**蓝牙打印机iOS版 SDK**

**说明文档**

目录

[1. 概述 4](#_Toc486188530)

[1.1 SDK的手机版本 4](#_Toc486188531)

[1.2 SDK的引入 4](#_Toc486188532)

[1.2.1 添加JQBlePrinter.framework 4](#_Toc486188533)

[1.2.2 配置依赖项 4](#_Toc486188534)

[1.3 SDK的使用 5](#_Toc486188535)

[1.3.1 获取蓝牙管理单例对象，并设置代理 5](#_Toc486188536)

[2. 蓝牙设备管理类 6](#_Toc486188537)

[2.1 蓝牙设备管理方法 6](#_Toc486188538)

[2.1.1 获取蓝牙设备管理单例对象 6](#_Toc486188539)

[2.1.2 开始扫描蓝牙设备 6](#_Toc486188540)

[2.1.3 开始扫描提供指定服务的蓝牙设备 6](#_Toc486188541)

[2.1.4 停止扫描 6](#_Toc486188542)

[2.1.5 连接指定名字的蓝牙设备 7](#_Toc486188543)

[2.1.6 连接蓝牙设备 7](#_Toc486188544)

[2.1.7 断开蓝牙连接 7](#_Toc486188545)

[2.1.8 向蓝牙设备写入字符串信息 7](#_Toc486188546)

[2.1.9 向蓝牙设备写入二进制数据 7](#_Toc486188547)

[2.1.10 判断是否连接蓝牙打印机 8](#_Toc486188548)

[2.1.11 读取蓝牙打印机的状态 8](#_Toc486188549)

[2.2 蓝牙设备管理代理协议 8](#_Toc486188550)

[2.2.1 发现蓝牙打印机 8](#_Toc486188551)

[2.2.2 扫描蓝牙打印机超时时间到 8](#_Toc486188552)

[2.2.3 已经和蓝牙打印机断开连接 8](#_Toc486188553)

[2.2.4 连接蓝牙打印机失败 9](#_Toc486188554)

[2.2.5 手机蓝牙状态更新 9](#_Toc486188555)

[2.2.6 蓝牙打印机状态更新 9](#_Toc486188556)

[2.2.7 蓝牙打印机状态枚举定义 9](#_Toc486188557)

[3. 蓝牙打印服务类 10](#_Toc486188558)

[3.1 获取打印服务单例对象 10](#_Toc486188559)

[3.2 打印文本 10](#_Toc486188560)

[3.3 初始化打印机 10](#_Toc486188561)

[3.4 选择加粗模式 10](#_Toc486188562)

[3.5 选择/取消下划线模式 10](#_Toc486188563)

[3.6 打印和行进 11](#_Toc486188564)

[3.7 水平制表符 11](#_Toc486188565)

[3.8 打印并走纸到左黑标处 11](#_Toc486188566)

[3.9 打印并回车 11](#_Toc486188567)

[3.10 设定右侧字符间距 11](#_Toc486188568)

[3.11 选择打印模式 12](#_Toc486188569)

[3.12 设置绝对打印位置 12](#_Toc486188570)

[3.13 选择位图模式打印图片 12](#_Toc486188571)

[3.14 设置默认行高 12](#_Toc486188572)

[3.15 设置行高 13](#_Toc486188573)

[3.16 设置水平制表符位置 13](#_Toc486188574)

[3.17 打印并进纸 13](#_Toc486188575)

[3.18 选择字体 13](#_Toc486188576)

[3.19 选择国际字符集 14](#_Toc486188577)

[3.20 选择/取消顺时针旋转90° 14](#_Toc486188578)

[3.21 设定相对打印位置 14](#_Toc486188579)

[3.22 选择对齐模式 14](#_Toc486188580)

[3.23 打印并向前走纸n行 15](#_Toc486188581)

[3.24 选择字符代码页 15](#_Toc486188582)

[3.25 选择字符大小 16](#_Toc486188583)

[3.26 定义并打印下载位图 16](#_Toc486188584)

[3.27 选择/取消黑白反显打印模式 16](#_Toc486188585)

[3.28 设定左边距 16](#_Toc486188586)

[3.29 设定横向和纵向移动单位 17](#_Toc486188587)

[3.30 设定打印区域宽度 17](#_Toc486188588)

[3.31 设定汉字模式 17](#_Toc486188589)

[3.32 设置汉字字符模式 17](#_Toc486188590)

[3.33 选择/取消汉字下划线模式 17](#_Toc486188591)

