数可以查询打印机状态
- ⑤ 其他函数可以控制钱箱、切刀、蜂鸣器等

D `LABEL_XXX`

以 `LABEL` 开头的函数，主要封装了标签指令，可以控制标签打印机打印。

- ① `LABEL_PageBegin` 初始化一个标签页
- ② `LABEL_DrawXXX` 在标签页上画内容
- ③ `LABEL_PageEnd` 结束标签页
- ④ `LABEL_PagePrint` 打印标签页

3 修改历史

时间	修改
2017 年 8 月 29 日	1 支持 WebSocket 网页打印 2 封装热敏指令集，标签指令集，页模式指令集。

二 函数说明

IO Function

createWebSocketIO

Syntax

```
function createWebSocketIO()
```

Parameters

Return value

返回可读写 WebSocket 的 Object

Remarks

```
io.open = function(address, port)
```

打开端口。Address 为打印机名称或 IP 地址，port 为端口号。

```
io.close = function()
```

关闭端口。发送是异步的，关闭端口的时候，仍有数据未送完，则会等待送完再关闭。

```
io.isOpened = function()
```

端口是否打开

```
io.write = function(data)
```

写入数据到打印机

```
io.read = null
```

从打印机读取数据，当前未实现该功能

Page Function

PAGE_PageEnter

选择页模式

Syntax

```
function PAGE_PageEnter(io)
```

Parameters

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

PAGE_PagePrint

页模式下打印页面内容

Syntax

```
function PAGE_PagePrint(io)
```

Parameters

Return value

如果指令写入成功，返回 **true**。否则，返回 **false**

Remarks

PAGE_PageExit

退出页模式

Syntax

```
function PAGE_PageExit(io)
```

Parameters

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

PAGE_SetPrintArea

页模式下设置打印区域

Syntax

```
function PAGE_SetPrintArea(io, left, top, right, bottom, direction)
```

Parameters

left
打印区域左上角 x 坐标

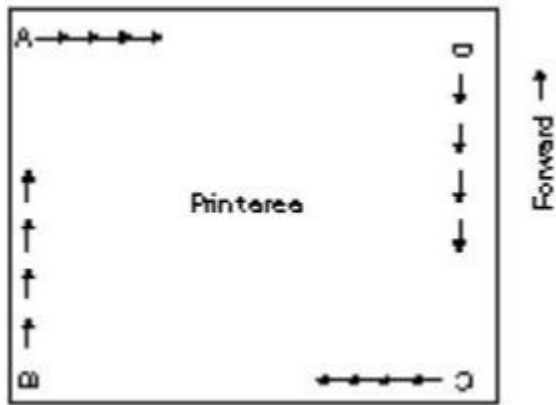
top
打印区域左上角 y 坐标

right
打印区域右下角 x 坐标

bottom
打印区域右下角 y 坐标

direction
打印区域方向

direction	打印方向	起始位置
0	自左向右	左上角(图中的 A)
1	自下向上	左下角(图中的 B)
2	自右向左	右下角(图中的 C)
3	自上向下	右上角(图中的 D)

