



# iOS Software Development Kit

## How to Use StarIO for POS Printers

This SDK contains an Xcode Objective-C project for iOS devices.

### Tools Needed:

- Xcode 7.0 or later
- StarIO iOS SDK

To use StarIO.framework version 3.14.0 or later, you need to add the following frameworks.

- External Accessory framework
- Core Bluetooth framework
- To upgrade StarIO.framework when you are using version 3.13.1 or earlier, you need to add Core Bluetooth framework into your project.

Please refer to [here](#) for more information.

**StarIO SDK Compatibility OS :        iOS 7.0 or later**

**StarIO SDK Compatibility Chart**

Device	CPU
<b>iPad 2</b>	Armv7
<b>iPad (3rd Generation)</b>	Armv7
<b>iPad (4th Generation)</b>	Armv7s
<b>iPad Air</b>	Arm64
<b>iPad Air 2</b>	Arm64
<b>iPad mini</b>	Armv7
<b>iPad mini 2</b>	Arm64
<b>iPad mini 3</b>	Arm64
<b>iPad mini 4</b>	Arm64
<b>12.9-inch iPad Pro</b>	Arm64
<b>9.7-inch iPad Pro</b>	Arm64
<b>iPhone 4s</b>	Armv7
<b>iPhone 5</b>	Armv7s
<b>iPhone 5s</b>	Arm64
<b>iPhone 5c</b>	Armv7s
<b>iPhone 6</b>	Arm64
<b>iPhone 6 Plus</b>	Arm64
<b>iPhone 6s</b>	Arm64
<b>iPhone 6s Plus</b>	Arm64
<b>iPhone 7</b>	Arm64
<b>iPhone 7 Plus</b>	Arm64
<b>iPod touch (5th Generation)</b>	Armv7
<b>iPod touch (6th Generation)</b>	Arm64

*Note:* iPad, iPhone, iPod, and iPod touch are trademarks of Apple Inc., registered in the U.S. and other countries. iPad Air and iPad mini are trademarks of Apple Inc. IOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

## Table of Contents

- ❖ [About this Manual](#)
- ❖ [Star Printer Compatibility Chart](#)
- ❖ [Connecting a Star Printer to an iOS Device](#)
- ❖ [Getting Started](#)
- ❖ [Using the SDK with Star Micronics POS Printers](#)
- ❖ [Overview of how this iOS SDK is designed](#)
- ❖ [StarIO Framework](#)
  - [How to add StarIO.framework into your project](#)
- ❖ [The StarIO Methods Overview](#)
  - [SMPort Class](#)
  - [SMBluetoothManager Class](#)
- ❖ [Functionality](#)
  - [Port Discovery](#)
  - [Help](#)
  - [Get Firmware Information](#)
  - [Get Status](#)
  - [Sample Receipt Printing](#)
  - [Open Cash Drawer](#)
  - [1D Barcodes](#)
  - [2D Barcodes](#)
  - [Cut](#)
  - [Text Formatting](#)
  - [Raster Graphics Printing](#)
  - [Image File Printing](#)
  - [AllReceipts](#)
  - [Bluetooth Pairing + Connect](#)
  - [Bluetooth Disconnect](#)
  - [Bluetooth Setting](#)
- ❖ [Tips for software application development when using StarIO](#)
- ❖ [Additional Resources](#)
- ❖ [SDK Version History](#)

## About this Manual

This manual is designed to help you understand StarIO and how to build an iOS application to interact with Star Micronics Thermal POS Printers. It is important to understand the basics of the Objective-C language. Although this SDK is for iOS, there are SDKs available for many different operating systems and programming languages at our website in the Developers section. Check the Developers section of our site for the newest SDKs, technical documentation, FAQs, and many more additional resources.

### Key Legend:

*Warning*



Explains potential issues

*Avoid Doing This*



Explains things not to do

*Note*



Provides important information and tips

### CAUTION:

- The information in this manual is subject to change without notice.
- STAR MICRONICS CO., LTD. has taken every measure to provide accurate information, but assumes no liability for errors or omissions.
- STAR MICRONICS CO., LTD. is not liable for any damages resulting from the use of information contained in this manual.
- Reproduction in whole or in part is prohibited.

© 2012 - 2017 Star Micronics Co., Ltd.

## Star Printer Compatibility Chart

The below chart summarizes the Star Printer Models supported on iOS Operating Systems.

### ◆ Line Mode / Raster Mode

✓ : Line Mode only    ✓ : Raster Mode only

Star Printer		Port Discovery	Get Firmware Information	Get Status	Sample Receipts	Sample Receipts (UTF-8)	Open Drawer	1D Barcodes	2D Barcodes	Cut Patterns	Text Formatting	Text Formatting (UTF-8)	Raster Graphics	Image File printing	AllReceipts	Bluetooth Pairing + Connect	Bluetooth Disconnect
Model	Interface																
TSP100IIIW	Ethernet	✓	✓ ***	✓	✓		✓						✓	✓	✓		
TSP100IIILAN	Ethernet	✓	✓ ***	✓	✓		✓						✓	✓	✓		
TSP100IIIBI	Bluetooth	✓	✓	✓	✓		✓						✓	✓	✓	✓	
TSP100IIIU	USB	✓	✓	✓	✓		✓						✓	✓	✓		
TSP100LAN	Ethernet	✓	✓ ***	✓	✓		✓						✓	✓	✓		
TSP650	Ethernet	✓	✓	✓	✓		✓	✓		✓	✓		✓	✓	✓		
TSP650II	Ethernet,	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	Bluetooth	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FVP10	Ethernet	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓		
	Bluetooth	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
TSP700II	Ethernet	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓		

	Bluetooth	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
TSP800II	Ethernet	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓		
	Bluetooth	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
TUP500	Ethernet	✓	✓	✓	✓		✓	✓	✓	✓	✓		✓	✓	✓		
TSP100U*	USB <Via an Apple AirPort>		✓ ** ***	✓	✓		✓						✓	✓	✓		
TSP100GT*	USB <Via an Apple AirPort>		✓ ** ***	✓	✓		✓						✓	✓	✓		
TSP100ECO*	USB <Via an Apple AirPort>		✓ ** ***	✓	✓		✓						✓	✓	✓		
SP700	Ethernet	✓	✓	✓	✓		✓			✓	✓						
	Bluetooth	✓	✓	✓	✓		✓			✓	✓					✓	✓

\* USB interface printer works via an Apple Airport Express. Please refer to the item of Ethernet and " Use share printer function with Apple AirPort Express" when connecting.

\*\* When using Apple AirMac Series with a USB printer, it returns an empty string.

\*\*\* It is impossible to get the firmware version of TSP100U, TSP100GT, TSP100LAN, TSP100ECO, TSP100IIIW and TSP100LAN.

**Note 1:** The TSP100 Series does not support the 1D Barcodes, 2D Barcodes, Cut Patterns, and Text Formatting samples in this application. These samples were written using Star Line Mode, while the TSP100 natively understands Raster Mode. Kindly reference Star Micronics' Raster Mode commands in the programming manual for the Cut and Text Formatting commands. It is recommended to print barcodes as graphics for this model.

**Note 2:** This SDK offers the most popular features, but not all printer functionality has been included (such as Presenter Options for the TUP500). The commands not included in this SDK application are available in the Line Mode Programming Manual.

## Connecting a Star POS Printer to an iOS Device

### Ethernet Interface

Star POS Printers ship with DHCP enabled by default. If your network supports DHCP, be sure to make the necessary configurations so that your Star POS Printer will automatically get an IP Address.

Use Star POS Printers with the #9100 Multi Session disabled. The setting can be confirmed by Test Print which can be executed by holding down the printer's feed button while turning the printer on.

Refer to "guidelines-ethernet\_en.pdf" for how to confirm and change the #9100 Multi Session setting and how to set the Static IP Address.

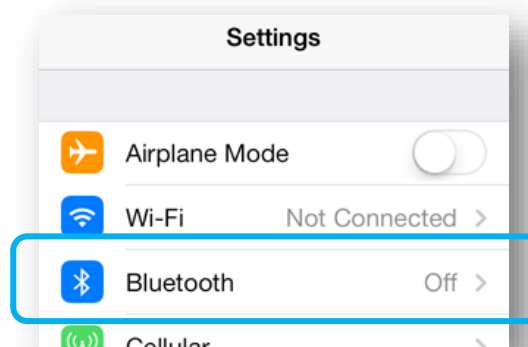
You can set up your POS Printers which can connect to TCP/IP by using Star Setting Utility\*(except for TSP100 Series) or TSP100III Utility\*(TSP100 and TSP100III). Please download it from App Store.

\* Star Setting Utility and TSP100III Utility do not support printers which are not assigned an IP Address. (IP Address: 0.0.0.0)

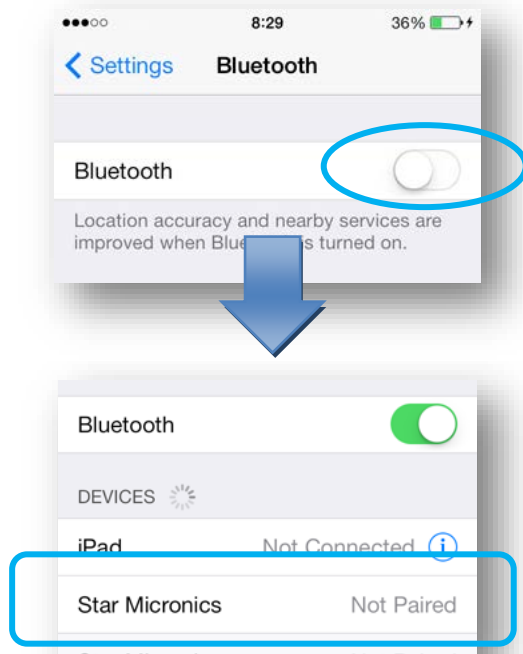
### Bluetooth Interface

#### ◆Pairing

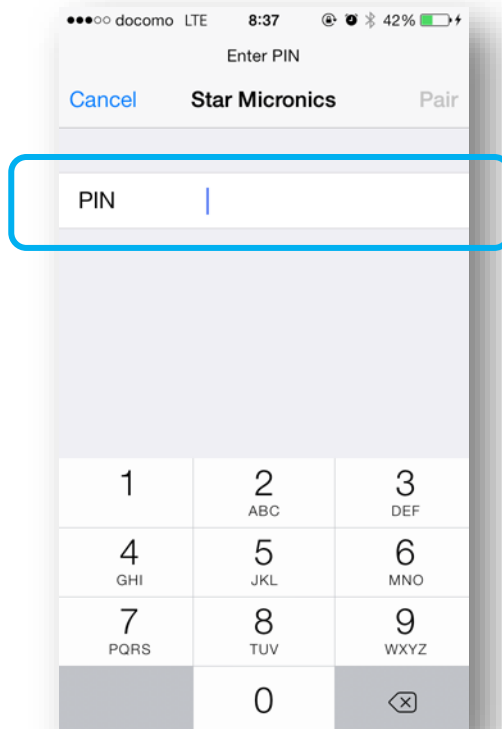
1. Place your Star POS device within connection range of the iOS device you want to pair with and turn the power on.  
If SSP is used by the Star POS device, press the PAIR button for 5 seconds or more to start pairing.
2. Tap Settings > Bluetooth.



3. Tap Bluetooth to turn it on. Your iOS device searches and displays the Bluetooth devices in range. Tap the Star portable printer you want to pair with.

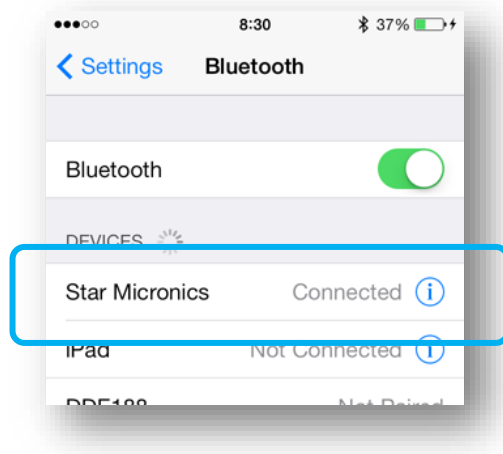


4. If a PIN code is used for Star Bluetooth device pairing, enter the PIN and tap Pair.





5. When the pairing is complete, you'll see this message.



◆How to change the Bluetooth Device Name

The Star Setting Utility can be downloaded from Apple App Store to change the iOS Port Name.

To confirm iOS Port Name, select [Settings]-[General]-[About] after Bluetooth pairing is established. The iOS Port Name will be shown under the Bluetooth address.

## USB Interface <TSP100IIIU only>

Connect the printer and the iOS device with a Lightning cable.

### ◆How to debug the USB printer

- The following descriptions, procedures and URL are subject to changes when Xcode versions upgrade or site updates. This document is based on our verification using Xcode 8.3 of MacOS 10.12.4 on April 4, 2017.

With USB printers, iOS device Lightning port is occupied by the communication with printer, so is not available for Xcode debug function.

However you can make use of the following method to confirm logs and to use Instruments:

#### 1. Log output

NSLog() output can be saved to files by executing the following code in the application (here it is saved to a file "xcode.log"):

```
NSString *tmpDirPath = NSTemporaryDirectory();
NSString *path = [NSString stringWithFormat:@"%@@xcode.log", tmpDirPath];

freopen([path cStringUsingEncoding:NSUTF8StringEncoding], "w+", stderr);
```

< How to extract the log >

- 1) Connect iOS device to mac with Lightning cable
- 2) Open Xcode, select "Window" - "Devices" on the main menu, and open "Devices" window
- 3) Select iOS device in the "Devices" pane in the left
- 4) Select/click the application to extract log in the "Installed Apps" and save the application in mac by clicking the gear-shaped icon and selecting "Download Container.."
- 5) Right-click the saved file and then select "Show Package Contents"  
You will see the content of the application then extract xcode.log from the tmp folder.

#### 2. Instruments

See Apple's Instruments User Guide (English) for how to use Instruments through network.

