star SDK

iOS Software Development Kit

How to Use StarIO for Printers(StarPRNT)

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.

❖ Portable Printer

When using StarPRNT emulation:

To use the StarPRNT emulation, set the emulation setting of the printer to “Star Line Mode”. To change the emulation, proceed as follows. (For SM-L200, it does not need to switch the emulation.)

◆ Switching over between StarPRNT and ESC/POS emulation

1. Turn the printer power and open the printer cover.
2. Press and hold the POWER button and the FEED button simultaneously. As soon as the ERROR lamp flashes five times, release the buttons. The emulation switchover takes place automatically.
3. After setting a paper, close the printer cover. The set emulation is printed out.
   ESC/POS : EMU = ESC/POS Mode
   StarPRNT : EMU = Star Line Mode

   If the emulation is not switched correctly, repeat the above steps 1 to 3.
   At that time, in step 2, make sure not to release the buttons until the lamp completes the 5th flash.
4. Please reboot the printer after switching the emulation.
   *It will be valid after rebooting the printer.

StarIO SDK Compatibility OS : iOS 7.0 or later

StarIO SDK Compatibility Chart

Device	CPU
iPad 2	Armv7
iPad (3rd Generation)	Armv7
iPad (4th Generation)	Armv7s
iPad Air	Arm64
iPad Air 2	Arm64
iPad mini	Armv7
iPad mini 2	Arm64
iPad mini 3	Arm64
iPad mini 4	Arm64
iPad Pro	Arm64
iPhone 4s	Armv7
iPhone 5	Armv7s
iPhone 5s	Arm64
iPhone 5c	Armv7s
iPhone 6	Arm64
iPhone 6 Plus	Arm64
iPhone 6s	Arm64
iPhone 6s Plus	Arm64
iPod touch (5th Generation)	Armv7
iPod touch (6th Generation)	Arm64

* As BLE is not supported, SM-L200 cannot use.

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

This manual is designed to help you understand StarIO and how to build an iOS application to interact with Star Micronics 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.

Star Printer Compatibility Chart

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

✓ : Line Mode, ESC/POS   ✓ : Raster Mode, ESC/POS   ✓ : Raster Mode only

Star Printer			Port Discovery	Get Firmware Information	Get Status	Sample Receipts	1D Barcodes	2D Barcodes	Text Formatting	Raster Graphics Text Printing	Image File Printing	MSR	AllReceipts	Bluetooth Setting
Model	**Interface**	**F/W Ver.**												
SM-S210i (JP model only)	Bluetooth	3.0 or later	✓	✓	✓	✓*2	✓	✓	✓	✓	✓	✓	✓	✓
SM-S220i (Excluding Jp model)	Bluetooth	3.0 or later	✓	✓	✓	✓*2	✓	✓	✓	✓	✓	✓	✓	✓
SM-S230i (Excluding JP model)	Bluetooth	1.0 or later	✓	✓	✓	✓*2	✓	✓	✓	✓	✓	✓	✓	✓
SM-T300i	Bluetooth	3.0 or later	✓	✓	✓	✓*2	✓	✓	✓	✓	✓	✓	✓	✓
SM-T400i	Bluetooth	3.0 or later	✓	✓	✓	✓*2	✓	✓	✓	✓	✓	✓	✓	✓

SM-L200	Bluetooth Low Energy [1]	1.0 or later	✓	✓	✓	✓[3]	✓	✓[4]	✓	✓	✓	✓[5]	✓	✓
mPOP	Bluetooth	1.0 or later	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	

*1 Bluetooth Low Energy is the communication method for connecting the SM-L series printer to an iOS device.

*2 When using Line Mode commands, printing in Russian and Simplified Chinese is not supported.

*3 When using Line Mode commands, Simplified Chinese will be supported from firmware version 1.1.

*4 PDF417 will be supported from firmware version 1.1.

*5 MSR capability will be supported from firmware version 2.0.

Connecting a Star POS Printer to an iOS Device

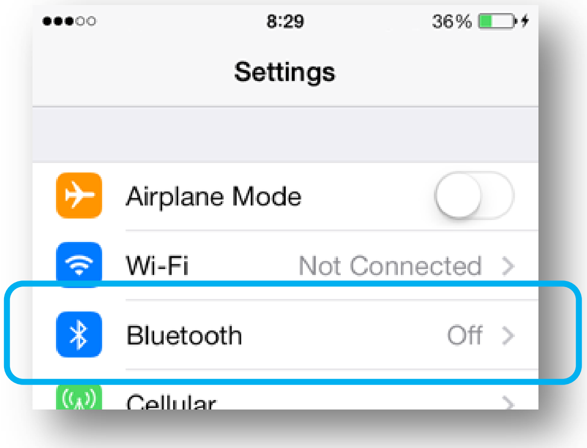
Bluetooth Interface

◆Pairing

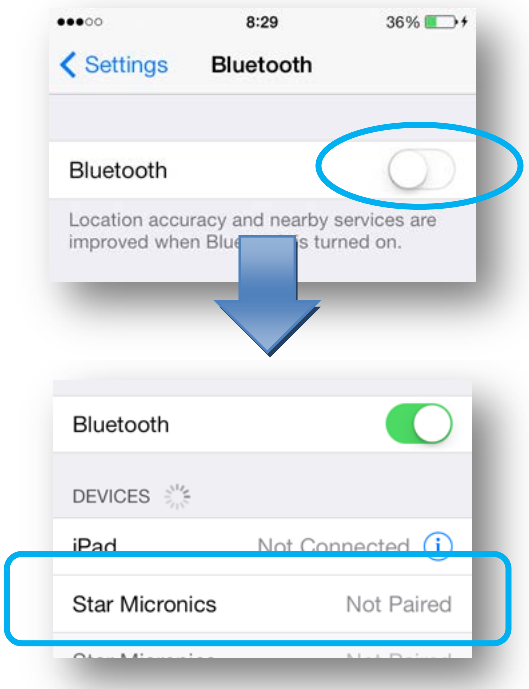
We recommend that you do not pair an iOS device with multiple Star printers at the same time.

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 paring.

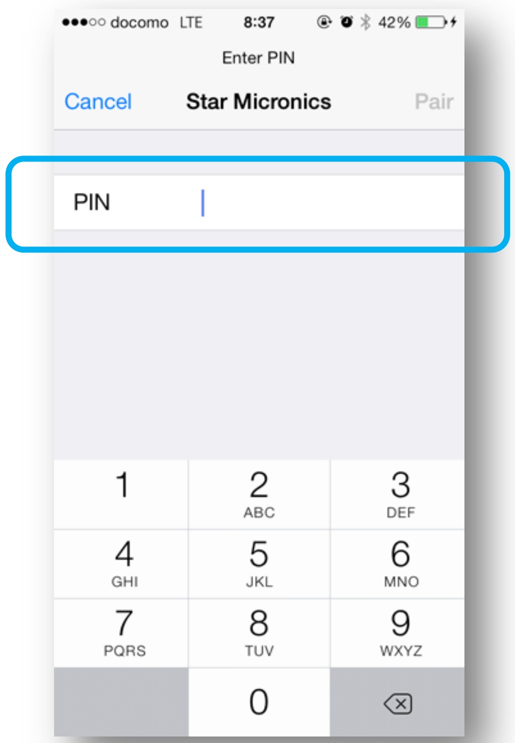
2. Tap Settings > Bluetooth.



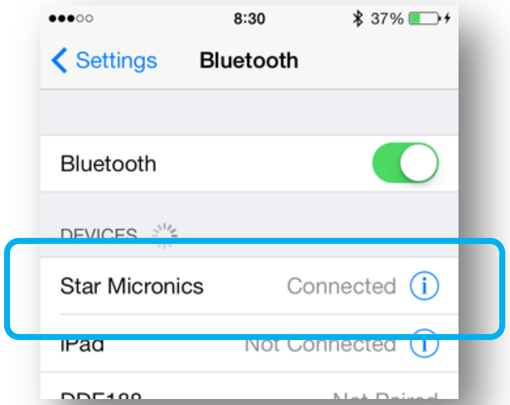
3. Tap Bluetooth to turn it on. Your iOS device searches and displays the Bluetooth devices in range. Tap the Star printer you want to pair with.


●●●○○ 8:29 36%
Settings
Bluetooth
Bluetooth
Location accuracy and nearby services are improved when Bluetooth is turned on.
Bluetooth
DEVICES
iPad Not Connected
Star Micronics Not Paired

4. Enter the PIN and tap Pair. (Portable Printer only)



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



◆**How to change the Bluetooth Device Name**

The Star Bluetooth 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.

Getting Started

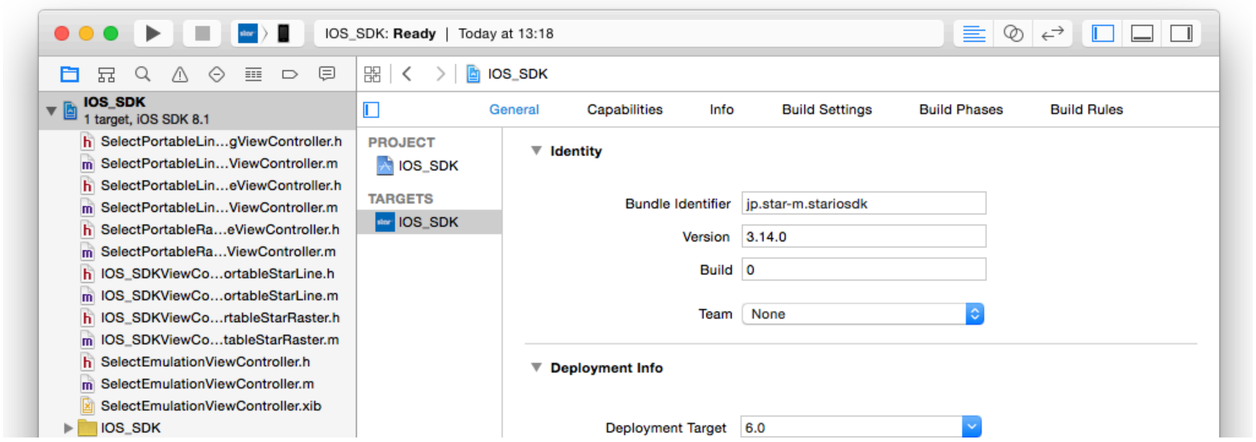
To build an iOS project, Xcode are needed. These tools are available in one package from the Apple Developer Site or Mac App Store. It is important to note that in order to produce applications that will actually run on an iOS device, you must be part of the Apple Developer Program, which requires a 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.


