[1 概述 2](#_Toc26387)

[1.1 SDK的手机版本 2](#_Toc2078)

[1.2 SDK的引入 2](#_Toc22521)

[1.2.1 添加JQBlePrinter.framework 2](#_Toc21603)

[1.2.2 配置依赖项 2](#_Toc19557)

[1.3 SDK的使用 2](#_Toc18703)

[1.3.1 获取蓝牙管理单例对象，并设置代理 2](#_Toc24500)

[2 蓝牙设备管理类 2](#_Toc1482)

[2.1 蓝牙设备管理方法 3](#_Toc26021)

[2.1.1 获取蓝牙设备管理单例对象 3](#_Toc24367)

[2.1.2 开始扫描蓝牙设备 3](#_Toc14582)

[2.1.3 开始扫描提供指定服务的蓝牙设备 3](#_Toc25437)

[2.1.4 停止扫描 3](#_Toc23584)

[2.1.5 连接指定名字的蓝牙设备 3](#_Toc27750)

[2.1.6 连接蓝牙设备 3](#_Toc25751)

[2.1.7 断开蓝牙连接 4](#_Toc30438)

[2.1.8 向蓝牙设备写入字符串信息 4](#_Toc11651)

[2.1.9 向蓝牙设备写入二进制数据 4](#_Toc26237)

[2.1.10 判断是否连接蓝牙打印机 4](#_Toc29903)

[2.1.11 读取蓝牙打印机的状态 4](#_Toc28427)

[2.2 蓝牙设备管理代理协议 5](#_Toc24049)

[2.2.1 发现蓝牙打印机 5](#_Toc20702)

[2.2.2 扫描蓝牙打印机超时时间到 5](#_Toc28653)

[2.2.3 已经和蓝牙打印机断开连接 5](#_Toc18222)

[2.2.4 连接蓝牙打印机失败 5](#_Toc4179)

[2.2.5 手机蓝牙状态更新 5](#_Toc17307)

[2.2.6 蓝牙打印机状态更新 6](#_Toc19275)

[2.2.7 蓝牙打印机状态枚举定义 6](#_Toc9873)

[3 蓝牙打印服务类 6](#_Toc30959)

[3.1 获取打印服务单例对象 6](#_Toc25522)

[3.2 页模式下打印 6](#_Toc4737)

[3.3 设置打印纸张大小（打印区域）的大小 6](#_Toc15038)

[3.4打印边框 7](#_Toc24125)

[3.5 打印线条 7](#_Toc12432)

[3.6 页模式下打印文本框 7](#_Toc3801)

[3.7 页模式下打印文本框 8](#_Toc4390)

[3.8 打印一维条码 9](#_Toc7157)

[3.9 打印二维码 9](#_Toc19155)

[3.10 打印图片 9](#_Toc771)

[3.11 定位到标签 10](#_Toc13320)