<https://developer.apple.com/library/content/documentation/DeveloperTools/Conceptual/InstrumentsUserGuide/WorkingwithTargets.html>

## Getting Started

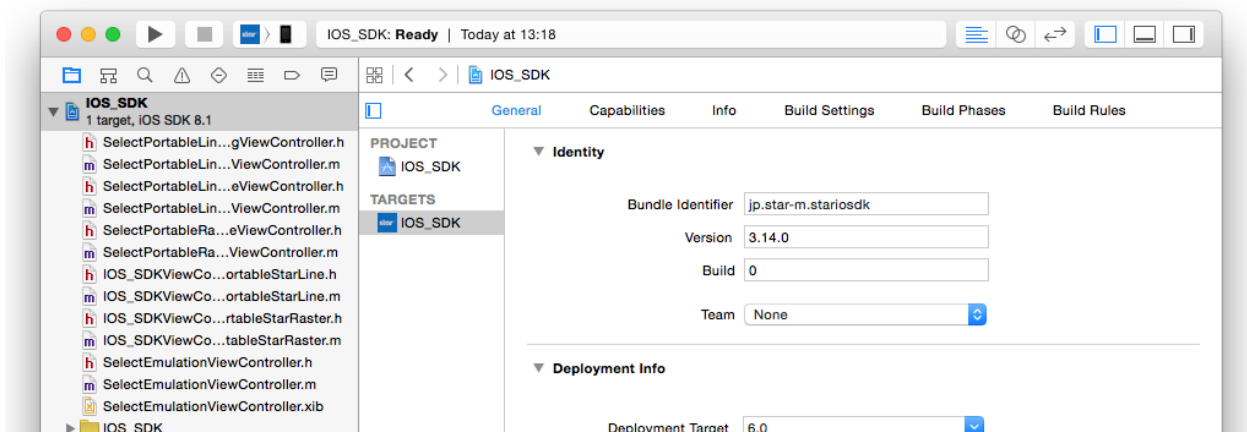
To build an iOS project, Xcode are needed. These tools are available in one package from the [Apple Developer Site](https://developer.apple.com) or Mac App Store. It is important to note that in order to produce applications that will actually run on an iOS device, you must be part of the Apple Developer Program, which requires a yearly subscription. While it is possible to obtain these tools from Mac App Store as stated above, your application will only be able to run in the iOS Simulator and will not install on an actual device.

It is assumed Xcode have already been installed on your Mac at this point. Should you need assistance or additional information, visit the [Resources](#) section of the Apple Developer Site.

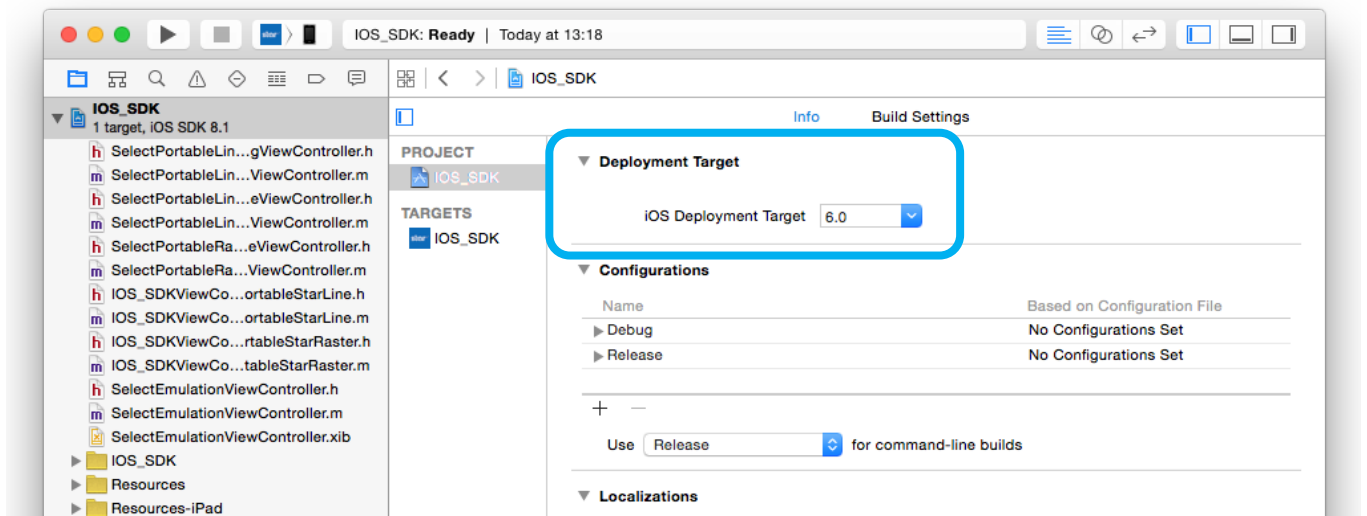
How to open the Star iOS SDK project in Xcode:



1. Unzip the Star iOS SDK folder and open it.

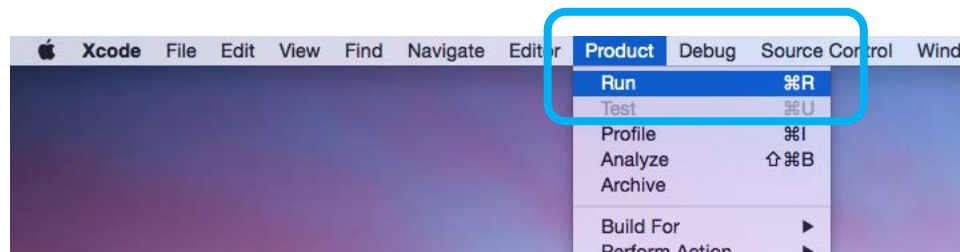


2. Open IOS\_SDK.xcodeproj.



- .3. Set the iOS Deployment Target to 7.0 or later.

Running the project:



1. Use the shortcut ⌘R or click Product in the top menu bar and then Run.

## Using the SDK with Star Micronics Printers

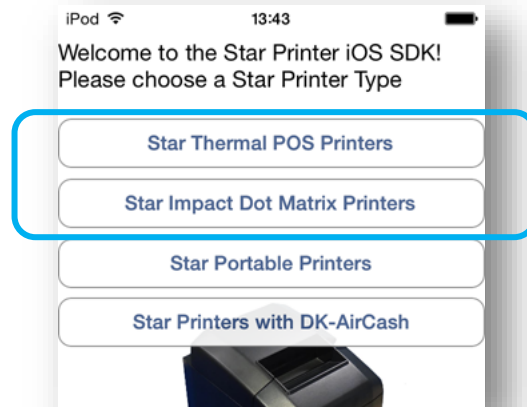
Please make sure you have a [compatible Star Micronics POS Printer Model](#).

### Port Name and Interface Relation:

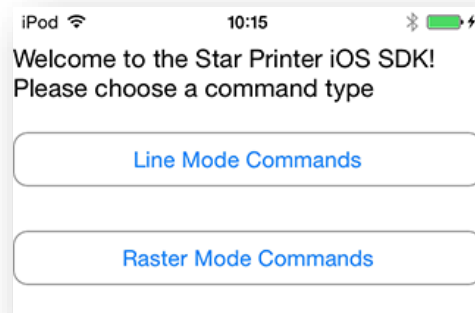
StarIO uses specific port names to identify what port will be used. These are very important to understand because not following the naming convention correctly will fail to communicate with the printer.

Interface	Port Name
Ethernet I/F (TCP/IP)	TCP: "IP Address"
Bluetooth	BT: "iOS Port Name" ✖
USB	USB: "iOS Port Name"

### Using a POS Printer



1. To use thermal printers, tap "Star Thermal POS Printers".  
And to use dot printers, tap "Star Impact Dot Matrix Printers".



2. Tap the desired command type to access the samples for that mode. The mode chosen results in which samples can be sent to the printer.

When "Star Impact Dot Matrix Printers" is selected, the screen will jump to the corresponding Line Mode Command page.

### Line Mode

Printers accept commands and print data line-by-line. The data is transferred to the printer in small pieces, allowing developers to customize receipt output with commands in any place they are needed. This mode alone can only make use of Device Fonts installed on the printer, which can be less visually appealing than TrueType Fonts.

[Line Mode is supported by these models:](#)

TSP650, TSP650II, FVP10, TSP700II, TSP800II, TUP500, SP700

### Raster Mode

Printers receive all print data graphically, allowing them to natively support the printing of engaging TrueType Fonts and output receipts at a lightning fast pace. Coding Raster commands is more complicated than Line Mode commands, as Raster commands require the entire receipt to be generated in graphical data before being sent to the printer.

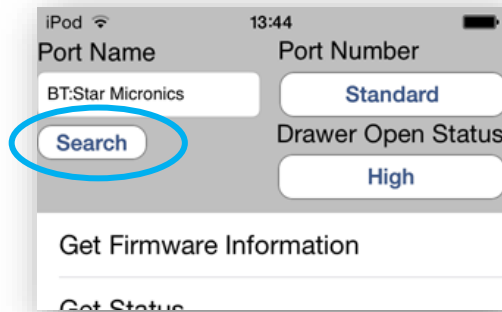
[Raster Mode is supported by these models:](#)

TSP100IIIU, TSP100IIIBI, TSP100IIIW, TSP100IIILAN, TSP100LAN, TSP100U, TSP100GT, TSP100ECO, TSP650, TSP650II, FVP10, TSP700II, TSP800II, TUP500

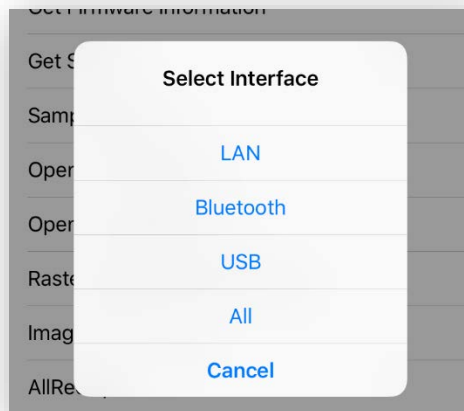
## Printers settings

### · To search for the printer and configure connection

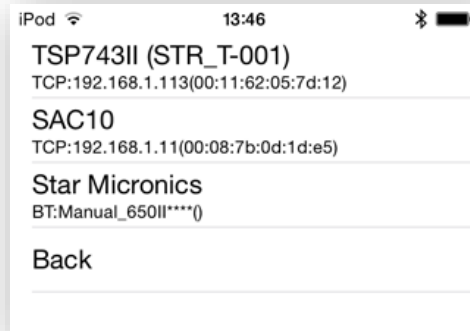
1. Tap “Search” to find all connected Star Printers.



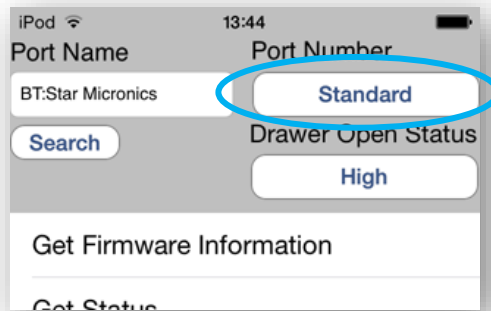
2. Tap the interface type of the printer you want to connect to.  
When tapping "LAN", the model names of ethernet printers you can connect to are displayed.  
When tapping "Bluetooth", the port names of paired printers you can connect to are displayed.  
When tapping "USB", the model names of usb printer connected by Lightning cable are displayed.



3. Tap the name of printer you want to connect to.



4. In Ethernet, configure the TCP Port by clicking the “Standard” dropdown if necessary. This sample application allows you to choose any port from 9100 to 9109. Configuring the port might be necessary when using a router such as Apple’s AirPort Express.

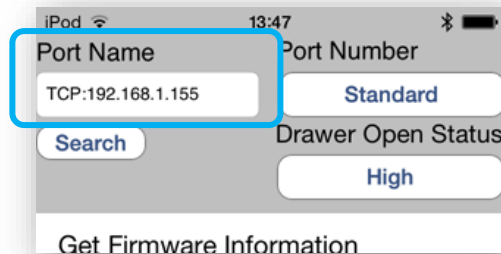




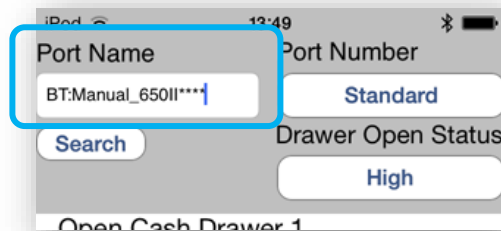
·To configure connection manually

1. Type the IP Address or the Bluetooth Device Name manually in the "PortName" field as shown below.

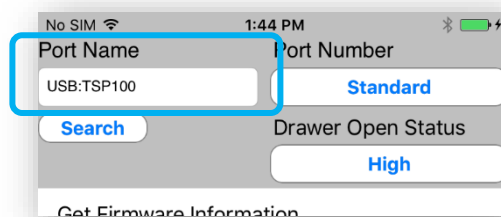
In Ethernet, if manually entered, type: **TCP:<IP Address>** \*Without brackets



In Bluetooth, if manually entered, type: **BT:<iOS Port Name>** \*Without brackets



In USB, if manually entered, type: **USB:<iOS Port Name>** \*Without brackets



## Overview of How This iOS SDK is Designed

This overview will touch briefly on key components of the SDK.

All functionality is located in the IOS\_SDK project and IOS\_SDK target.

Run the program from the IOS\_SDKViewController.m file; this source code is the starting point for both POS and Mobile Printers.

See how specific functions work by clicking on the other source files. For example, “code128.m” corresponds to the 1D barcode Code128 in the GUI.

