

1 概述.....	2
1.1 SDK 的手机版本.....	2
1.2 SDK 的引入.....	2
1.2.1 添加 JQBlePrinter.framework.....	2
1.2.2 配置依赖项.....	2
1.3 SDK 的使用.....	2
1.3.1 获取蓝牙管理单例对象，并设置代理.....	2
2 蓝牙设备管理类.....	3
2.1 蓝牙设备管理方法.....	3
2.1.1 获取蓝牙设备管理单例对象.....	3
2.1.2 开始扫描蓝牙设备.....	3
2.1.3 开始扫描提供指定服务的蓝牙设备.....	3
2.1.4 停止扫描.....	3
2.1.5 连接指定名字的蓝牙设备.....	3
2.1.6 连接蓝牙设备.....	4
2.1.7 断开蓝牙连接.....	4
2.1.8 向蓝牙设备写入字符串信息.....	4
2.1.9 向蓝牙设备写入二进制数据.....	4
2.1.10 判断是否连接蓝牙打印机.....	4
2.1.11 读取蓝牙打印机的状态.....	5
2.2 蓝牙设备管理代理协议.....	5
2.2.1 发现蓝牙打印机.....	5
2.2.2 扫描蓝牙打印机超时时间到.....	5
2.2.3 已经和蓝牙打印机断开连接.....	5
2.2.4 连接蓝牙打印机失败.....	5
2.2.5 手机蓝牙状态更新.....	6
2.2.6 蓝牙打印机状态更新.....	6
2.2.7 蓝牙打印机状态枚举定义.....	6
3 蓝牙打印服务类.....	6
3.1 获取打印服务单例对象.....	6
3.2 页模式下打印.....	6
3.3 设置打印纸张大小（打印区域）的大小.....	7
3.4 打印边框.....	7
3.5 打印线条.....	7
3.6 页模式下打印文本框.....	8
3.7 页模式下打印文本框.....	8
3.8 打印一维条码.....	9
3.9 打印二维码.....	9
3.10 打印图片.....	10
3.11 定位到标签.....	10

1 概述

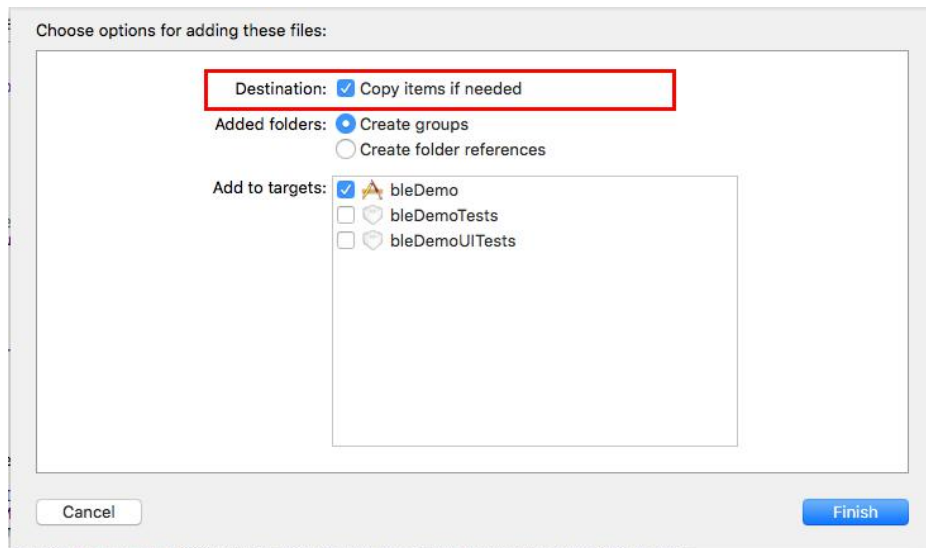
1.1 SDK 的手机版本

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

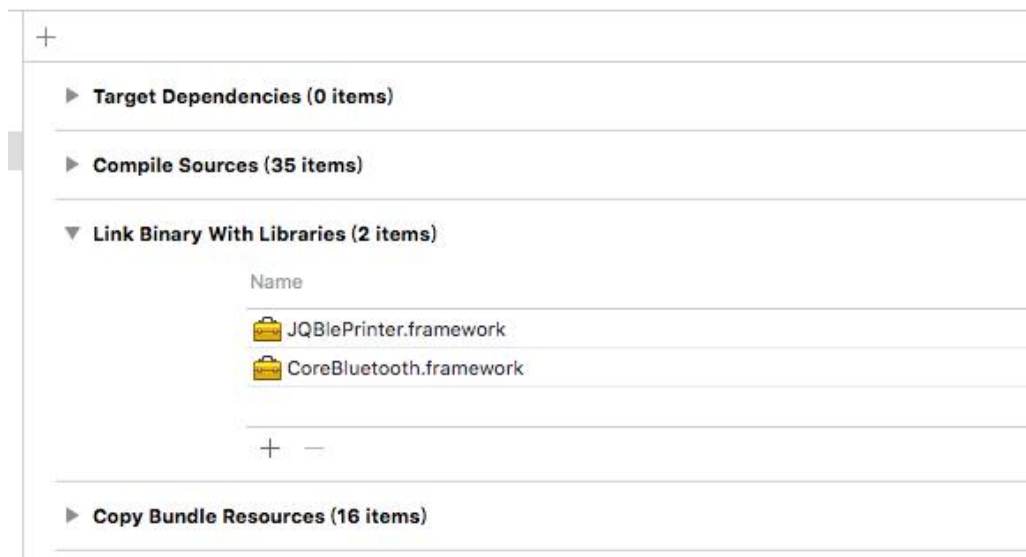
1.2.1 添加 JQBlePrinter.framework

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
 * @returninstancetype: 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(NSUInteger, 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 是否有下划线: 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 是否有下划线: 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 二维码 宽度(2-6)
 * @param lel 二维码 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;
```