

IOS Bluetooth 4.0 (BLE) API Development Manual

V1.0

Revision

Date	Version	Instruction	Editor
Nov.8th, 2019	1.0	Establishment	Qi Zhengfeng

Content

1. Summary..... 1

2. Explanation..... 1

3. API Interface Description..... 1

1. Summary

In this file, it describes the development and programming methods between the printer with Bluetooth 4.0 and the device with IOS. ESC/POS commands and CPCL commands are supported. It also supplies the Functions, DEMO and Source Codes for Characters Printing, Barcode Printing, Picture Printing, etc to be reference for users.

2. Explanation

BTDemoIVT is the source code of Demo and api; SPRTPrint.h is the header file of api; SPRTPrint.m is the Functions Set; all functions in Demo will call this interface to print by printer. The below are API interface explanations.

3. API Interface Description

```
//
//  SPRTPrint.h
//  BTDemo
//
//  Created by SPpp on 14-10-31.
//
//value of Status Checking Command
#define DLE_EOT_1 1
#define DLE_EOT_3 3
#define DLE_EOT_4 4

// printing successfully or not    qzfeng 2015/12/23
#define DLE_EOT_5 5

//2-dimension barcode type
#define POS_BT_PDF417      0
#define POS_BT_DATAMATRIX  1
#define POS_BT_QRCODE      2
//characters position for 1-dimension barcode
#define POS_BT_HT_NONE     0
#define POS_BT_HT_UP       1
#define POS_BT_HT_DOWN     2
#define POS_BT_HT_BOTH     3
//1-dimension barcode type
#define POS_BT_UPCA        65
#define POS_BT_UPCE        66
```

```

#define POS_BT_JAN13      67
#define POS_BT_JAN8       68
#define POS_BT_CODE39     69
#define POS_BT_ITF        70
#define POS_BT_CODABAR    71
#define POS_BT_CODE93     72
#define POS_BT_CODE128    73

#define BUF_SIZE          8*1024
#define GET_NUM            100

#import <CoreBluetooth/CoreBluetooth.h>
#import <CoreBluetooth/CBService.h>
#import <UIKit/UIKit.h>
#import <Foundation/Foundation.h>
static int head=-1,tail=-1;
//static BOOL timerStarted = NO;
static Byte sndBuf[BUF_SIZE];
// static BOOL isBufferedWrite = NO;
static BOOL isBufferedWrite = YES;           // qzfeng 2016/05/10
static BOOL taskInRunning = NO;
static NSMutableData *lastData = nil;

```

```

@interface SPRTPrint : NSObject
/*****
Function Name: printtxt
Function: Send text data or control command to the printer
Parameter: data or control command

Return:
YES - succeed
NO - failed
*****/
+ (BOOL) printTxt:(NSString *)data;

```

```

/*****
Function Name: printbin
Function: Send binary data or control board to the printer
Parameter: data or control command

Return:
YES - succeed
NO - failed

```

*****/

+ (BOOL) printBin:(NSData *)data;

/******

Function Name: sendCheckOfflineCmd, Sending Command: DLE EOT n (0x10 0x04 1)

Function: Send command of checking the printer offline or not. Limited by ble4.0, the program can't receive the return from printer by order, but can only handle the checking result in response function "didDataReceived"

Parameter: null

Return:

YES - succeed

NO - failed

*****/

+ (BOOL) sendCheckOfflineCmd;

/******

Function Name: sendCheckPaperOutCmd, Sending command: DLE EOT n (0x10 0x04 4)

Function: Send command of checking the paper out or not. Limited by ble4.0, the program can't receive the return from printer by order, but can only handle the checking result in response function "didDataReceived"

Parameter: null

Return:

YES - succeed

NO - failed

*****/

+ (BOOL) sendCheckPaperOutCmd;

/******

Function Name: SendCheckErrorCmd, Sending command: DLE EOT n (0x10 0x04 3)

Function: Send command of checking the printer with error or not. Limited by ble4.0, the program can't receive the return from printer by order, but can only handle the checking result in response function "didDataReceived"

Parameter: null

Return:

YES - succeed

NO - failed

*****/

+ (BOOL) sendCheckErrorCmd;

/******

Function Name: printAlignXXX, Sending command: ESC a n(0/0x30:left; 1/0x31:middle; 1/0x32:right)

Function: Adjust the alignment way

Parameter: null

Return:

YES - succeed

NO - failed

*****/

+ (BOOL) printAlignLeft;

+ (BOOL) printAlignCenter;

+ (BOOL) printAlignRight;

/******

Function Name: print2DBarCode,

Sending command of choosing type: GS Z n(1d 5a n choose 0:pdf417,1:datamatrix,2:qr-code)

Sending command of printing: ESC Z v r k nL nH d1...dn(1b 5a v r k nL nH d1...dn print, the detailed parameter can be referred in Commands Manual)

Function: Printing 2D barcode

Parameter: type:0-pdf417,1-datamatrix,2-qr-code

v、r: the meaning is different to the different type barcode.