It is important to note that not every function is available for both printer types. The first page of each SDK manual shows which functions are supported. They are listed again here for convenience:

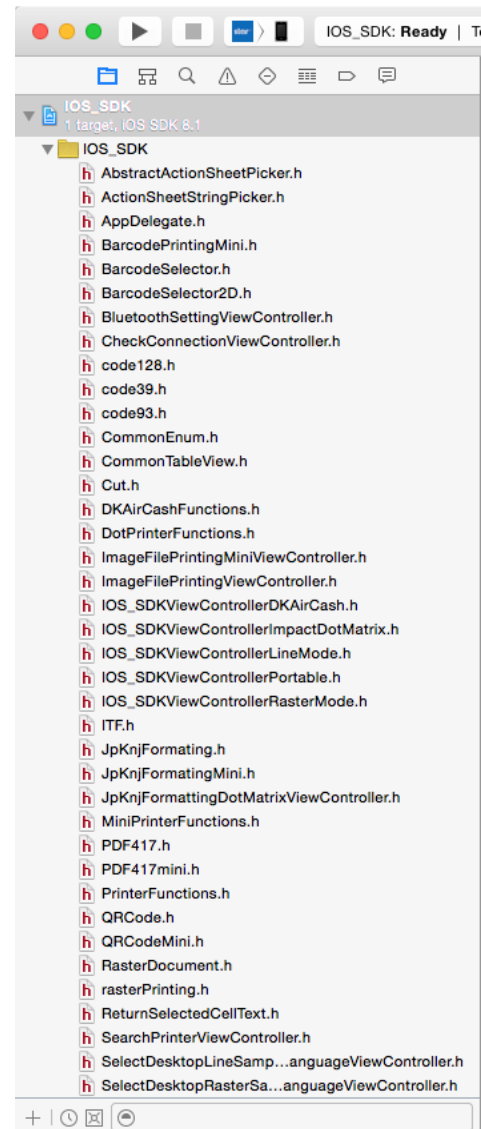
### POS Printers

- No Magnetic Stripe Reader support

### Portable Printers

- No Cash Drawer support
- No Cut Pattern support

Source files containing “Mini” are sample codes for portable printer models with ESC/POS mode.  
StarBitmap.m applies to both printer types.

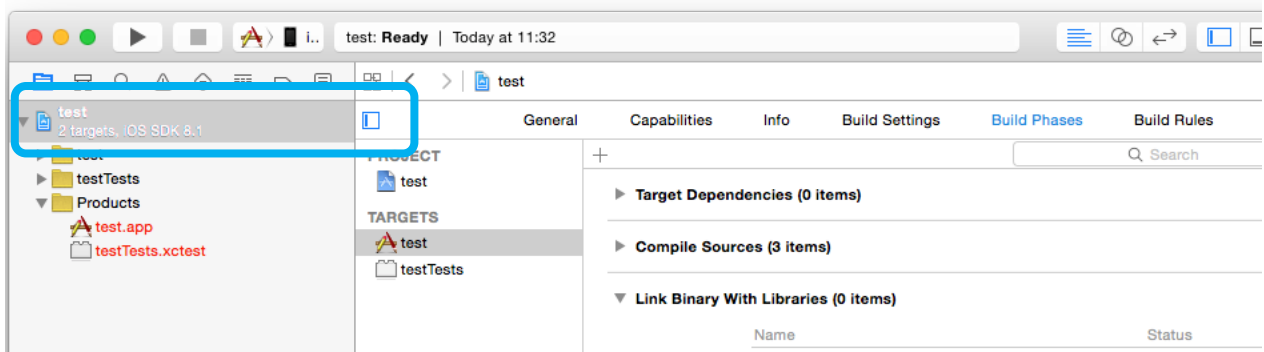


## The StarIO Framework

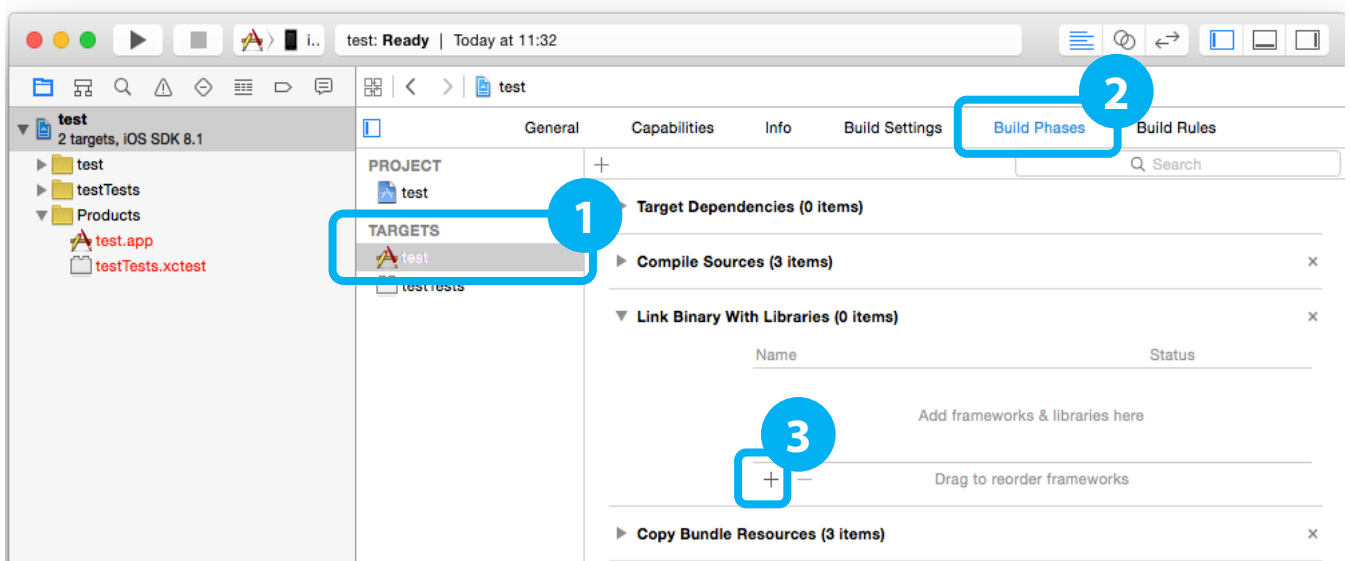
The StarIO framework is already included when the Star iOS SDK is loaded in Xcode; there's no need to include it again when testing our SDK. However, when you are building your own application, it is necessary to add the StarIO framework into it to utilize the StarIO methods.

### ◆When building a new application

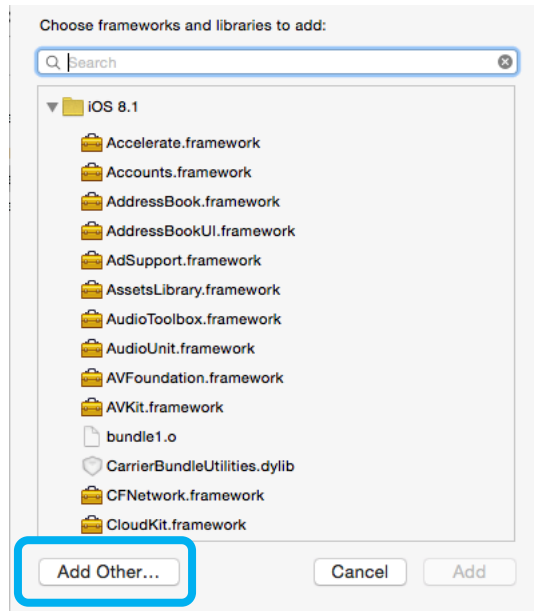
#### 1. Add StarIO.framework into your project



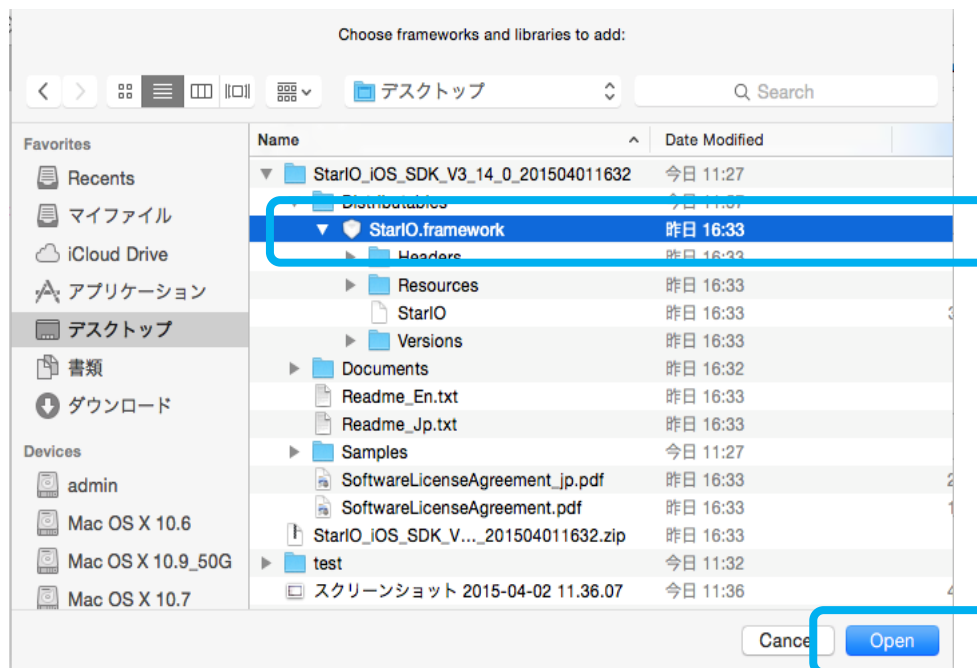
#### 1. Click the created project.



#### 2. Open a target, click the Build Phases tab, click the + of Link Binary With Libraries.

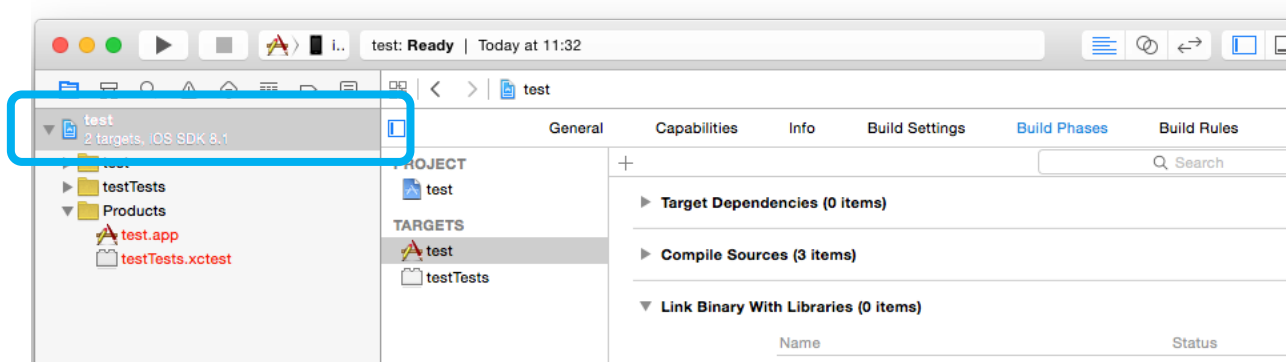


3. Click the **Add Other...** button.

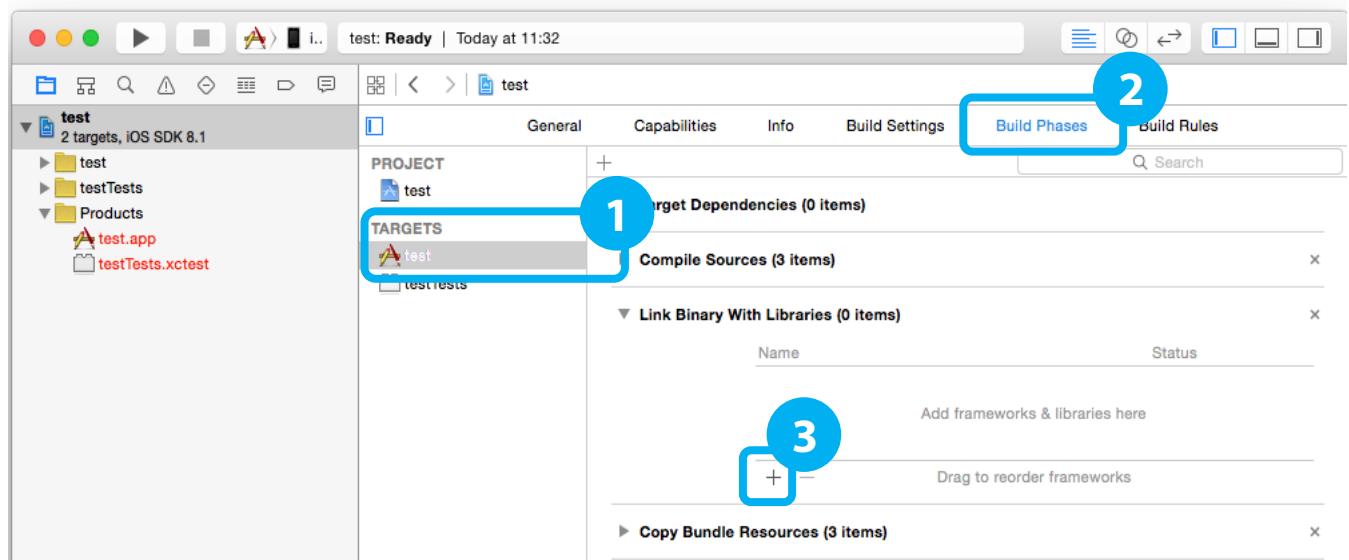


4. Browse to the location of where the Star iOS SDK was unzipped and select StarIO.framework. Then click Open.
5. The framework is added to your project and all StarIO methods are now available to you.