IOS_SDK: Ready | Today at 13:18

IOS_SDK
1 target, iOS SDK 8.1
SelectPortableLin...gViewController.h
SelectPortableLin...ViewController.m
SelectPortableLin...eViewController.h
SelectPortableLin...ViewController.m
SelectPortableRa...eViewController.h
SelectPortableRa...ViewController.m
IOS_SDKViewCo...ortableStarLine.h
IOS_SDKViewCo...ortableStarLine.m
IOS_SDKViewCo...rtableStarRaster.h
IOS_SDKViewCo...tableStarRaster.m
SelectEmulationViewController.h
SelectEmulationViewController.m
SelectEmulationViewController.xib
IOS_SDK

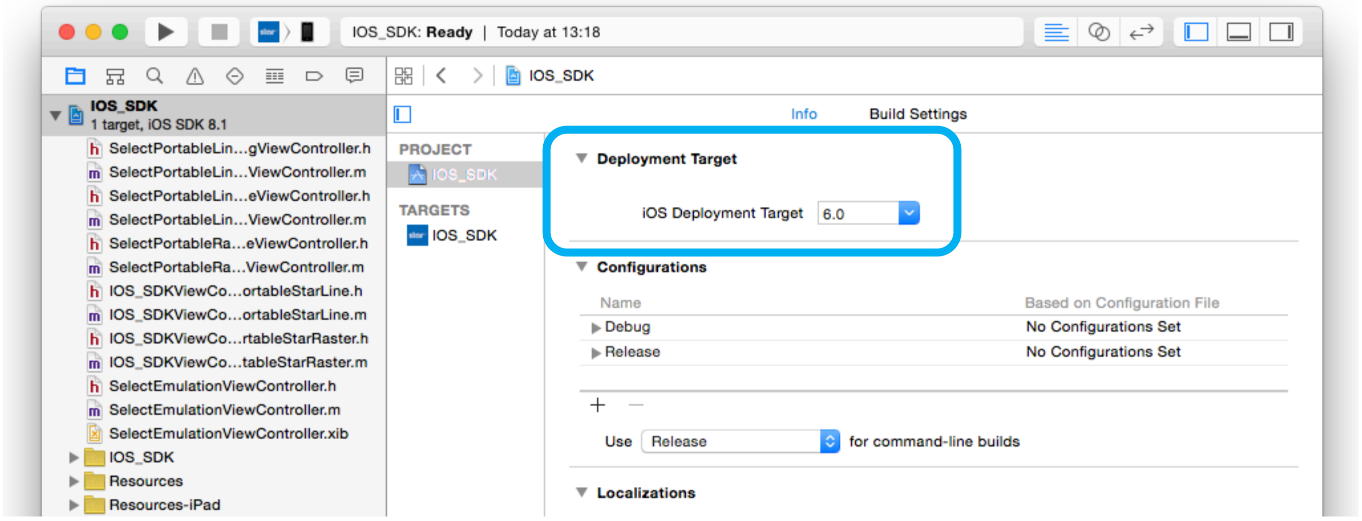
General | Capabilities | Info | Build Settings | Build Phases | Build Rules

PROJECT
IOS_SDK
TARGETS
IOS_SDK

Identity
Bundle Identifier: jp.star-m.stariosdk
Version: 3.14.0
Build: 0
Team: None

Deployment Info
Deployment Target: 6.0

3. Set the iOS Deployment Target to 6.0 or later.



IOS_SDK: **Ready** | Today at 13:18

IOS_SDK
1 target, iOS SDK 8.1

- SelectPortableLin...gViewController.h
- SelectPortableLin...ViewController.m
- SelectPortableLin...eViewController.h
- SelectPortableLin...ViewController.m
- SelectPortableRa...eViewController.h
- SelectPortableRa...ViewController.m
- IOS_SDKViewCo...ortableStarLine.h
- IOS_SDKViewCo...ortableStarLine.m
- IOS_SDKViewCo...rtableStarRaster.h
- IOS_SDKViewCo...tableStarRaster.m
- SelectEmulationViewController.h
- SelectEmulationViewController.m
- SelectEmulationViewController.xib
- IOS_SDK
- Resources
- Resources-iPad

PROJECT: IOS_SDK
TARGETS: IOS_SDK

Info | Build Settings

Deployment Target

iOS Deployment Target: 6.0

Configurations

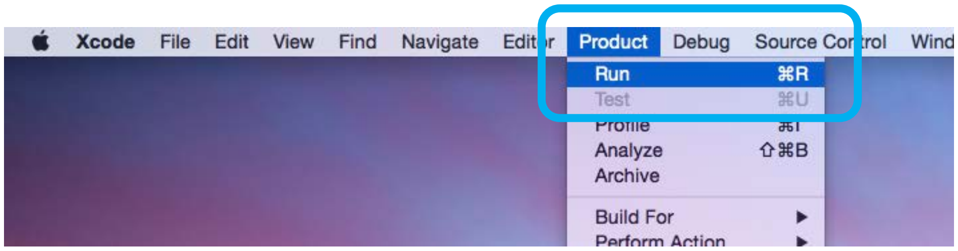
Name	Based on Configuration File
Debug	No Configurations Set
Release	No Configurations Set

Use Release for command-line builds

Localizations

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 Printer Model.

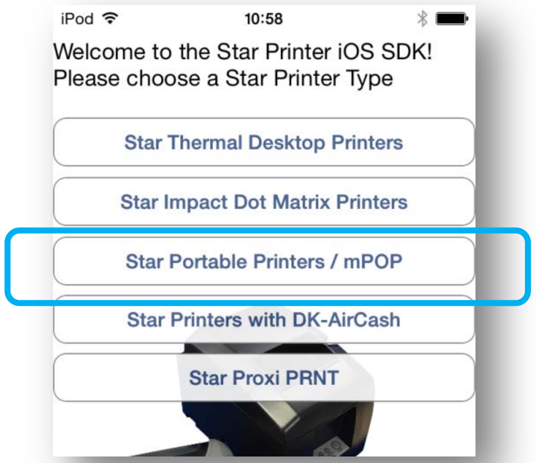
Port Name and Interface Relation:
StarIO uses specific port names to identify what port will be used. These are very important to understand because not following the naming convention correctly will fail to communicate with the printer.

Interface	Port Name
Bluetooth	BT:"iOS Port Name"
Bluetooth Low Energy	BLE:"Device Name" ***** BLE:"MAC Address"

*****When getting the printer device name using searchPriner method for the first time, sometimes portName will be @"BLE:".
In those cases, please connect the printer using getPort method.
Once you have got the Device name, searchPrinter method works correctly.

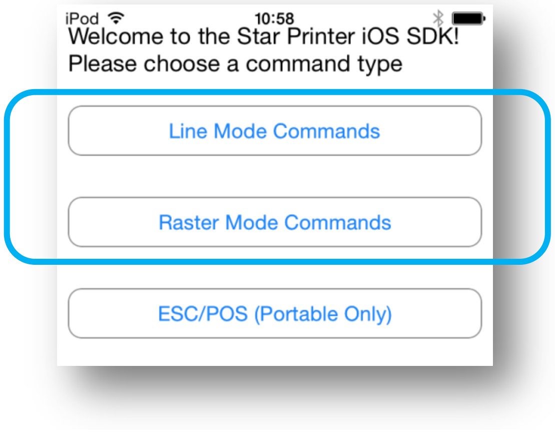
Using a Printer

1. Tap "Star Portable Printers / mPOP".


iPod
10:58
Star Thermal Desktop Printers
Star Impact Dot Matrix Printers
Star Portable Printers / mPOP
Star Printers with DK-AirCash
Star Proxi PRNT

Welcome to the Star Printer iOS SDK!
Please choose a Star Printer Type

2. Tap “Line Mode Commands” or “Raster Mode Commands”.
   The mode chosen results in which samples can be sent to the printer.



Welcome to the Star Printer iOS SDK!
Please choose a command type

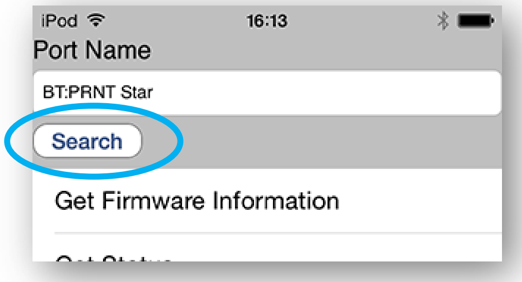
Line Mode
Printers accept commands and print data line-by-line. The data is transferred to the printer in small pieces, allowing developers to customize receipt output with commands in any place they are needed. This mode alone can only make use of Device Fonts installed on the printer, which can be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

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.

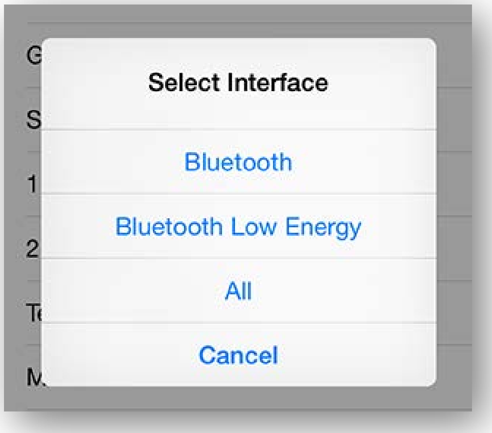
Bluetooth and Bluetooth Low Energy Printers

･To search for the printer and configure connection

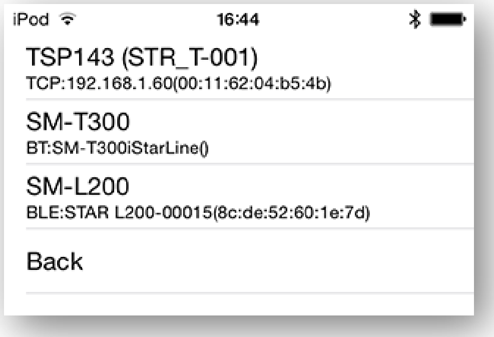
1. Tap “Search” to find all connected Star Bluetooth and Bluetooth Low Energy Printers.



2. Tap the interface type of the printer you want to connect to.
   In case of "Bluetooth", the port names of paired printers you can connect to are displayed.
   In case of “Bluetooth Low Energy”, the printers located nearby are detected and listed.