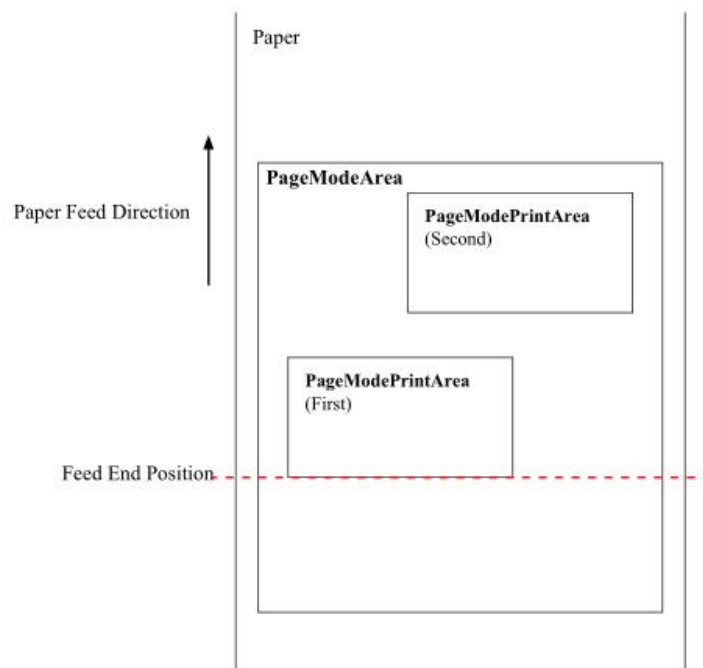


Return value

如果指令写入成功，返回 **true**。否则，返回 **false**

Remarks

下图展示了打印区域的概念



PAGE_DrawString

画文本

Syntax

```
function PAGE_DrawString(io, pszString, x, y, nWidthScale, nHeightScale, nFontType, nFontStyle)
```

Parameters

pszString

要打印的内容

x

指定水平方向的起始点位置离打印区域左边界的点数。（横坐标）

支持左对齐，居中，右对齐

传入 x 见下表

x	含义
-1	左对齐
-2	居中对齐
-3	右对齐
大于等于 0	横坐标

y

指定垂直方向的起始点位置离打印区域上边界的点数。（纵坐标）

nWidthScale

指定宽度放大倍数 [0,7]

nHeightScale

指定高度放大倍数 [0,7]

nFontType

字体类型

0 标准字体

1 压缩字体

nFontStyle

指定字体风格，可以为下表中的一个或者若干个（相加即可）

Value	Meaning
0x00	正常

0x08	加粗
0x80	1 点粗的下划线
0x100	2 点粗的下划线
0x200	倒置（只在行首有效）
0x400	反显（黑底白字）
0x1000	每个字符顺时针旋转 90 度

Return value

如果指令写入成功，返回 `true`。否则，返回 `false`

Remarks

页模式 `Draw` 函数并不立刻打印，只是填在页面中，直到调用了 `PAGE_PagePrint`，才开始打印。

PAGE_DrawRect

画矩形

Syntax

```
function PAGE_DrawRect(io, x, y, nWidth, nHeight, nColor)
```

Parameters

x

指定水平方向的起始点位置离打印区域左边界的点数。（横坐标）

y

指定垂直方向的起始点位置离打印区域上边界的点数。（纵坐标）

nWidth

指定矩形宽度

nHeight

指定矩形高度

nColor

指定矩形颜色

0 白色

1 黑色

Return value

如果指令写入成功，返回 **true**。否则，返回 **false**

Remarks

如果想画线，只需要把宽度设置为 **1**（若想画粗一点的线，可设置大一点）即可。

注意：不要画太大区域矩形，否则电源撑不住打印机会复位。

PAGE_DrawBarcode

画条码

Syntax

```
function PAGE_DrawBarcode(io, pszContent, x, y, nBarcodeUnitWidth, nBarcodeHeight, nHriFontType, nHriFontPosition, nBarcodeType)
```

Parameters

pszContent

条码内容

x

指定水平方向的起始点位置离打印区域左边界的点数。（横坐标）

y

指定垂直方向的起始点位置离打印区域上边界的点数。（纵坐标）

nBarcodeUnitWidth

指定条码的基本元素宽度。
可以为以下列表中所列值（n）之一。

n	单基本模块宽度（连续型）	双基本模块宽度（离散型）	
		窄元素宽度	宽元素宽度
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
6	0. 75mm	0. 75mm	1.875mm

nBarcodeHeight

条码高度

nHriFontType

指定 HRI（Human Readable Interpretation）字符的字体类型。
可以为以下列表中所列值之一。

Value	Meaning
0x00	标准 ASCII

0x01	压缩 ASCII
------	----------

nHriFontPosition

指定 HRI（Human Readable Interpretation）字符的位置。

可以为以下列表中所列值之一。

Value	Meaning
0x00	不打印
0x01	只在条码上方打印
0x02	只在条码下方打印
0x03	条码上、下方都打印

nBarcodeType

可以为以下列表中所列值之一。

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

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

PAGE_DrawQRCode

画二维码

Syntax

```
function PAGE_DrawQRCode(io, pszContent, x, y, nQRCodeUnitWidth,  
nVersion, nEcLevel)
```