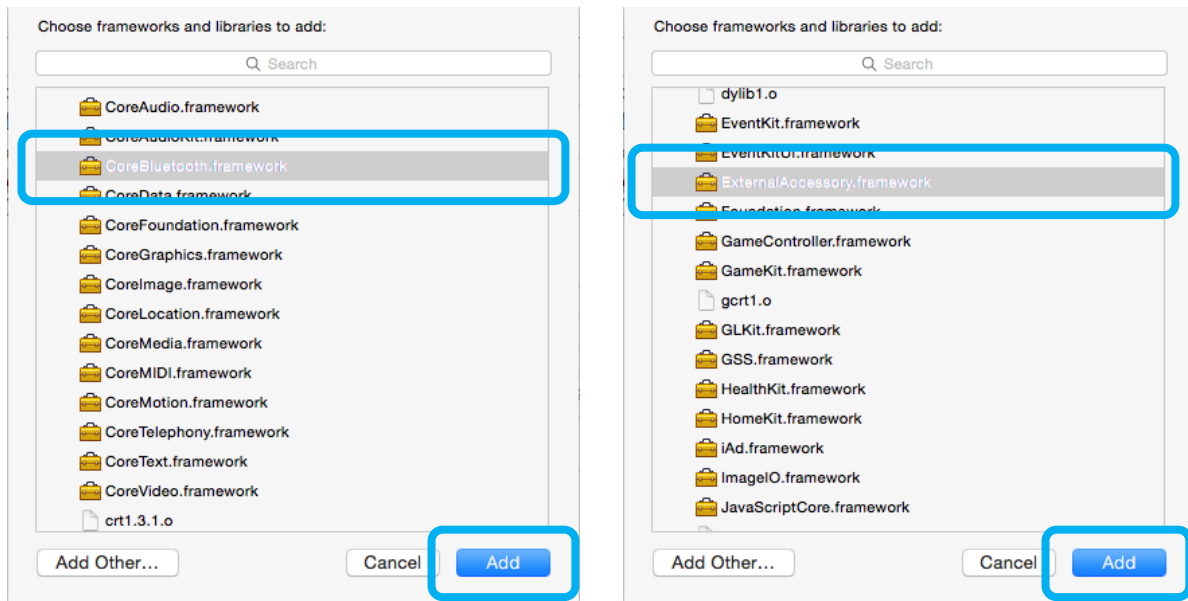
## 2. Add other framework into your project.



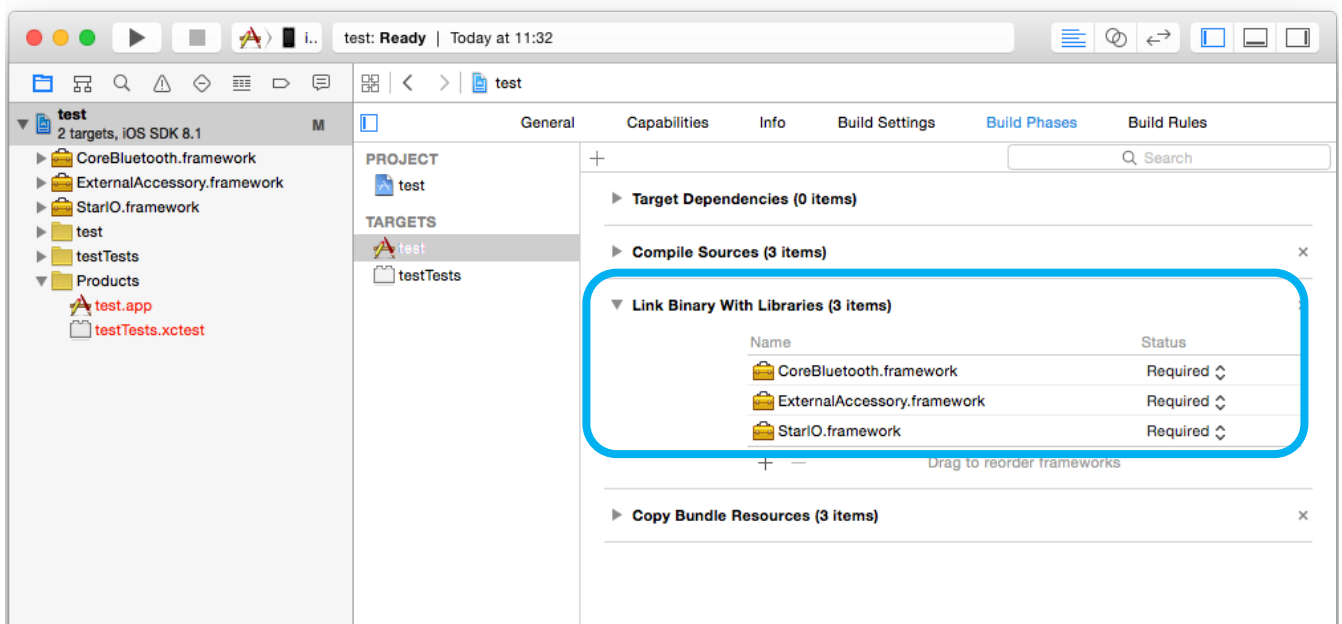
### 1. Click the created project.



### 2. Open a target, click the Build Phases tab, click the + of Link Binary With Libraries.



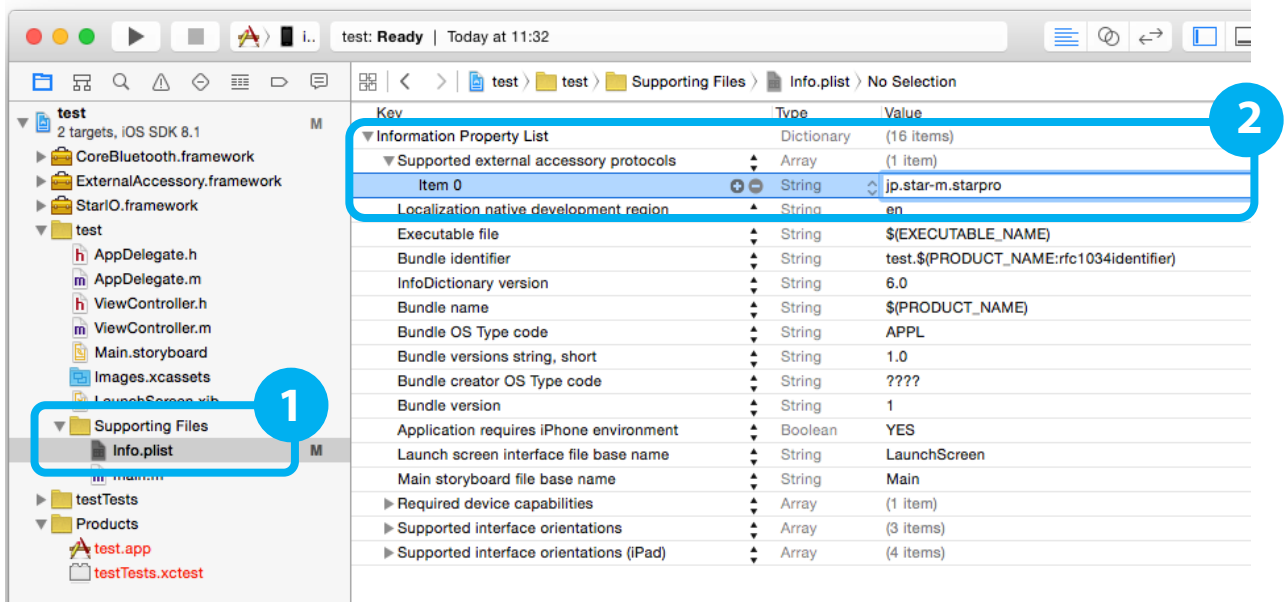
3. External Accessory framework and Core Bluetooth framework are added respectively. Select the framework, click the Add button.



4. Check if the necessary framework has been added.

### 3. Edit information property list (Bluetooth/USB printer only)

*Note: Please do not apply this, if you are using Ethernet interface.*



1. Click on the information property list file (default: Info.plist file).
2. Add the Supported external accessory protocols key. Click the triangle of this key and set the value for the Item 0 to jp.star-m.starpro.
4. You have finished editing the information property list.

### ◆Version up of StarIO.framework

---

1. Delete StarIO.framework from your project.
2. Copy new StarIO.framework
3. Clean the Xcode project.  
-Open the Xcode project and select [Build]-[Clean] from the menu.
4. Build the Xcode project.



To refer to the new StarIO.framework without deleting the existing StarIO.framework, surely confirm the “framework search path” setting of the Xcode project.

If the old path of the StarIO.framework remains in front of the “framework search path”, the previous StarIO.framework will be used.



## The StarIO Methods Overview

### SMPort Class:

#### ● Property

<code>portName</code>	Acquires the printer port name.
<code>portSettings</code>	Acquires the port settings.
<code>timeoutMillis</code>	Acquires and specifies the timeout time for internal control and API.
<code>endCheckedBlockTimeoutMillis</code>	Acquires and specifies the timeout time for endCheckedBlock method.

`-(NSString *)portName`

Specifies the port of the printer.

`-(NSString *)portSettings`

Acquires the port settings.

`-(u_int32_t)timeoutMillis`

Acquires and specifies the timeout time for internal control and API. (unit: millisecond)

`@property(assign, readonly) u_int32_t endCheckedBlockTimeoutMillis`

It obtains and sets endCheckedBlock method timeout value [unit: ms]

If it takes long time to print, stand-by time for print completion in endCheckedBlock method can be extended by increasing this value.

Default value is the timeout value designated by getPort method.



Timeout length is 10 seconds if specified less than 10 seconds.

When [Data timeout function setting] is used in the portSetting parameter of the getPort method, it should be at least 3 seconds longer than the time specified for the data timeout function. If it is set to less than 3 seconds, it will be controlled internally to be 3 seconds longer automatically.

## ●Method

### getPort

```
+ (SMPort *) getPort: (NSString *) portName : (NSString *) portSettings : (u_int32_t) TimeoutMillis)
```

GetPort is what you will be using to “open” the port to the printer.

#### Parameters:

portName        - Specify the communication port to the printer.

*Ex.    @"TCP:192.168.1.2" ( In Ethernet )*  
        *@"BT:StarMicronics" ( In Bluetooth )*  
        *@"USB:TSP100" (In USB)*

[Use share printer function with Apple AirPort Express]

Set AirPort Express IP Address for portName.

*Ex. @"TCP:192.168.1.2"*

portSettings    - Use the blank (@""") for portSettings.

- Specify @"d[value]" for portSettings to set the Data timeout function\* ON.

#### \* Data timeout function

In case an error occurs while printing or no data is sent to the printer during a predetermined time, the data canceling function will be executed.

To set the time value, specify from 0 to 255 (unit: second) in [value].

The default value is 3 seconds. If a value outside of the range is specified, it will be ignored.

This function prevents next print data from being printed incorrectly when the Bluetooth connection is disconnected during data transmission.



The data timeout function is supported by TSP650II and TSP800 with F/W version 2.0 or later , TSP700II with F/W version 5.0 and TSP100IIIBI with F/W version 1.0.

*Ex. portSettings = "d127" ... In case the data canceling function is executed when no data is sent for 127 seconds during printing*

timeoutMillis - Specify the timeout time for internal control and API.

*Note: this parameter guarantees that all of the below APIs will complete in a bounded amount of time, but does NOT guarantee the exact timeout length).*

[Use share printer function with Apple AirPort Express]

Set port number for **portSettings**.

Increase the port number in sequential order from 9100 to 9109 until communication is successful.

*Ex. @"9100"*



When [Data timeout function setting] is used in the portSetting parameter, it should be at least 3 seconds longer than the time specified for the data timeout function. If it is set to less than 3 seconds, the timeout time of the endCheckedBlock method will be controlled internally to be 3 seconds longer automatically.

#### Returns:

An instance of SMPort class. It returns "nil" if it fails to generate communication port.



After executing getPort, please do not forget releasePort before executing the next getPort.  
Otherwise the communication may return nil.

```
//The following would be an actual usage of getPort:
```

```
SMPort *port = nil;
NSString *portName = @"TCP:192.168.0.5";
NSString *portSettings = @"";
@try
{
    port = [SMPort getPort:portName :portSettings :10000];
}
@catch (PortException)
{
    //There was an error opening the port
}
```



Always use a `try`, `catch` when using `getPort`. If the port cannot be opened because of connection problems, your program will crash unless you use a `try`, `catch` like the above example.



For the Bluetooth I/F, close a port when it is not in communication with a printer for 30 seconds or more.  
It is recommended to open and close a port per transaction.

## searchPrinter

```
+ (NSArray *) searchPrinter;
+ (NSArray *) searchPrinter: (NSString *) target
```

searchPrinter detects printers in LAN and paired Bluetooth printers and returns search result as NSArray..

NSArray of return value includes instance of PortInfo Class.

PortInfo class of return value includes, PortName, MAC address(Ethernet model only), ModelName and you can get them by portName, macAddress, and modelName property.

And you can use Port Name as Argument value of getPort.

When the Argument value of target is specified, it detects either Ethernet printers or Bluetooth printers.

### Parameters:

Target - When @"TCP:" is specified, Ethernet printers will be detected.

When @"BT:" is specified, Bluetooth printers will be detected.



This API do not guarantee the discovery of devices.

iOS6 is required to specify the MAC address in Bluetooth.

Other iOS versions cannot use this function.

//The following would be an actual usage of searchPrinter:

```
NSArray *portArray = [[SMPort searchPrinter] retain];
for (int i = 0; i < portArray.count; i++) {
    PortInfo *port = [portArray objectAtIndex:i];
    NSLog(@"Port Name: %@", port.portName);
    NSLog(@"MAC Address : %@", port.macAddress);
    NSLog(@"Model Name: %@", port.modelName);
}
[portArray release];
```

The above example shows both of printers in LAN and Bluetooth printers being detected and search result being output to the log.

## readPort

```
- (u_int32_t) readPort: (u_int8_t *) readBuffer : (u_int32_t *) offset : (u_int32_t) size;
```

This method reads data from the device. Only use this if you really need to read raw bytes from the printer.



**Do not use this method to read raw status.**  
Use `getParsedStatus::` for getting status.

### Parameters:

readbuffer - A Byte Array buffer into which data is read.

offset - specifies where to begin writing data into the readBuffer[]

size - Total number of bytes to read.

### Returns:

The number of bytes that were actually read. Under some interface types, this function will succeed even when no data was read in. Your application should call this function a limited number of times until the expected data has been read in or until an application determined retry threshold has been reached.

### Throws:

[PortException](#) - when a communication failure occurs

## releasePort

```
+ (void) releasePort: (SMPort *) port;
```

This function closes a connection to the port specified.

### Parameters:

port - [StarIOPort](#) type representing a previously initialized port.



After executing `getPort`, please do not forget `releasePort` before executing the next `getPort`.  
Otherwise the communication may return nil.

## writePort

```
-(u_int32_t) writePort: (u_int8_t const *) writeBuffer: (u_int32_t) offset: (u_int32_t) size;
```

This method writes data to the device. Use this to print to the printer, send commands, etc. The following is an example of how to use this method:

To check the completion of printing, run `beginCheckedBlock` before and `endCheckedBlock` after this method.

See the sample code [here](#).

