Star Micronics Cloud Services
iOS SDK User's Manual

Aug 24, 2015

General Outline

This document provides information about the Star Micronics Cloud Services iOS SDK,

showing guidelines for our customers to build the Star Micronics Cloud Services applications.

Version History

Release Date	Update
Aug 24, 2015	1st Release

About this Manual

This manual is designed to help you understand SMCloudServices.framework and how to build an iOS application.
Communication with printers and print data generation make use of the StarIO.framework and StarIO_Extension.framework. **When you would like to know StarIO.framework and StarIO_Extension.framework details, please download StarIO iOS SDK and Star mPOP iOS SDK, and refer to the document included in the package.**
It is important for the readers of this document to have understood the basics of the Objective-C language, because this document is prepared for such application developers.
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.

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.

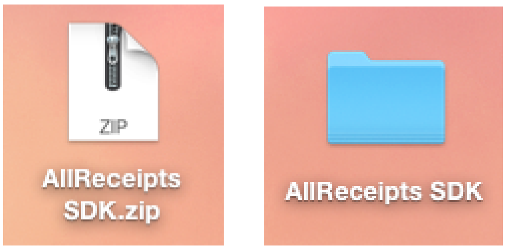
Contents

1 Getting Started

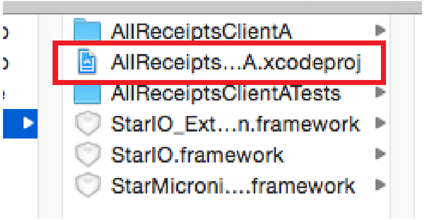
To build an iOS project, Xcode are needed. These tools are available in one package from the Apple Developer Site or Mac App Store. It is important to note that in order to produce applications that will actually run on an iOS device, you must be part of the Apple Developer Program, which requires a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

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

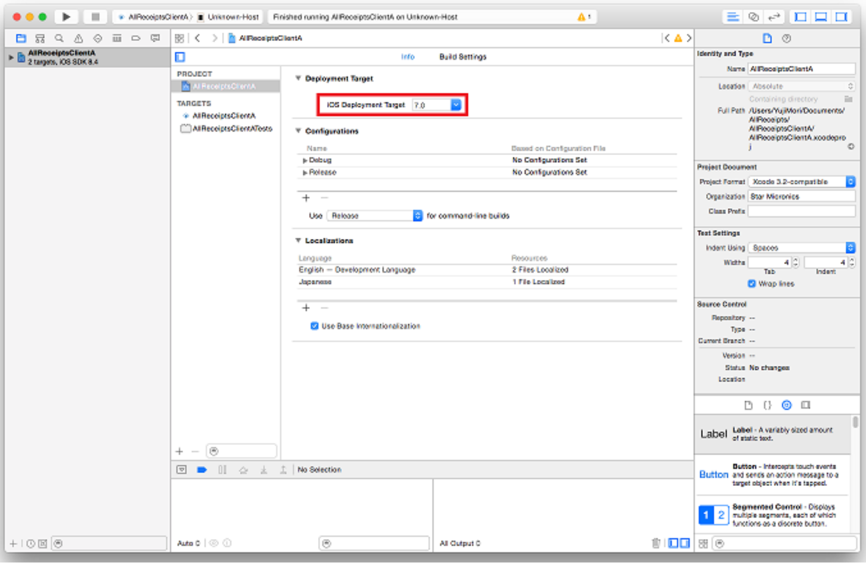


2. Open AllReceiptsClientA.xcodeproj.



- AllReceiptsClientA
- AllReceipts…A.xcodeproj
- AllReceiptsClientATests
- StarIO_Ext…n.framework
- StarIO.framework
- StarMicroni….framework

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



AllReceiptsClientA 〉 Unknown-Host
Finished running AllReceiptsClientA on Unknown-Host

AllReceiptsClientA
2 targets, iOS SDK 8.4

Info | Build Settings

PROJECT
AllReceiptsClientA
TARGETS
AllReceiptsClientA
AllReceiptsClientATests

Deployment Target
iOS Deployment Target 7.0

Configurations

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

Use Release for command-line builds

Localizations

Language	Resources
English — Development Language	2 Files Localized
Japanese	1 File Localized

☑ Use Base Internationalization

No Selection

Identity and Type
Name AllReceiptsClientA
Location Absolute
Containing directory
Full Path /Users/YujiMori/Documents/AllReceipts/AllReceiptsClientA/AllReceiptsClientA.xcodeproj

Project Document
Project Format Xcode 3.2-compatible
Organization Star Micronics
Class Prefix

