TSC Swift Library Instructions

To WiFi

1. openport(a,b)

Description: Open the socket.

Parameter:

a: String: IP address.

b: String: Port number.

To Bluetooth

2. openport_mfi()

Description: Start the Bluetooth port.

Parameter: none

To BLE

3. (NSMutableArray) searchBLEDevice(a)

Description: Search BLE Device during specific time.

Parameter:

a: Integer: Search time, like: 5

Need to use var variable to receive the NSMutableArray containing discovered devices.

4. openport_ble(a)

Description: Start the BLE port.

Parameter:

a: CBPeripheral: Device discovered by **searchBLEDevice** function

5. closeport(int delay)

Description: Close printer port by specific delay time.

Parameter:

Int; Delay time, like: 5

6. setup(a,b,c,d,e,f,g)

Description: Set up label width, label height, print speed, print density, sensor type, gap/black mark vertical distance, gap/black mark shift distance

Parameter:

a: string, sets up label width; unit: mm
b: string, sets up label height; unit: mm
c: string, sets up print speed, (selectable print speeds vary on different printer models)
1.0: sets print speed at 1.0"/sec
1.5: sets print speed at 1.5"/sec
2.0: sets print speed at 2.0"/sec
3.0: sets print speed at 3.0"/sec
4.0: sets print speed at 4.0"/sec
6.0: sets print speed at 6.0"/sec
8.0: sets print speed at 8.0"/sec
10.0: sets print speed at 10.0"/sec
12.0: sets print speed at 12.0"/sec
d: string, sets up print density
0~15，the greater the number, the darker the printing
e: string, sets up the sensor type to be used
0: signifies that vertical gap sensor is to be used
1: signifies that black mark sensor is to be used
f: string, sets up vertical gap height of the gap/black mark; unit: mm
g: string, sets up shift distance of the gap/black mark; unit:: mm; in the case of the average label, set this parameter to be 0.

7. clearbuffer()
Description: Clear
Parameter: None

8. barcode(a,b,c,d,e,f,g,h,I)
Description: Use built-in bar code formats to print
Parameter:
a: string; the starting point of the bar code along the X direction, given in points (of 200 DPI, 1 point=1/8 mm; of 300 DPI, 1point=1/12 mm)
b: string; the starting point of the bar code along the Y direction, given in points (of 200 DPI, 1 point=1/8 mm; of 300 DPI, 1 point=1/12 mm)
c: string
128 Code 128, switching code subset A, B, C
automatically
128M Code 128, switching code subset A, B, C
manually.

EAN128 Code 128, switching code subset A, B, C
automatically
25 Interleaved 2 of 5
25C Interleaved 2 of 5 with check digits
39 Code 39
39C Code 39 with check digits
93 Code 93
EAN13 EAN 13
EAN13+2 EAN 13 with 2 digits add-on
EAN13+5 EAN 13 with 5 digits add-on
EAN8 EAN 8
EAN8+2 EAN 8 with 2 digits add-on
EAN8+5 EAN 8 with 5 digits add-on
CODA Codabar
POST Postnet
UPCA UPC-A
UPCA+2 UPC-A with 2 digits add-on
UPCA+5 UPC-A with 5 digits add-on
UPCE UPC-E
UPCE+2 UPC-E with 2 digits add-on
UPCE+5 UPC-E with 5 digits add-on
d: string; sets up bar code height, given in points
e: string, sets up whether to print human recognizable interpretation (text)
or not.
0: prints no interpretation
1: prints interpretation
f: string; sets up rotation degrees
0: rotates 0 degree
90: rotates 90 degrees
180: rotates180 degrees
270: rotates 270 degrees
g: string; sets up narrow bar ratio, refer to TSPL user's manual
h: string; sets up wide bar ratio, refer to TSPL user's manual
I: string; bar code content

9. printerfont(a,b,c,d,e,f,g)

Description: Use printer built-in fonts to print
Parameter:

a: string; the starting point of text (character string) along the X direction, given in points
(of 200 DPI, 1 point=1/8 mm; of 300 DPI, 1 point=1/12 mm)
b: string; the starting point of text (character string) along the Y direction, given in points
(of 200 DPI, 1 point=1/8 mm; of 300 DPI, 1 point=1/12 mm)
c: string; built-in font type name, 12 kinds in sum
1: 8*/12 dots
2: 12*20 dots
3: 16*24 dots
4: 24*32 dots
5: 32*48 dots
TST24.BF2: Traditional Chinese 24*24 (Customized Font)
TST16.BF2: Traditional Chinese 16*16 (Customized Font)
TTT24.BF2: Traditional Chinese 24*24 (Telecommunication Code) (Customized Font)
TSS24.BF2: Simplified Chinese 24*24 (Customized Font)
TSS16.BF2: Simplified Chinese 16*16 (Customized Font)
K: Japan, Korean font 24*24, (Customized Font)
L: Japan Korean font 16*16 (Customized Font)
d: string; sets up the rotation degree of the text (character string)
0: rotates 0 degree
90: rotate 90 degrees
180: rotate 180 degrees
270: rotate 270 degrees
e: string; sets up the magnification rate of text (character string) along the X direction, range: 1~8
f: string; sets up the magnification rate of text (character string) along the Y direction, range: 1~8
g: string; prints the content of text (character string)

10. sendcommand(command)
Description: Sends raw data to printer
Parameter: Refer to TSPL for details

11. printlabel(a,b)
Description: Print label content
Parameter:

a: string; sets up the number of label sets
b: string, sets up the number of print copies

12. formfeed()
Description: Skip to next page (of label); this function is to be used after setup
Parameter: None

13. nobackfeed()
Description: disable the backfeed function
Parameter: None

14. windowsfont(a,b,c,d,e,f,g,h)
Description: Use Windows font to print text
Parameter:
a: Integer, the starting point of the text along the X direction, given in dots
b: Integer, the starting point of the text along the Y direction, given in dots
c: Integer, the font height, given in points.
d:Integer, rotation in counter clockwise direction
0 -> 0 degree
90-> 90 degree
180-> 180 degree
270-> 270 degree
e: Integer, font style
0-> Normal
1-> Italic
2-> Bold
3-> Bold and Italic
f: Integer, font with underline
1-> Without underline
g: String, font type face. Specify the true type font name. For example: Arial, Times new Roman.
h: String, text to be printed

15. downloadPCX(a,b)
Description: Download mono PCX graphics file to the printer
Parameter:
a: String; file name (including file retrieval path)
b: String, names of files that are to be downloaded in the printer memory (Please

use capital letters)

16. printer_status(int a)

Description: Response the printer status.

Parameter:

a: Int: Delay time.

Need to use var variable to receive the printer return the status.

17. (NSString*) smartbattery_status (a)

Description: Response the smart battery status.

Parameter:

a: Integer; type of return status.

0 : serial number	3 : temperature	6 : replacement threshold
1 : voltage	4 : discharged counts	7 : health
2 : capacity	5 : manufacture date	8 : maximum capacity

WiFi Example

```
import tscswift

var wireless = WiFi()
wireless.openport("10.0.10.194" as CFString, portnumber: 9100)

//var status = wireless.printer_status(delay: 2)
//<00> Normal, <01>Head opened, <10>Pause

wireless.sendcommand("DIRECION 1\r\n")
//wireless.setup("100", height: "120", speed: "4", density: "10", sensor: "0",
vertical: "2", offset: "0")
wireless.sendcommand("SIZE 100 mm, 120 mm\r\n")
wireless.sendcommand("SPEED 4\r\n")
wireless.sendcommand("DENSITY 10\r\n")
wireless.sendcommand("GAP 2 mm, 0 mm\r\n")
//wireless.sendcommand("BLINE 2 mm, 0 mm\r\n")//black mark

wireless.clearBuffer()

//Using downloaded font to print text
/*wireless.sendcommand("CODEPAGE UTF-8\r\n")
wireless.sendcommand_utf8("TEXT 30,30,\"ARIALUNI.TTF\",0,8,8,\"中文
English にほンゴ한국어\"\r\n")*/
wireless.sendcommand("TEXT 30,30,\"2\",0,2,2,\"1234567\"\r\n")
wireless.printerfont("30", y: "100", fontName: "2", rotation: "0",
magnificationRateX: "2", magnificationRateY: "2", content: "Printer Font
Test!")
wireless.windowsfont(30, y: 170, height: 32, rotation: 0, style: 0,
withUnderline: 0, fontName: "Arial", content: "中文 English にほンゴ한국어")
//fontName: Please refer to "iosfonts.com"

wireless.barcode("30", y: "300", barcodeType: "39", height: "70", readable:
"1", rotation: "0", narrow: "2", wide: "6", code: "12345")

//Download PCX
var mainBunle = Bundle.main
```

```
var absolutePath = mainBunle.path(forResource: "UL", ofType: "pcx")
wireless.downloadpcx(absolutePath!, name: "UL.PCX")
wireless.sendcommand("PUTPCX 300,250,\"UL.PCX\"\r\n")

wireless.printlabel(1, copies: 1)

wireless.closeport(4)
```