Data: barcode content

The below is the detailed explanation of commands:

PDF417 2D barcode

$1 \leq v \leq 30$ is the characters numbers in each line. Because the paper width for different models is different, the max value of v should be within the max numbers allowed by the model.

$0 \leq r \leq 8$ means the Error Correction Level

DATA MATRIX 2D barcode

$0 \leq v \leq 144$ means the bitmap height (0: choose automatically)

$8 \leq r \leq 144$ means the bitmap width (when v=0, invalid)

QR CODE 2D barcode

$0 \leq v \leq 40$ means the bitmap version (0: choose automatically)

r =76,77,81,72 means the Error Correction Level (L:7%, M:15%,Q:25%,H:30%)

·parameter k, n(nL, nH), d the meaning of parameter

$1 \leq k \leq 6$ means the enlargement times in portrait direction

$1 \leq n \leq 65535$ means the data length of printed barcode to be n, nL, nH is the lowest and highest bit of n ($n = dL + dH \times 256$)

$0 \leq dn \leq 255$ means the barcode data

Return:

YES - succeed

NO - failed

```
*****/
+ (BOOL) print2DBarcode:(int)type para1:(int)v para2:(int)r para3:(int)k
content:(NSData*)data;
```

```
/******
```

Function name: print1DBarcode,

Sending command of setting width: GS W n(1d 77 n, $2 \leq n \leq 6$)

Sending command of setting height: GS h n(1d 68 n, $1 \leq n \leq 255$)

Sending command of barcode characters position: GS H n (1d 48 n, 0: don't print; 1: print on the barcode; 2: print under the barcode; 3: print both on and under the barcode)

Sending command of printing: GS k m d1...dk nul(1d 6b m d1...dk 00)

or GS k m n d1...dn(1d 6b m n d1...dn)

Function: print 1D barcode

Parameter:

Type: barcode type. Refer to the below explanation of m barcode type.

W: barcode width, $2 \leq n \leq 6$

H: barcode height, $1 \leq n \leq 255$

Position: barcode characters position

0: don't print; 1: print on the barcode; 2: print under the barcode; 3: print both on and under the barcode

Data: barcode content

Other explanations for barcode type:

[range] ① $0 \leq m \leq 6$ (the value range of k and d depends on the barcode type)

② $65 \leq m \leq 73$ (the value range of k and d depends on the barcode type)

[description] Choose one type of barcode and print

m to choose the barcode type, as below:

M barcode type, characters numbers d

① 0 UPC-A $11 \leq k \leq 12$ $48 \leq d \leq 57$

1 UPC-E $11 \leq k \leq 12$ $48 \leq d \leq 57$

2 JAN13 (EAN13) $12 \leq k \leq 13$ $48 \leq d \leq 57$

3 JAN 8 (EAN8) $7 \leq k \leq 8$ $48 \leq d \leq 57$

4 CODE39 $1 \leq k \leq 255$ $45 \leq d \leq 57$, $65 \leq d \leq 90$, 32, 36,37,43

5 ITF $1 \leq k \leq 255$ $48 \leq d \leq 57$

6 CODABAR $1 \leq k \leq 255$ $48 \leq d \leq 57$, $65 \leq d \leq 68$, 36, 43,45,46,47,58

② 65 UPC-A $11 \leq n \leq 12$ $48 \leq d \leq 57$

66 UPC-E $11 \leq n \leq 12$ $48 \leq d \leq 57$

67 JAN13 (EAN13) $12 \leq n \leq 13$ $48 \leq d \leq 57$

68 JAN 8 (EAN8) $7 \leq n \leq 8$ $48 \leq d \leq 57$

69 CODE39 $1 \leq n \leq 255$ $45 \leq d \leq 57$, $65 \leq d \leq 90$, 32, 36,37,43

d1 = dk = 42

70 ITF $1 \leq n \leq 255$ $48 \leq d \leq 57$
 71 CODABAR $1 \leq n \leq 255$ $48 \leq d \leq 57$ $65 \leq d \leq 68$, 36,43,45,46,47 58
 72 CODE93 $1 \leq n \leq 255$ $0 \leq d \leq 127$
 73 CODE128 $2 \leq n \leq 255$ $0 \leq d \leq 127$