Text Settings
Indent Using Spaces
Widths 4 Tab 4 Indent
☑ Wrap lines

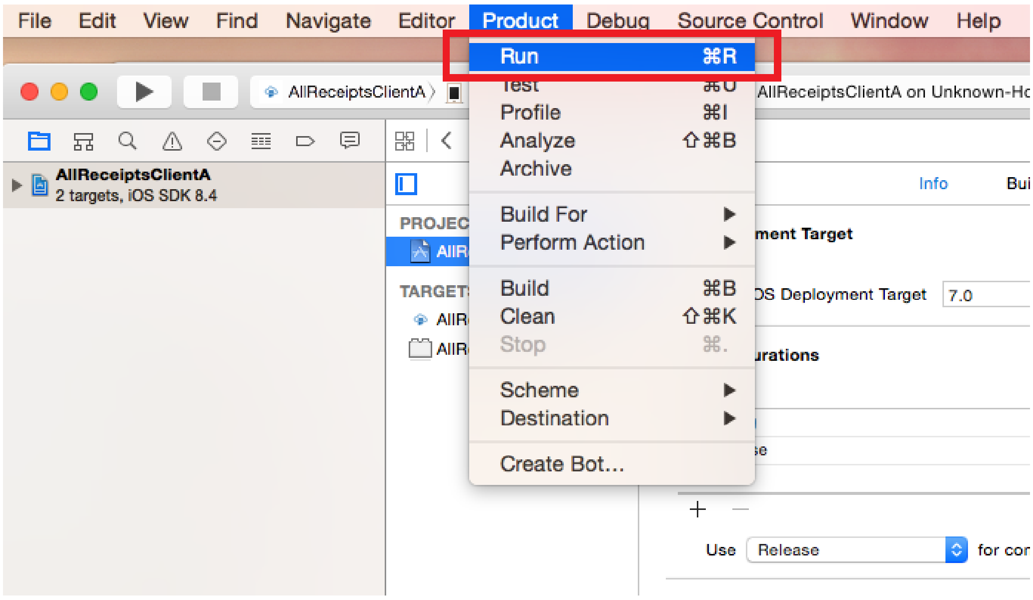
Source Control
Repository --
Type --
Current Branch --
Version --
Status No changes
Location

Label Label - A variably sized amount of static text.
Button Button - Intercepts touch events and sends an action message to a target object when it's tapped.
1 2 Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Auto ◇
All Output ◇

Running the project

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

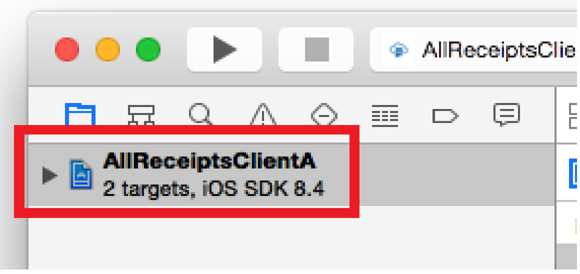


2 How to configuration AllReceipts Application

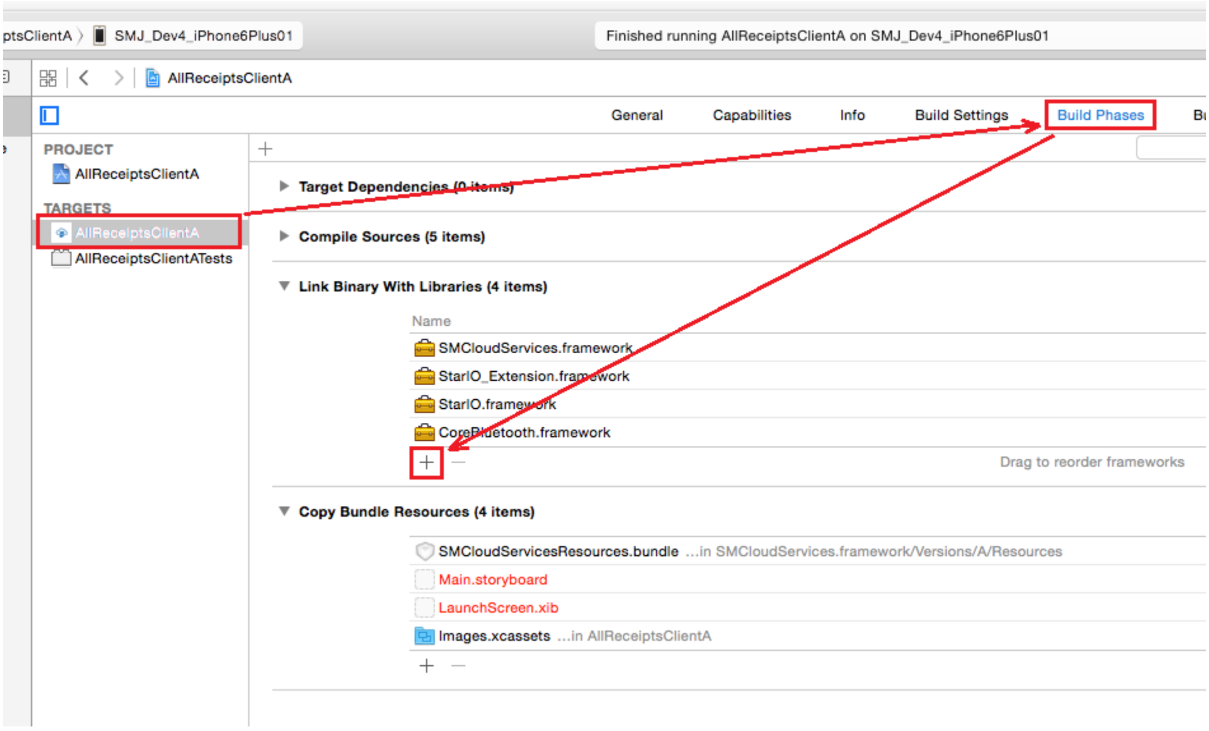
The SMCloudServices.framework, StarIO.framework and StarIO_Extension.framework is already included when the Star AllReceipts iOS SDK is loaded in Xcode; there's no need to include it again when testing our SDK. However, when you are building your own application, it is necessary to add the framework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