※Remember to use a Try, Catch for safe programming practices.

The SDK has code in “PrintTextWithPortName” that will show you how to verify data transmission to the printer.

### Parameters:

`writeBuffer` - Contains the output data in a byte array.

`offset` - Specifies where to begin pulling data from `writeBuffer` .

`size` - Number of bytes to write.

### Returns:

The number of bytes that were actually written. Under some interface types, this function will succeed even when no data was written out. Your application should call this function a limited number of times until all the data has been written out or until an application determined retry threshold has been reached.

### Throws:

`PortException` - when a communication failure occurs

## getParsedStatus

```
-(void) getParsedStatus: (void *) starPrinterStatus: (u_int32_t) level;
```

This method retrieves detailed status from the printer with StarIO.

### Returns:

[StarPrinterStatus](#) structure giving the current device status

### Throws:

[PortException](#) - when a communication failure occurs

This method uses a class structure that is included with StarIO called [StarPrinterStatus](#). This structure gives the printer's status in both boolean and binary form. Create the [StarPrinterStatus](#) object in your project by doing the following:

```
StarPrinterStatus_2 printerStatus;
[port getParsedStatus:&printerStatus :2];
if (printerStatus.offline == SM_TRUE)
{
    if (printerStatus.coverOpen == SM_TRUE) {
        //There was a cover open error
    }
    else if (printerStatus.receiptPaperEmpty == SM_TRUE) {
        //There was a receipt paper empty error
    }
    else {
        //There was a offline error
    }
}
else {
    //If False, then the printer is online.
}
```



Status List of the class structure **StarPrinterStatus**

Member name	Contents	Type	Detail
<b>blackMarkError</b>	Black Mark Error	SM_BOOLEAN	" SM_TRUE " : Black mark error occurs. " SM_FALSE " : Black mark error does not occur. When you set printer to Black mark, and print to not Black mark paper, this error occurs.
<b>compulsionSwitch</b>	Compulsion SW	SM_BOOLEAN	You can check status of CashDrawer (Open or Close) " SM_TRUE " : Compulsion SW is pressed. " SM_FALSE " : Compulsion SW is not pressed.
<b>coverOpen</b>	Cover Status	SM_BOOLEAN	You can check status of Cover " SM_TRUE " : Cover is opened. " SM_FALSE " : Cover is closed.
<b>cutterError</b>	Auto-cutter Error	SM_BOOLEAN	You can check status of Cutter " SM_TRUE " : Cutter error occurs. " SM_FALSE " : Cutter error does not occur.
<b>etbAvailable</b>	ETB available or not	SM_BOOLEAN	" SM_TRUE " : available to use " SM_FALSE " : not available to use
<b>etbCounter</b>	ETB Counter	UCHAR	You can get current value of ETB
<b>headThermistorError</b>	Head Thermistor Error	SM_BOOLEAN	You can check status of Head Thermistor. " SM_TRUE " : Head thermistor detects an abnormal value. " SM_FALSE " : Head thermistor does not detect an abnormal value.
<b>offline</b>	ONLINE/OFFLINE Status	SM_BOOLEAN	You can check status of Online or offline. " SM_TRUE " : Printer is Offline. " SM_FALSE " : Printer is Online
<b>overTemp</b>	Stopped by high head temperature	SM_BOOLEAN	" SM_TRUE " : Printer is stopped by head temperature. " SM_FALSE " : Printer is not stopped by head temperature.
<b>presenterPaperJamError</b>	Presenter Paper Jam Error	SM_BOOLEAN	You can check status of Paper Jam in Presenter. " SM_TRUE " : Paper jam occurs in presenter . " SM_FALSE " : Paper jam does not occur in presenter .
<b>presenterState</b>	Presenter Paper Position	UCHAR	You can check status of Presenter. 0 : State where there is no paper in presenter 1 : State where paper is supplied (loop state) 3 : State where paper is discharged (Can be pulled out) 6 : State where paper is recovered 7 : State where paper is pulled out.
<b>raw</b>	Byte column of status	UCHAR[63]	Byte column of status (example : HEX 23 86 00 00 00 00 00 00 00)
<b>rawLength</b>	raw length	CHAR	raw length
<b>receiptPaperEmpty</b>	Paper end	SM_BOOLEAN	" SM_TRUE " : Paper end. " SM_FALSE " : Paper exist.
<b>receiptPaperNearEmptyInner</b>	Paper Near-end (Inner Side)	SM_BOOLEAN	" SM_TRUE " : Paper near-end. " SM_FALSE " : Paper does not near-end.
<b>receiveBufferOverflow</b>	Receive Buffer Overflow	SM_BOOLEAN	You can check status of recieved Buffer. " SM_TRUE " : Received buffer is full. " SM_FALSE " : Received buffer is not full.
<b>unrecoverableError</b>	Non-recoverable Error	SM_BOOLEAN	" SM_TRUE " : Unrecoverable error occurs. " SM_FALSE " : Unrecoverable error does not occur.  Unrecoverable error : Head Thermistor Error, Auto-cutter Error, Electric Voltage Error and etc.)
<b>voltageError</b>	Electric Voltage Error	SM_BOOLEAN	" SM_TRUE " : Printers detects an abnormal power supply voltage. " SM_FALSE " : Printers does not detect an abnormal power supply voltage.

Class structure **StarPrinterStatus** Supported

Member name	TSP100 Series *	FVP10	TSP 650	TSP 650 II	TSP 700 II	TSP 800 II	TUP 500
<b>blackMarkError</b>		✓			✓	✓	✓
<b>compulsionSwitch</b>	✓	✓	✓	✓	✓	✓	
<b>coverOpen</b>	✓	✓	✓	✓	✓	✓	✓
<b>cutterError</b>	✓	✓	✓	✓	✓	✓	✓
<b>etbAvailable</b>	✓	✓	✓	✓	✓	✓	✓
<b>etbCounter</b>	✓	✓	✓	✓	✓	✓	✓
<b>headThermistorError</b>							✓
<b>offline</b>	✓	✓	✓	✓	✓	✓	✓
<b>overTemp</b>	✓	✓	✓	✓	✓	✓	✓
<b>presenterPaperJamError</b>							✓
<b>presenterState</b>							✓
<b>raw</b>	✓	✓	✓	✓	✓	✓	✓
<b>rawLength</b>	✓	✓	✓	✓	✓	✓	✓
<b>receiptPaperEmpty</b>	✓	✓	✓	✓	✓	✓	✓
<b>receiptPaperNearEmptyInner</b>		✓	✓	✓	✓	✓	✓
<b>receiveBufferOverflow</b>		✓	✓	✓	✓	✓	✓
<b>unrecoverableError</b>	✓	✓	✓	✓	✓	✓	✓
<b>voltageError</b>							✓

\*TSP100Series: TSP100U, TSP100GT, TSP100LAN, TSP100ECO, TSP100IIHW, TSP100IIILAN, TSP100IIIBI, TSP100IIIU

## beginCheckedBlock

```
-(void) beginCheckdBlock: (void *) starPrinterStatus: (u_int32_t) level;
```

This method is used in combination with endCheckedBlock and checks the completion of printing. beginCheckedBlock must be run just before sending print data.

### Parameters:

- starPrinterStatus - a pointer to StarPrinterStatus structure  
(Possible to specify StarPrinterStatus, StarPrinterStatus\_1 of StarPrinterStatus\_2. Normally StarPrinterStatus\_2 is specified.)
- level - the level of StarPrinterStatus structure  
(Possible to specify a value of 0,1 or 2. Normally 2 is specified.)

See the sample code [here](#).



F/W Version 3.0 or later is required for TSP650 and TUP500.

## endCheckedBlock

```
-(void) endCheckdBlock: (void *) starPrinterStatus: (u_int32_t) level;
```

This method is used together with the beginCheckedBlock method in a set.

It monitors printer status and when the transferred data is printed completely, returns control. In case of being transferred other kind of data than print data, when its command is processed in the printer, it returns the control.

In case that printing is not completed before the timeout (\*) or printer error occurs during printing, it returns PortException.

(\*) To timeout value, endCheckedBlockTimeoutMillis property is applied. Default value is the timeout value designated by getPort. Please adjust the endCheckedBlockTimeoutMillis value to be longer than printing time. Timeout length is specified by getPort, endCheckedBlockTimeoutMillis or is 10 seconds if specified less than 10 seconds.

### Parameters:

starPrinterStatus - a pointer to StarPrinterStatus structure  
(Possible to specify StarPrinterStatus, StarPrinterStatus\_1 of StarPrinterStatus\_2. Normally StarPrinterStatus\_2 is specified.)

level - the level of StarPrinterStatus structure  
(Possible to specify a value of 0,1 or 2. Normally 2 is specified.)

### Returns:

StarPrinterStatus structure giving the current device status

### Throws:

PortException - when a communication failure\* occurs

\*Examples) - An error sending the command (such as Off-Line)  
- No response for the completion of printing from a printer within the timeout



F/W Version 3.0 or later is required for TSP650 and TUP500.

```

unsigned char command[] = {0x41, 0x42, 0x43, 0x44, 0x1B, 0x7A, 0x00, 0x1B, 0x64, 0x02};
uint bytesWritten = 0;

StarPrinterStatus_2 starPrinterStatus;

SMPort *port = nil;

@try
{
    port = [SMPort getPort:@"BT:" :@"":10000];

    //Start checking the completion of printing
    [port beginCheckedBlock:&starPrinterStatus :2];

    if (starPrinterStatus.offline == SM_TRUE)
    {
        //There was an error writing to the port
    }

    while (bytesWritten < sizeof (command)) {
        bytesWritten += [port writePort: command : bytesWritten : sizeof (command) - bytesWritten];
    }

    //End checking the completion of printing
    [port endCheckedBlock:&starPrinterStatus :2];

    if (starPrinterStatus.offline == SM_TRUE)
    {
        //There was an error writing to the port
    }
}
@catch (PortException)
{
    //There was an error writing to the port
}
@finally
{
    [SMPort releasePort:port];
}

```

## disconnect

```
-(BOOL) disconnect
```

This method disconnects the specified Bluetooth device.

After the disconnection, the Bluetooth device can be connected by other iOS terminals.

This method fails in the following cases:

- when the disconnection has not been completed within the timeout specified by `getPort`
- when the disconnection function is not supported by a printer (such like portable printers).

This method has no effect on Ethernet devices.

### Returns:

It returns YES when succeeded and NO when failed.

It always returns YES when it was run with the Ethernet device.

## getFirmwareInformation

```
-(NSDictionary *) getFirmwareInformation:
```

This method gets a firmware Information of the printer.

### Returns:

It returns `NSDictionary` as an acquisition result.

Gets a model name from the return value by setting the Key to `@modelName`.

Gets a firmware version from the return value by setting the Key to `@firmwareVersion`.

### Throws:

`StarIOPortException` - when a communication failure occurs

### Note:

- If it failed to get information, it returns an empty string.
- It is impossible to get the firmware version of TSP100U, TSP100GT, TSP100LAN, TSP100ECO, TSP100IIW and TSP100IIILAN.
- When using Apple AirMac Express with a USB printer, it returns an empty string.

## StarIOVersion

```
+(NSString *) StarIOVersion
```

This method gets the StarIO version.

### Returns:

StarIO version

## SMBluetoothManager Class:

SMBluetoothManager Class specifies various settings of the Bluetooth interface.  
It can not be used with SMPort Class.

### ●Property

<b>portName</b>	Acquires the portName of the device to be connected.
<b>deviceType</b>	Acquires the type of the device to be connected.
<b>opened</b>	Shows whether the port is opened.
<b>deviceName</b>	Acquires and specifies the current Bluetooth device name.
<b>iOSPortName</b>	Acquires and specifies the port name to be used with the StarIO.
<b>autoConnect</b>	Acquires and specifies the setting (Valid or Invalid) of the autoconnection function.
<b>security</b>	Acquires the Bluetooth security setting (SPP or PIN Mode)
<b>pinCode</b>	Specifies the PIN Code to be used for pairing.

`@property(nonatomic, readonly) NSString *portName`

Creates an instance of SMBluetoothManager.

`@property(nonatomic, readonly) SMDeviceType deviceType`

Acquires the type of the device to be connected.

`@property(nonatomic, readonly) BOOL opened`

Shows whether the port is opened.

It returns YES if the open method was successful.

Then it will return NO when the close method is called.

`@property(nonatomic, retain) NSString *deviceName`

Acquires and specifies the current Bluetooth device name.

The current setting is read when the loadSetting method is called.

To set it, run the apply method after changing this property.

*Valid number of characters:* 1 to 16

*Valid characters:* 0-9, a-z, A-Z

; : ! ? # \$ % & , . @ \_ - = Space / \* + ~ ^ [ { ( ) } ] | \



#### `@property(nonatomic, retain) NSString *iOSSPortName`

Acquires and specifies the iOS port name to be used with the StarIO.  
The current setting is read when the loadSetting method is called.  
To set it, run the apply method after changing this property.

*Valid number of characters:* 1 to 16

*Valid characters:* 0-9, a-z, A-Z  
; : ! ? # \$ % & , . @ \_ - = Space / \* + ~ ^ [ { ( ) } ] | \

#### `@property(nonatomic, assign) BOOL autoConnect`

Acquires and specifies the setting of the auto connection function.  
The current setting is read when the loadSetting method is called.  
To set it, run the apply method after changing this property.



Set to NO when the security setting is set to PIN code mode.

#### `@property(nonatomic, assign) SMBluetoothSecurity security`

Acquires and specifies the Bluetooth security setting(SSP or PIN code mode).  
The current setting is read when the open method is called.  
To set it, run the apply method after changing this property.



Set the autoConnect property to NO when specifies the PIN code mode.

#### `@property(nonatomic, retain) NSString *pinCode`

Specifies the PIN code of the Bluetooth interface.  
It can not acquire the current setting.  
Set to nil when the PIN code is not changed.

*Valid number of characters:* 4 to 16

*Valid characters:* 0-9, a-z, A-Z

## ●Method

### initWithPortName : deviceType

```
-(id) initWithPortName: (NSString *) portName deviceType: (SMDeviceType) deviceType
```

This method is used to create an instance of SMBluetoothManager.