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



TSP143 (STR_T-001)
TCP:192.168.1.60(00:11:62:04:b5:4b)

SM-T300
BT:SM-T300iStarLine()

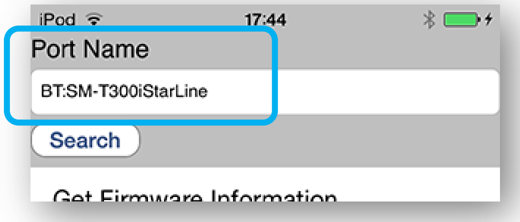
SM-L200
BLE:STAR L200-00015(8c:de:52:60:1e:7d)

Back

· **To configure connection manually**

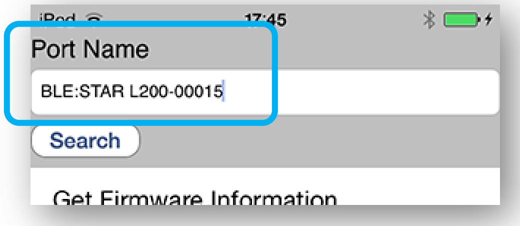
1. Type the IP Address or the Bluetooth Device Name manually in the "PortName" field as shown below.

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



BT:SM-T300iStarLine

In Bluetooth Low Energy, if manually entered, type: BLE:<Device Name>
or
BLE:<MAC Address>
*Without brackets



BLE:STAR L200-00015

Overview of How This iOS SDK is Designed

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

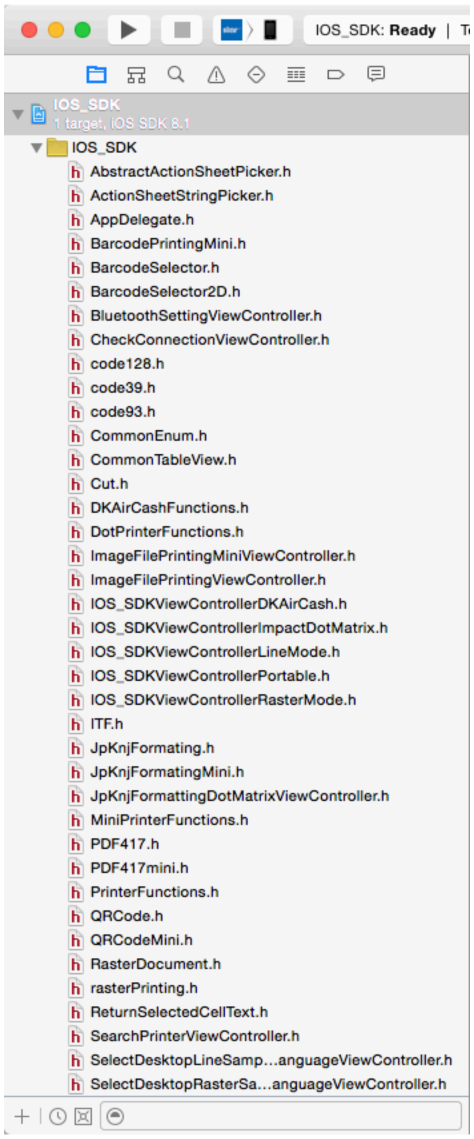
All functionality is located in the IOS_SDK project and IOS_SDK target.

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.

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


IOS_SDK: **Ready** | T

IOS_SDK
1 target, IOS SDK 8.1

- IOS_SDK
  - AbstractActionSheetPicker.h
  - ActionSheetStringPicker.h
  - AppDelegate.h
  - BarcodePrintingMini.h
  - BarcodeSelector.h
  - BarcodeSelector2D.h
  - BluetoothSettingViewController.h
  - CheckConnectionViewController.h
  - code128.h
  - code39.h
  - code93.h
  - CommonEnum.h
  - CommonTableView.h
  - Cut.h
  - DKAirCashFunctions.h
  - DotPrinterFunctions.h
  - ImageFilePrintingMiniViewController.h
  - ImageFilePrintingViewController.h
  - IOS_SDKViewControllerDKAirCash.h
  - IOS_SDKViewControllerImpactDotMatrix.h
  - IOS_SDKViewControllerLineMode.h
  - IOS_SDKViewControllerPortable.h
  - IOS_SDKViewControllerRasterMode.h
  - ITF.h
  - JpKnjFormating.h
  - JpKnjFormatingMini.h
  - JpKnjFormattingDotMatrixViewController.h
  - MiniPrinterFunctions.h
  - PDF417.h
  - PDF417mini.h
  - PrinterFunctions.h
  - QRCode.h
  - QRCodeMini.h
  - RasterDocument.h
  - rasterPrinting.h
  - ReturnSelectedCellText.h
  - SearchPrinterViewController.h
  - SelectDesktopLineSamp…anguageViewController.h
  - SelectDesktopRasterSa…anguageViewController.h

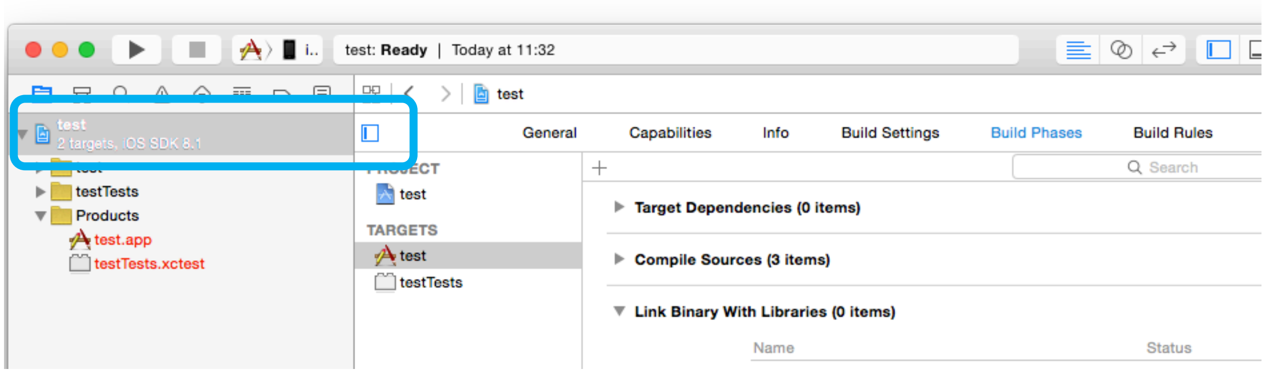
The StarIO Framework

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.

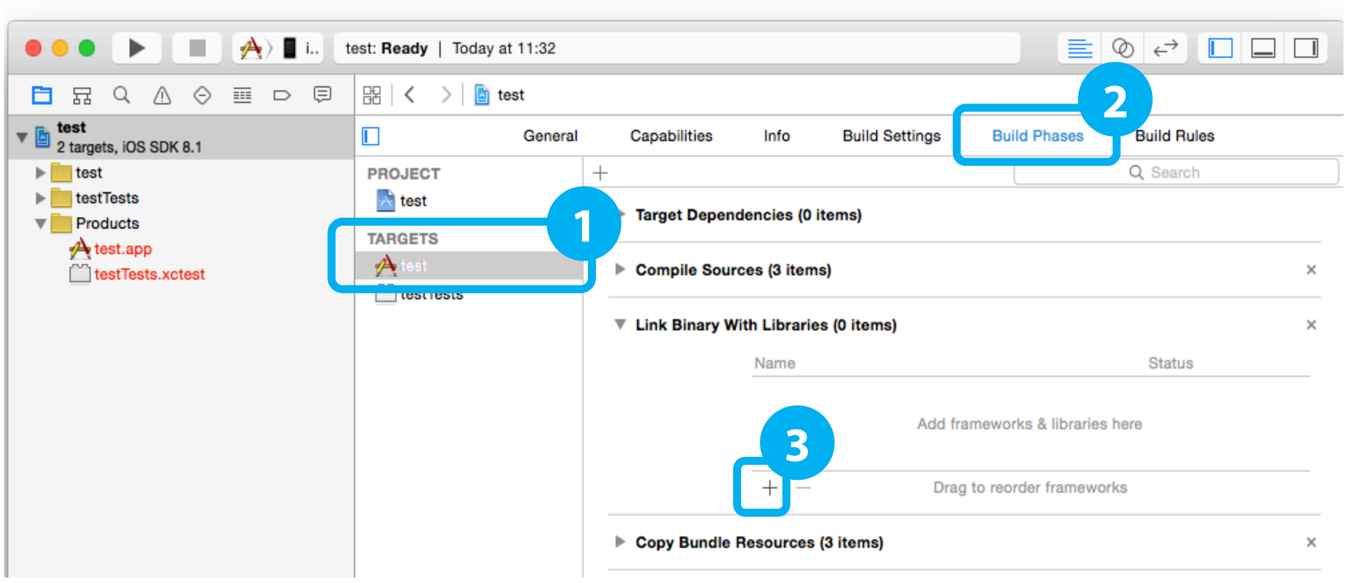
◆When building a new application

1. Add StarIO.framework into your project

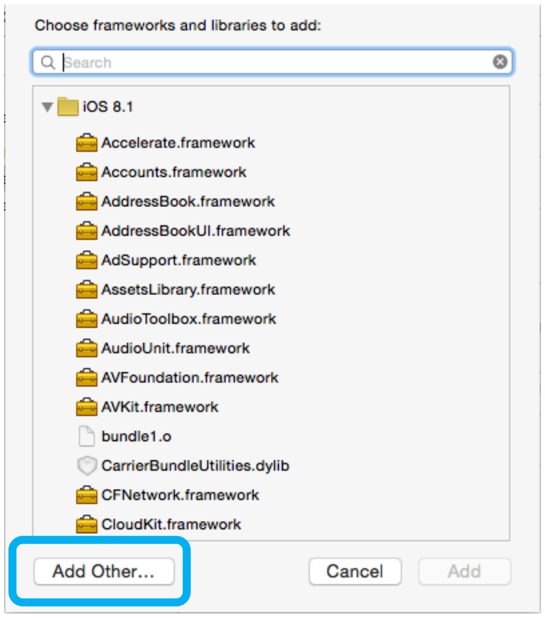
1. Click the created project.