2.1 Add SMCloudServices.framework

1. Click on the project.

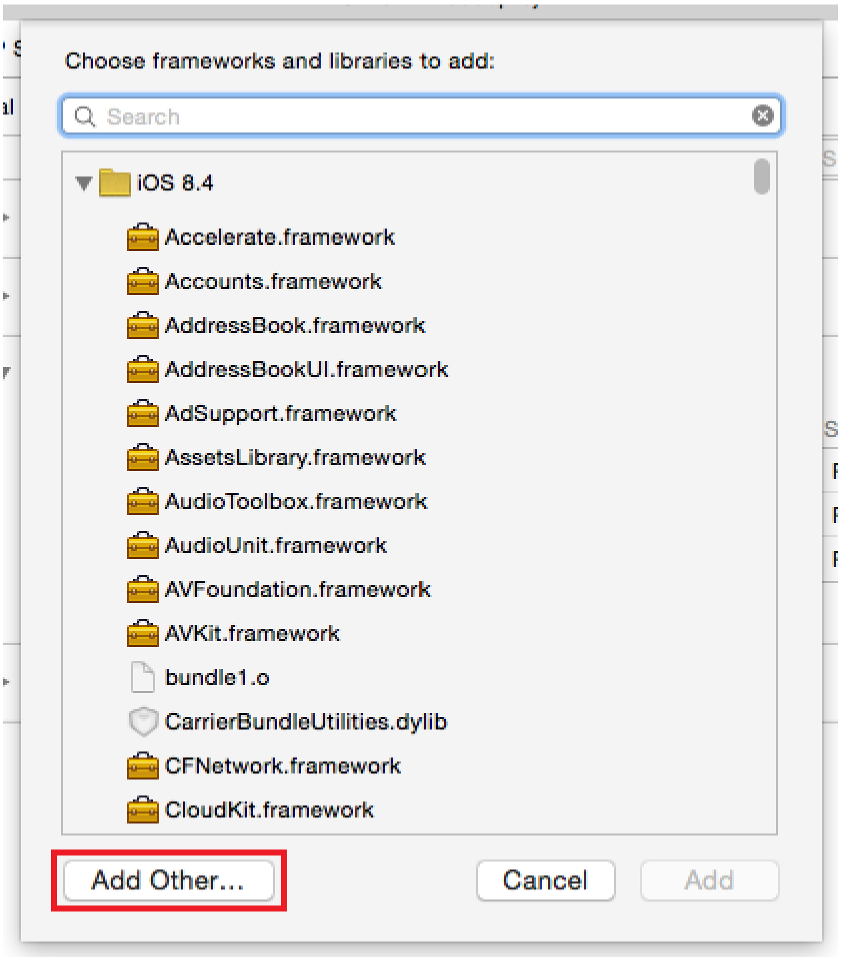


2. Open a target. Click the Build Phases tab and click the +.

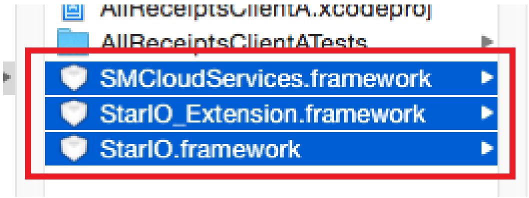


ptsClientA
SMJ_Dev4_iPhone6Plus01
Finished running AllReceiptsClientA on SMJ_Dev4_iPhone6Plus01
AllReceiptsClientA
General
Capabilities
Info
Build Settings
Build Phases
Bu
PROJECT
AllReceiptsClientA
TARGETS
AllReceiptsClientA
AllReceiptsClientATests
+
Target Dependencies (0 items)
Compile Sources (5 items)
Link Binary With Libraries (4 items)
Name
SMCloudServices.framework
StarIO_Extension.framework
StarIO.framework
CoreBluetooth.framework
+ —
Drag to reorder frameworks
Copy Bundle Resources (4 items)
SMCloudServicesResources.bundle ...in SMCloudServices.framework/Versions/A/Resources
Main.storyboard
LaunchScreen.xib
Images.xcassets ...in AllReceiptsClientA
+ —

3. Click the Add Other… button.



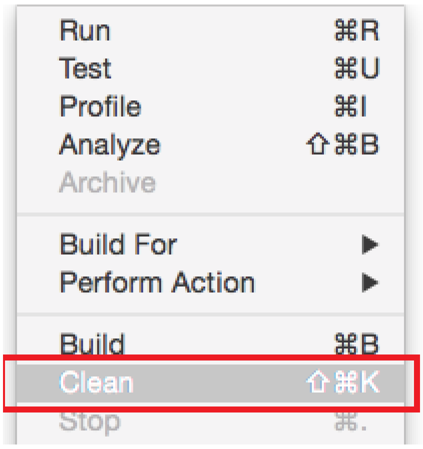
4. Browse to the location of where the Star AllReceipts iOS SDK was unzipped and select SMCloudServices.framework, StarIO.framework and StarIO_Extension.framework. Then click Open.