Return:

YES - succeed

NO - failed

*****/

+ (BOOL) print1DBarCode:(int)type width:(int)w height:(int)h txtpositon:(int)positon
 content:(NSData*)data;

**

* Convert to code c128

* qzfeng 2015/11/11

* data is the input string, which must be numbers, dataBytes is the converted byte, and
 retDataBytesLen is the length of the converted byte

**/

+ (BOOL) getCodeCByte: (NSData*)data content3:(Byte *)dataBytes retLen:(int
 *)retDataBytesLen;

/**

* Print 128 barcode with code c

* 15/11/10 fudaohui

* Note: data is code b character set, dataBytes is data of code c character set

**/

+ (BOOL) print128BarCode: (int)w height2:(int)h txtpositon2:(int)positon
 content0:(NSData*)data content2:(Byte *)dataBytes length2:(int)dataBytesLen;

*****/

Function name: buffedWriteCtrl

Function: switch on/off printing with buffer

Due to the limitations of the Bluetooth bottom layer, the size of data which sent directly to the printer can only about 2KB in a task. If the printed date size exceeds this limit, the subsequent data cannot be sent successfully. The buffer mode should be switched on, the data is buffered and then printed by a timed task.

Parameter:

YES Switch on printing with buffer

NO Switch off printing with buffer

Return:

no

*****/

+ (void) buffedWriteCtrl:(BOOL)isBuffed;

/******

Function name: printBitMap,

Bitmap command: ESC * m nl nh d1 ... dn (1b 2a m nl nh dl ... dk)

Function: print a bitmap, please refer to the reference manual for specific instructions

Parameter:

mode 0 : 8 points single density 1: 8 points double density 32: 24 points single density 33: 24 points double density

bm bitmap data

Return:

YES succeed

NO failed

*****/

+ (BOOL) printBitMap:(int)mode bitmap:(NSData*)bm;

/******

Function name: setLineHeight

Bitmap command: ESC 3 n (1b 33 n)

Function: Set height of line(or line spacing)

Parameter:

0 ≤ n ≤ 255, the system default value is 32, equivalent to 4mm or 1/6 inch

Return:

YES succeed

NO failed

*****/

+ (BOOL) setLineHeight:(int)n;

/******

Function name: restoreDefaultLineHeight

Bitmap command: ESC 2 (1b 32)

Function: Restore the default height of line

Parameter:

no

Return:

YES succeed

NO failed

*****/

+ (BOOL) restoreDefaultLineHeight;

/******

Function name: setChineseWordFormat

Bitmap command: FS! N (1c 21 n)

Function: Set the printing format of Chinese characters: double height, double width,

underline

Parameter:

isDoubleHeight NO: clear double height setting
YES: set to double height
isDoubleWidth NO: clear double width setting
YES: set to double width
isUnderLine NO: clear underline setting
YES: set to underline printing

Return:

YES succeed
NO failed

*****/

+ (BOOL) setChineseWordFormat:(BOOL)isDoubleHeight
doubleWidth:(BOOL)isDoubleWidth underline:(BOOL)isUnderLine;

/*****

Function name: setAsciiWordFormat

Bitmap command: ESC! N (1b 21 n)

Function: Set the printing format of standard ascii font A (12X24) or compressed ascii font B (9X17)

Parameter:

type 0: Standard ascii font A (12X24)
1: Compressed ascii font B (9X17)
isbold NO: clear bold setting
YES: set to bold printing
isDoubleHeight NO: Clear double height setting
YES: set to double height
isDoubleWidth NO: Clear double width setting
YES: set to double width
isUnderLine NO: clear underline setting
YES: set to underline printing

Return:

YES succeed
NO failed

*****/

+ (BOOL) setAsciiWordFormat:(int)type bold:(BOOL)isbold
doubleHeight:(BOOL)isDoubleHeight doubleWidth:(BOOL)isDoubleWidth
underline:(BOOL)isUnderLine;

/*****

Function name: putInBuf, please do not call it directly, it will be called automatically when printing with buffer

Function: Put the print data into the sending buffer

Parameter:

Data or control instructions to be printed

Return:

YES succeed

NO failed, insufficient sending buffer space

*****/

+ (BOOL) putInBuf:(NSData *)data;

/******

Function name: getFromBuf, please don't call it directly, it will be called automatically when printing with buffer

Function: fetch data from the buffer to the printer, up to 100 bytes

Parameter:

Get the data or control instructions to be printed

Return:

Fetches bytes

*****/

+ (NSData *)getFromBuf;

/******

Function name: SendTask, please do not call it directly, it will be called automatically when printing with buffer

Function: timed task of printing with buffer, execute once every 0.1 seconds, sending 100 bytes each time. If it fails, it will resend after 2 seconds waiting time.