[3.34 定义自定义汉字 18](#_Toc486188592)

[3.35 选择/取消汉字倍高倍宽 18](#_Toc486188593)

[3.36 打印并走纸到右黑标处 18](#_Toc486188594)

[3.37 走纸到标签处 18](#_Toc486188595)

[3.38 打印光栅位图 18](#_Toc486188596)

[3.39 设置参数打印条码 19](#_Toc486188597)

[3.40 打印二维码 20](#_Toc486188598)

# 概述

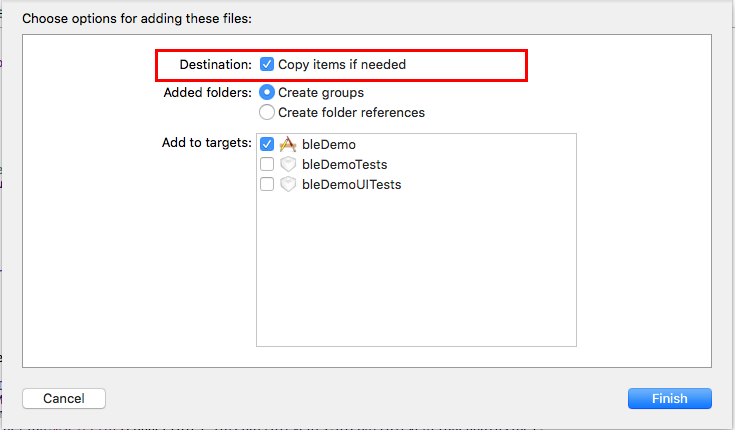
## SDK的手机版本

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

## SDK的引入

### 添加JQBlePrinter.framework

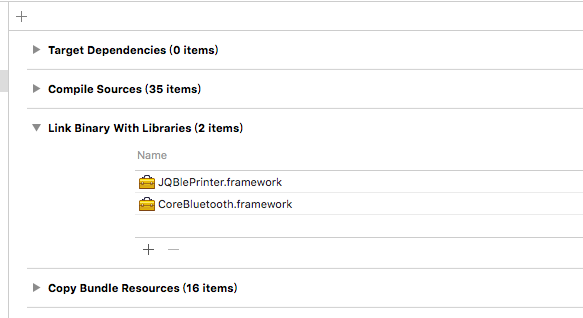
将JQBlePrinter.framework拖动到你的工程中，并勾选”Copy items if needed”。



### 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。



## SDK的使用

### 获取蓝牙管理单例对象，并设置代理

self.bleManager = [BleDeviceManager bleManager];

self.bleManager.delegate = self;

# 蓝牙设备管理类

## 蓝牙设备管理方法

### 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;

### 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;

### 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;

### 停止扫描

/\*\*

\* 停止扫描

\*/

- (void)stopScan;

### 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;

### 连接蓝牙设备

/\*\*

\* 连接蓝牙设备

\* @param peripheral 蓝牙设备对象

\*/

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

### 断开蓝牙连接

/\*\*

\* 断开蓝牙连接

\* @param peripheral 蓝牙设备对象

\*/

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

### 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;

### 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;

### 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;

### 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;

## 蓝牙设备管理代理协议

### 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;

### 扫描蓝牙打印机超时时间到

/\*\*

\* 已经连接上蓝牙打印机

\*/

- (void)didConnectPeripheral;

### 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;

### 连接蓝牙打印机失败

/\*\*

\* 连接蓝牙打印机失败

\*/

- (void)didFailToConnectPeripheral;

### 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;

### 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;

### 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

};

# 蓝牙打印服务类

## 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;

## 打印文本

/\*\*

\* 1、打印文本。

\* @param text 表示所要打印的文本内容。

\*/

- (Boolean)esc\_print\_text:(NSString \*)text;

## 初始化打印机

/\*\*

\* 2、初始化打印机。

\* 使所有设置恢复到打印机开机时的默认值模式。

\*/

- (Boolean)esc\_reset;

## 选择加粗模式

/\*\*

\* 3、选择加粗模式。

\*

\* @param b b为true时选择加粗模式，b为false时取消加粗模式。

\*/

- (Boolean)esc\_bold:(Boolean)b;

## 选择/取消下划线模式

/\*\*

\* 4、选择/取消下划线模式。

\* @param n 当n=1或n=49时选择下划线模式且设置为1点宽，当n=2或n=50时选择下划线模式且设置为2点宽，当n取其他值时取消下划线模式。

\*/

- (Boolean)esc\_underline:(NSInteger)n;