AllReceiptsClientA.xcodeproj
AllReceiptsClientATests
SMCloudServices.framework
StarIO_Extension.framework
StarIO.framework

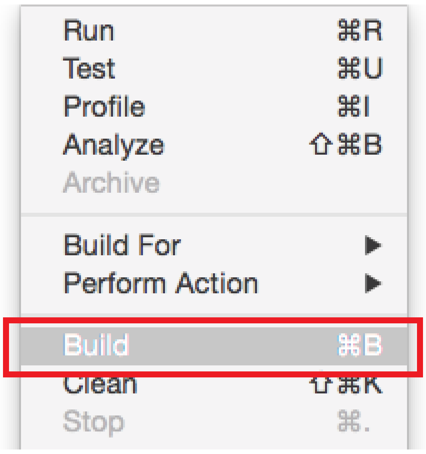
5. The framework is added to the project.

★Version up of SMCloudServices.framework, StarIO.framework and StarIO_Extension.framework

1. Delete SMCloudServices.framework, StarIO.framework and StarIO_Extension.framework from your project.
2. Copy new SMCloudServices.framework, StarIO.framework and StarIO_Extension.framework
3. Clean the Xcode project.



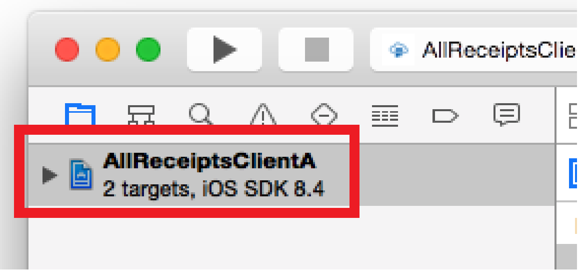
4. Build the Xcode project.



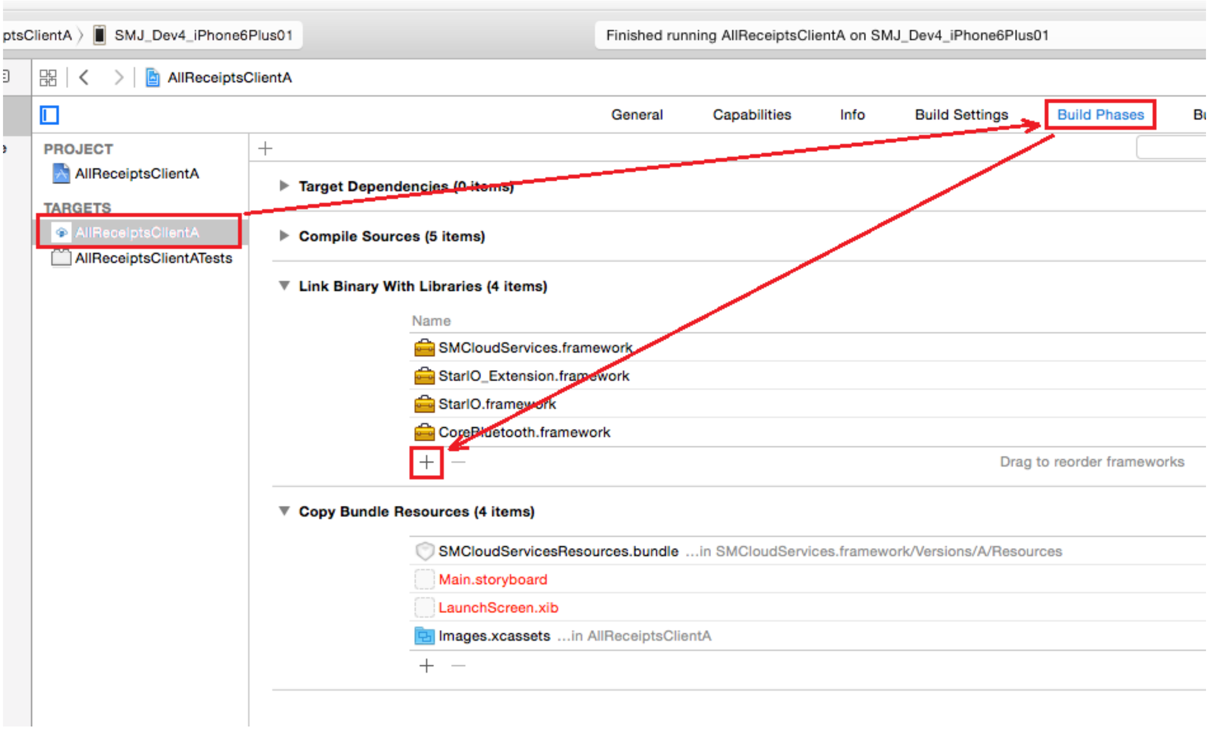
⚠ **To refer to the new SMCloudServices.framework, StarIO.framework and StarIO_Extension.framework without deleting the existing SMCloudServices.framework, StarIO.framework and StarIO_Extension.framework, surely confirm the "framework search path" setting of the Xcode project.**
If the old path of the SMCloudServices.framework, StarIO.framework and StarIO_Extension.framework remains in front of the "framework serach path", the previous SMCloudServices.framework, StarIO.framework and StarIO_Extension.framework will be used.

