

# Label Printer

---

LPAPI Interface Description

iOS Version

V6.2

**DothanTech**

**11<sup>th</sup> April 202**

**Content**

Content .....	2
Document Modification Records .....	3
1. Brief Introduction .....	5
1.1 Introduction to use .....	5
1.2 Demo.....	5
2. Interface list.....	5

## Document Modification Records

SN	Version	Descriptions	Modifier	Date
1	V0.1	Complete the first internal version of the document framework.	Cai Junjie	2016-07-18
2	V0.9	Improve document format and fix minor issues.	Yang Lingmei	2016-10-24
3	V1.0	Release first official version Ver 1.0.	Cai Junjie	2016-10-25
4	V2.0	Release second official version Ver 2.0. Add a method to print barcode.	Cai Junjie	2017-04-10
5	V3.0	Release third official version Ver 3.0. Optimize the method of generate barcode and QR Code.	Cai Junjie	2017-06-23
6	V4.0	Release fourth official version Ver 4.0. Optimize the way of data importing.	Cai Junjie	2017-08-04
7	V4.1	Fix the problem of font size	Cai Junjie	2017-08-14
8	V4.2	Add a way to print images.	Cai Junjie	2017-09-11
9	V4.3	Add method to get list of printers.	Cai Junjie	2017-11-01
10	V4.4	Add a method to print the view.	Cai Junjie	2017-12-15
11	V4.5	Add a method for setting print parameters.	Cai Junjie	2018-01-05
12	V4.6	Add the parameter setting of Error correction level of QR code, Word spacing of text, Line spacing of text, Bold text lines.	Cai Junjie	2018-01-25
13	V4.7	Add method to get printer details. Add display of printer list.	Cai Junjie	2018-03-15
14	V4.8	Add a parameter setting for automatic line feed of text. Add a method to print the text following the previous one.	Cai Junjie	2018-03-27
15	V5.0	Optimize the printing process. Add method to get preview image.	Cai Junjie	2018-06-07

16	V5.1	Add parameter settings for encoding mode of barcode, and support 15 kinds of encoding modes. Add the callback method to get the printing status. Add the method of upgrading printer firmware.	Cai Junjie	2018-10-18
17	V5.2	Optimize printing results.	Cai Junjie	2019-03-22
18	V5.3	Add a method to print images directly.	Cai Junjie	2019-04-01
19	V5.4	Fixed some printer scanning and connectivity issues.	Cai Junjie	2019-05-22
19	V5.5	Compatible with iOS 13.	Cai Junjie	2019-11-05
20	V5.6	Optimize the way of determining the connected printer when the specified printer name is not set.	Cai Junjie	2019-12-06
21	V5.7	Add compatibility for one-inch and half-inch printer.	Cai Junjie	2020-07-09
22	V5.8	Compatible with iOS 14.	Cai Junjie	2020-09-28
23	V5.9	Fix an issue with Bluetooth connections.	Cai Junjie	2020-11-25
24	V5.10	Open control settings for background printing.	Cai Junjie	2021-01-21
25	V5.11	Add compatibility with new printer models.	Cai Junjie	2021-04-12
26	V5.12	Compatible with iOS 14.5.	Cai Junjie	2021-05-17
27	V5.13	Add compatibility with 3 inch thermal transfer printer	Cai Junjie	2021-05-27
28	V5.14	Fix some bugs	Cai Junjie	2021-12-17
29	V6.0	Improve Bluetooth speed for some printer models.	Cai Junjie	2022-08-25
30	V6.1	Compatible with iOS 16.	Cai Junjie	2022-09-22
31	V6.2	Split static libraries between real machine and emulator	Cai Junjie	2023-04-11

## 1. Brief Introduction

dtplib interface is a JavaScript interface based on the dtplib Print Assistant provided by the label printer manufacturer. The purpose is to allow users to quickly and seamlessly complete the operation of the printer, shorten the development cycle and speed up the development process.

The dtplib interface has a wide range of system compatibility, supporting Windows, Kirin OS, UOS, Ubuntu, Debian and other mainstream desktop operating systems.

### 1.1 Introduction to use

1. This print interface requires an iPhone or iPad running iOS 11.0 or later.
2. Configure the project when accessing, add the -ObjC field to the Other Linker Flags parameter list under the Build Settings item. Please note the upper and lower case.
3. The folder contains static libraries for both "real" and "emulator", so please use them differently because of the different architectures.