Parameters

pszContent

二维码文本

x

指定水平方向的起始点位置离打印区域左边界的点数。(横坐标)

y

指定垂直方向的起始点位置离打印区域上边界的点数。(纵坐标)

nQRCodeUnitWidth

QR 码单元宽度，范围[1,16]。

QR 码单元宽度越大，QR 码越大。

nVersion

QR 码版本。0 表示自动计算版本。

QR 码版本越大，能编码的字符就越多，QR 码也越大。

nEcLevel

QR 码纠错等级。[1,4]

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

PAGE_DrawImageRGBA

画位图

Syntax

```
function PAGE_DrawImageRGBA(io, img, w, h, x, y, nBinaryAlgorithm)
```

Parameters

img

图像数据，RGBA 格式数组。

h

图像的宽度

w

图像的高度

x

指定水平方向的起始点位置离打印区域左边界的点数。（横坐标）
支持左对齐，居中，右对齐
传入 x 见下表

x	含义
-1	左对齐
-2	居中对齐
-3	右对齐
大于等于 0	横坐标

y

指定垂直方向的起始点位置离打印区域上边界的点数。（纵坐标）

nBinaryAlgorithm

- 二值化算法：
- 0 抖动算法（对彩色图效果较好）
 - 1 平均阈值（对黑白图效果较好）

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

Pos Function

POS_TextOut

打印文本

Syntax

```
function POS_TextOut(io, str, x, nWidthScale, nHeightScale, nFontType, nFontStyle)
```

Parameters

str

要打印的内容。

x

指定水平方向的起始点位置离打印区域左边界的点数。（横坐标）
支持左对齐，居中，右对齐

传入 x 见下表

x	含义
-1	左对齐
-2	居中对齐
-3	右对齐
大于等于 0	横坐标

nWidthScale

指定宽度放大倍数 [0,7]

nHeightScale

指定高度放大倍数 [0,7]

nFontType

字体类型

0 标准字体

1 压缩字体

nFontStyle

指定字体风格，可以为下表中的一个或者若干个（相加即可）

Value	Meaning
-------	---------

0x00	正常
0x08	加粗
0x80	1 点粗的下划线
0x100	2 点粗的下划线
0x200	倒置（只在行首有效）
0x400	反显（黑底白字）
0x1000	每个字符顺时针旋转 90 度

Return value

如果指令写入成功，返回 `true`。否则，返回 `false`

Remarks

`POS_TextOut` 并不立刻打印，需要调用 `POS_FeedXXX` 系列函数，才会把行缓冲区中的内容打印出来。或者也可以在字符串末尾加上换行，例如“`abcdefg\r\n`”。

POS_SetBarcode

打印条码

Syntax

```
function POS_SetBarcode(io, str, x, nBarcodeUnitWidth, nBarcodeHeight, nHriFontType, nHriFontPosition, nBarcodeType)
```

Parameters

str

条码内容（不同条码类型可编码的字符范围、长度不同，使用时请注意）

x

支持左对齐，居中，右对齐

传入 x 见下表

x	含义
-1	左对齐
-2	居中对齐
-3	右对齐

nBarcodeUnitWidth

指定条码的基本元素宽度。

可以为以下列表中所列值（n）之一。

n	单基本模块宽度 (连续型)	双基本模块宽度（离散型）	
		窄元素宽度	宽元素宽度
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
6	0. 75mm	0. 75mm	1.875mm

nBarcodeHeight

条码高度

nHriFontType

指定 HRI（Human Readable Interpretation）字符的字体类型。

可以为以下列表中所列值之一。

Value	Meaning
0x00	标准 ASCII
0x01	压缩 ASCII

nHriFontPosition

指定 HRI（Human Readable Interpretation）字符的位置。
可以为以下列表中所列值之一。

Value	Meaning
0x00	不打印
0x01	只在条码上方打印
0x02	只在条码下方打印
0x03	条码上、下方都打印

nBarcodeType

可以为以下列表中所列值之一。

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

Return value

如果指令写入成功，返回 **true**。否则，返回 **false**

Remarks

POS_PrintQRcode

打印二维码（QR 码）

Syntax

```
function POS_PrintQRcode(io, str, x, nUnitWidth, nVersion, nECCLevel)
```

