

# iOSPrinterSDK 开发文档

## 一、简介

此开发包封装了 BLE 蓝牙通信功能和常用的 ESC/POS 指令集

1、静态库名称: libPrintfESCLib.a

2、头文件名称:

Head File Name	Discription
PrinterBluetoothManager.h	BLE 蓝牙通信
PrintfESCManager.h	ESC/POS 指令封装
ESCTable.h	表格打印封装

PrinterBluetoothManager.h 头文件提供的方法如下:

1. 全局单例 PrinterBluetoothManager:

a) 获取单个实例

```
+(instancetype) printerBluetoothManagerInstance;
```

2. 开始或者停止扫描蓝牙设备

a) 开始扫描

```
-(void)beginScan;
```

b) 停止扫描

```
-(void)stopScan;
```

3. 连接蓝牙设备相关

a) 连接设备

```
-(void)connect:(CBPeripheral *)peripheral;
```

b) 连接上一次连接的蓝牙, 默认不连接

```
-(void)setIsOpenDefaultConnect:(BOOL)defaultConnect;
```

c) 是否已经连接蓝牙

```
-(BOOL)isConnected;
```

4. 发送数据

a) 发送数据给打印机

```
-(BOOL)writeNSData:(NSData *)nsData;
```

b) 发送数据给打印机并通过 block 回调读取结果

```
-(void)writeAndRead:(NSData *)nsData readNSData:(ReadNSData)
```

```
readNSData;
```

5. 蓝牙相关回调函数

a) 扫描到设备的回调闭包

```
-(void)blePeripheralFound:(void (^)(CBPeripheral *peripheral, NSNumber *rssi))block;
```

b) 连接设备成功的回调闭包

```
-(void)blePeripheralConnected:(void (^)(CBCentralManager *central, CBPeripheral *peripheral))block;
```

- b) 设备断开的回调闭包
  - (void)blePeripheralDisconnected:(void (^)(CBCentralManager \*central,CBPeripheral \*peripheral))block;
- c) 读取打印设备数据回调
  - @property (strong, nonatomic) ReadNSData \_Nullable readNSData;

PrintfESCManager.h 头文件提供的方法如下:

1. 全局单例 PrintfESCManager:
  - a) 获取单个实例
    - +(instancetype) createNew;
2. 控制打印机相关
  - a) 初始化打印机
    - (void)initPrinter;
  - b) 打印并换行
    - (void)setPrint;
  - c) 打印并走纸 n 行
    - (void)setPrintAndFeedRow:(int)row;
  - d) 设置绝对打印位置
    - (void)setAbsolutePrintPosition:(int)where;
  - e) 选择对齐方式
    - (void)setSelectJustification:(int)n;
    - 当 n 为 0 时: 左对齐
    - 当 n 为 1 时: 中间对齐
    - 当 n 为 2 时: 右对齐
  - f) 设置左边距
    - (void)setLeftMargin:(int)left;
3. 文本打印相关
  - a) 打印文本
    - (void)setText:(NSString\*)content;
  - b) 设置默认行间距
    - (void)setDefaultLineSpace;
  - c) 设置行间距
    - (void)setLineSpace:(int)space;
  - d) 选择打印模式
    - (void)setPrintMode:(int)mode;
    - 0: 标准 ASCII 码字体 (12 × 24)
    - 1: 压缩 ASCII 码字体 B (9 × 17)
    - 16: 选择倍高模式
    - 32: 选择倍宽模式
    - 多个模式, 相加即可

e) 选择/取消下划线模式

`-(void)setUnderlineMode:(int)n;`

当 n 为 0 时：取消下划线模式

当 n 为 1 时：选择下划线模式

f) 选择/取消加粗模式

`-(void)setBoldMode:(int)mode;`

当 n 为 0 时，取消加粗模式

当 n 为 1 时，选择加粗模式

g) 选择字体

`-(void)setSelectFont:(int)font;`

当 n 位为 0 时，选择标准 ASCII 码字体 (12 × 24)

当 n 位为 1 时，选择压缩 ASCII 码字体 (9 × 17)

#### 4. 条码打印相关

a) 打印条码

`-(void)setBarCodeWithType:(int)type barcodeStringPosition:(int)position  
barcodeHeightInDot:(int)height barcodeWidth:(int)width  
barcodeContent:(NSString*)content;`

**type**: 条码类型。0: UPC-A, 1: UPC-B, 2: EAN13, 3: EAN8, 4: CODE39,  
5: ITF, 6: CODABAR, 7: CODE93, 8: CODE128

**barcodeStringPosition**: 条码字符打印位置。0: 不打印, 1: 条码上方,  
2: 条码下方, 3: 条码上、下方都打印。

**barcodeHeightInDot**: 条码高度。高度范围  $1 \leq n \leq 255$ 。

**barcodeWidth**: 条码宽度。条码宽度范围  $2 \leq n \leq 6$ 。

b) 打印二维码

`-(void)setQRCode:(NSString *)content;`

#### 5. 图片打印相关

a) 打印图片

`-(void)setImage:(UIImage *)image left:(int)left;`

#### 6. 其他

a) 打满虚线, 58mm

`-(void)setPlusLine_58;`

b) 打满虚线, 80mm

`-(void)setPlusLine_80;`

c) 打印标题和内容, 分两边

`-(void)setTwoColumn:(NSString *)title content:(NSString *)content  
paperMM:(int)paperMM;`

**title**: 标题, 即左边的文字

**content**: 内容, 即右边的文字

**paperMM**: 纸张大小, 58mm 打印机传 48, 80mm 打印机传 72

d) 打印表格

`-(void)setTable:(ESCTable*)table;`

e) 获取打印数据

```
-(NSData *)getData;  
f) 清空打印数据  
-(void)clearData;
```

ESCTable.h 头文件提供的方法如下:

1. 表格初始化

```
-(instancetype)init:(NSString*)column  
regularExpression:(NSString*)regularExpression columnWidthArray:(int*)array  
columnWidthLength:(int)length;  
column: 表格标题头  
regularExpression: 上面 column 每个标题头的切割符号  
columnWidthArray: 每个标题头（每列）所占的字节数，1 个英文符号一个字节，  
一个中文两个字节。58mm 打印机，一行可以打印 32 个字节  
columnWidthLength: 有多少列
```

2. 设置 Table 中列数据的对其方式，默认右对齐

```
-(void)setColumnAlignRight:(BOOL)right;
```

3. 添加一行数据

```
-(void)addRow:(NSString*)row;  
添加一行数据。数据格式与表头格式一致。若某一单元格的数据超出限定的字符宽度，会自动换行打印，若需要手动换行，可在需要换行处加” \n”。
```

4. 获取表格生成的文本内容

```
-(NSString*)getTableText;
```