## 打印和行进

/\*\*

\* 5、打印和行进。

\* 基于当前的行间距，打印缓冲区内的数据并走纸一行。

\*/

- (Boolean)esc\_print\_formfeed;

## 水平制表符

/\*\*

\* 6、水平制表符。

\* 将打印位置移动至下一水平制表符位置。

\*/

- (Boolean)esc\_next\_horizontal\_tab;

## 打印并走纸到左黑标处

/\*\*

\* 7、打印并走纸到左黑标处。

\* 将打印缓冲区中的数据全部打印出来并走纸到左黑标处。

\*/

- (Boolean)esc\_left\_black\_label;

## 打印并回车

/\*\*

\* 8、打印并回车。

\* 该指令等同于LF指令，既打印缓冲区内的数据并走纸一字符行。

\*/

- (Boolean)esc\_print\_enter;

## 设定右侧字符间距

/\*\*

\* 9、设定右侧字符间距。

\* @param n 当n＜0时设定右侧字符间距为0，当n＞255时设定右侧字符间距为【255×（水平或垂直移动单位）】,

\* 当0≤n≤255时设定右侧字符间距为【n×（水平或垂直移动单位）】。

\*/

- (Boolean)esc\_right\_space:(NSInteger)n;

## 选择打印模式

/\*\*

\* 10、选择打印模式。

\* @param n 当n=0时选择字符字体A，当n=1时选择字符字体B，当n=2时表示选择字符字体C，当n=3时表示选择字符字体D；

\* 当n=8时选择字符加粗模式，当n=16时选择字符倍高模式，当n=32时选择字符倍宽模式，当n=128时选择字符下划线模式。

\* 此命令字体、加粗模式、倍高模式、倍宽模式、下划线模式同时设置。若要多种效果叠加，只需将相应的值相加即可

\* （例如若要B字体加粗，只需将n=1+8即n=9传入）。

\*/

- (Boolean)esc\_print\_mode:(NSInteger)n;

## 设置绝对打印位置

/\*\*

\* 11、设置绝对打印位置。

\* 将当前位置设置到距离行首（nL+nH×256）×（横向或纵向移动单位）处。当nL＜0或nL＞255时将nL设置为0，当nH＜0或nH＞255时将nH设置为0。

\*

\*/

- (Boolean)esc\_absolute\_print\_position:(NSInteger)nL nH:(NSInteger)nH;

## 选择位图模式打印图片

/\*\*

\* 12、选择位图模式打印图片。

\* @param m m表示位图模式。当m=1时位图模式为8点双密度，当m=32时位图模式为 24点单密度，当m=33时位图模式为24点双密度，

\* 除m=1,32,33之外位图模式都为8点单密度。

\* @param picName picName为要打印的位图。由于打印纸宽度有限，图片不可太大。

\*/

- (Boolean)esc\_bitmap\_mode:(NSUInteger)m image:(UIImage \*)image;

## 设置默认行高

/\*\*

\* 13、设置默认行高。

\* 将行间距设为约 3.75mm{30/203"}。

\*/

- (Boolean)esc\_default\_line\_height;

## 设置行高

/\*\*

\* 14、设置行高

\* 设置行高为[n×纵向或横向移动单位]英寸。

\* @param n n表示行高值。当n＜0时设置行高为0，当n＞255时设置行高为255[n×纵向或横向移动单位]英寸，

\* 当0≤n≤255时设置行高为[n×纵向或横向移动单位]英寸。

\*/

- (Boolean)esc\_line\_height:(NSInteger)n;

## 设置水平制表符位置

/\*\*

\* 15、设置水平制表符位置。

\* @param n n的长度表示横向跳格数，n[k]表示第k个跳格位置的值。当n的长度大于32时，只取前32个值；当n[k]大于等于n[k-1]时忽略该命令。

\* 当n[k]≤0或n[k]≥255时，忽略该命令。

\*/

- (Boolean)esc\_horizontal\_tab\_position:(NSArray \*)n;

## 打印并进纸

/\*\*

\* 16、打印并进纸。

\* @param n 当0≤n≤255时打印缓冲区数据并进纸【n×纵向或横向移动单位】英寸。当n＜0时进纸0，当n＞255时进纸【255×纵向或横向移动单位】英寸。

\*/

- (Boolean)esc\_print\_formfeed:(NSInteger)n;

## 选择字体

/\*\*

\* 17、选择字体。

\* @param n 当n=1或n=49时选择字体B，当n=2或n=50时选择字体C，当n=3或n=51时选择字体D，当n为其他值时选择字体A。

\*/

- (Boolean)esc\_font:(NSInteger)n;