2.2 Add CoreBluetooth.framework

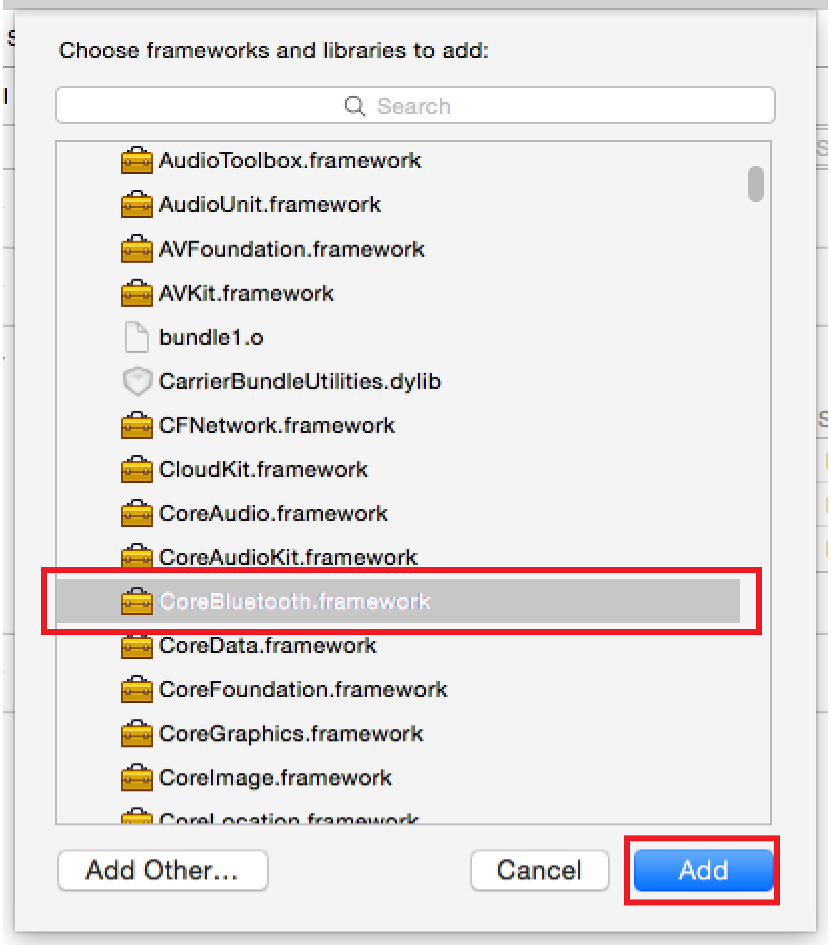
1. Click on the project.



2. Open a target. Click the Build Phases tab and click the +.


ptsClientA 〉 SMJ_Dev4_iPhone6Plus01
Finished running AllReceiptsClientA on SMJ_Dev4_iPhone6Plus01
AllReceiptsClientA
General
Capabilities
Info
Build Settings
Build Phases
PROJECT
AllReceiptsClientA
TARGETS
AllReceiptsClientA
AllReceiptsClientATests
Target Dependencies (0 items)
Compile Sources (5 items)
Link Binary With Libraries (4 items)
Name
SMCloudServices.framework
StarIO_Extension.framework
StarIO.framework
CoreBluetooth.framework
Drag to reorder frameworks
Copy Bundle Resources (4 items)
SMCloudServicesResources.bundle ...in SMCloudServices.framework/Versions/A/Resources
Main.storyboard
LaunchScreen.xib
Images.xcassets ...in AllReceiptsClientA

3. Select CoreBluetooth.framework and click the Add button.

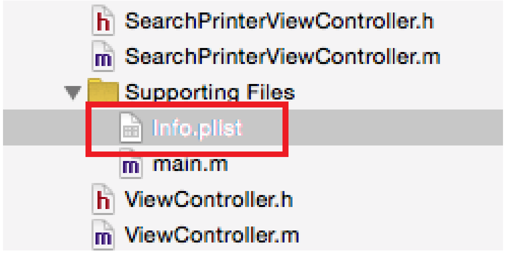


Choose frameworks and libraries to add:

4. The framework is added to the project.

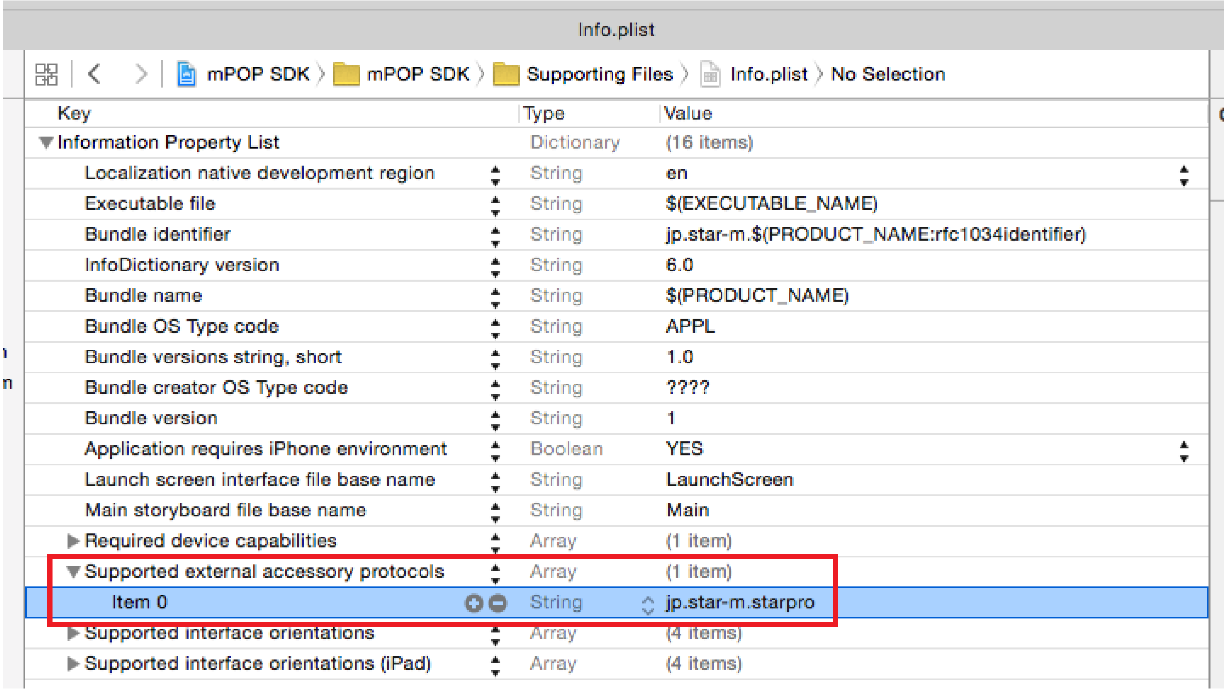
2.3 Edit information property list

1. Click on the information property list file.



SearchPrinterViewController.h
SearchPrinterViewController.m
Supporting Files
Info.plist
main.m
ViewController.h
ViewController.m

2. Add the Supported external accessory protocols key. Then click the triangle of this key and set the value for the Item * to jp.star-m.starpro.



Info.plist

mPOP SDK 〉 mPOP SDK 〉 Supporting Files 〉 Info.plist 〉 No Selection

Key	Type	Value
▼ Information Property List	Dictionary	(16 items)
Localization native development region	String	en
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	jp.star-m.$(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 external accessory protocols	Array	(1 item)
Item 0	String	jp.star-m.starpro
▶ Supported interface orientations	Array	(4 items)
▶ Supported interface orientations (iPad)	Array	(4 items)

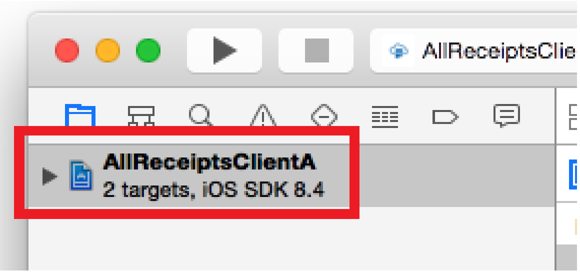
3. You have finished editing the information property list.