Parameters

str

二维码文本

x

支持左对齐，居中，右对齐
传入 x 见下表

x	含义
-1	左对齐
-2	居中对齐
-3	右对齐

nUnitWidth

QR 码单元宽度，范围[1,16]。
QR 码单元宽度越大，QR 码越大。

nVersion

QR 码版本。0 表示自动计算版本。
QR 码版本越大，能编码的字符就越多，QR 码也越大。

nECCLevel

QR 码纠错等级。[1,4]

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

POS_PrintImageRGBA

打印图片

Syntax

```
function POS_PrintImageRGBA(io, img, w, h, x, nBinaryAlgorithm)
```

Parameters

img
图片像素数据，RGBA 格式数组

w
图片的宽度

h
图片的高度

x
支持左对齐，居中，右对齐
传入 x 见下表

x	含义
-1	左对齐
-2	居中对齐
-3	右对齐

nBinaryAlgorithm
二值化算法：
0 抖动算法（对彩色图效果较好）
1 平均阈值（对黑白图效果较好）

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

POS_Reset

复位打印机。会清空设置。

Syntax

```
function POS_Reset(io)
```

Parameters

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

POS_PrintSelfTestPage

打印自检页

Syntax

```
function POS_PrintSelfTestPage(io)
```

Parameters

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

POS_FeedAndCutPaper

进纸到切刀位置并切纸

Syntax

```
function POS_FeedAndCutPaper(io)
```

Parameters

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

POS_FullCutPaper

直接切纸（全切）

Syntax

```
function POS_FullCutPaper(io)
```

Parameters

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

POS_HalfCutPaper

直接切纸（半切）

Syntax

```
function POS_HalfCutPaper(io)
```

Parameters

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

POS_Beep

蜂鸣器鸣叫

Syntax

```
function POS_Beep(io, nBeepCount, nBeepMillis)
```

Parameters

nBeepCount

鸣叫次数

nBeepMillis

蜂鸣时间：100ms 为单位

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

POS_KickOutDrawer

打开钱箱

Syntax

```
function POS_KickDrawer(io, nDrawerIndex, nHightLevelTime,  
nLowLevelTime)
```

Parameters

nID

钱箱编号

0 钱箱引脚 2

1 钱箱引脚 5

nHighLevelTime

钱箱脉冲高电位 ms 时间

nLowLevelTime

钱箱脉冲低电位 ms 时间

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

POS_FeedLine

Syntax

```
function POS_FeedLine(io)
```

Parameters

Return value

如果指令写入成功，返回 **true**。否则，返回 **false**

Remarks

打印机进纸一行

POS_FeedLines

Syntax

```
function POS_FeedLines(io, lines)
```

Parameters

`lines`

进纸行数

Return value

如果指令写入成功，返回 `true`。否则，返回 `false`

Remarks

进纸 `lines` 行

POS_FeedDots

Syntax

```
function POS_FeedDots(io, dots)
```

Parameters

nDot

进纸点数

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

进纸 `dots` 点。一般情况下，1mm 有 8 个点。

POS_SetMotionUnit

设置水平和垂直移动单位

Syntax

```
function POS_SetMotionUnit(io, nHorizontalMU, nVerticalMU)
```

Parameters

`nHorizontalMU`
水平移动单位

`nVerticalMU`
垂直移动单位

Return value

如果指令写入成功，返回 `true`。否则，返回 `false`

Remarks

当 `nHorizontalMU` 和 `nVerticalMU` 都设置为 200 时，1 点就是 0.125mm。

POS_SetLineHeight

设置行高

Syntax

```
function POS_SetLineHeight(io, nHeight)
```

Parameters

nHeight

行高

Return value

如果指令写入成功，返回 **true**。否则，返回 **false**

Remarks

POS_SetCharRightSpacing

设置字符右边空白

Syntax

```
function POS_SetCharRightSpacing(io, nDistance)
```

Parameters

nDistance

字符右边空白

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

Label Function

LABEL_PageBegin

描述：指示一个 Page 页面的开始，并设置 Page 页的大小，参考点坐标和页面旋转角度。

Syntax

```
function LABEL_PageBegin(io, startx, starty, width, height, rotate)
```