#### Parameters:

- |            |   |
|------------|---|
| portName   | - the port name of the device to be connected<br>Ex. "BT:Star Micronic" |
| deviceType | - the type of the device to be connected<br>SMDeviceTypeDesktopPrinter  |

#### Returns:

It returns Instance of SMBluetoothManager when succeeded.  
It returns nil when failed.

### open

```
-(BOOL) open
```

This method is used to open connection to the Bluetooth printer.  
Get the current settings by loadSetting method after conducting open method.

#### Returns:

It returns YES when succeeded and NO when failed.

### loadSetting

```
-(BOOL) loadSetting
```

This method is gets the value specified from the star Bluetooth device.

#### Returns:

It returns YES when succeeded and NO when failed.

## apply

-(BOOL) apply

This method is used to apply the property values of deviceName, iOSEPortName, autoConnect, security and pinCode.

### Returns:

It returns YES when succeeded and NO when failed.



The values applied with this method are effective after turning the device off and on and pairing again.

## close

-(void) close

This method is used to close communication with the printer.

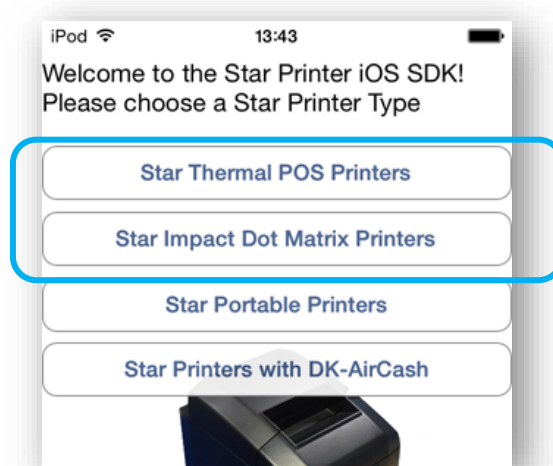
## StarIO iOS SDK Functionality

Overview of this SDK functionality and StarIO Printer Commands

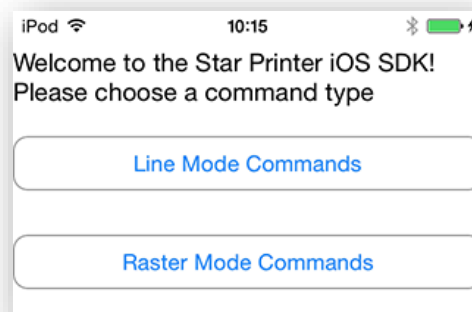
All of these commands can be found in the Star Line Mode Command Manual.

This SDK also has page and section references to the Line Mode Manual for more information so please download and study it if you need more detail on a specific command.

### Choosing a Printer and Communication Type



1. To use thermal printers, tap "Star Thermal POS Printers".  
And to use dot printers, tap "Star Impact Dot Matrix Printers".



2. Select "Line Mode Commands" or "Raster Mode Commands". [The difference is detailed here.](#)  
When "Star Impact Dot Matrix Printers" is selected, the screen will jump to the corresponding Line Mode Command page.

## Supported Samples by Command Type

Line Mode Command Samples Include:

[Port Discovery](#)

[Get Firmware Information](#)

[Get Status](#)

[Sample Receipt](#)

[Open Cash Drawer](#)

[1D Barcodes](#)

[2D Barcodes](#)

[Cut](#)

[Text Formatting](#)

[Bluetooth Pairing + Connect](#)

[Bluetooth Disconnect](#)

[Bluetooth Setting](#)

Raster Mode Command Samples Include:

[Port Discovery](#)

[Get Firmware Information](#)

[Get Status](#)

[Sample Receipt](#)

[Open Cash Drawer](#)

[Raster Graphics Text Printing](#)

[Image File Printing](#)

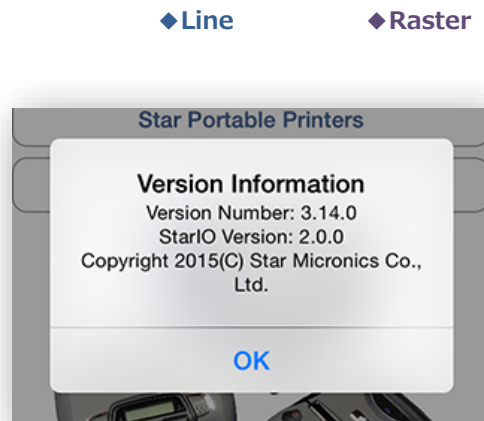
[AllReceipts](#)

[Bluetooth Pairing + Connect](#)

[Bluetooth Disconnect](#)

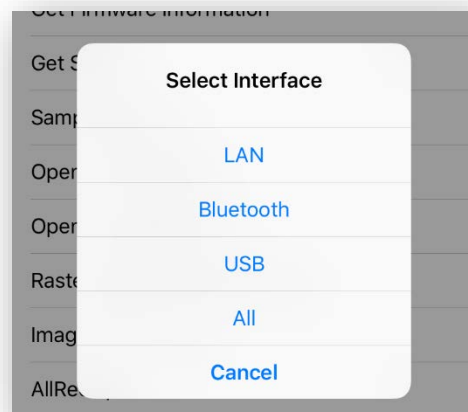
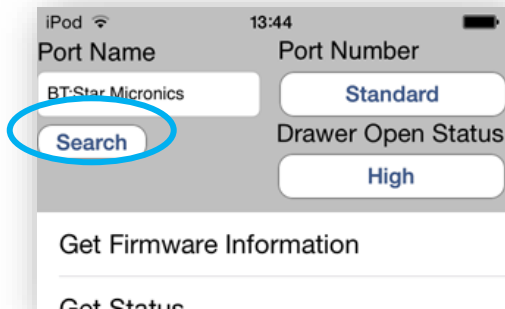
[Bluetooth Setting](#)

## Get StarIO Version



Tap “About”, displays StarIO version.

## Port Discovery



Automatically detects which Star Micronics Printers are connected to the network. Tap the printer to connect to it. [This feature is documented in greater detail here](#). USB printers do not support this feature.

\* USB printers connected via an AirPort are not supported.

## Help

◆Line

◆Raster

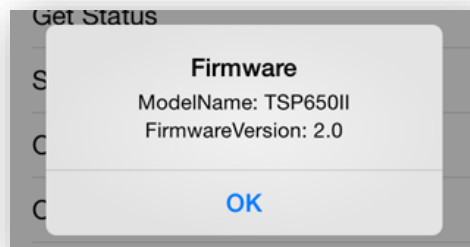


Help produces information on network port settings.

## Get Firmware Information

◆Line

◆Raster



Displays firmware information of the printer specified by Port Name.

When using TSP100LAN, TSP100IIIW and TSP100IIILAN it displays TSP100LAN for Model Name only.

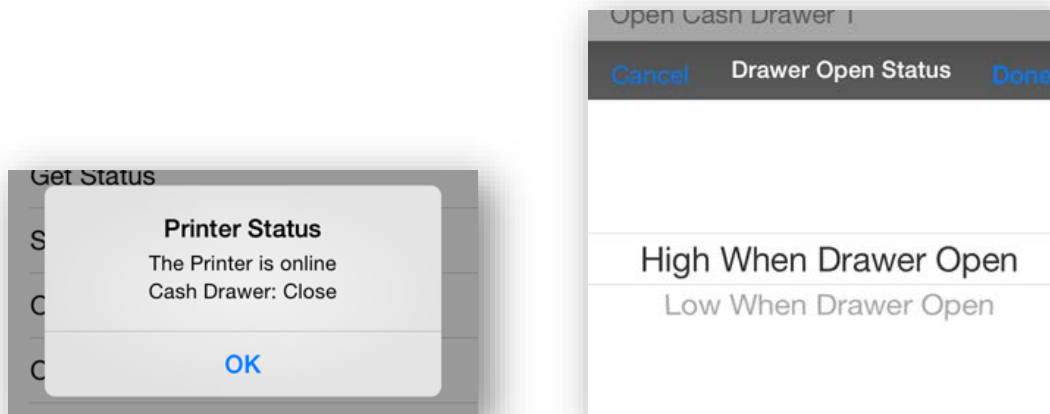
When using TSP100U, TSP100GT and TSP100ECO it displays TSP100 for Model Name only.

It is impossible to get the firmware version of TSP100U, TSP100GT, TSP100LAN, TSP100ECO, TSP100IIIW and TSP100IIILAN.

## Get Status

◆ Line

◆ Raster



### Drawer Open Status

Select the Sensor Active setting according to the specification of a peripheral device to be used.

### StarPrinterStatus

```
public boolean retrieveStatus()
```

offline

other

[See status return values here](#)

false = printer online; true = printer offline

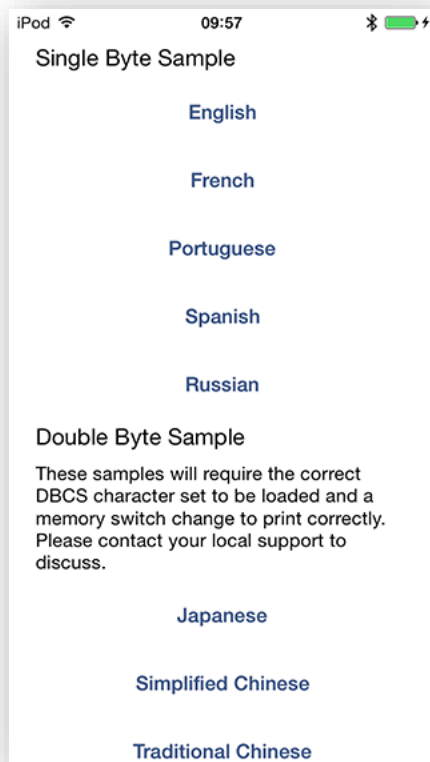
[See status return values here](#)



## Sample Receipt

◆ Line

◆ Raster

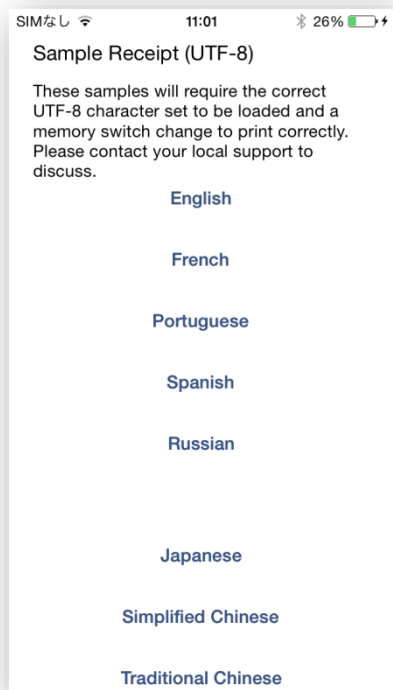


Prints a premade sample receipt in the chosen command type. "Sample Receipt" outputs a receipt in English, while "JP Sample Receipt" outputs one in Japanese.

Select the sample's width and tap "OK" to print it. This part of the source code is heavily commented to demonstrate how receipts can be fully customized.

Sample Receipt (UTF-8)

◆Line



Prints a premade sample receipt in the chosen command type. “Sample Receipt” outputs a receipt in English, while “JP Sample Receipt” outputs one in Japanese. Select the sample’s width and tap “OK” to print it. This part of the source code is heavily commented to demonstrate how receipts can be fully customized.



Sample Receipt (UTF-8) is supported by TSP650II with F/W version 3.0 or later. The table below shows information about languages which are supported by other models and F/W versions.

Japanese <sup>*1</sup> (JP models only)	TSP650II F/W Ver. 1.2 or later
Simplified Chinese <sup>*1</sup> (China/America & Europe models only)	TSP700II F/W Ver. 4.1 or later
Traditional Chinese <sup>*1</sup> (TW models only)	
Japanese <sup>*1</sup> (JP models only)	TSP800II F/W Ver. 1.6 or later
Simplified Chinese <sup>*1</sup> (China/America & Europe models only)	FVP10 F/W Ver. 1.5 or later

<sup>\*1</sup> To use UTF-8 with models other than TSP650II with F/W 3.0 or later, you have to change the memory switch setting of your printer. Set the bit4 of Bank0 to ON. You can change the memory switch setting by using StarPRNT Intelligence software. Please [here](#) to download.

## Open Cash Drawer1/2

◆ Line

◆ Raster

No additional screen. The printer will open the cash drawer if one is connected.

BEL

Opens the cash drawer1 (no value input needed)

SUB

Opens the cash drawer2 (no value input needed)



It is impossible to execute Open Cash Drawer1 and Open Cash Drawer2 at the same time.

## 1D Barcodes

◆Line

<<CODE39>>



### Configure 1D Barcode

ESC b n1 n2 n3 n4 d1 ... dk RS

n1 = Barcode Type

0 = UPC-E \*      1 = UPC-A \*      2 = JAN/EAN8 \*      3 = JAN/EAN13 \*  
4 = Code39      5 = ITF      6 = Code128      7 = Code93      8 = NW-7 \*

n2 = Under-bar character selection and added line feed selection

1 = No added under-bar characters & Executes line feed after printing barcode  
2 = Adds under-bar characters & Executes line feed after printing barcode  
3 = No added under-bar characters & doesn't line feed after printing barcode  
4 = Adds under-bar characters & doesn't line feed after printing barcode

n3 = Specifies the size of the narrow and wide barcode lines

n4 = Barcode height (dot count)

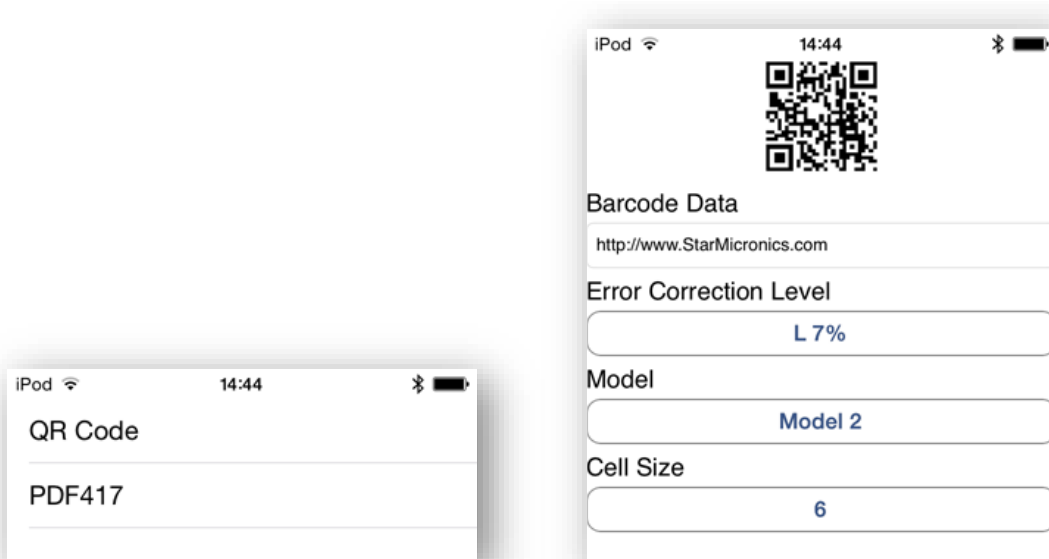
\* These barcodes are supported by Star POS Printers, but no example is in the sample application.