**1 概述**

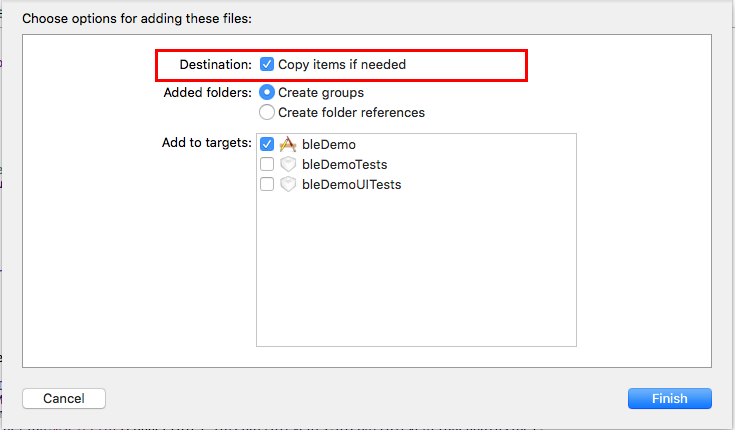
**1.1 SDK的手机版本**

支持iOS7及以上操作系统的iPhone/iPad，处理器支持armV7、armV7s、arm64

**1.2 SDK的引入**

**1.2.1 添加JQBlePrinter.framework**

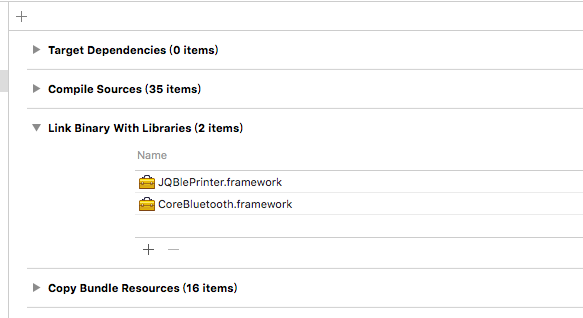
将JQBlePrinter.framework拖动到你的工程中，并勾选”Copy items if needed”。



**1.2.2 配置依赖项**

导入SDK后，切换到Build Phases选项卡中，在Link Binary With Libraries中添加

CoreBluetooth.framework。



**1.3 SDK的使用**

**1.3.1 获取蓝牙管理单例对象，并设置代理**

self.bleManager = [JQBleDeviceManager bleManager];

self.bleManager = self;

**2 蓝牙设备管理类**

**2.1 蓝牙设备管理方法**

**2.1.1 获取蓝牙设备管理单例对象**

/\*\*

\* 获取蓝牙设备管理单例对象

\* @return instancetype：蓝牙设备管理单例对象

\*/

+ (instancetype)bleManager;

**2.1.2 开始扫描蓝牙设备**

/\*\*

\* 开始扫描蓝牙设备

\* @param timeout 扫描超时时间

\*/

- (void)findPeripherals:(int)timeout;

**2.1.3 开始扫描提供指定服务的蓝牙设备**

/\*\*

\* 开始扫描提供指定服务的蓝牙设备

\* @param serviceArray 提供指定服务UUID的集合

\* @param timeout 扫描超时时间

\*/

- (void)findPeripherals:(NSArray \*)serviceArray timreOut:(NSTimeInterval)timeout;

**2.1.4 停止扫描**

/\*\*

\* 停止扫描

\*/

- (void)stopScan;

**2.1.5 连接指定名字的蓝牙设备**

/\*\*

\* 连接指定名字的蓝牙设备

\* @param bleName 要连接蓝牙设备的名字

\*/

- (void)connectBlePrint:(NSString \*)bleName;

**2.1.6 连接蓝牙设备**

/\*\*

\* 连接蓝牙设备

\* @param peripheral 蓝牙设备对象

\*/

- (void)connectPeripheral:(CBPeripheral \*)peripheral;

**2.1.7 断开蓝牙连接**

/\*\*

\* 断开蓝牙连接

\* @param peripheral 蓝牙设备对象

\*/

- (void)disconnect:(CBPeripheral \*)peripheral;

**2.1.8 向蓝牙设备写入字符串信息**

/\*\*

\* 向蓝牙设备写入字符串信息，自动进行分段发送

\* @param peripheral 已经连接的蓝牙设备对象

\* @param message 要发送的字符串信息

\*/

- (void)write:(CBPeripheral \*)peripheral message:(NSString \*)message;

**2.1.9 向蓝牙设备写入二进制数据**

/\*\*

\* 向蓝牙设备写入二进制数据，自动进行分段发送

\*

\* @param peripheral 已经连接的蓝牙设备对象

\* @param data 要发送的二进制数据

\*/

- (void)write:(CBPeripheral \*)peripheral data:(NSData \*)data;

**2.1.10 判断是否连接蓝牙打印机**

/\*\*

\* 判断是否连接蓝牙打印机

\* @return true：已经连接 false：未连接

\*/

- (BOOL)isConnectBle;

**2.1.11 读取蓝牙打印机的状态**

/\*\*

\* 读取蓝牙打印机的状态

\* @param timeout 读取状态超时时间，以S为单位

\* @param success 成功读取状态的block

\* @param fail 超时时间到，未获取蓝牙打印机状态的block

\*/

- (void)readBlePrintStatus:(NSTimeInterval)timeout

success:(void (^)(JQBlePrintStatus blePrintStatus))success

fail:(void (^)(void))fail;