Parameters

`startx`

页面起始点 x 坐标

`starty`

页面起始点 y 坐标

`width`

页面页宽

`startx + width` 的范围为 $[1, 384]$ 。编写 SDK 的时候，该打印机一行的打印点数为 384 点。如果你不确定每行打印点数，请参考打印机规格书。一般来说有 384, 576, 832 这三种规格。

`height`

页面页高

`starty + height` 的范围 $[1, 936]$ 。编写 SDK 的时候，限制是 936，但是这个值并不确定，这和打印机的资源有关。即便如此，也不建议把页高设置过大。建议页宽和页高设置和标签纸匹配即可。

`rotate`

页面旋转。`rotate` 的取值范围为 $\{0, 1\}$ 。为 0，页面不旋转打印，为 1，页面旋转 90 度打印。

Return value

如果指令写入成功，返回 `true`。否则，返回 `false`

Remarks

LABEL_PageEnd

描述：指示一个 Page 页面的结束。

Syntax

```
function LABEL_PageEnd(io)
```

Parameters

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

LABEL_PagePrint

描述：将 Page 页上的内容打印到标签纸上。

Syntax

```
function LABEL_PagePrint(io, num)
```

Parameters

num

打印的次数，1-255。

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

LABEL_PageFeed

描述：走纸到标签缝隙处。

Syntax

```
function LABEL_PageFeed(io)
```

Parameters

Return value

如果指令写入成功，返回 **true**。否则，返回 **false**

Remarks

LABEL_PageCalibrate

描述：校准标签纸

Syntax

```
function LABEL_PageCalibrate(io)
```

Parameters

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

LABEL_DrawPlainText

描述：在 Page 页面上指定位置绘制文本。只能单行打印。

Syntax

```
function LABEL_DrawPlainText(io, startx, starty, font, style, str)
```

Parameters

startx

定义文本起始位置 x 坐标，取值范围：[0, Page_Width-1]

starty

定义文本起始位置 y 坐标，取值范围：[0, Page_Height-1]

font

选择字体，有效值范围为{16, 24, 32, 48, 64, 80, 96}，当前打印机只可以使用 24。

style

字符风格。

数据位	定义
0 加粗标志位：	置 1 字体加粗，清零则字体不加粗。
1 下划线标志位：	置 1 文本带下划线，清零则无下划线。
2 反白标志位：	置 1 文本反白(黑底白字)，清零不反白。
3 删除线标志位：	置 1 文本带删除线，清零则无删除线。
[5,4] 旋转标志位：	00 旋转 0° ；
	01 旋转 90° ；
	10 旋转 180° ；
	11 旋转 270° ；
[11,8] 字体宽度放大倍数；	
[15,12] 字体高度放大倍数；	

str

字符串数据流

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

LABEL_DrawLine

描述：在 Page 页指定两点间绘制一条直线段。

Syntax

```
function LABEL_DrawLine(io, startx, starty, endx, endy, width, color)
```

Parameters

startx

直线段起始点 x 坐标值，取值范围：[0, Page_Width-1]。

starty

直线段起始点 y 坐标值，取值范围：[0, Page_Height-1]。

endx

直线段终止点 x 坐标值，取值范围：[0, Page_Width-1]。

endy

直线段终止点 y 坐标值，取值范围：[0, Page_Height-1]。

width

直线段线宽，取值范围：[1, Page_Height-1]。

color

直线段颜色，取值范围：{0, 1}。

当 Color 为 1 时，线段为黑色。

当 Color 为 0 时，线段为白色。

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

LABEL_DrawBox

描述：在 Page 页指定位置绘制指定大小的矩形框。

Syntax

```
function LABEL_DrawBox(io, left, top, right, bottom, borderwidth, bordercolor)
```