Note: 1D Barcode samples are not available for Raster Mode in this application. When using Raster Mode, barcodes must be sent as a graphic data.

## 2D Barcodes

◆Line

<<QR Code>>



### Select QR Code

There are 5 commands below that are very important to printing a good QR Code.

- |                                  |                                |
|----------------------------------|--------------------------------|
| (1) Set QR Code Model #          | ESC GS y S 0 n                 |
| (2) Set QR Code Correction Level | ESC GS y S 1 n                 |
| (3) Set QR Code Cell Size        | ESC GS y S 2 n                 |
| (4) Set QR Code Data             | ESC GS y D 1 NUL nL nH d1...dk |
| (5) Print the QR Code            | ESC GS y P                     |

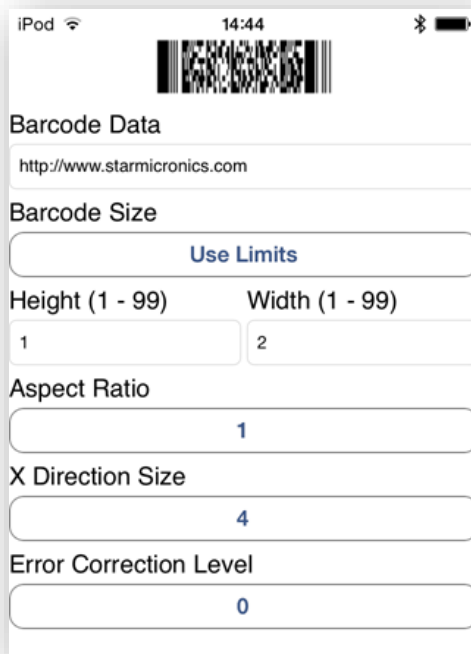
This is the order in which commands need to be sent to print the QR Code:

**QR Model + QR Correction Level + QR Cell Size + QR Data + Print QR Code**

Refer to the Line Mode Programming Manual for a listing of all QR Code commands.

Note: 1D Barcode samples are not available for Raster Mode in this application. When using Raster Mode, barcodes must be sent as a graphic data.

<<PDF417>>



iPod 14:44

Barcode Data  
http://www.starmicronics.com

Barcode Size  
Use Limits

Height (1 - 99) Width (1 - 99)  
1 2

Aspect Ratio  
1

X Direction Size  
4

Error Correction Level  
0

### Select PDF417

Please visit page 3-120 in the Line Mode Spec Manual for more details on PDF417

- |  |                               |
|--|-------------------------------|
| (1) Set PDF417 barcode size            | ESC GS x S 0 n p1 p2          |
| (2) Set PDF417 ECC (Security Level)    | ESC GS x S 1 n                |
| (3) Set PDF417 module X direction size | ESC GS x S 2 n                |
| (4) Set PDF417 module aspect ratio     | ESC GS x S 3 n                |
| (5) Set PDF417 barcode data            | ESC GS x D nL nH d1 d2 ... dk |
| (6) Print PDF417 barcode               | ESC GS x P                    |

This is the order in which commands need to be sent to print the PDF417 barcode:

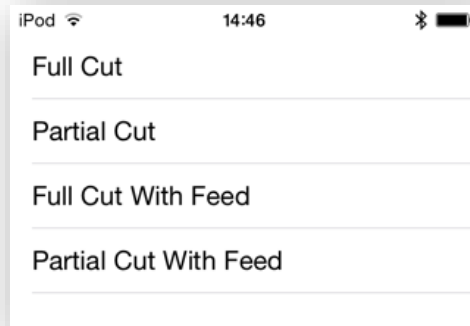
PDF417 Size + PDF417 ECC + PDF417 X-dim + PDF417 Ratio + PDF417 Data + Print PDF417

Refer to the Line Mode Programming Manual for a listing of all PDF417 commands.

Note: 1D Barcode samples are not available for Raster Mode in this application. When using Raster Mode, barcodes must be sent as a graphic data.

## Cut

◆Line



### Full Cut

ESC d 0

Full Cut at Current Position

### Partial Cut

ESC d 1

Partial Cut at Current Position

### Full Cut with Feed

ESC d 2

Feed and then Full Cut Position

### Partial Cut with Feed

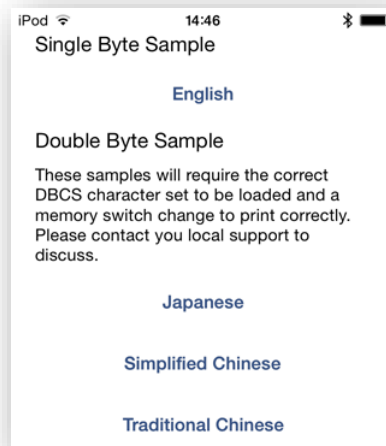
ESC d 3

Feed and then Partial Cut

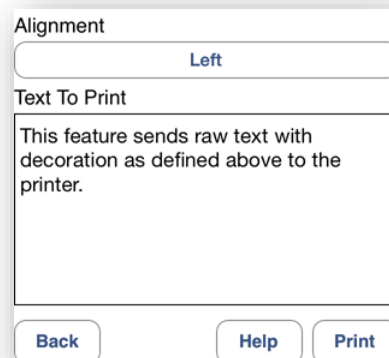
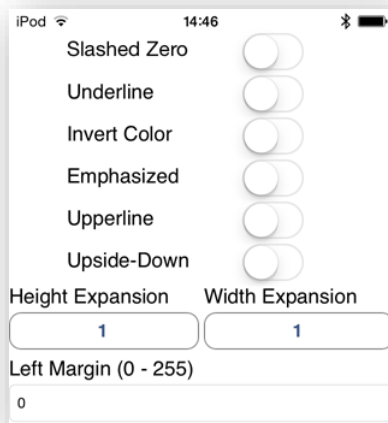
Note: Cut samples are not available for Raster Mode in this application. Please refer to the Star Micronics Line Mode Programming Manual for Raster Mode cut commands.

## Text Formatting

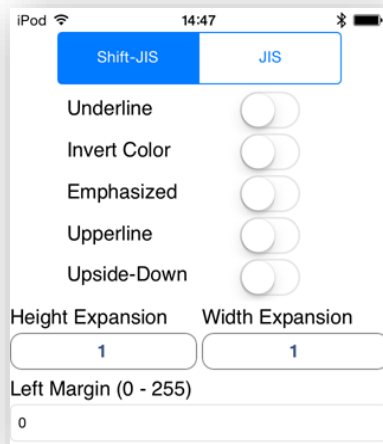
◆Line



### Ex.) Thermal Printer / English



### Ex.) Thermal Printer / Japanese





## Text Formatting (continued from above)

### Slashed Zero (only English)

ESC / 1 = on                      ESC / 0 = off

### JIS Mode (only Japanese)

ESC p = on                      ESC q = off [Default]

### Shift JIS Mode (only Japanese)

ESC \$ 1 = on                      ESC \$ 0 = off [Default]

### Underline

ESC - 1 = on                      ESC - 0 = off [Default]

### Invert Color (B/W) (Thermal printer)

ESC 4 = on                      ESC 5 = off [Default]

### Two Color (Dot printer)

ESC 4 = Red                      ESC 5 = Black [Default]

### Emphasized (Bold)

ESC E = on                      ESC F = off [Default]

### Upperline

ESC \_ 1 = on                      ESC \_ 0 = off [Default]

### Upside-Down

SI = on                      DC2 = off [Default]

### Character Expansion (Thermal printer)

Height Expansion              ESC h n               $0 \leq n \leq 5$

Width Expansion              ESC W n               $0 \leq n \leq 5$

### Character Expansion (Dot printer)

Height Expansion(only English)    ESC h 1 = on    ESC h 0 = off [Default]

Width Expansion                      ESC W 1 = on    ESC W 0 = off [Default]

### Left Margin

ESC l n                       $0 \leq n \leq 255$

### Alignment

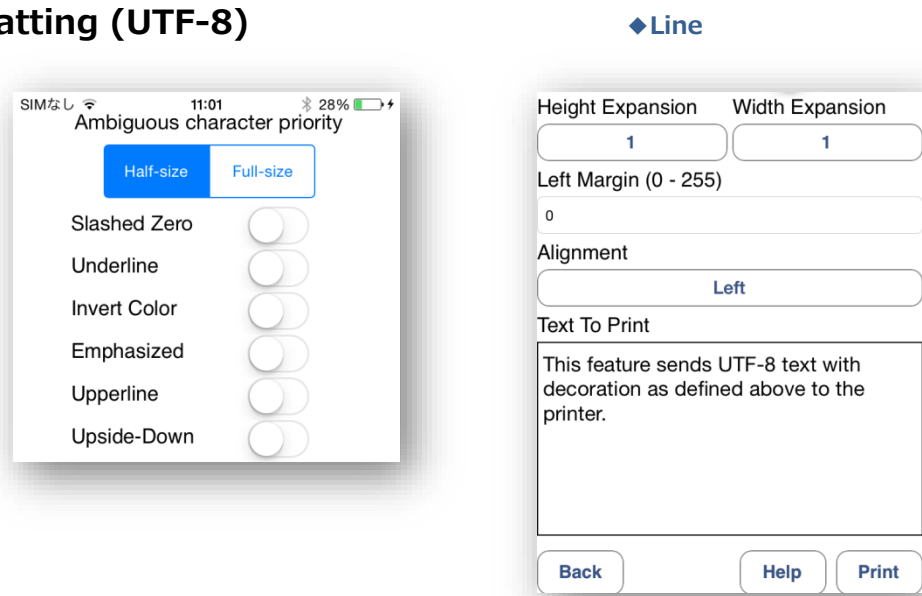
Left [Default]                      ESC GS a 0

Center                      ESC GS a 1

Right                      ESC GS a 2

Note: Raster Mode receives graphical data only so this sample is not compatible. For a data formatting example in this mode, refer to [Raster Graphical Text Printing](#).

Text Formatting (UTF-8)



Text Formatting (continued from above)

**Ambiguous character priority**  
ESC GS ) U pL pH fn 0 = Half-size [Default]  
ESC GS ) U pL pH fn 1 = Full-size

**Slashed Zero**  
ESC / 1 = on                      ESC / 0 = off [Default]

**Underline**  
ESC - 1 = on                      ESC - 0 = off [Default]

**Invert Color**  
ESC 4 = on                      ESC 5 = off [Default]

**Emphasized (Bold)**  
ESC E = on                      ESC F = off [Default]

**Upperline**  
ESC \_ 1 = on                      ESC \_ 0 = off [Default]

**Upside-Down**  
SI = on                      DC2 = off [Default]

**Character Expansion**  
Height Expansion              ESC h n              0 ≤ n ≤ 5  
Width Expansion              ESC W n              0 ≤ n ≤ 5

**Left Margin**  
ESC I n                      0 ≤ n ≤ 255

**Alignment**

Left [Default]	ESC GS a 0
Center	ESC GS a 1
Right	ESC GS a 2



Text Formatting (UTF-8) is supported by TSP650II with F/W version 3.0 or later.

## Raster Graphical Text Printing

◆ Raster

iPod 14:47

Text To Print

This is the text that will be printed on any desktop thermal printer. That includes the tsp100. This will not print on the dot matrix printers.

Font

Thonburi

Font Style

Thonburi-Bold

Size 17.00

Printer Width 3 inch

Compression ☒

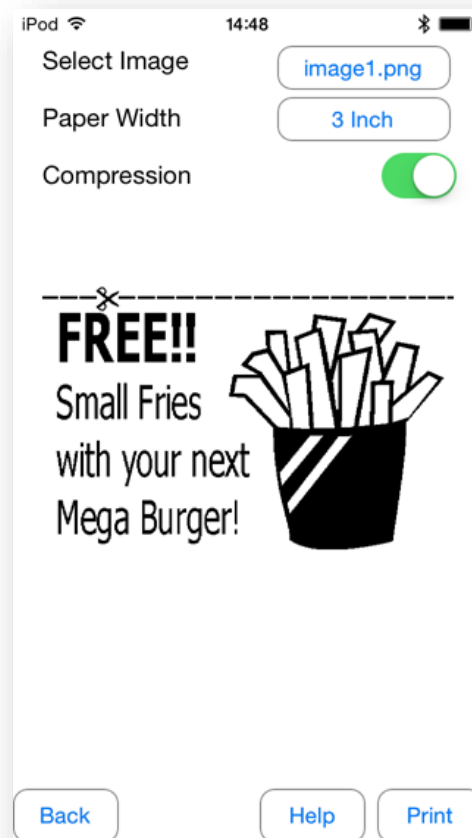
Refer to the Line Mode Programming Manual for a listing of all Raster commands.

**Note:**

- Line Mode cannot process graphical data so this sample is not compatible. For a data formatting example in this mode, refer to [Text Formatting](#).