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

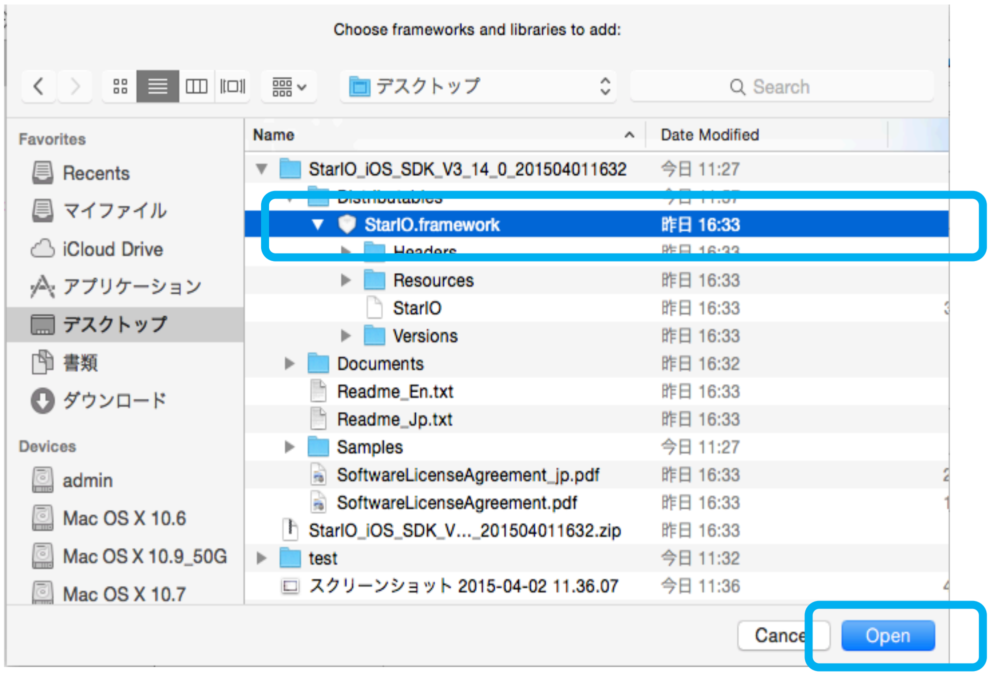

test: Ready | Today at 11:32
test
2 targets, iOS SDK 8.1
test
testTests
Products
test.app
testTests.xctest
PROJECT
TARGETS
1
2
3
General
Capabilities
Info
Build Settings
Build Phases
Build Rules
Target Dependencies (0 items)
Compile Sources (3 items)
Link Binary With Libraries (0 items)
Name
Status
Add frameworks & libraries here
Drag to reorder frameworks
Copy Bundle Resources (3 items)

3. Click the Add Other… button.


Choose frameworks and libraries to add:

- iOS 8.1
  - Accelerate.framework
  - Accounts.framework
  - AddressBook.framework
  - AddressBookUI.framework
  - AdSupport.framework
  - AssetsLibrary.framework
  - AudioToolbox.framework
  - AudioUnit.framework
  - AVFoundation.framework
  - AVKit.framework
  - bundle1.o
  - CarrierBundleUtilities.dylib
  - CFNetwork.framework
  - CloudKit.framework

4. Browse to the location of where the Star iOS SDK was unzipped and select StarIO.framework. Then click Open.

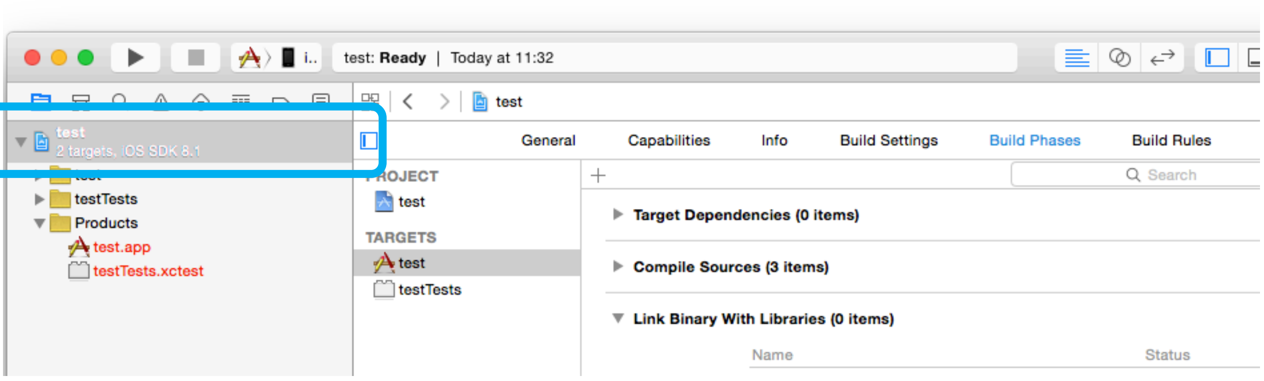

Choose frameworks and libraries to add:

Name	Date Modified
StarIO_iOS_SDK_V3_14_0_201504011632	今日 11:27
Distributables	今日 11:27
StarIO.framework	昨日 16:33
Headers	昨日 16:33
Resources	昨日 16:33
StarIO	昨日 16:33
Versions	昨日 16:33
Documents	昨日 16:32
Readme_En.txt	昨日 16:33
Readme_Jp.txt	昨日 16:33
Samples	今日 11:27
SoftwareLicenseAgreement_jp.pdf	昨日 16:33
SoftwareLicenseAgreement.pdf	昨日 16:33
StarIO_iOS_SDK_V..._201504011632.zip	昨日 16:33
test	今日 11:32
スクリーンショット 2015-04-02 11.36.07	今日 11:36

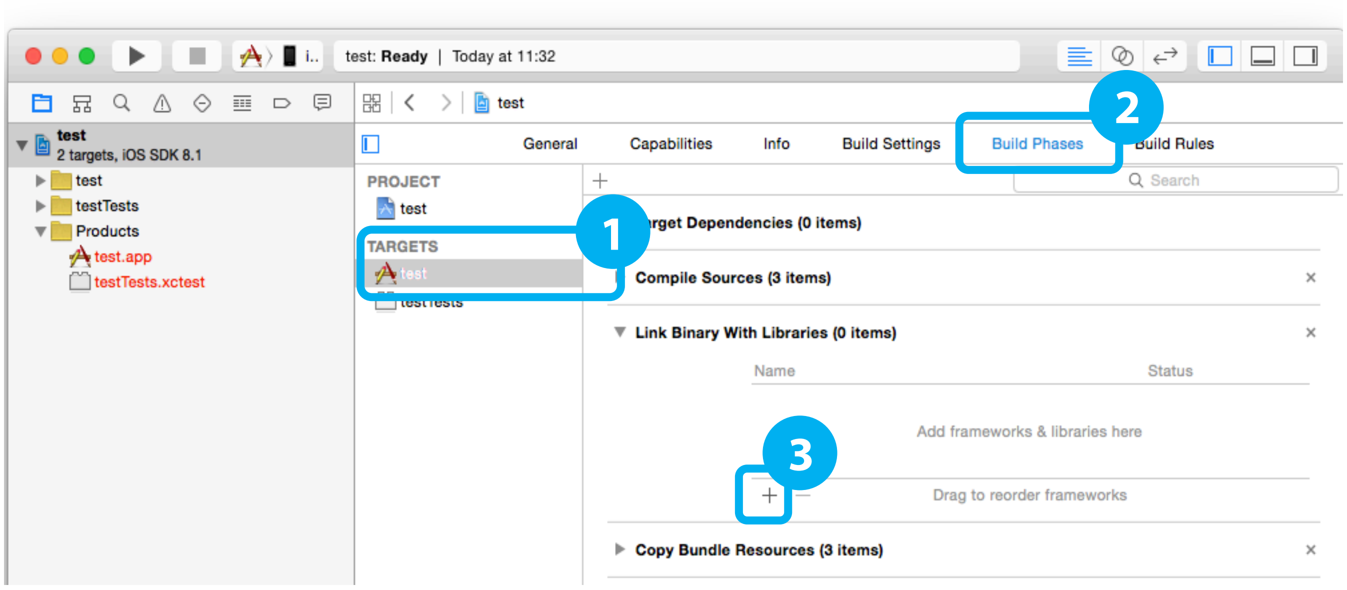
5. The framework is added to your project and all StarIO methods are now available to you.

2. Add other framework into your project

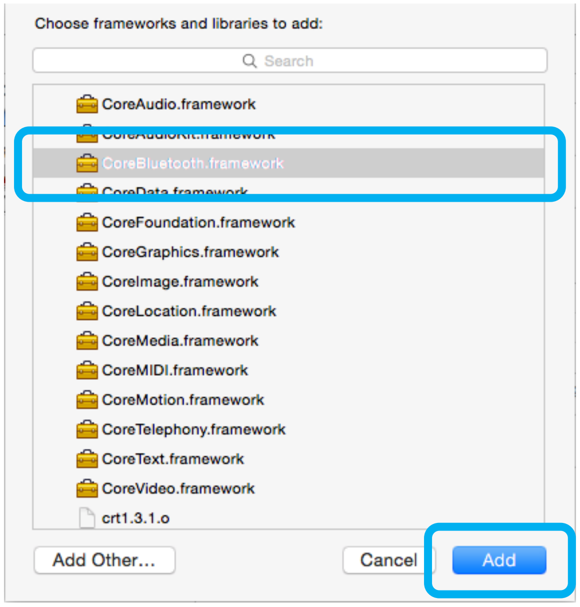
1. Click the created project.