**2.2 蓝牙设备管理代理协议**

**2.2.1 发现蓝牙打印机**

/\*\*

\* 发现蓝牙打印机

\* @param peripheral 已发现的蓝牙打印机对象

\*/

- (void)peripheralFound:(CBPeripheral \*)peripheral;

**2.2.2 扫描蓝牙打印机超时时间到**

/\*\*

\* 已经连接上蓝牙打印机

\*/

- (void)didConnectPeripheral;

**2.2.3 已经和蓝牙打印机断开连接**

/\*\*

\* 已经和蓝牙打印机断开连接

\*/

- (void)didDisconnectPeripheral;

**2.2.4 连接蓝牙打印机失败**

/\*\*

\* 连接蓝牙打印机失败

\*/

- (void)didFailToConnectPeripheral;

**2.2.5 手机蓝牙状态更新**

/\*\*

\* 手机蓝牙状态更新

\* @param central 中心设备对象

\*/

- (void)didUpdatecentralManagerState:(CBCentralManager \*)central;

**2.2.6 蓝牙打印机状态更新**

/\*\*

\* 蓝牙打印机状态更新

\* @param blePrintStatus 蓝牙打印机状态

\*/

- (void)didUpdateBlePrintStatus:(JQBlePrintStatus)blePrintStatus;

**2.2.7 蓝牙打印机状态枚举定义**

typedef NS\_ENUM(NSInteger, JQBlePrintStatus) {

JQBlePrintStatusNoPaper = 0x01, // 缺纸

JQBlePrintStatusOverHeat = 0x02, // 打印头过热

JQBlePrintStatusBatteryLow = 0x04, // 电量低

JQBlePrintStatusPrinting = 0x08, // 正在打印中

JQBlePrintStatusCoverOpen = 0x10, // 纸仓盖未关闭

JQBlePrintStatusNoError, // 其他值，没有错误

JQBlePrintStatusOk, // 打印完毕

};

**3 蓝牙打印服务类**

**3.1 获取打印服务单例对象**

/\*\*

\* 打印服务单例对象

\* @return 打印服务单例对象

\*/

+ (instancetype)CPCLManager;

**3.2 页模式下打印**

/\*\*

\* 页模式下打印

\* @param horizontal 0:正常打印，不旋转；

\* 1：整个页面顺时针旋转180°后，再打印

\* @param skip 0：打印结束后不定位，直接停止；

\* 1：打印结束后定位到标签分割线，如果无缝隙，最大进纸30mm 后停止

\*/

- (void)print:(int)horizontal skip:(int)skip;

**3.3 设置打印纸张大小（打印区域）的大小**

/\*\*

\* 设置打印纸张大小（打印区域）的大小

\* @param pageWidth 打印区域宽度

\* @param pageHeight 打印区域高度

\*/

- (void)pageSetup:(int)pageWidth pageHeight:(int)pageHeight;

**3.4打印边框**

/\*\*

\* 打印边框

\* @param lineWidth 边框线条宽度

\* @param top\_left\_x 矩形框左上角x坐标

\* @param top\_left\_y 矩形框左上角y坐标

\* @param bottom\_right\_x 矩形框右下角x坐标

\* @param bottom\_right\_y 矩形框右下角y坐标

\*/

- (void)drawBox:(int)lineWidth

top\_left\_x:(int)top\_left\_x

top\_left\_y:(int)top\_left\_y

bottom\_right\_x:(int)bottom\_right\_x

bottom\_right\_y:(int)bottom\_right\_y;

**3.5 打印线条**

/\*\*

\* 打印线条

\* @param lineWidth 线条宽度

\* @param start\_x 线条起始点x坐标

\* @param start\_y 线条起始点y坐标

\* @param end\_x 线条结束点x坐标

\* @param end\_y 线条结束点y坐标

\* @param fullline true:实线 false: 虚线

\*/

- (void)drawLine:(int)lineWidth

start\_x:(int)start\_x

start\_y:(int)start\_y

end\_x:(int)end\_x

end\_y:(int)end\_y

fullline:(BOOL)fullline;