3 How to use AllReceipts with RegistrationView

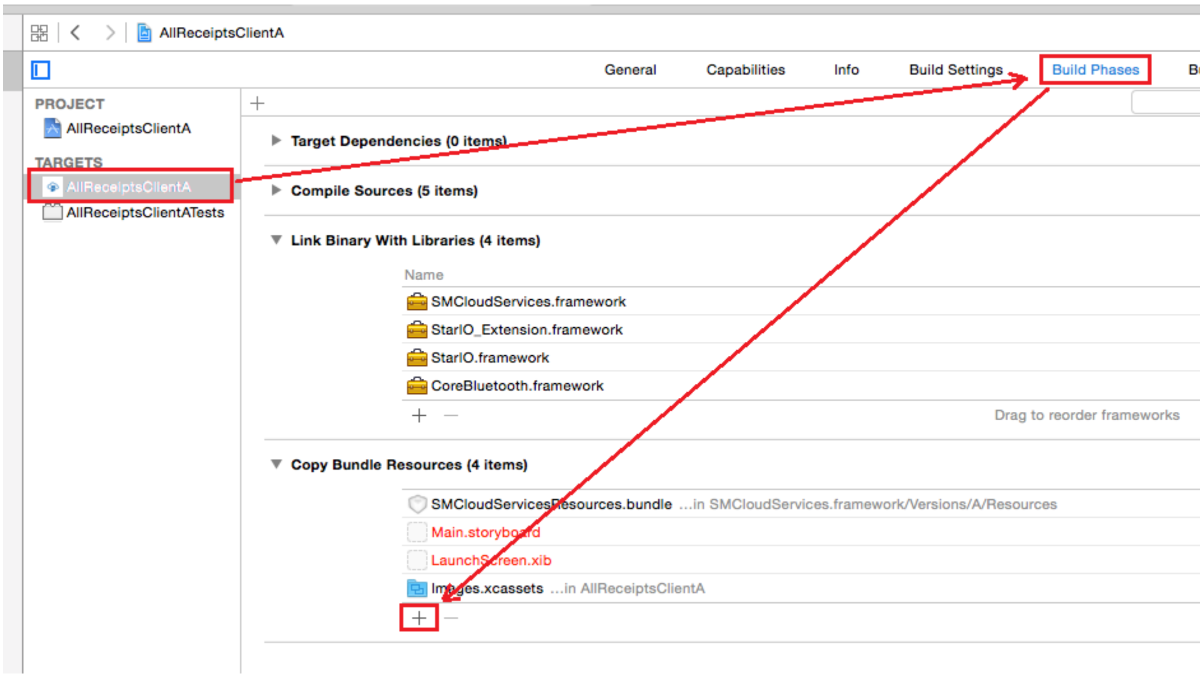
To use the AllReceipts with RegistrationView the application under development needs a user interface to show RegistrationView for device registration to the Star Micronics Cloud Services. The Star AllReceipts iOS SDK project includes the settings to use AllReceipts with RegistrationView; so you can directly test the SDK as it is. However, when you are building your own application, it is necessary to add settings to the project for AllReceipts with RegistrationView.

3.1 Append SMCloudServicesResources.bundle

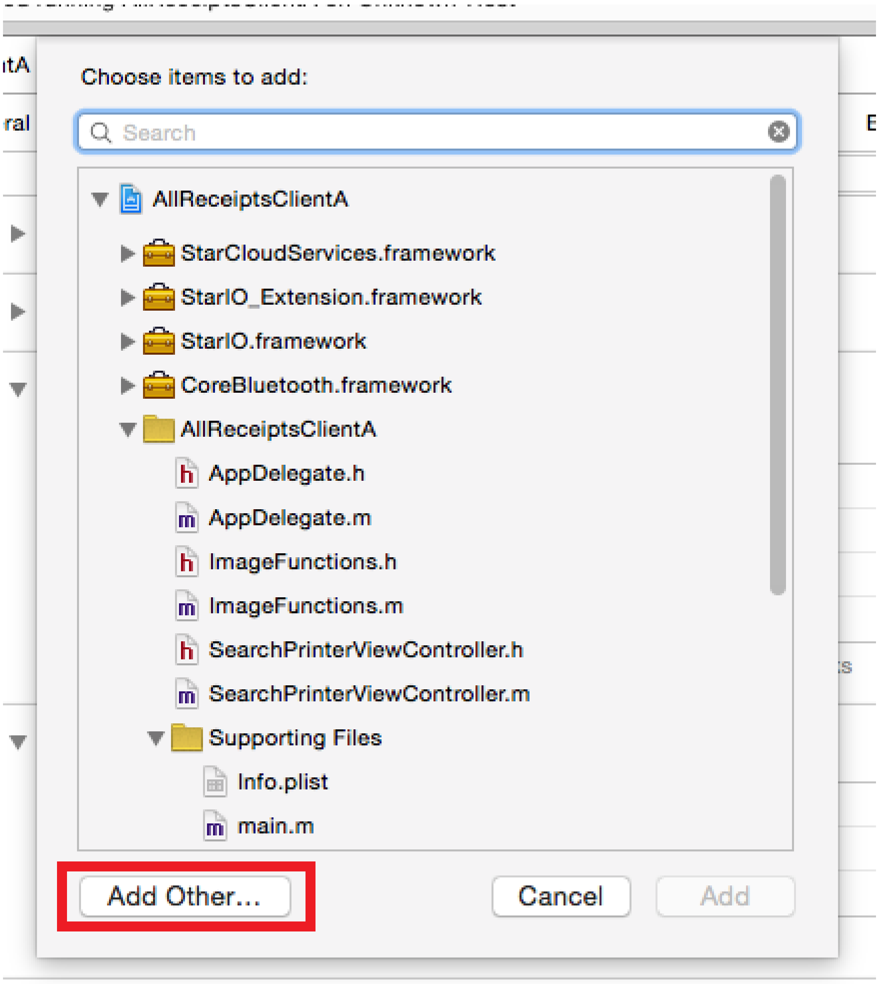
1. Click on the project.