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

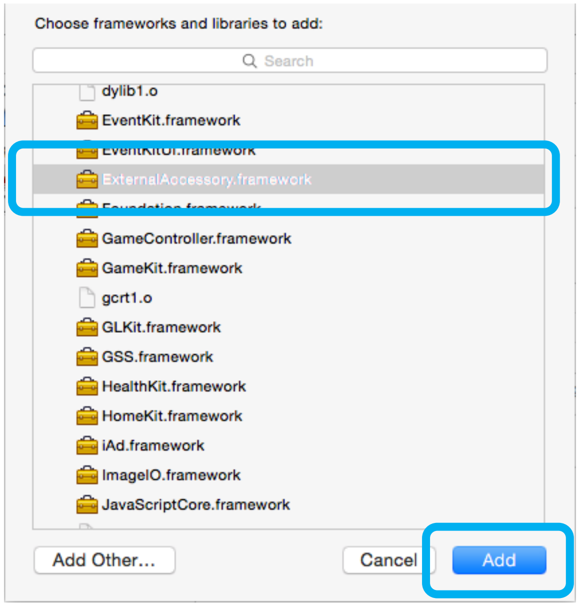

test: Ready | Today at 11:32
test
2 targets, iOS SDK 8.1
testTests
Products
test.app
testTests.xctest
General
Capabilities
Info
Build Settings
Build Phases
Build Rules
PROJECT
TARGETS
Search
1
2
3
Target Dependencies (0 items)
Compile Sources (3 items)
Link Binary With Libraries (0 items)
Name
Status
Add frameworks & libraries here
Drag to reorder frameworks
Copy Bundle Resources (3 items)

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



Choose frameworks and libraries to add:

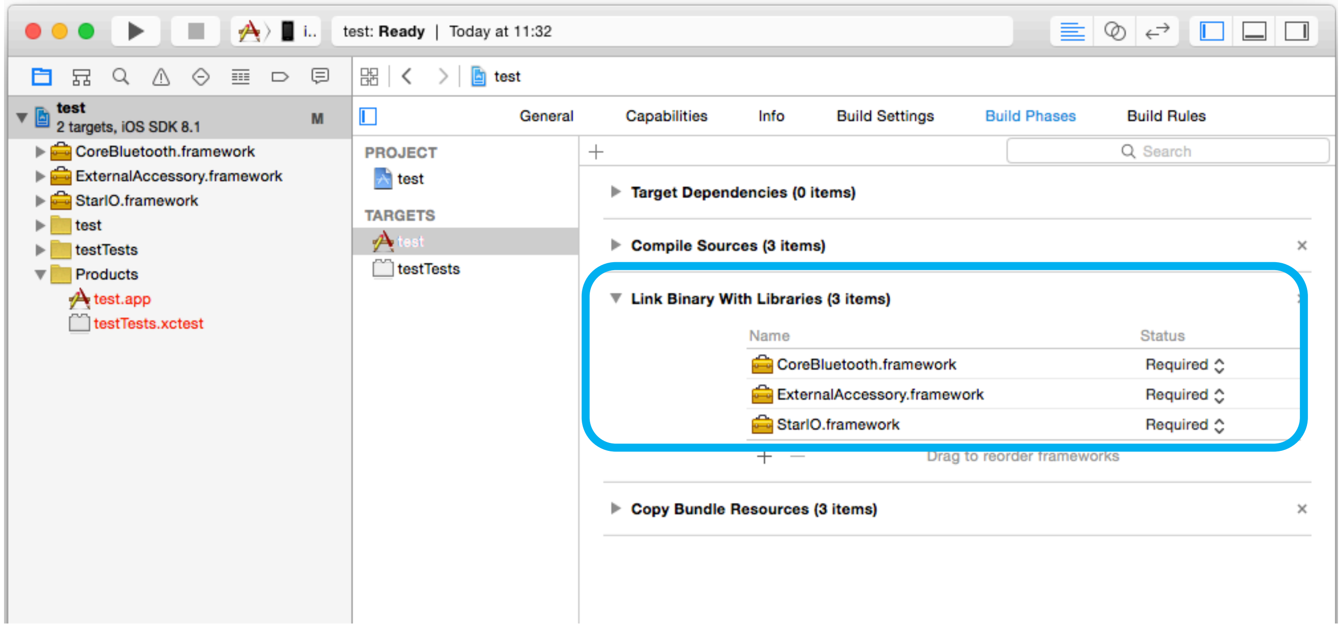
CoreAudio.framework
CoreAudioKit.framework
CoreBluetooth.framework
CoreData.framework
CoreFoundation.framework
CoreGraphics.framework
CoreImage.framework
CoreLocation.framework
CoreMedia.framework
CoreMIDI.framework
CoreMotion.framework
CoreTelephony.framework
CoreText.framework
CoreVideo.framework
crt1.3.1.o



Choose frameworks and libraries to add:

dylib1.o
EventKit.framework
EventKitUI.framework
ExternalAccessory.framework
Foundation.framework
GameController.framework
GameKit.framework
gcrt1.o
GLKit.framework
GSS.framework
HealthKit.framework
HomeKit.framework
iAd.framework
ImageIO.framework
JavaScriptCore.framework

4. Check if the necessary framework has been added.



test: Ready | Today at 11:32

test
2 targets, iOS SDK 8.1
CoreBluetooth.framework
ExternalAccessory.framework
StarIO.framework
test
testTests
Products
test.app
testTests.xctest

Target Dependencies (0 items)

Compile Sources (3 items)

Link Binary With Libraries (3 items)

Name	Status
CoreBluetooth.framework	Required
ExternalAccessory.framework	Required
StarIO.framework	Required

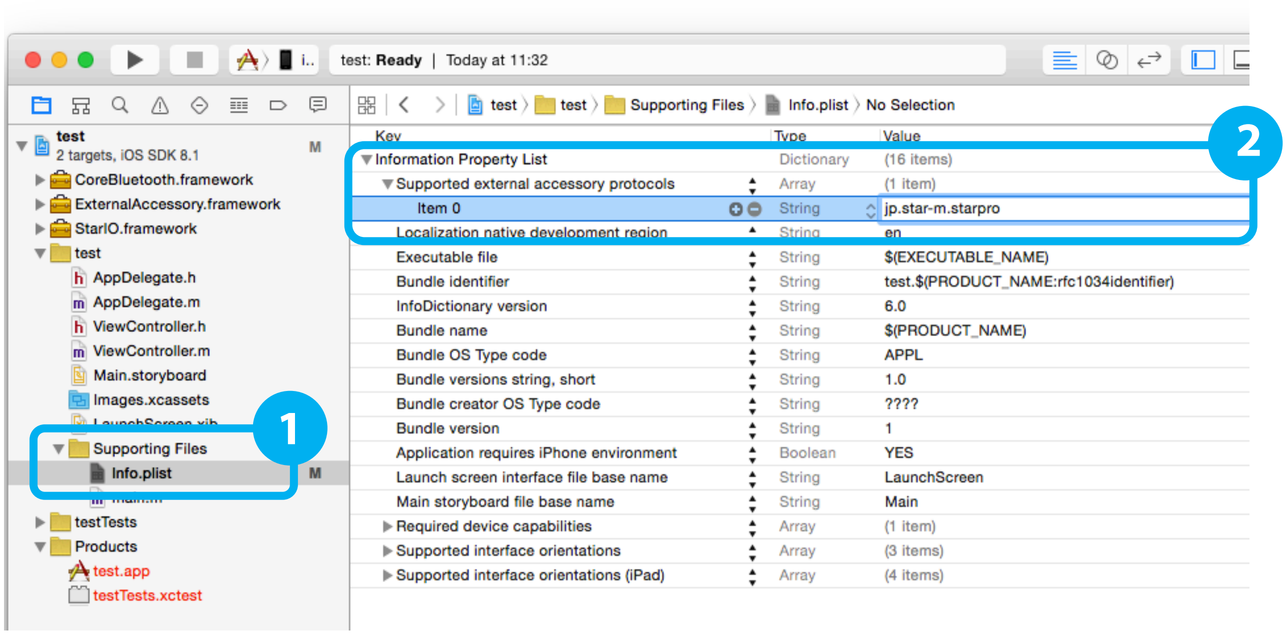
Drag to reorder frameworks

Copy Bundle Resources (3 items)

3. Edit information property list (Bluetooth printer only)

Note: **Please do not apply this, if you are not using Bluetooth ineterface.**

1. Click on the information property list file (default: Info.plist file).



Key	Type	Value
▼Information Property List	Dictionary	(16 items)
▼Supported external accessory protocols	Array	(1 item)
Item 0	String	jp.star-m.starpro
Localization native development region	String	en
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	test.$(PRODUCT_NAME:rfc1034identifier)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle creator OS Type code	String	????
Bundle version	String	1
Application requires iPhone environment	Boolean	YES
Launch screen interface file base name	String	LaunchScreen
Main storyboard file base name	String	Main
▶Required device capabilities	Array	(1 item)
▶Supported interface orientations	Array	(3 items)
▶Supported interface orientations (iPad)	Array	(4 items)

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 serach path", the previous StarIO.framework will be used.

The StarIO Methods Overview

SMPort Class:

●Property

portName	Acquires the printer port name.
portSettings	Acquires the port settings.
timeoutMillis	Acquires and specifies the timeout time for internal control and API.
endCheckedBlockTimeoutMillis	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, readwrite) 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.

●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. @”BT:PRNT Star” (In Bluetooth)

@”BT:00:11:62:1b:4d:f4”(To specify the MAC address in Bluetooth)

@”BLE:STAR L200-00001” (In Bluetooth Low Energy)

@”BLE:8C:DE:52:60:30:C6”(To specify the MAC address in Bluetooth Low Energy)

Note: iOS6 is- required to specify the MAC address in Bluetooth. Other iOS versions cannot use this function.

portSettings – Specify @”portable” for portSettings.

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).

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 = @"BT:PRNT Star";
NSString *portSettings = @"portable";
@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.

Notification in case of SM-L Series

It could take some time when an iOS device tries to connect to a printer via "Bluetooth Low Energy". When it fails to connect, please keep retrying until success. If the connection time must be reduced, please design your application as the connection to a printer always keeps opening.

*In this case, the printer cannot be detected by any other applications and devices.

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 Bluetooth printers or Bluetooth Low Energy printers.

Parameters:

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

When @"BLE:" is specified, Bluetooth Low Energy 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.

When getting the printer device name using searchPriner method for the first time, sometimes portName will be @"BLE:".
In those cases, please connect the printer using getPort method.
Once you have got the Device name, searchPrinter method works correctly..

```
//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 Bluetooth printers and Bluetooth Low Energy 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_int_32_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:

Bluetooth:

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.

Bluetooth Low Energy:

It returns a transmission data size when it succeeded and "0" when it failed.

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 opne 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
blackMarkError	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.
compulsionSwitch	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.
coverOpen	Cover Status	SM_BOOLEAN	You can check status of Cover " SM_TRUE " : Cover is opened. " SM_FALSE " : Cover is closed.
cutterError	Auto-cutter Error	SM_BOOLEAN	You can check status of Cutter " SM_TRUE " : Cutter error occurs. " SM_FALSE " : Cutter error does not occur.
etbAvailable	ETB available or not	SM_BOOLEAN	" SM_TRUE " : available to use " SM_FALSE " : not available to use
etbCounter	ETB Counter	UCHAR	You can get current value of ETB
headThermistorError	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.
offline	ONLINE/OFFLINE Status	SM_BOOLEAN	You can check status of Online or offline. " SM_TRUE " : Printer is Offline. " SM_FALSE " : Printer is Online
overTemp	Stopped by high head temperature	SM_BOOLEAN	" SM_TRUE " : Printer is stopped by head temperature. " SM_FALSE " : Printer is not stopped by head temperature.
presenterPaperJamError	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 .
presenterState	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.
raw	Byte column of status	UCHAR[63]	Byte column of status (example : HEX 23 86 00 00 00 00 00 00 00)
rawLength	raw length	CHAR	raw length
receiptPaperEmpty	Paper end	SM_BOOLEAN	" SM_TRUE " : Paper end. " SM_FALSE " : Paper exist.
receiptPaperNearEmptyInner	Paper Near-end (Inner Side)	SM_BOOLEAN	" SM_TRUE " : Paper near-end. " SM_FALSE " : Paper does not near-end.
receiveBufferOverflow	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.
unrecoverableError	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.)
voltageError	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

mamber name	SM-T300	SM-S210i *	SM-S220i **	SM-S230i **	SM-T300i	SM-T400i	mPOP
blackMarkError	✓	✓	✓	✓	✓	✓	
compulsionSwitch							✓
coverOpen	✓	✓	✓	✓	✓	✓	✓
cutterError							✓
etbAvailable	✓	✓	✓	✓	✓	✓	✓
etbCounter	✓	✓	✓	✓	✓	✓	✓
headThermistorError							✓
offline	✓	✓	✓	✓	✓	✓	✓
overTemp	✓	✓	✓	✓	✓	✓	✓
presenterPaperJamError							
presenterState							
raw	✓	✓	✓	✓	✓	✓	✓
rawLength	✓	✓	✓	✓	✓	✓	✓
receiptPaperEmpty	✓	✓	✓	✓	✓	✓	✓
receiptPaperNearEmptyInner							
receiveBufferOverflow							
unrecoverableError							✓
voltageError							✓

*** JP Model only ** Models for US and Europe only**

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.

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 (*1) or printer error occurs during printing, it returns PortException.

(*1) 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

