

# iOS POS Program Manual

## Bluetooth

### 1. Defined in the POSBLEManager Class

#### 1.1 sharedInstance

Singleton.

- (void) sharedInstance

#### 1.2 startScan

Search for bluetooth printers.

- (void) startScan

#### 1.3 stopScan

Stop searching for bluetooth printers.

- (void) stopScan

#### 1.4 connectDevice

Connect a Printer.

- (void) connectDevice: (CBPeripheral \*) peripheral

[Parameter]

-peripheral: Device.

#### 1.5 disconnectRootPeripheral

Disconnect a Printer.

- (void) disconnectRootPeripheral

#### 1.6 writeCommandWithData

Data printing.

- (void) writeCommandWithData: (NSData \*) data

[Parameter]

- data: Data to print.

## 1.7 printerCheck

Check printer connect status.

- (BOOL) printerCheck

[Return value]

- BOOL: YES or NO.

## 1.8 printerStatus

Printer status.

- (void) printerStatus: (POSBLEPrinterStatusBlock \*) statusBlock

[Parameter]

- statusBlock: Status callback method.

## POSBLEManagerDelegte

### ★ POSbleUpdatePeripheralList

Discover peripherals.

- (void) POSbleUpdatePeripheralList: (NSArray \*) peripherals RSSIList:

(NSArray \*) rssiList

[Parameter]

- peripherals: Peripheral List.

- rssiList: Rssi List.

### ★ POSbleConnectPeripheral

Peripheral connected succeeded.

- (void) POSbleConnectPeripheral: (CBPeripheral \*) peripheral

[Parameter]

- peripheral: Device.

### ★ POSbleFailToConnectPeripheral

Peripheral connected failed.

- (void) POSbleFailToConnectPeripheral: (CBPeripheral \*) peripheral  
error: (NSError \*) error

[Parameter]

- peripheral: Device.
- error: Connect error.

## ★ POSbleDisconnectPeripheral

Peripheral disconnected.

- (void) POSbleDisconnectPeripheral: (CBPeripheral \*) peripheral

[Parameter]

- peripheral: Device.

## ★ POSbleWriteValueForCharacteristic

Send data succeeded.

- (void) POSbleWriteValueForCharacteristic: (CBCharacteristic \*)  
character error: (NSError \*) error

[Parameter]

- character: Device attributes.
- error: Send error.

## ★ POSbleReceiveValueForCharacteristic

receive data from printer.

- (void) POSbleReceiveValueForCharacteristic: (CBCharacteristic \*)  
character error: (NSError \*) error

[Parameter]

- character: Device attributes.
- error: Receive error.

# WiFi

## 2. Defined in the POSWIFIManager Class

## 2.1 sharedInstance

Singleton.

- (void) sharedInstance

## 2.2 connectWithHost

Connect a Printer.

- (void) connectWithHost: (NSString \*) hostStr port: (UInt16 \*) port

[Parameter]

- hostStr: Connect printer ip address.

- port: Port Number.

## 2.3 disconnect

Disconnect a Printer.

- (void) disconnect

## 2.4 writeCommandWithData

Data printing.

- (void) writeCommandWithData: (NSData \*) data

[Parameter]

- data: Data to print.

## 2.5 printerCheck

Check printer connect status.

- (BOOL) printerCheck

[Return value]

- BOOL: YES or NO.

## 2.6 printerStatus

Printer status.

- (void) printerStatus: (POSBLEPrinterStatusBlock \*) statusBlock

[Parameter]

- statusBlock: Status callback method.

## 2.7 createUdpSocket

Create udp socket.

- (BOOL) createUdpSocket

[Return value]

- BOOL: YES or NO

## 2.8 closeUdpSocket

Close udp socket.

- (void) closeUdpSocket

## 2.9 sendFindCmd

Search printers with udp socket data.

- (void) sendFindCmd:

(POSWIFIManagerFoundPrinterBlock)printerBlock

[Parameter]

- printerBlock: Callback method

## 2.10 connectWithMac

Connect with mac address.

- (void) connectWithMac: (NSString \*)macStr

[Parameter]

- macStr: Mac address text

## 2.11 writeCommandWithData

Write command to printer with subpackage.

- (void) writeCommandWithData: (NSData \*)data subpackageLength:

(int)subpackageLength writeCallback:

(POSWIFIManagerWriteBlock)writeBlock

[Parameter]

- data: Print data.

- subpackageLength: Data subpackage length.

- writeCallback: Callback method.

## 2.12 setIPConfigWithIP

Set ip configuration.