## 选择国际字符集

/\*\*

\* 18、选择国际字符集。

\* @param n 当n≤0或n＞13时选择America字符集，当n=1时选择France字符集，当n=2时选择German字符集，当n=3时选择UK字符集，

\* 当n=4时选择Denmar字符集，当n=5时选择Sweden字符集，当n=6时选择Italy字符集，当n=7时选择Spain I字符集，当n=8时选择Japan字符集，

\* 当n=9时选择Norway字符集，当n=10时选择Denmar字符集，当n=11时选择Spain II字符集，当n=12时选择Latin字符集，当n=13时选择Korea字符集。

\*/

- (Boolean)esc\_national\_character\_set:(NSInteger)n;

## 选择/取消顺时针旋转90°

/\*\*

\* 19、选择/取消顺时针旋转90°。

\* @param n 当n=1或n=49时设置90°顺时针旋转模式，当n=2或n=50时设置180°顺时针旋转模式，当n=3或n=51时设置270°顺时针旋转模式，

\* 当n取其他值时取消旋转模式。

\*/

- (Boolean)esc\_rotate:(NSInteger)n;

## 设定相对打印位置

/\*\*

\* 20、设定相对打印位置。

\* 将打印位置从当前位置移至（nL+nH×256）×（水平或垂直运动单位）。当nL＜0时设置nL=0，当nL＞255时设置nL=255。

\* 当nH＜0时设置nH=0，当nH＞255时设置nH=255。

\*/

- (Boolean)esc\_relative\_print\_position:(NSInteger)nL nH:(NSInteger)nH;

## 选择对齐模式

/\*\*

\* 21、选择对齐模式。

\* @param n 当n=1或n=49时选择居中对齐，当n=2或n=50时选择右对齐，当n取其他值时选择左对齐。

\*/

- (Boolean)esc\_align:(NSInteger)n;

## 打印并向前走纸n行

/\*\*

\* 22、打印并向前走纸n行。

\* @param n 当n＜0时进纸0行，当n＞255时进纸255行，当0≤n≤255时进纸n行。

\*/

- (Boolean)esc\_print\_formfeed\_row:(NSInteger)n;

## 选择字符代码页

/\*\*

\* 23、选择字符代码页。

\* @param n 当n=1时选择Page 1 Katakana，当n=2时选择Page 2 Multilingual(Latin-1) [CP850]，当n=3时选择Page 3 Portuguese [CP860]，

\* 当n=4时选择Page 4 Canadian-French [CP863]，当n=5时选择Page 5 Nordic [CP865]，当n=6时选择Page 6 Slavic(Latin-2) [CP852]，

\* 当n=7时选择Page 7 Turkish [CP857]，当n=8时选择Page 8 Greek [CP737]，当n=9时选择Page 9 Russian(Cyrillic) [CP866]，

\* 当n=10时选择Page 10 Hebrew [CP862]，当n=11时选择Page 11 Baltic [CP775]，当n=12时选择Page 12 Polish，

\* 当n=13时选择Page 13 Latin-9 [ISO8859-15]，当n=14时选择Page 14 Latin1[Win1252]，当n=15时选择Page 15 Multilingual Latin I + Euro[CP858]，

\* 当n=16时选择Page 16 Russian(Cyrillic)[CP855]，当n=17时选择Page 17 Russian(Cyrillic)[Win1251]，当n=18时选择Page 18 Central Europe[Win1250]，

\* 当n=19时选择Page 19 Greek[Win1253]，当n=20时选择Page 20 Turkish[Win1254]，当n=21时选择Page 21 Hebrew[Win1255]，

\* 当n=22时选择Page 22 Vietnam[Win1258]，当n=23时选择Page 23 Baltic[Win1257]，当n=24时选择Page 24 Azerbaijani，

\* 当n=30时选择Thai[CP874]Thai[CP874]，当n=40时选择Page 25 Arabic [CP720]，当n=41时选择Page 26 Arabic [Win 1256]，

\* 当n=42时选择Page 27 Arabic (Farsi)，当n=43时选择Page 28 Arabic presentation forms B，当n=50时选择Page 29 Page 25 Hindi\_Devanagari，

\* 当n=252时选择Page 30 Japanese[CP932]，当n=253时选择Page 31 Korean[CP949]，当n=254时选择Page 32 Traditional Chinese[CP950]，