Due to the limitations of the Bluetooth bottom layer, the size of data which sent directly to the printer can only about 2KB in a task. If the printed data size exceeds this limit, the subsequent data cannot be sent successfully. The buffer mode should be switched on, the data is buffered and then printed by a timed task.

*****/

+ (void) SendTask;

/******

Function name: isBuffEmpty, please do not call it directly, it will be called automatically when printing with buffer

Function: Determine whether the printer buffer is empty

Return:

YES The buffer is empty

NO The buffer is not empty

*****/

+ (BOOL) isBuffEmpty;

```

/*****
Function name: printPng
Function: print png pictures
Parameter:
    File name to be printed
Return:
    YES   The buffer is empty
    NO    The buffer is not empty
*****/
+ (BOOL) printPNG_JPG:(NSString *)filename offset:(int) xoff;

```

```

/*****
Function name: cutPaper
Function: Cut paper. Note that this function should be called at the beginning of the line,
for example, there is a carriage return or line feed on the previous line
Parameter:
    mode: paper cutting mode
    0, 1, 48, 49: half cut
    65, 66: Paper feeding (cutting position + [n × (vertical movement unit) inches]) and half cut
    dis: paper feeding distance, the unit is vertical movement unit (the default value of
vertical movement unit is a point of the printer, which can be set by GS P command)
Return:
    YES succeed
    NO failed
*****/
+ (BOOL) cutPaper:(int) mode feed_distance:(int) dis;

```

```

/*****
/*
The following is the realization of CPCL interface function to complete the L31 label
printing function;
qzfbeg begin 2015/12/07
*/
*****/

```

```

/*****
Function name: print
Function: Print with labels
Parameter:
horizontal:
0: normal printing, no rotation;
1: After the entire page rotates 180 degrees clockwise, then print

```

skip:

0: No positioning after printing, stop directly;

1: Position the label dividing line after printing, if there is no gap, stop after 30mm paper feeding

Return:

None

```
*****/
```

+ (void) print:(int) horizontal skipNum:(int) skip;

```
/******
```

Function name: pageSetup

Function: Set the size of the printing paper (printing area)

Parameter:

pageWidth: the width of printing area

pageHeight: the height of printing area

Return:

None

```
*****/
```

+ (void) pageSetup:(int) pageWidth pageHeightNum:(int) pageHeight;

```
/******
```

Function name: drawBox

Function: Printed border

Parameter:

lineWidth: Border line width

top_left_x: The x coordinate of the upper left corner of the rectangle

top_left_y: The y coordinate of the upper left corner of the rectangle

bottom_right_x: The x coordinate of the lower right corner of the rectangle

bottom_right_y: The y coordinate of the lower right corner of the rectangle

Return

None

```
*****/
```

+ (void) drawBox:(int) lineWidth leftX:(int) top_left_x leftY:(int) top_left_y rightX:(int) bottom_right_x rightY:(int) bottom_right_y;

```
/******
```

Function name: drawLine

Function: Print lines

Parameter:

lineWidth: Line width
 start_x: The x coordinate of line starting point
 start_y: The y coordinate of line starting point
 end_x: The x coordinate of the end point of the line
 end_y: The y coordinate of the end point of the line
 fullline: true:solid line false: dotted line

Return:
 None

```

*****/
+ (void) drawLine:(int) lineWidth startX:(int) start_x startY:(int) start_y endX:(int) end_x
endY:(int) end_y isFullline:(Boolean) fullline;

```

```

/*****

```

Function Name: drawText

Function: Print text

Fuction:

text_x Starting abscissa
 text_y Starting ordinate
 text Printed text content
 fontSize Font size

1:16 dot;

2:24 dot;

3:32 dot;

4:24 dot matrix doubled

5:32 dot matrix doubled

6:24 dot matrix enlarged twice

7:32dot matrix enlarged twice

Others: 24dot

rotate (Rotation angle):