## Image File Printing

◆ Raster



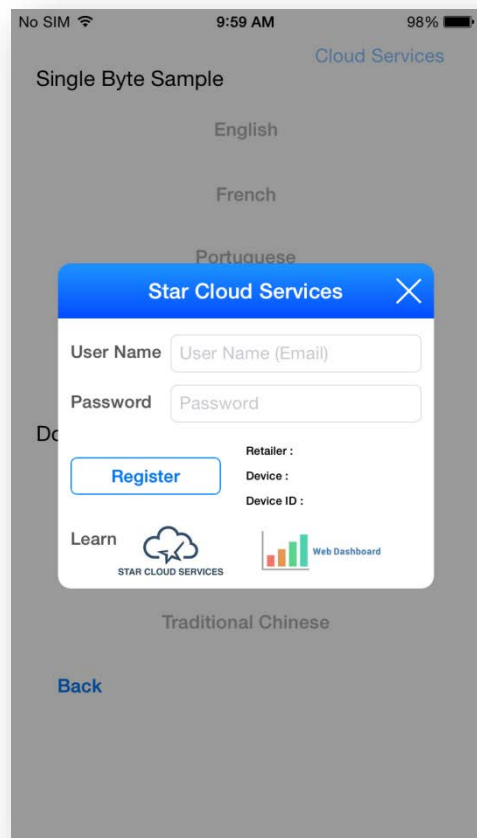
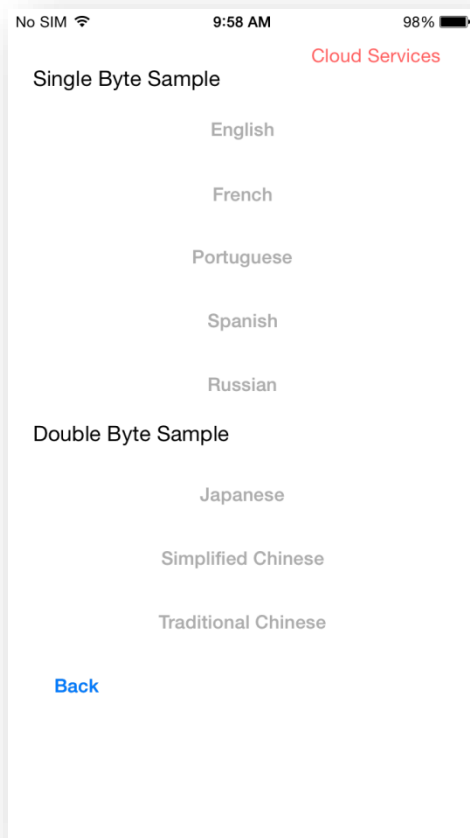
Use the dropdown box to select one of four different sample images to print via Raster Graphics. Note: The images in this sample are pre-formatted for 80mm wide receipts. If the printer in use is smaller or wider than 80mm, the image will not be automatically scaled.

Raster Mode converts all print data into image data and then outputs it to the printer. This enables Star Printers to print at high speeds, regardless of outputting receipts with only text or text and logos/coupons. As there are many options on how to customize output in Raster Mode, refer to the Line Mode Programming Manual for a listing of all Raster commands. These commands are also conveniently listed right on the Android device by tapping the Help button on the screen.

Using "Compression API" method may improve through put.

Note: This sample is not available in Line Mode.

## AllReceipts



When you tap "Cloud Services", you can get "Registration View" for the device registration to Star Cloud Services.

For details of AllReceipts and Star Cloud Services, refer to the StarPRNT iOS SDK Manual.

## Bluetooth Pairing + Connect

◆ Line

◆ Raster

All Bluetooth devices you can connect to are displayed.

You can pair with and connect to a device by tapping it, when the security setting is set to SSP (Default). When a PIN code is used for a secured setting, only the connection function is available.

You must pair with a device in advance before using this function.

No StarIO functionality is used.

## Bluetooth Disconnect

◆ Line

◆ Raster

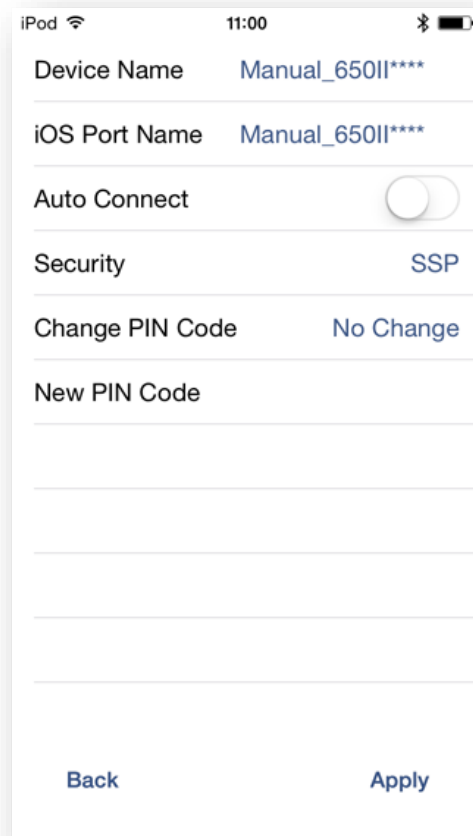
All devices you have paired with and connected to are displayed. You can disconnect a device to allow it to be connected to other host devices.

This function uses the disconnect method.

## Bluetooth Setting

◆ Line

◆ Raster



iPod 11:00

Device Name Manual\_650II\*\*\*\*

iOS Port Name Manual\_650II\*\*\*\*

Auto Connect ☐

Security SSP

Change PIN Code No Change

New PIN Code

Back Apply

Connects to the Bluetooth device which is specified for PortName and changes various settings of the Bluetooth interface.



The values applied with this method are effective after turning the device off and on and pairing again.



## Tips for App Development when using StarIO

Star Micronics prides itself as the industry leader in great POS products and with great power comes great responsibility. Below is a tips section just to help you get on the fast track to software development with StarIO.

**TIP #1:** If you are going to be coding a large project, create a class to abstract all the printing methods into class(s) instead of having the code reside in the main code block. This will help with code reusability and will also save you time in the long run from having to find one line of code in the main code. By having StarIO only reside in the class(s), you will be fully taking advantage of object oriented programming.

**TIP #2:** Know what the differences and definitions of (ASCII & Unicode), (Hex & Decimal), and (Byte & Char) are. A byte is normally 8-bits long which would be 8 digits of binary (1s and 0s). These bytes are just 8 bits of binary data but bytes can also be int or char. The three different variable types basically hold the data in the same way but there are slight differences. Try to code with Bytes instead of Chars, ints, or strings when choosing a variable to contain your print job data. ASCII to Unicode and vice versa conversions are sometimes unsecure so make sure you know what and how the encoding class works with these. Big mistakes made in Unicode are culture-sensitive search and casing, surrogate pairs, combining characters, and normalization.

**TIP #3:** HEX DUMP MODE! If you are debugging and your application seems to have a bug in it use hex dump mode on the printer. This is the best way to verify what is being sent out of the computer is being received correctly. To put the printer in hex dump mode, turn the printer off, open the cover to the paper, hold the feed button down, turn the printer back on, close the cover, let go of the feed button. Hex dump mode is a sure fire way to verify hex data is sent correctly. When in hex dump mode, printer functions will not work.

**TIP #4:** Do not waste time trying to reverse engineer StarIO command codes. All the available StarIO commands are available in the Thermal Line Mode Spec Manual and that is the best resource to use when researching a specific StarIO command. This SDK & Manual was built to help you (The Developer) have a very easy job ahead of you to program for Star Printers.

**TIP #5:** If there is a command that is not covered in this SDK but you wish to see a code snippet of that command in use then visit our Developers' section for a possible code block that matches your needs.

**TIP #6:** Looking for an Android printing SDK? Visit our [Developers section](#) to get access to Star developer tools for these environments.

## Additional Resources

This section will share resources that will help you develop good software with StarIO.

Please get the programmers manual for Star Portable Printers from the link below.

### [Star Micronics Developers Network](#)

Browse Star Micronics' FAQs, look up information, etc.

The Developers Network gets you access to:

- Updated Versions of this Manual and Source Code
- Getting Started Advice and Industry Information
- Star Micronics Printer Drivers
- Technical Questions/Support

### [Apple Developer Site](#)

The official Apple development resource.

### [Apple Developer Site Resources](#)

Peruse Apple's library of documentation for developers.

### [Unicode.org](#)

The Unicode Consortium - Good place to learn more about Unicode.

### [1D Barcodes](#)

Barcode Island is a great resource for specs on 1D barcodes.

### [2D Barcodes](#)

Great place for information on 2D Barcodes, [QR Codes](#), and [PDF417](#)

### [Code Pages](#)

Learn about Code Pages here.

## ASCII Table Resource

ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol
0	0	NUL	16	10	DLE	32	20	(space)	48	30	0
1	1	SOH	17	11	DC1	33	21	!	49	31	1
2	2	STX	18	12	DC2	34	22	"	50	32	2
3	3	ETX	19	13	DC3	35	23	#	51	33	3
4	4	EOT	20	14	DC4	36	24	\$	52	34	4
5	5	ENQ	21	15	NAK	37	25	%	53	35	5
6	6	ACK	22	16	SYN	38	26	&	54	36	6
7	7	BEL	23	17	ETB	39	27	'	55	37	7
8	8	BS	24	18	CAN	40	28	(	56	38	8
9	9	TAB	25	19	EM	41	29	)	57	39	9
10	A	LF	26	1A	SUB	42	2A	*	58	3A	:
11	B	VT	27	1B	ESC	43	2B	+	59	3B	;
12	C	FF	28	1C	FS	44	2C	,	60	3C	<
13	D	CR	29	1D	GS	45	2D	-	61	3D	=
14	E	SO	30	1E	RS	46	2E	.	62	3E	>
15	F	SI	31	1F	US	47	2F	/	63	3F	?
ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol
64	40	@	80	50	P	96	60	`	112	70	p
65	41	A	81	51	Q	97	61	a	113	71	q
66	42	B	82	52	R	98	62	b	114	72	r
67	43	C	83	53	S	99	63	c	115	73	s
68	44	D	84	54	T	100	64	d	116	74	t
69	45	E	85	55	U	101	65	e	117	75	u
70	46	F	86	56	V	102	66	f	118	76	v
71	47	G	87	57	W	103	67	g	119	77	w
72	48	H	88	58	X	104	68	h	120	78	x
73	49	I	89	59	Y	105	69	i	121	79	y
74	4A	J	90	5A	Z	106	6A	j	122	7A	z
75	4B	K	91	5B	[	107	6B	k	123	7B	{
76	4C	L	92	5C	\	108	6C	l	124	7C	
77	4D	M	93	5D	]	109	6D	m	125	7D	}
78	4E	N	94	5E	^	110	6E	n	126	7E	~
79	4F	O	95	5F	_	111	6F	o	127	7F	

*Use this to compare hex values to symbol (ASCII) values.*

## SDK Package Version History

Release Date	SDK Package Version	Update
Mar. 2017	3.17.0	<ul style="list-style-type: none"> <li>- Added support of TSP100IIIU</li> <li>- The performance when the getPort method failed on Ethernet was improved.</li> </ul>
Jan. 2017	3.16.2	<ul style="list-style-type: none"> <li>- Added iOS 10 support</li> <li>- Added support of TSP100IIIBI</li> </ul>
Mar. 2016	3.16.0	<ul style="list-style-type: none"> <li>- Added AllReceipts support</li> </ul>
Jan. 2016	3.15.a	<ul style="list-style-type: none"> <li>- Added support of TSP100IIIW and TSP100IIILAN</li> </ul>
Jul. 2015	3.14.1	<ul style="list-style-type: none"> <li>- Add support of UTF-8 on Sample Receipt and Text Formatting.</li> </ul>
Apr. 2015	3.14.0	<ul style="list-style-type: none"> <li>-iOS 5.x End of support</li> </ul>
Mar. 18 2015	3.13.1	<ul style="list-style-type: none"> <li>- Deleted a 32-bit separate build settings.</li> </ul>
Sep. 8 2014	3.12.0	<ul style="list-style-type: none"> <li>- Add support of Dot Matrix Printers(SP700)</li> <li>- Added StarIO Version method</li> <li>- Added Image File Printing function</li> </ul>
Jul. 24 2014	3.10.3	<ul style="list-style-type: none"> <li>- Added Data timeout function</li> <li>- Fixed a problem with multiple execution of getPort method in Bluetooth connection.</li> </ul>
Mar. 10 2014	3.10.0	<ul style="list-style-type: none"> <li>-Added getFirmwareInformation method</li> <li>-Added Open Cash Drawer2 function</li> </ul>
Nov. 19 2013	3.9.0	<ul style="list-style-type: none"> <li>-Added SMBluetoothManager Class</li> <li>-Added endCheckedBlockTimeoutMillis Property</li> <li>-iPad Air, iPad mini(2nd Generation), iPhone5S, iPhone 5C support</li> <li>-iOS 4.3 End of support</li> <li>-Added 64-bit build</li> </ul>
Sep. 19 2013	3.8.0	<ul style="list-style-type: none"> <li>- Added iOS 7 support</li> </ul>
Jul. 3 2013	3.7.1	<ul style="list-style-type: none"> <li>- Added Star Printer Status List</li> </ul>
Jun. 16 2013	3.6.0	<ul style="list-style-type: none"> <li>- Added support for SearchPrinter</li> </ul>
Mar. 14 2013	3.4.0a	<ul style="list-style-type: none"> <li>- Fixed a bug in the sample code</li> </ul>

Jan. 21 2013	3.3.0	- Added support for Bluetooth Interface
Nov. 2 2012	3.2.0	- Updated StarIO SDK Compatibility Chart
Oct. 2 2012	3.1.0	- Added iOS 6 support - Added iPhone 5 support
Aug. 31 2012	3.0.0	- Changed SDK UI to tree structure - Added Port Discovery support (POS Printers) - Added Apple AirPort Express support - Added Sample Receipt Printing support
May. 30 2012	2.3.0	- Fixed hang up in application when Mobile Printer Sleep Mode is enabled (Mobile Printers Only)
Apr. 27 2012	2.2.0	- Updated default StarIO Port Class name - Added support for Retina Display - Added ARC support
Jan. 12 2012	2.1.0	- Added Japanese Text Formatting Sample - Added Japanese Manuals
Dec. 9 2011	2.0.0	- Updated Sample Printing
Jun. 3 2011	1.2.1	- Initial Release



<http://www.star-micronics.com/support/>