Parameters

left

矩形框左上角 x 坐标值，取值范围：[0, Page_Width-1]。

top

矩形框左上角 y 坐标值。取值范围：[0, Page_Height-1]。

right

矩形框右下角 x 坐标值。取值范围：[0, Page_Width-1]。

bottom

矩形框右下角 y 坐标值。取值范围：[0, Page_Height-1]。

borderwidth

矩形框线宽。

bordercolor

矩形框线颜色，取值范围{0, 1}。当 Color = 1 时，绘制黑色矩形框，Color = 0 时，绘制白色矩形框。

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

LABEL_DrawRectangel

描述：在 Page 页指定位置绘制矩形块。

Syntax

```
function LABEL_DrawRectangel(io, left, top, right, bottom, color)
```

Parameters

left

矩形块左上角 x 坐标值，取值范围：[0, Page_Width-1]。

top

矩形块左上角 y 坐标值。取值范围：[0, Page_Height-1]。

right

矩形块右下角 x 坐标值。取值范围：[0, Page_Width-1]。

bottom

矩形块右下角 y 坐标值。取值范围：[0, Page_Height-1]。

color

矩形块颜色，取值范围：{0, 1}。当 Color 为 1 时，矩形块为黑色。当 Color 为 0 时，矩形块为白色。

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

LABEL_DrawBarcode

描述：在 Page 页指定位置绘制一维条码。

Syntax

```
function LABEL_DrawBarcode(io, startx, starty, type, height, unitwidth, rotate, str)
```

Parameters

startx

条码左上角 x 坐标值，取值范围：[0, Page_Width-1]。

starty

条码左上角 y 坐标值，取值范围：[0, Page_Height-1]。

type

标识条码类型，取值范围：[0, 29]。各值定义如下：

type	类型	长度	条码值范围（十进制）
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-	偶数 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",	//可以根据数据切换编码模式-> !096 - !105	
12	"EAN128",	// 自动切换编码模式	
13	"25C",	// 25C Check use mod 10-> 奇数先在前面补 0, 10 的倍数-[(奇数位的数字之和<从左至右)+(偶数位数字之和)*3]	
14	"39C",	//39 碼的檢查碼必須搭配「檢查碼相對值對照表」，如表所示，將查出的相對值累加後再除以 43，得到的餘數再查出相對的編碼字元，即為檢查碼字元。	
15	"39",	//Full ASCII 39 Code，特殊字符用两个可表示的字来表示，39C 同样是包含 Full ASCII，注意宽窄比处理	
16	"EAN13+2",	// 附加码与主码间隔 7-12 单位，起始为 1011 间隔为 01，($_{0*10+_1}$) Mod 4-> 0--AA 1--AB 2--BA 3--BB	
17	"EAN13+5",	//附加码部分同上，模式($_{0+_2+_4}$)*3+($_{1+_3}$)*9) mod 10 ->"bbaaa", "babaa", "baaba", "baaab", "abbaa", "aabba", "aaabb", "ababa", "abaab", "aabab"	
18	"EAN8+2",	// 同 EAN13+2	
19	"EAN8+5",	// 同 EAN13+5	

```

20      "POST", // 详见规格说明，是高低条码，不是宽窄条码
21      "UPCA+2", // 附加码见 EAN
22      "UPCA+5", // 附加码见 EAN
23      "UPCE+2", // 附加码见 EAN
24      "UPCE+5", //附加码见 EAN
25      "CPOST", // 测试不打印。。。
26      "MSIC", //将检查码作为数据再计算一次检查码
27      "PLESSEY", // 测试不打印。。。
28      "ITF14", // 25C 变种， 第一个数前补 0，检查码计算时需扣除最后一个数，
但仍填充为最尾端
29      "EAN14"