```
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];
}
```

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.

StarIOVersion

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

This method gets the StarIO version.

Returns:

StarIO version

SMBluetoothManager Class:

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

●Property

portName	Acquires the portName of the device to be connected.
deviceType	Acquires the type of the device to be connected.
opened	Shows whether the port is opened.
deviceName	Acquires and specifies the current Bluetooth device name.
iOSPortName	Acquires and specifies the port name to be used with the StarIO.
autoConnect	Acquires and specifies the setting (Valid or Invalid) of the autoconnection function.
security	Acquires the Bluetooth security setting (SPP or PIN Mode)
pinCode	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.
This name is displayed when you are pairing via Bluetooth.
When using Bluetooth Low Energy, it is used as a connection port name for communication.
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 / * + ~ ^ [{ ( ] }) | \

In case of Bluetooth Low Energy, the changed Bluetooth names are effective after turning the device off and on and connecting again.

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

Acquires and specifies the iOS port name to be used with the StarIO for Bluetooth communication. This function si not used with Bluetooth Low Energy.
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.
This function is available only for Bluetooth.
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(No security or PIN code mode).
This function is available only for Bluetooth.
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.
This function is available only for Bluetooth.
It can not acquire the current setting.
Set to nil when the PIN code is not changed.

Valid number of characters:	4 to 16 [*1] [*1] The PIN code for SM-L200 must be 4 digits.
Valid characters:	0-9, a-z, A-Z[*2] [*2] Supported characters for SM-L200 are only “0“ - “9”.

● 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 Ex. "BT:PRNT Star"
deviceType	– the type of the device to be connected SMDeviceTypePortablePrinter(Specify portable printer)

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 or Bluetooth Low Energy 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, iOSPortName, 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 paring again.

close