- (void)setIPConfigWithIP: (NSString \*)ip Mask:(NSString \*)mask  
Gateway:(NSString \*)gateway DHCP:(BOOL)dhcp

[Parameter]

- ip: ip address.
- mask: mask address.
- gateway: gateway address.
- dhcp: YES or NO.

## 2.13 setWiFiConfigWithIP

Set ip configuration.

- (void)setWiFiConfigWithIP: (NSString \*)ip Mask:(NSString \*)mask  
Gateway:(NSString \*)gateway ssid:(NSString \*)ssid password:  
(NSString \*)password encrypt:(NSUInteger)encrypt

[Parameter]

- ip: ip address.
- mask: mask address.
- gateway: gateway address.
- ssid: ssid number.
- password: wifi password.
- encrypt: wifi encrypt method.

## 2.14 printerSN

Printer SN Code.

- (void) printerSN: (POSWiFiPrinterSNBlock \*) snBlock

[Parameter]

- snBlock: Serial Number callback method.

## POSWIFIManagerDelegate

## ★ POSwifiConnectedToHost

Connected a Printer.

- (void) POSwifiConnectedToHost: (NSString \*) host port: (UInt16) port  
[Parameter]
- host: IP Address.
- port: Port Number.

## ★ POSwifiDisconnectWithError

Disconnected a Printer.

- (void) POSwifiDisconnectWithError: (NSError \*) error  
[Parameter]
- error: error.

## ★ POSwifiWriteValueWithTag

Print successful.

- (void) POSwifiWriteValueSuccessWithTag:(long)tag

## ★ POSwifiReceiveValueForData

receive data from printer.

- (void) POSwifiReceiveValueForData:(NSData \*)data  
[Parameter]
- data: Printer data.

## ★ POSwifiUDPSearchFoundPrinter

Search printer through udp data.

- (void) POSwifiUDPSearchFoundPrinter:(PrinterProfile \*)foundPrinter  
[Parameter]
- foundPrinter: Printer found.

## ★ POSwifiUDPSearchFinishDiscovery

Search printer finished.

- (void) POSwifiReceiveValueForData

## ESC/POS

### 3. Defined in the PosCommand Class

#### 3.1 initializePrinter

Initialize the printer.

- (NSData \*) initializePrinter

[Return value]

- NSData \*: Send data.

#### 3.2 printAndFeedLine

Print and wrap.

- (NSData \*) printAndFeedLine

[Return value]

- NSData \*: Send data.

#### 3.3 printAndBackStandardModel

Print and go back to standard mode.

- (NSData \*) printAndBackStandardModel

[Return value]

- NSData \*: Send data.

#### 3.4 canglePrintDataByPageModel

Cancel print data in page mode.

- (NSData \*) canglePrintDataByPageModel

[Return value]

- NSData \*: Send data.

#### 3.5 printUnderPageModel

Printing in page mode.



- (NSData \*) printUnderPageModel

[Return value]

- NSData \*: Send data.

### 3.6 setDefaultLineSpace

Set default line spacing.

- (NSData \*) setDefaultLineSpace

[Return value]

- NSData \*: Send data.

### 3.7 setLeftSpaceWithnL

Set left margin.

- (NSData \*) setLeftSpaceWithnL:(int) nL andnH:(int) nH

[Parameter]

- nL: 0 ~ 255

- nH: 0 ~ 255

(nL+nH\*256) print area width

[Return value]

- NSData \*: Send data.

### 3.8 selectPagemodel

Select page mode.

- (NSData \*) selectPagemodel

[Return value]

- NSData \*: Send data.

### 3.9 selectFont

Select font.

- (NSData \*) selectFont: (int) n

[Parameter]

- n:

0, 48	ASCII (12 * 24)
1, 49	ASCII (9 * 17)

[Return value]

- NSData \*: Send data.

### 3.10 selectInternationalCharacterSets

Select international character set.

- (NSData \*) selectInternationalCharacterSets: (int) n

[Parameter]

- n:

0	America
1	France
2	Germany
3	England
4	Denmark 1
5	Sweden
6	Italy
7	Spain 1
8	Japan
9	Norway
10	Denmark 2
11	Spain 2
12	Latin America
13	Korea
14	Slovenia / Croatia
15	China

[Return value]

- NSData \*: Send data.

### 3.11 selectStandardModel

Select standard mode

- (NSData \*) selectStandardModel

[Return value]

- NSData \*: Send data.

### 3.12 printAndFeedForwardWhitN

Print and advance paper n lines.

- (NSData \*) printAndFeedForwardWhitN: (int) n

[Parameter]

- n: 0~255.

[Return value]

- NSData \*: Send data.

### 3.13 selectOrCanceInvertPrintModel

Select or cancel the black and white reverse print mode.