**3.6 页模式下打印文本框**

/\*\*

\* 页模式下打印文本框

\* @param text\_x 起始横坐标

\* @param text\_y 起始纵坐标

\* @param text 打印的文本内容

\* @param fontSize 字体大小 1：16点阵；2：24点阵；3：32点阵；4：24点阵放大一倍；5：32点阵放大一倍; 6：24点阵放大两倍；7：32点阵放大两倍；其他：24点阵

\* @param rotate 旋转角度: 0：不旋转； 1：90度； 2：180°； 3:270°

\* @param bold 是否粗体: false：否； true：是

\* @param reverse 是否反白：false：不反白；true：反白

\* @param underline 是有有下划线：false:没有；true：有

\*/

- (void)drawText:(int)text\_x

text\_y:(int)text\_y

text:(NSString \*)text

fontSize:(int)fontSize

rotate:(int)rotate

bold:(int)bold

reverse:(BOOL)reverse

underline:(BOOL)underline;

**3.7 页模式下打印文本框**

/\*\*

\* 页模式下打印文本框

\* @param text\_x 起始横坐标

\* @param text\_y 起始纵坐标

\* @param width 文本框宽度

\* @param height 文本框高度

\* @param text 打印的文本内容

\* @param fontSize 字体大小 1：16点阵；2：24点阵；3：32点阵；4：24点阵放大一倍；5：32点阵放大一倍; 6：24点阵放大两倍；7：32点阵放大两倍；其他：24点阵

\* @param rotate 旋转角度: 0：不旋转； 1：90度； 2：180°； 3:270°

\* @param bold 是否粗体: false：否； true：是

\* @param reverse 是否反白：false：不反白；true：反白

\* @param underline 是有有下划线：false:没有；true：有

\*/

- (void)drawText:(int)text\_x

text\_y:(int)text\_y

width:(int)width

height:(int)height

str:(NSString \*)str

fontsize:(int)fontsize

rotate:(int)rotate

bold:(int)bold

underline:(BOOL)underline

reverse:(BOOL)reverse;

**3.8 打印一维条码**

/\*\*

\* 打印一维条码

\* @param start\_x 一维码起始横坐标

\* @param start\_y 一维码起始纵坐标

\* @param text 内容

\* @param type 条码类型：0：CODE39； 1：CODE128；2：CODE93 3：CODEBAR； 4：EAN8；5：EAN13；6：UPCA; 7:UPC-E; 8:ITF

\* @param rotate 旋转角度: 0：不旋转；1：90度； 2：180°；3:270°

\* @param linewidth 条码线宽度

\* @param height 条码高度

\*/

- (void)drawBarCode:(int)start\_x

start\_y:(int)start\_y

text:(NSString \*)text

type:(int)type

rotate:(int)rotate

linewidth:(int)linewidth

height:(int)height;

**3.9 打印二维码**

/\*\*

\* 打印二维码

\* @param start\_x 二维码起始横坐标

\* @param start\_y 二维码起始纵坐标

\* @param text 二维码内容

\* @param rotate 旋转角度：0：不旋转； 1：90度； 2：180°； 3:270°

\* @param ver QrCode宽度(2-6)

\* @param lel QrCode纠错等级(0-20)

\*/

- (void)drawQrCode:(int)start\_x

start\_y:(int)start\_y

text:(NSString \*)text

rotate:(int)rotate

ver:(int)ver

lel:(int)lel;

**3.10 打印图片**

/\*\*

\* 打印图片

\* @param start\_x 图片起始点横坐标

\* @param start\_y 图片起始点纵坐标

\* @param picName 图片名字

\*/

- (void)drawGraphic:(int)start\_x

start\_y:(int)start\_y

picName:(NSString \*)picName;

**3.11 定位到标签**

/\*\*

\* 定位到标签

\*/

- (void)feed;