```
-(void) close
```

This method is used to close communication with the printer.

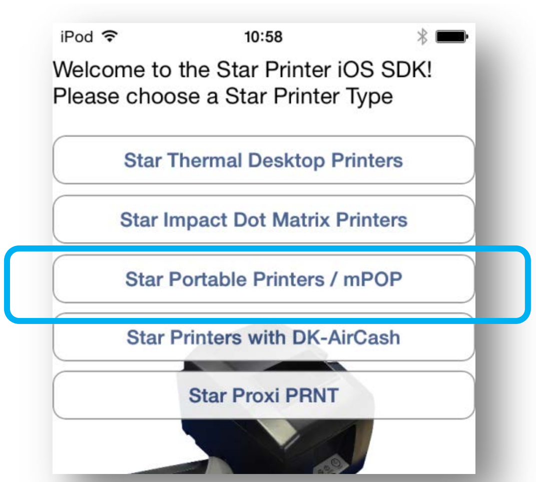
StarIO iOS SDK Functionality

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. Tap "Star Portable Printers / mPOP".

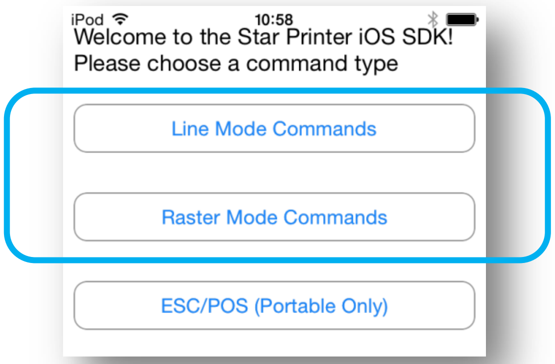


iPod 10:58

Welcome to the Star Printer iOS SDK!
Please choose a Star Printer Type

Star Thermal Desktop Printers
Star Impact Dot Matrix Printers
Star Portable Printers / mPOP
Star Printers with DK-AirCash
Star Proxi PRNT

2. Tap “Line Mode Commands” or “Raster Mode Commands”.
   The mode chosen results in which samples can be sent to the printer.



iPod 10:58

Welcome to the Star Printer iOS SDK!
Please choose a command type

Supported Samples by Command Type

Line Mode Command Samples Include:

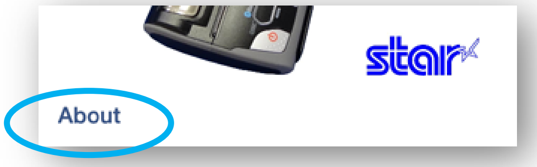
- Port Discovery
- Get Firmware Information
- Get Status
- Sample Receipt
- 1D Barcodes
- 2D Barcodes
- Text Formatting
- Bluetooth Setting

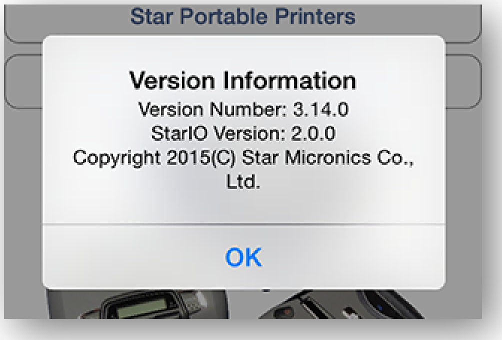
Raster Mode Command Samples Include:

- Port Discovery
- Get Firmware Information
- Get Status
- Sample Receipt
- Raster Graphics Text Printing
- Image File Printing
- AllReceipts
- Bluetooth Setting

Get StarIO Version

◆Line ◆Raster



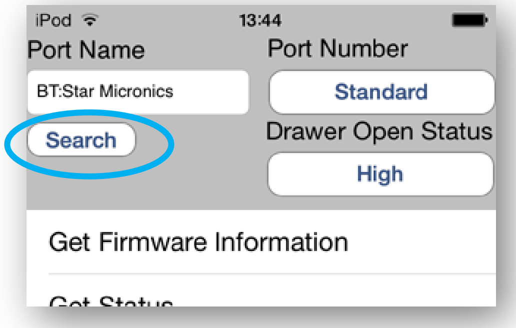


Version Information
Version Number: 3.14.0
StarIO Version: 2.0.0
Copyright 2015(C) Star Micronics Co., Ltd.

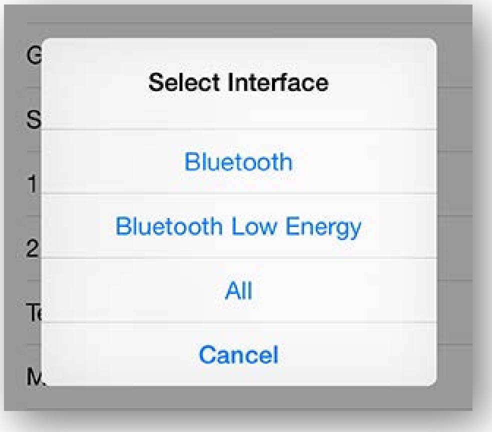
Tap "About", displays StarIO version.

Port Discovery

◆Line ◆Raster



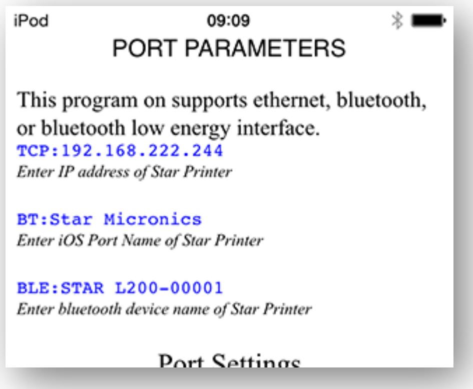
Port Name: BT:Star Micronics
Port Number: Standard
Drawer Open Status: High



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.

Help

◆Line ◆Raster


iPod 09:09

PORT PARAMETERS

This program on supports ethernet, bluetooth, or bluetooth low energy interface.

TCP:192.168.222.244

Enter IP address of Star Printer

BT:Star Micronics

Enter iOS Port Name of Star Printer

BLE:STAR L200-00001

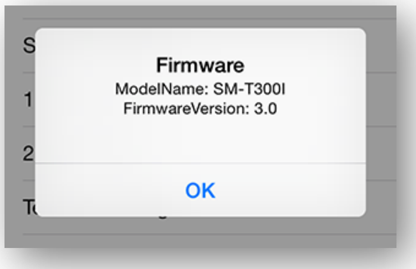
Enter bluetooth device name of Star Printer

Port Settings

Help produces information on network port settings.

Get Firmware Information

◆Line ◆Raster


Firmware

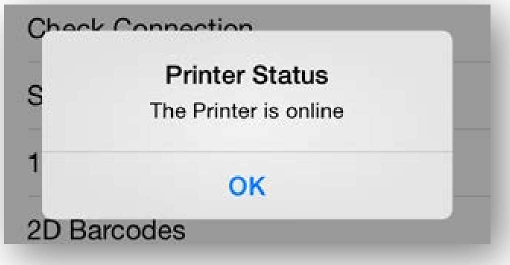
ModelName: SM-T300I

FirmwareVersion: 3.0

Displays firmware information of the printer specified by Port Name.

Get Status

◆Line ◆Raster



Printer Status

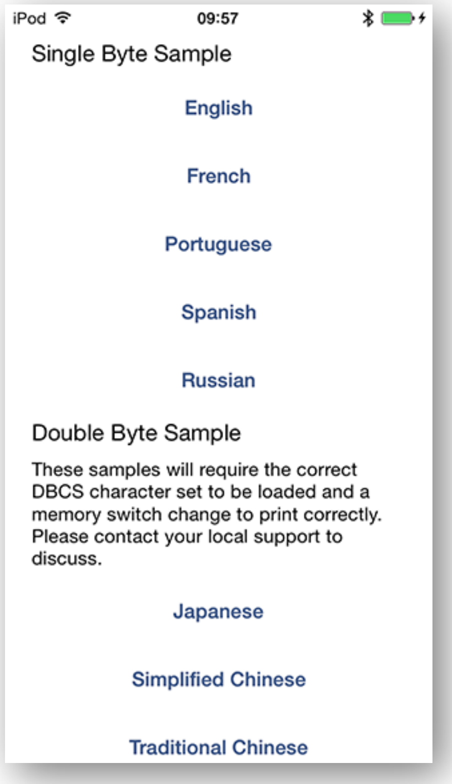
The Printer is online

StarPrinterStatus

public boolean retreiveStatus()	See status return values here
offline	false = printer online; true = printer offline
other	See status return values here

Sample Receipt

◆Line ◆Raster



iPod 09:57

Single Byte Sample

English

French

Portuguese

Spanish

Russian

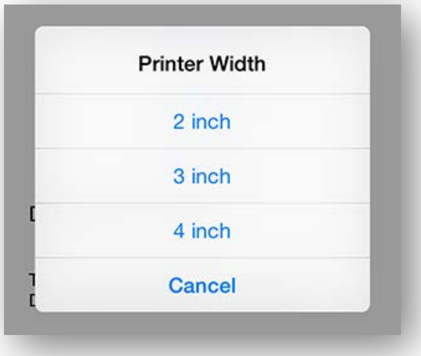
Double Byte Sample

These samples will require the correct DBCS character set to be loaded and a memory switch change to print correctly. Please contact your local support to discuss.

Japanese

Simplified Chinese

Traditional Chinese



Printer Width

2 inch

3 inch

4 inch

Cancel

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.

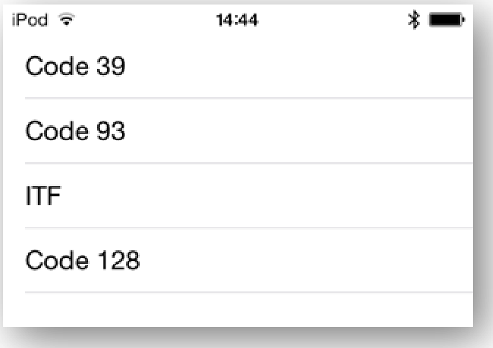
When using Line Mode commands, printing in Russian and Simplified Chinese is not supported with SM-210i, SM-S220i, SM-S230i, SM-T300i and SM-T400i.

When using Line Mode commands, Simplified Chinese will be supported from firmware version 1.1 for SM-L200.

1D Barcodes

◆Line

<<CODE39>>



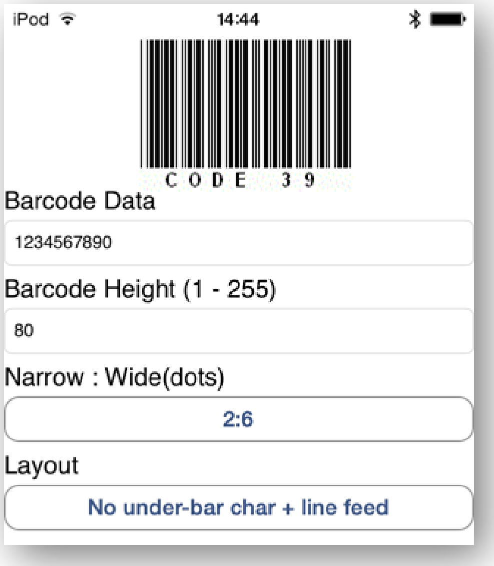
iPod 14:44

Code 39

Code 93

ITF

Code 128



iPod 14:44

Barcode Data

1234567890

Barcode Height (1 - 255)

80

Narrow : Wide(dots)

2:6

Layout

No under-bar char + line feed

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)

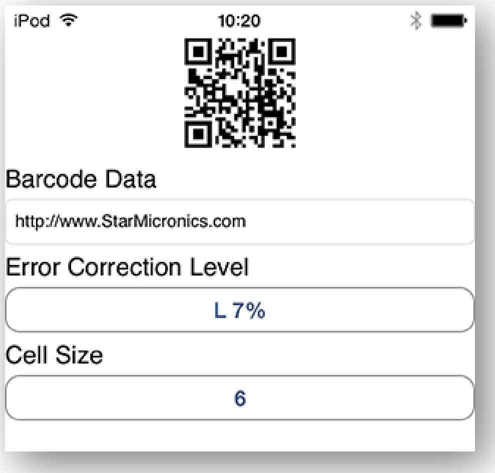
* These barcodes are supported by Star POS Printers, but no example is in the sample application.

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＞＞



iPod 10:20

Barcode Data

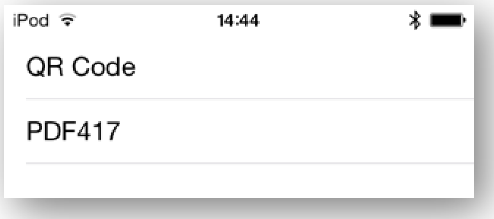
http://www.StarMicronics.com

Error Correction Level

L 7%

Cell Size

6



iPod 14:44

QR Code

PDF417

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

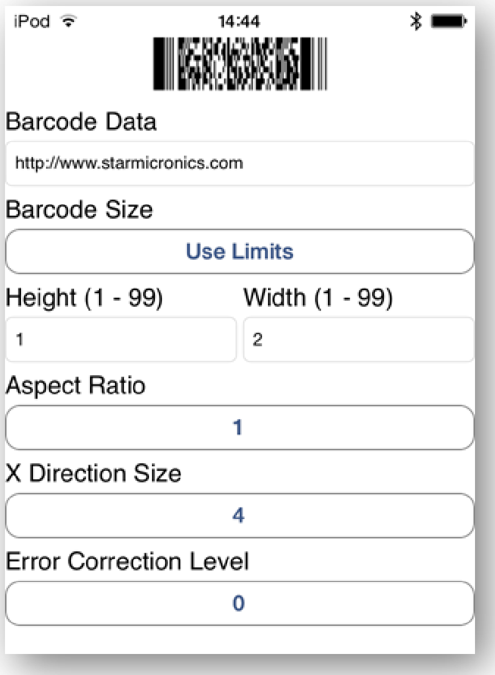
This is the order in which commands need to be sent to print the QR Code:

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＞＞



Barcode Data
http://www.starmicronics.com
Barcode Size
Use Limits
Height (1 - 99): 1
Width (1 - 99): 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

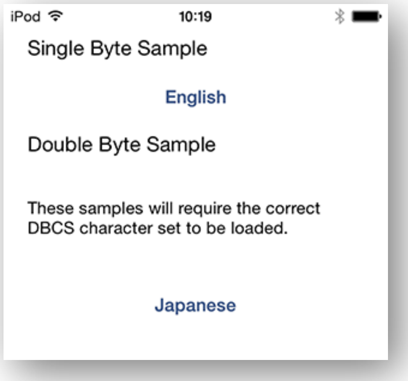
Refer to the Line Mode Programming Manual for a listing of all PDF417 commands.

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.

⚠ PDF417 will be supported from firmware version 1.1 for SM-L200.

Text Formatting

◆Line


iPod 10:19

Single Byte Sample

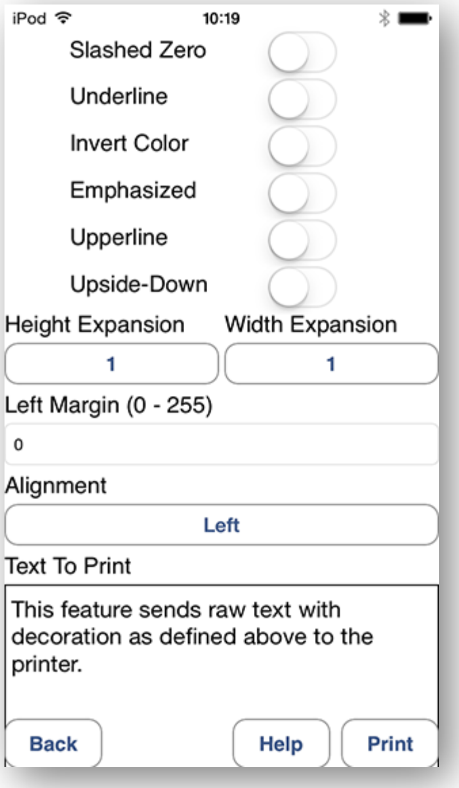