\* 当n=255时选择Page 33 Simplified Chinese[CP936]。

\* 当n取其他值时选择else if(n == 252) esc\_character\_code\_page[2] = 0x01。

\*/

- (Boolean)esc\_character\_code\_page:(NSInteger)n;

## 选择字符大小

/\*\*

\* 24、选择字符大小。

\* @param n 当n=2时2倍高，当n=3时3倍高，当n=4时4倍高，当n=20时2倍宽，当n=30时3倍宽，当n=40时4倍宽，当n=22时2倍宽高，当n=33时3倍宽高，

\* 当n=44时4倍宽高，当n取其他值时1倍宽高。

\*/

- (Boolean)esc\_character\_size:(NSInteger)n;

## 定义并打印下载位图

/\*\*

\* 25、定义并打印下载位图。

\* @param x x表示位图的横向点数（1≤x≤255），

\* @param y y表示位图的纵向点数（1≤y≤48）。

\* @param data data的长度等于x\*y\*8（1≤x\*y≤1536），表示位图字节数，除以上取值外其他取值均忽略此命令。

\* @param m m表示打印下载位图的模式，当m=1或m=49时设置倍宽模式，当m=2或m=50时设置倍高模式，当m=3或m=51时设置倍宽倍高模式，

\* 当m取其他值时设置普通模式打印所下载的位图。

\*/

- (Boolean)esc\_define\_print\_download\_bitmap:(NSInteger)x y:(NSInteger)y data:(NSArray \*)data mode:(NSInteger)m;

## 选择/取消黑白反显打印模式

/\*\*

\* 26、选择/取消黑白反显打印模式。

\* @param b 当b为true时选择黑白反显打印模式，当b为false时取消黑白反显打印模式。

\*/

- (Boolean)esc\_black\_white\_reverse:(Boolean)b;

## 设定左边距

/\*\*

\* 27、设定左边距。

\* 当0≤nL≤255且0≤nH≤255时，将左边距设为【(nL+nH×256)×(水平移动单位)】。当nL和nH取其他值时将左边距设为0。

\*/

- (Boolean)esc\_left\_margin:(NSInteger)nL nH:(NSInteger)nH;

## 设定横向和纵向移动单位

/\*\*

\* 28、设定横向和纵向移动单位。

\* 当0≤x≤255且0≤y≤255时分别将水平和垂直移动单位设为25.4/x毫米和25.4/y毫米。当x和y取其他值时取x=0和Y=0。

\*/

- (Boolean)esc\_move\_unit:(NSInteger)x y:(NSInteger)y;

## 设定打印区域宽度

/\*\*

\* 29、设定打印区域宽度。

\* 当0≤nL≤255且0≤nH≤255时,将打印区域宽度设为（nL+nH×256）×（水平移动单位）。当nL和nH取其他值时取nL=0和nH=0。

\*/

- (Boolean)esc\_print\_area\_width:(NSInteger)nL nH:(NSInteger)nH;

## 设定汉字模式

/\*\*

\* 30、设定汉字模式。

\* @param b 当b为true时选择汉字模式，当b为false时取消汉字模式。

\*/

- (Boolean)esc\_chinese\_mode:(Boolean)b;

## 设置汉字字符模式

/\*\*

\* 31、设置汉字字符模式。

\* @param n 当n=4时选择倍宽，当n=8时选择倍高，当n=128时选择下划线，当n=12时选择倍高倍宽，当n=132时选择倍宽下划线，当n=136时选择倍高下划线，

\* 当n=140时选择倍宽倍高下划线，当n取其他值时不选择倍高倍宽下划线。

\* 倍高、倍宽、下划线模式同时设置。

\*/

- (Boolean)esc\_chinese\_character\_mode:(NSInteger)n;

## 选择/取消汉字下划线模式

/\*\*

\* 32、选择/取消汉字下划线模式。

\* @param n 当n=1或n=49时选择汉字下划线（1点宽），当n=2或n=50时选择汉字下划线（2点宽），当n为其他值时不加下划线。

\*/

- (Boolean)esc\_chinese\_character\_underline\_mode:(NSInteger)n;

## 定义自定义汉字

/\*\*

\* 33、定义自定义汉字。

\* @param c2 c2表示自定义字符编码第二个字节,取值范围为A1H≤c2≤FEH，第一个字节为FEH，

\* @param data data表示自定义汉字的数据，1表示打印一个点，0表示不打印点。

\* data的长度为72，若data的长度不等于72或data的每个元素值出现小于0或大于255的情况，则忽略该命令。

\*/

- (Boolean)esc\_define\_chinese\_character:(NSInteger)c2 data:(NSArray \*)data;