- (NSData \*) selectOrCanceInvertPrintModel: (int) n

[Parameter]

- n: 1: select; 0: cancel.

[Return value]

- NSData \*: Send data.

### 3.14 setHorizontalAndVerticalMoveUnitWithX

Set horizontal and vertical movement units.

- (NSData \*) setHorizontalAndVerticalMoveUnitWithX: (int) x andY:(int) y

[Parameter]

- x: horizontal, 0~255

- y: vertical, 0~255

[Return value]

- NSData \*: Send data.

### 3.15 setBarcodeHeight

Set barcode height.

- (NSData \*) setBarcodeHeight: (int) n

[Parameter]

- n: 1~255, default: 162.

[Return value]

- NSData \*: Send data.

### 3.16 setBarcodeWidth

Set barcode width.

- (NSData \*) setBarcodeWidth: (int) n

[Parameter]

- n: 2~6, default: 3.

[Return value]

- NSData \*: Send data.

### 3.17 printBarcodeWithM

Print barcode.

- (NSData \*) printBarcodeWithM: (int) m andContent: (NSString \*)  
content useEncoding: (NSStringEncoding) strEncoding

[Parameter]

- m:

0	UPC-A
1	UPC-E
2	JAN13 (EAN13)
3	JAN8 (EAB13)
4	CODE39
5	ITF

6	CODABAR
---	---------

- content: barcode content.
- strEncoding: Character Encoding.

[Return value]

- NSData \*: Send data.

### 3.18 printRasteBmpWithM

Print raster bitmap.

- (NSData \*) printRasteBmpWithM: (PrintRasterType) m andImage: (UIImage \*) image andType: (BmpType) type andPaperHeight: (int) paperheight

[Parameter]

- m:

RasterNolmorWH	Normal
RasterDoubleWidth	Double width
RasterDoubleHeight	Double height
RasterDoubleWH	Double width and height

- image: image
- type: Dithering or Threshold.
- paperheight: Paper height.

[Return value]

- NSData \*: Send data.

### 3.19 selectAlignment

Set alignment.

- (NSData \*) selectAlignment: (int) n

[Parameter]

- n:

0, 48	left alignment
-------	----------------

1, 49	center alignment
2, 50	right alignment

[Return value]

- NSData \*: Send data.

### 3.20 selectHRICharactersPrintPosition

Set HRI Character print position.

- (NSData \*) selectHRICharactersPrintPosition: (int) n

[Parameter]

- n:

0, 48	none
1, 49	above
2, 50	below
3, 51	above and below

[Return value]

- NSData \*: Send data.

### 3.21 setPrintAreaWidthWithnL

Set the print area width.

- (NSData \*) setPrintAreaWidthWithnL: (int) nL and nH:(int)nH

[Parameter]

- nL:  $0 \leq nL \leq 255$

- nH:  $0 \leq nH \leq 255$

(nL+nH\*256) print area width

[Return value]

- NSData \*: Send data.

### 3.22 printQRCode

Print QR code.

- (NSData \*) printQRCode: (int) n level:(int)errLevel code: (NSString\*)code useEnCodeing: (NSStringEncoding)strEncoding

[Parameter]

- n: unit size 1~16
- errLevel:error correct level 48~51

48	error correct level L (7%)
49	error correct level M (15%)
50	error correct level Q (25%)
51	error correct level H (30%)

- strEncoding: Character Encoding.

[Return value]

- NSData \*: Send data.

### 3.23 creatCashBoxContorPulseWithM

Generate cash box control pulses.

- (NSData \*) creatCashBoxContorPulseWithM: (int) m andT1:(int) t1 andT2:(int) t2

[Parameter]

- m:

0, 48	pin 2
0, 49	pin 5

- t1: 0 ~ 255, default 30
- t2: 0 ~ 255, default 255

Cash drawer open pulse high level time [t1 \* 2ms], low level [t2 \* 2ms],  
if t2 < t1, low level [t1 \* 2ms]

[Return value]

- NSData \*: Send data.

### 3.24 setBeeper

Set printer beeper.

- (NSData \*) setBeeper: (BOOL) onSwitch

[Parameter]

- onSwitch: YES or NO

[Return value]

- NSData \*: Send data.

### 3.25 setCutterAndBeeper

Set printer beeper count after cutter.

- (NSData \*) setCutterAndBeeper: (int) count interval:(int)interval

[Parameter]

- count: beeper count

- interval: interval \* 50ms

[Return value]

- NSData \*: Send data.

### 3.26 setCashBoxOpen

Set printer cash box open.

- (NSData \*) setCashBoxOpen

[Return value]

- NSData \*: Send data.