### 1.2 Demo

Refer to the LPAPIDemo project code.

## 2. Interface list

### 1.+ (void)setPrintPageGapType:(int)gapType

【Function】 Set printer paper type

【Parameter list】 gapType: paper type.

0: continuous paper;

1: positioning holes (if positioning holes are not supported, then adjust to gap paper);

2: Gap paper;

3: Black mark paper.

### 2.+ (void)setPrintPageGapLength:(int)gapLength

【Function】 Set printer gap length (skip distance)

【Parameter list】 gapLength: paper gap length in millimeters.

### 3.+ (void)setPrintDarkness:(int)darkness

【Function】 Set printer print darkness

【Parameter list】 darkness:

0: set with the printer;

1: lightest;

6: normal;

15: darkest

### 4.+ (void)setPrintSpeed:(int)speed

【Function】 Set printer print speed

【Parameter list】 speed:

0: set with printer;

1: slowest;

3: normal;

5: fastest.

### 5.+ (void)scanPrinters:(void(^)(NSArray \*scannedPrinterNames))completion

【Function】 Get the list of searched printers.

【Parameter list】 completion: operation after searching, get a list of printers.

**6.+ (void)openPrinter:(NSString \*)printerName**

**completion:(DzDidOpenedPrinterBlock)completion**

【Function】 Turns on the printer with the specified name.

【Parameter list】 printerName: printer name. If it is an empty string, opens the first supported printer on the current client system. Enter printer model directly is supported.

【Parameter list】 completion: operation after connecting to the printer, provide feedback on whether the connection was successful or not.

**7.+ (PrinterInfo \*)connectingPrinterDetailInfo**

【Function】 read the details of the currently connected printer.

【Return Value】 details of printer

**8.+ (void)closePrinter**

【Function】 Turns off the open printer.

**9.+ (BOOL) startDraw:(CGFloat)width height:(CGFloat)height**

**orientation:(int)orientation**

【Function】 With the specified parameters, start drawing, then call endDraw to end the drawing, and finally call " print " to print.

【Parameter list】 width, height: width and height of the label (in millimeters).

【Parameter list】 orientation: the direction of label is printed.

0: no rotation;

90: rotate 90 degrees clockwise;

180: rotate 180 degrees;

270: rotate 90 degrees counterclockwise.

**10.+ (UIImage \*)endDraw**

【Function】 end the draw and also return the printed image.

【Return Value】 image of the label to be printed.

**11.+ (UIImage \*)previewImage**

【Function】 get a preview image without rotation direction.

【Return Value】 preview of the label image.

**12.+ (void)print:(void(^)(BOOL isSuccess))completion**

【Function】 to submit print data for actual printing, the endDraw method must be called first.

【Parameter list】 completion: operation after printing, provide feedback on whether the print was successful or not.

**13.+ (BOOL)setItemOrientation:(int)orientation**

【Function】 set rotation angle of the print element.

【Parameter list】 orientation:

0: no rotation;

90: rotate 90 degrees clockwise;

180: rotate 180 degrees;

270: rotate 90 degrees counterclockwise.

**14. + (BOOL)setItemHorizontalAlignment:(int)alignment**

【Function】 set the horizontal alignment of the print element.

【Parameter list】 alignment: Horizontal alignment.

0: Horizontal left;      1: Horizontal center;      2: Horizontal right.

15.+ **(BOOL)setItemVerticalAlignment:(int)alignment**

【Function】 set the vertical alignment of the print element.

【Parameter list】 alignment: vertical alignment.

0: vertical top;      1: vertical center;      2: vertical bottom.

16.+ **(BOOL)drawText:(NSString \*)text x:(CGFloat)x y:(CGFloat)y width:(CGFloat)width height:(CGFloat)height fontHeight:(CGFloat)fontHeight**

【Function】 print text.

【Parameter list】 text: text content.

【Parameter list】 x, y, width, height: position and size of the printed object, in millimeters.

【Parameter list】 fontHeight: the font height of the text, in millimeters.

17.+ **(BOOL)drawBarcode:(NSString \*)text x:(CGFloat)x y:(CGFloat)y width:(CGFloat)width height:(CGFloat)height**

【Function】 print barcode

【Parameter list】 text: barcode content.

【Parameter list】 x, y, width, height: position and size of the printed object, in millimeters.