## 选择/取消汉字倍高倍宽

/\*\*

\* 34、选择/取消汉字倍高倍宽。

\* @param b 当b为true时选择汉字倍高倍宽模式，当b为false时取消汉字倍高倍宽模式。

\*/

- (Boolean)esc\_chinese\_character\_twice\_height\_width:(Boolean)b;

## 打印并走纸到右黑标处

/\*\*

\* 35、打印并走纸到右黑标处。

\*/

- (Boolean)esc\_print\_to\_right\_black\_label;

## 走纸到标签处

/\*\*

\* 36、走纸到标签处。

\*/

- (Boolean)esc\_print\_to\_label;

## 打印光栅位图

/\*\*

\* 37、打印光栅位图。

\* @param m m表示光栅位图模式，当m=1或m=49时选择倍宽模式，当m=2或m=50时选择倍高模式，当m=3或m=51时选择倍宽倍高模式。

\* data表示要打印的光栅位图的数据，data的长度等于(xL+xH\*256)\*(yL+yH\*256)，表示要打印的光栅位图数据长度，

\* 当xL<0或xL>255或xH<0或xH>255或yL<0或yL>255或yH<0或yH>255或data的长度不等于((xL+xH\*256)\*(yL+yH\*256))或((xL+xH\*256)\*(yL+yH\*256))等于0时忽略该命令。

\*/

-(Boolean)esc\_print\_grating\_bitmap:(NSInteger)m xL:(NSInteger)xL xH:(NSInteger)xH yL:(NSInteger)yL yH:(NSInteger)yH data:(NSArray \*)data;

## 设置参数打印条码

/\*\*

\* 38、设置参数打印条码。

\* @param HRI\_position HRI\_position表示HRI字符打印位置(当HRI\_position=1或HRI\_position=49时HRI字符显示在条形码上方；

\* 当HRI\_position=2或HRI\_position=50时HRI字符显示在条形码下方；当HRI\_position取其他值时HRI字符不显示)。

\* @param HRI\_font HRI\_font表示HRI字符字体（当HRI\_font=1或HRI\_font=49时选择字体B，当HRI\_font取其他值时选择字体A）。

\* @param width width表示条码宽度（当width=2时设置条形码宽度为2，当width=3时设置条形码宽度为3，当width取其他值时设置条形码宽度为1），

\* @param height height表示条码高度（当1<=height<=255时设置条码高度为height，当height取其他值时设置条码高度为162），

\* @param type type表示条码类型（当type=0或type=65时选择条码类型为UPC-A，当type=1或type=66时选择条码类型为UPC-E，

\* 当type=2或type=67时选择条码类型为EAN13，当type=3或type=68时选择条码类型为EAN8，当type=4或type=69时选择条码类型为CODE39，

\* 当type=5或type=70时选择条码类型为ITF，当type=6或type=71时选择条码类型为CODABAR，当type=7或type=72时选择条码类型为CODE93，

\* 当type=8或type=73时选择条码类型为CODE128），

\* @param content content表示条码内容（UPC-A（长度为11、12）、UPC-E（长度为7、8、11、12）、EAN13（长度为12、13）、EAN8（长度为7、8）、

\* ITF（长度为大于2的偶数）只支持数字；

\* CODE39（长度大于1且小于255，支持数字、英文、空格、‘$’、‘%’、‘\*’、‘+’、‘-’、‘.’、‘/’）；

\* CODE93（长度大于1且小于255，支持数字、英文、空格、‘$’、‘%’、‘+’、‘-’、‘.’、‘/’）；

\* CODABAR（长度大于2且小于255，支持数字、英文ABCDabcd、‘$’、‘+’、‘-’、‘.’、‘/’、‘:’）；

\* CODE128（长度大于2且小于255，支持所有英文）。

\*/

- (Boolean)esc\_barcode\_1d:(NSInteger)HRI\_position HRI\_font:(NSInteger)HRI\_font width:(NSInteger)width height:(NSInteger)height type:(NSInteger)type content:(NSString \*)content;

## 打印二维码

/\*\*

\* 39、打印二维码。

\* @param type type表示二维码类型，当type=0时选择PDF417，当type=2时选择DATAMATRIX，当type取其他值时选择QRCODE。

\* @param content content表示要打印的二维码内容。

\*/

- (Boolean)esc\_print\_barcode\_2d:(NSInteger)type content:(NSString \*)content;