English

Double Byte Sample

These samples will require the correct DBCS character set to be loaded.

Japanese

English


iPod 10:19

Slashed Zero

Underline

Invert Color

Emphasized

Upperline

Upside-Down

Height Expansion: 1

Width Expansion: 1

Left Margin (0 - 255): 0

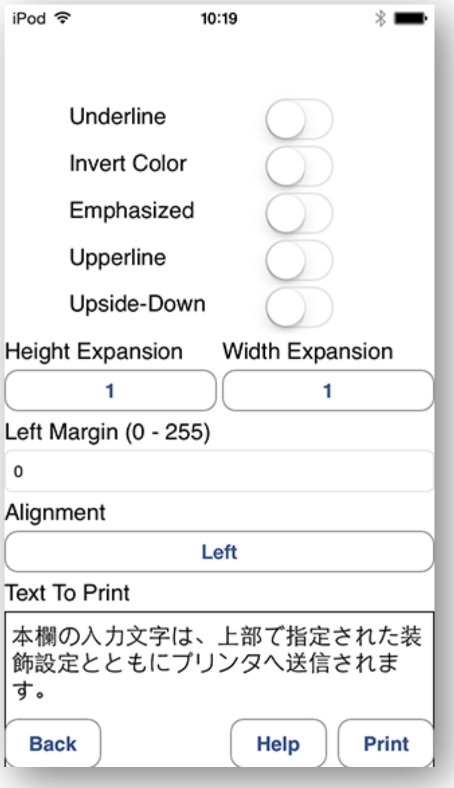
Alignment: Left

Text To Print

This feature sends raw text with decoration as defined above to the printer.

Back Help Print

Japanese


iPod 10:19

Underline

Invert Color

Emphasized

Upperline

Upside-Down

Height Expansion: 1

Width Expansion: 1

Left Margin (0 - 255): 0

Alignment: Left

Text To Print

本欄の入力文字は、上部で指定された装飾設定とともにプリンタへ送信されます。

Back Help Print

Text Formatting (continued from above)

Slashed Zero (only ASCII)

ESC / 1 = on	ESC / 0 = off

Underline

ESC – 1 = on	ESC – 0 = off [Default]

Invert Color (B/W)

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 (Excluding SM-L200)
Width Expansion	ESC W n	0 ≤ n ≤ 2 (SM-L200)

Left Margin

ESC l n	0 ≤ n ≤ 255

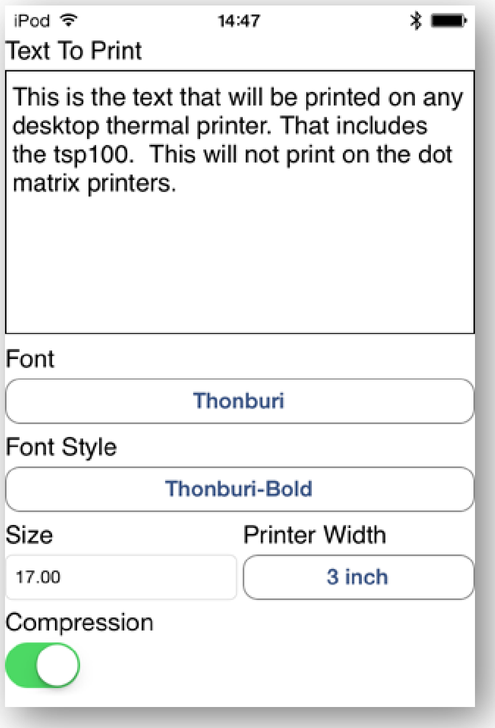
Alignment

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.

Raster Graphical Text Printing

◆Raster



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

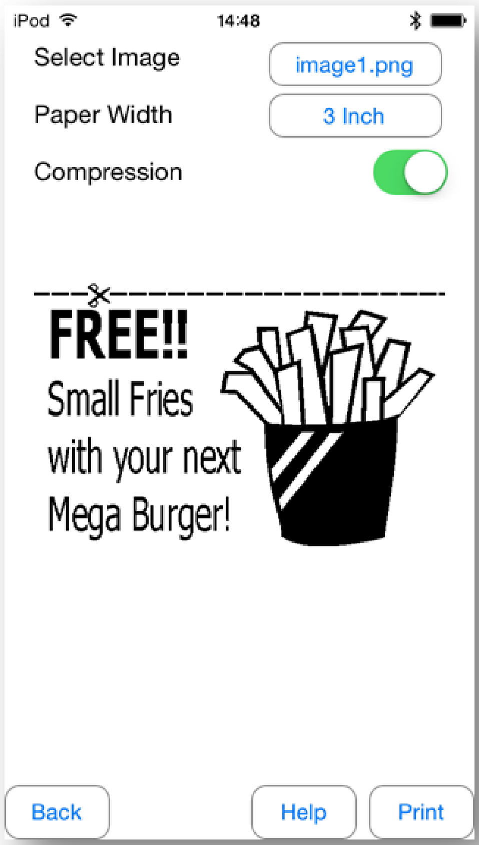
Compression

Refer to the mStar 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



Select Image: image1.png
Paper Width: 3 Inch
Compression

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.

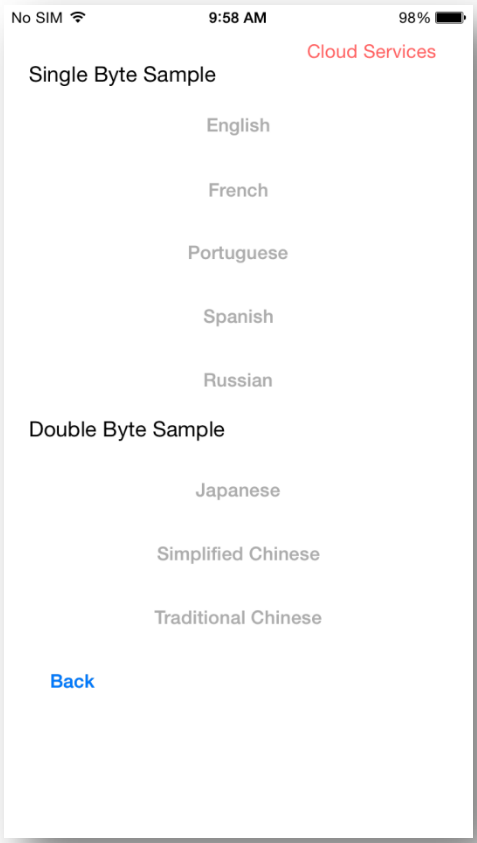
Raster Mode converts all print data into image data and then outputs it to the printer. This enables Star Printers to print at high speeds, regardless of outputting receipts with only text or text and logos/coupons. As there are many options on how to customize output in Raster Mode, refer to the Line 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.

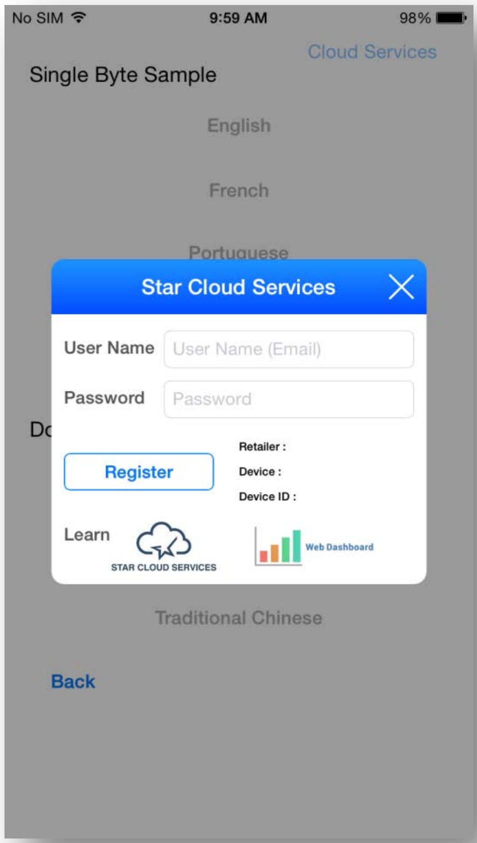
Using "Compression API" method may improve through put.

Note: This sample is not available in Line Mode.

AllReceipts

◆Raster



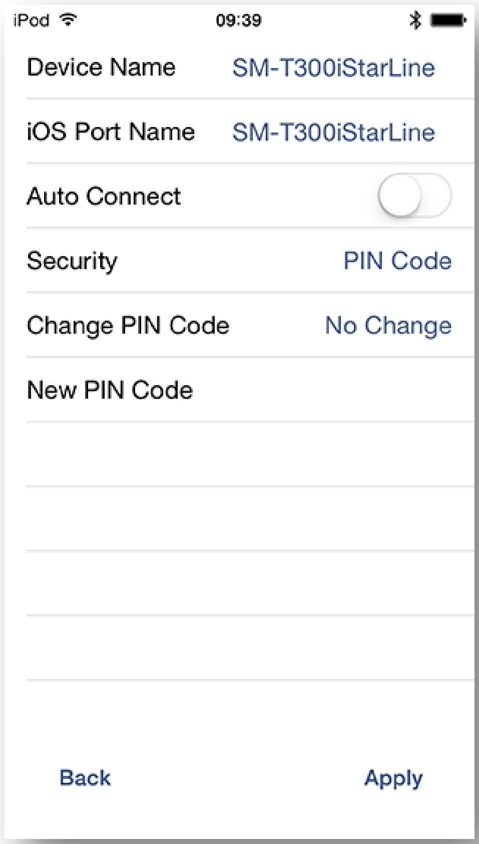


When you tap “Cloud Services”, you can get “Registration View” for the device registration to Star Cloud Services.

For details of AllReceipts and Star Cloud Services, refer to the Star_AllReceipts_iOS_SDK_Manual.

Bluetooth Setting

◆Line ◆Raster



Device Name	SM-T300iStarLine
iOS Port Name	SM-T300iStarLine
Auto Connect	
Security	PIN Code
Change PIN Code	No Change
New PIN Code	

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 paring again. In case of Bluetooth Low Energy, the values are effective after turning the device off and on and then connecting 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: 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 #4: 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 #5: 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
Apr. 2015	3.14.0	- Manual newly issued
Aug. 2015	3.15.0	- Added mPOP support
Mar. 2016	3.16.0	- Added AllReceipts support

Star Micronics is a global leader in the manufacturing of small printers. We apply over 50 years of knowhow and innovation to provide elite printing solutions that are rich in stellar reliability and industry-respected features. Offering a diverse line of Thermal, Hybrid, Mobile, Kiosk and Impact Dot Matrix printers, we are obsessed with exceeding the demands of our valued customers every day.

We have a long history of implementations into Retail, Point of Sale, Hospitality, Restaurants and Kitchens, Kiosks and Digital Signage, Gaming and Lottery, ATMs, Ticketing, Labeling, Salons and Spas, Banking and Credit Unions, Medical, Law Enforcement, Payment Processing, and more!

High Quality POS Receipts, Interactive Coupons with Triggers, Logo Printing for Branding, Advanced Drivers for Windows, Mac and Linux, Complete SDK Packages, Android, iOS, Blackberry Printing Support, OPOS, JavaPOS, POS for .NET, Eco-Friendly Paper and Power Savings with Reporting Utility, ENERGY STAR, MSR Reading, *future*PRNT, StarPRNT… How can Star help you fulfill the needs of your application?

Don't just settle on hardware that won't work as hard as you do. Demand everything from your printer. Demand a Star!

Star Micronics Worldwide

Star Micronics Co., Ltd.
536 Nanatsushinya
Shimizu-ku, Shizuoka 424-0066
Japan
+81-54-347-2163
http://www.star-m.jp/eng/index.htm

Star Micronics America, Inc.
65 Clyde Road. Suite G
Somerset, NJ 08873
USA
1-848-216-3300
http://www.starmicronics.com

Star Micronics EMEA
Star House
Peregrine Business Park, Gomm Road
High Wycombe, Buckinghamshire HP13 7DL
UK
+44-(0)-1494-471111
http://www.star-emea.com

Star Micronics Southeast Asia Co., Ltd.
Room 2902C. 29th Fl. United Center Bldg.
323 Silom Road, Silom Bangrak, Bangkok 10500
Thailand
+66-2-631-1161 x 2
http://www.starmicronics.co.th/