18.+ **(BOOL)drawQRCode:(NSString \*)text x:(CGFloat)x y:(CGFloat)y width:(CGFloat)width**

【Function】 print QR Code

【Parameter list】 text: QR Code content.

【Parameter list】 x, y, width, height: position and size of the printed object, in millimeters.

19.+ **(BOOL)drawImage:(NSString \*)file threshold:(int)threshold**

【Function】 print image files. Local files or URL network files are supported.

【Parameter list】 file: local file full path name or URL network file.

【Parameter list】 size of the print object is the same as the label.

【Parameter list】 threshold:

the gray value of the image into a black and white image for printing.

1 to 255, gray value  $\geq$  the threshold value will be treated as white points and not printed; 256 means that the image will be printed after converting it into a gray image, conversion of grayscale dots to black and white dots will be done by the printer driver;

0/257 means that the original color of the image will be used directly for printing, and the conversion of the original The conversion of the original color to black and white is done by the printer driver.

Print QR code images, it is recommended that the grayscale threshold be set to 192;

Print images such as logos, it is recommended that the grayscale threshold be set to 0.

20.+ **(BOOL)drawImageWithImage:(UIImage \*)image threshold:(int)threshold**

【Function】 print image

【Parameter list】 image: image to be printed.

【Parameter list】 size of the print object is the same as the label.

【Parameter list】 threshold: same as above.

21.+ **(BOOL)drawLine:(CGFloat)x y:(CGFloat)y width:(CGFloat)width**

**height:(CGFloat)height**

【Function】 print line

【Parameter list】 x, y, width, height: position and size of the printed object, in millimeters.

**22.+ (BOOL)drawRectangleWithX:(CGFloat)x y:(CGFloat)y width:(CGFloat)width height:(CGFloat)height lineWidth:(CGFloat)lineWidth isFilled:(CGFloat)isFilled**

【Function】 print rectangle

【Parameter list】 x, y, width, height: position and size of the printed object, in millimeters.

【Parameter list】 lineWidth: width of the line, in millimeters.

【Parameter list】 isFilled: whether to fill the rectangle.

**23.+ (BOOL)drawRoundRectangleWithX:(CGFloat)x y:(CGFloat)y width:(CGFloat)width height:(CGFloat)height lineWidth:(CGFloat)lineWidth radius:(CGFloat)radius isFilled:(CGFloat)isFilled**

【Function】 print round rectangle

【Parameter list】 x, y, width, height: position and size of the printed object, in millimeters.

【Parameter list】 lineWidth: width of the line, in millimeters.

【Parameter list】 radius: radius of the corner, in millimeters.

【Parameter list】 isFilled: whether to fill the round rectangle.

**24.+ (BOOL)drawEllipseWithX:(CGFloat)x y:(CGFloat)y width:(CGFloat)width height:(CGFloat)height lineWidth:(CGFloat)lineWidth isFilled:(CGFloat)isFilled**

【Function】 print ellipse

【Parameter list】 x, y, width, height: position and size of the printed object, in millimeters.

【Parameter list】 lineWidth: width of the line, in millimeters.

【Parameter list】 isFilled: whether to fill the ellipse.

**25.+ (BOOL)drawCircleWithX:(CGFloat)x y:(CGFloat)y width:(CGFloat)width height:(CGFloat)height lineWidth:(CGFloat)lineWidth isFilled:(CGFloat)isFilled**

【Function】 print circle

【Parameter list】 x, y, width, height: position and size of the printed object, in millimeters.

【Parameter list】 lineWidth: width of the line, in millimeters.

【Parameter list】 isFilled: whether to fill the circle.

**26.+ (BOOL)drawView:(UIView \*)view**

【Function】 print view

【Parameter list】 view: the view to be printed.

【Parameter list】 lineWidth: width of the line, in millimeters.

**27.+ (void)upgradeDeviceWithFilePath:(NSString \*)filePath**

【Function】 upgrade the printer firmware with an upgrade file.

【Parameter list】 filePath: file path.

**28.+ (void)printImage:(UIImage \*)image completion:(void(^)(BOOL isSuccess))completion**

【Function】 print image directly

【Parameter list】 image: the image to be printed.

【Parameter list】 completion: operation after printing, provide feedback on whether the print was successful or not.

The above are commonly used interfaces, the rest could be found in the LPAPI.h file.