```

height

定义条码高度。

unitwidth

定义条码码宽。取值范围：[1, 4]。

各值定义如下：

Width 取值 多级条码单位宽度 (mm) 二进制条码窄线条宽度 二进制条码宽线条宽度

1 0.125 0.125 0.25

2 0.25 0.25 0.50

3 0.375 0.375 0.75

4 0.50 0.50 1.0

rotate

表示条码旋转角度。取值范围：[0, 3]。各值定义如下：

Rotate 取值 定义

0 条码不旋转绘制。

1 条码旋转 90° 绘制。

2 条码旋转 180° 绘制。

3 条码旋转 270° 绘制。

str

文本字符数据流

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

LABEL_DrawQRCode

描述：在 Page 页指定位置绘制 QRCode 码。

Syntax

```
function LABEL_DrawQRCode(io, startx, starty, version, ecc, unitwidth, rotate, str)
```

Parameters

startx

QRCode 码左上角 x 坐标值，取值范围：[0, Page_Width-1]。

starty

QRCode 码左上角 y 坐标值，取值范围：[0, Page_Height-1]。

version

指定字符版本。取值范围：[0,20]。当 version 为 0 时，打印机根据字符串长度自动计算版本号。

ecc

指定纠错等级。取值范围：[1, 4]。各值定义如下：

ECC 纠错等级

1 L: 7%，低纠错，数据多。

2 M: 15%，中纠错

3 Q: 优化纠错

4 H: 30%，最高纠错，数据少。

unitwidth

QRCode 码码块，取值范围：[1, 4]。各值定义与一维条码指令输入参数 UniWidth 相同。

rotate

QRCode 码旋转角度，取值范围：[0, 3]。各值定义与一维条码指令输入参数 Rotate 相同。

str

QRCode 文本字符数据流

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

LABEL_DrawPDF417

描述：在 Page 页指定位置绘制 PDF417 条码。

Syntax

```
function LABEL_DrawPDF417(io, startx, starty, colnum, lwratio, ecc, unitwidth, rotate, str)
```

Parameters

startx

PDF417 码左上角 x 坐标值，取值范围：[0, Page_Width-1]。

starty

PDF417 码左上角 y 坐标值，取值范围：[0, Page_Height-1]。

colnum

ColNum 为列数，表述每行容纳多少码字。一个码字为 17*UnitWidth 个点。行数由打印机自动产生，行数范围限定为 3~90。ColNum 的取值范围：[1,30]。

lwratio

宽高比。取值范围：[3,5]。

ecc

纠错等级，取值范围：[0..8]。

ecc 取值 纠错码数 可存资料量（字节）

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 码码块，取值范围：[1, 3]。各值定义与一维条码指令输入参数 UniWidth 相同。

rotate

PDF417 码旋转角度，取值范围：[0, 3]。各值定义与一维条码指令输入参数 Rotate 相同。

str

PDF417 文本字符数据流。

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks

LABEL_DrawImageRGBA

描述：在 Page 页指定位置绘制位图。

Syntax

```
function LABEL_DrawImageRGBA(io, startx, starty, width, height, style,
img, nBinaryAlgorithm)
```

Parameters

startx

位图左上角 x 坐标值，取值范围：[0, Page_Width]。

starty

位图左上角 y 坐标值，取值范围：[0, Page_Height]。

width

位图的像素宽度。

height

位图的像素高度。

style

位图打印特效，各位定义如下：

位 定义

0 反白标志位，置 1 位图反白打印，清零正常打印。

[2:1] 旋转标志位：

00 旋转 0° ；

01 旋转 90° ；

10 旋转 180° ；

11 旋转 270°

[7:3] 保留。

[11:8] 位图宽度放大倍数。

[12:15] 位图高度放大倍数。

img

位图数据。RGBA 格式数组。

nBinaryAlgorithm

二值化算法，彩色图片无法直接打印，需要二值化为黑白图像才可以打印。

0 使用抖动算法

1 使用平均阈值算法

Return value

如果指令写入成功，返回 true。否则，返回 false

Remarks