Bluetooth Example

```
import tscswift

var bt = Bluetooth()
bt.openport_mfi()

//var status = bt.printer_status(delay: 2)
//<00> Normal, <01>Head opened, <10>Pause

bt.sendcommand("DIRECION 1\r\n")
//bt.setup("100", height: "120",
speed: "4", density: "10", sensor: "0", vertical:"2", offset: "0")
bt.sendcommand("SIZE 100 mm, 120 mm\r\n")
bt.sendcommand("SPEED 4\r\n")
bt.sendcommand("DENSITY 10\r\n")
bt.sendcommand("GAP 2 mm, 0 mm\r\n")
//bt.sendcommand("BLINE 2 mm, 0 mm\r\n")//black mark

bt.clearBuffer()

//Using downloaded font to print text
/*bt.sendcommand("CODEPAGE UTF-8\r\n")
bt.sendcommand_utf8("TEXT
30,30,\"ARIALUNI.TTF\",0,8,8,\"中文English にほンゴ한국어\"\r\n")*/
bt.sendcommand("TEXT 30,30,\"2\",0,2,2,\"1234567\"\r\n")
bt.printerfont("30", y: "100", fontName: "2", rotation: "0", magnificationRateX:
"2", magnificationRateY: "2", content: "Printer Font Test!")
bt.windowsfont(30, y: 170, height: 32, rotation: 0, style: 0, withUnderline: 0,
fontName: "Arial", content: "中文 English にほンゴ한국어")
//fontName: Please refer to "iosfonts.com"

bt.barcode("30", y: "300", barcodeType: "39", height: "70", readable: "1",
rotation: "0", narrow: "2", wide: "6", code: "12345")

//Download PCX
var mainBunle = Bundle.main
var absolutePath =mainBunle.path(forResource: "UL", ofType: "pcx")
```

```
bt.downloadpcx(absolutePath!,
name: "UL.PCX")
bt.sendcommand("PUTPCX 300,250,\"UL.PCX\"\r\n")

bt.printlabel(1, copies: 1)

bt.closeport(4)
```

BLE Example

```
import tscswift
import CoreBluetooth
import tscswift

var ble = BLE()

for i in 0...deviceList.count-1
{
	var device = deviceList[i] as! CBPeripheral
	if(device.name == "Alpha-3R") //BLE Name
	{
		ble.openport_ble(device)
	}
}

//var status = ble.printer_status(2.0)
//<00> Normal, <01>Head opened, <10>Pause

ble.sendcommand("DIRECION 1\r\n")
//ble.setup("100", height: "120", speed: "4", density: "10", sensor: "0", vertical:
"2", offset: "0")
ble.sendcommand("SIZE 100 mm, 120 mm\r\n")
ble.sendcommand("SPEED 4\r\n")
ble.sendcommand("DENSITY 10\r\n")
ble.sendcommand("GAP 2 mm, 0 mm\r\n")
//ble.sendcommand("BLINE 2 mm, 0 mm\r\n")//black mark

ble.clearBuffer()

//Using downloaded font to print text
/*ble.sendcommand("CODEPAGE UTF-8\r\n")
ble.sendcommand_utf8("TEXT 30,30,\"ARIALUNI.TTF\",0,8,8,\"中文English
にほンゴ한국어\"\r\n")*/
ble.sendcommand("TEXT 30,30,\"2\",0,2,2,\"1234567\"\r\n")
ble.printerfont("30", y: "100", fontName: "2", rotation: "0", magnificationRateX:
"2", magnificationRateY: "2", content: "Printer Font Test!")
```

```
//ble.windowsfont(30, y: 170, height: 32, rotation: 0, style: 0, withUnderline: 0,
fontName: "Arial", content: "中文English にほンゴ한국어")
//fontName: Please refer to "iosfonts.com"

ble.barcode("30", y: "300", barcodeType: "39", height: "70", readable: "1",
rotation: "0", narrow: "2", wide: "6", code: "12345")

//Download PCX
/*var mainBunle = Bundle.main
var absolutePath = mainBunle.path(forResource: "UL", ofType: "pcx")
ble.downloadpcx(absolutePath!, name: "UL.PCX")
ble.sendcommand("PUTPCX 300,250,\"UL.PCX\"\r\n")*/

ble.printlabel(1, copies: 1)

ble.closeport(2)
//ble.closeport(10) //With Windowsfont
//ble.closeport(30)//With Windowsfont & downloadPCX
```