0:no rotation; 1:Rotate 90 degrees; 2:180°; 3:270°

bold or not

0:No; 1:Yes

underline or not

false:No ;true:Yes

reverse or not

false:No;true:Yes

Return:

No

```

*****/
+ (void) drawText:(int) text_x textY:(int) text_y textStr:(NSString *) text fontSizeNum:(int)

```

fontSize rotateNum:(int) rotate isBold:(int) bold isUnderLine:(Boolean) underline
isReverse:(Boolean) reverse;

/*****

Function name:drawText

Function: Print text box

Parameter:

text_x Starting abscissa

text_y Starting ordinate

width Text box width

height Text box height

text Printed text content

fontSize :

1:16 dot;

2:24 dot;

3:32 dot;

4:24 dot matrix doubled;

5:32 dot matrix doubled;

6:24 dot matrix enlarged twice;

7:32dot matrix enlarged twice;

Others:24 dot

rotate (Rotation angle):

0:No rotation; 1:90 rotate 90degree; 2:180°; 3:270°

bold or not

0:No; 1:Yes

underline or not

false:No;true:Yes

reverse or not

false: No;true:Yes

Return:

None

*****/

+ (void) drawText:(int) text_x textY:(int) text_y widthNum:(int) width heightNum:(int) height
textStr:(NSString *) text fontSizeNum:(int) fontSize rotateNum:(int) rotate isBold:(int) bold
isUnderLine:(Boolean) underline isReverse:(Boolean) reverse;

/*****

Function name:drawBarCode

Function: Print 1D barcode

Parameter:

start_x One-dimensional code starting abscissa

start_y One-dimensional code starting ordinate

text Content

type:
 0:CODE39; 1:CODE128;
 2:CODE93; 3:CODEBAR;
 4:EAN8; 5:EAN13;
 6:UPCA; 7:UPC-E;
 8:ITF
 Linewidth: barcode width
 Height: barcode height

Return:
 None

```

*****/
+ (void) drawBarCode:(int) start_x startY:(int) start_y textStr:(NSString *) text typeNum:(int)
type roateNum:(int) rotate lineWidthNum:(int) linewidth heightNum:(int) height;

```

```

/*****

```

Function name:drawQrCode
 Function: Print 2D barcode
 Parameter:
 start_x Starting abscissa of 2D barcode
 start_y Starting ordinate of 2D barcode
 text Content
 rotate:
 0:No; 1:90; 2:180°; 3:270°
 ver QrCode width(2-6)
 lel QrCode error correction level (0-20)

Return:
 None

```

*****/
+ (void) drawQrCode:(int) start_x startY:(int) start_y textStr:(NSString *) text roateNum:(int)
rotate verNum:(int) ver lelNum:(int) lel;

```

```

/*****

```

Function name: drawGraphic
 Function: Print image
 Parameter:
 start_x Image starting point x coordinate
 start_y Image starting point y coordinate
 bmp_size_x Image width
 bmp_size_y Image height
 bmp Image

Return:

None

*****/

```
// + (void) drawGraphic:(int) start_x startY:(int) start_y bmpSizeX:(int) bmp_size_x  
bmpSizeY:(int) bmp_size_y bmpBitMap:(Bitmap) bmp;
```

/******

Function name:printerStatus

Function: Get printer status

Parameter:

None

Return:

"ok": success;

"no_paper": paper out;

"cover_open": printer cover is opened;

"connect_failed": communication failed;

*****/

```
+(NSString *) printerStatus;
```

/******

Function name:feed

Function: Position label

Parameter:

No

Return:

None

*****/

```
+ (void) feed;
```

/******

Function name:isChinese

Function: Determine whether it is Chinese

Parameter:

txtBytes: Byte data;

idx: Index position;

Return:

true: Chinese; false:English;

```
*****/
+ (Boolean) isChinese:(Byte *) textBytes index:(int)idx;

/*****
/*   qzfeng end 2015/12/07 */
*****/

/*****
Author:wangdongsai2016-6-28
Function name:drawBmp
Function: Read bmp data and encapsulate it into cpcl command to send print pictures
Parameter:
    startX: Horizontal starting coordinate;
    start_y: Vertical starting coordinate;
    bmp_Path: bmp Picture path;
Return:
    YES Succeed
    NO Failed

*****/
+ (BOOL) drawBmp:(int) start_x startY:(int) start_y bmpPath:(NSString *) bmp_Path;

@end
```