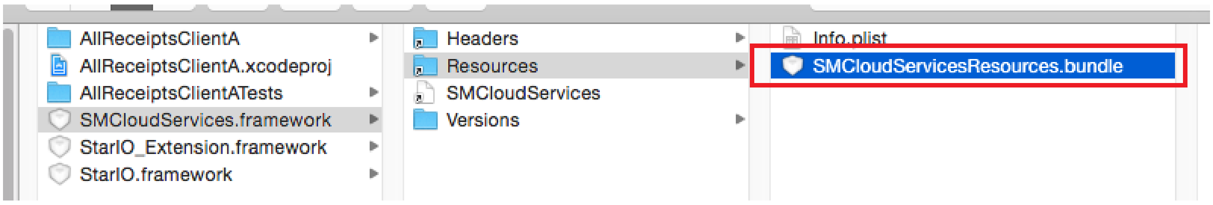
2. Open a target. Click the Build Phases tab and click the + (**Copy Bundle Resources**).


AllReceiptsClientA
General
Capabilities
Info
Build Settings
Build Phases
B
PROJECT
AllReceiptsClientA
TARGETS
AllReceiptsClientA
AllReceiptsClientATests
Target Dependencies (0 items)
Compile Sources (5 items)
Link Binary With Libraries (4 items)
Name
SMCloudServices.framework
StarIO_Extension.framework
StarIO.framework
CoreBluetooth.framework
Drag to reorder frameworks
Copy Bundle Resources (4 items)
SMCloudServicesResources.bundle ...in SMCloudServices.framework/Versions/A/Resources
Main.storyboard
LaunchScreen.xib
Images.xcassets ...in AllReceiptsClientA

3. Click the Add Other… button.



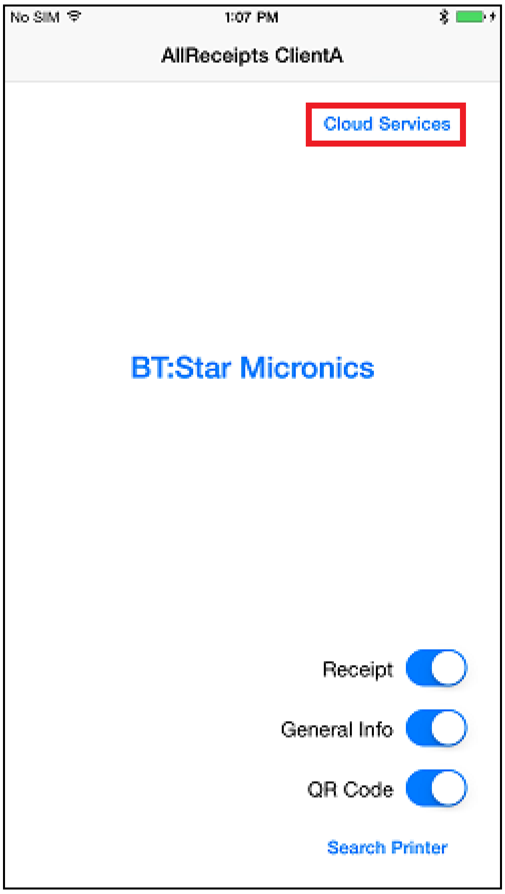
4. Browse to the location of where the Star AllReceipts iOS SDK was unzipped and select SMCloudServicesResources.bundle. Then click Open.


AllReceiptsClientA
AllReceiptsClientA.xcodeproj
AllReceiptsClientATests
SMCloudServices.framework
StarIO_Extension.framework
StarIO.framework
Headers
Resources
SMCloudServices
Versions
Info.plist
SMCloudServicesResources.bundle

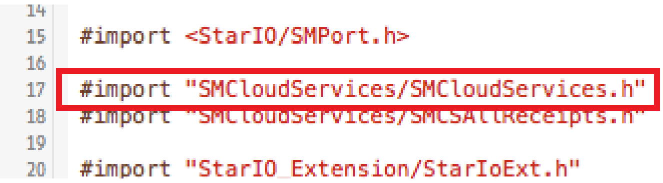
5. The bundle is added to the project.

3.2 Arrangement of the user interface to indicate View

1. For example arrange UIButton.



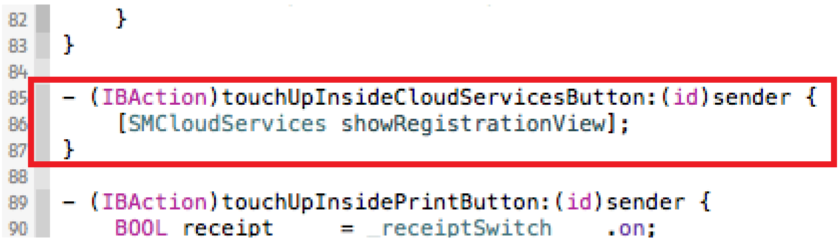
2. Import the SMCloudServicess.h.



```
#import <StarIO/SMPort.h>

#import "SMCloudServices/SMCloudServices.h"
#import "SMCloudServices/SMCSAllReceipts.h"

#import "StarIO_Extension/StarIoExt.h"
```

#import "SMCloudServices/SMCloudServices.h"

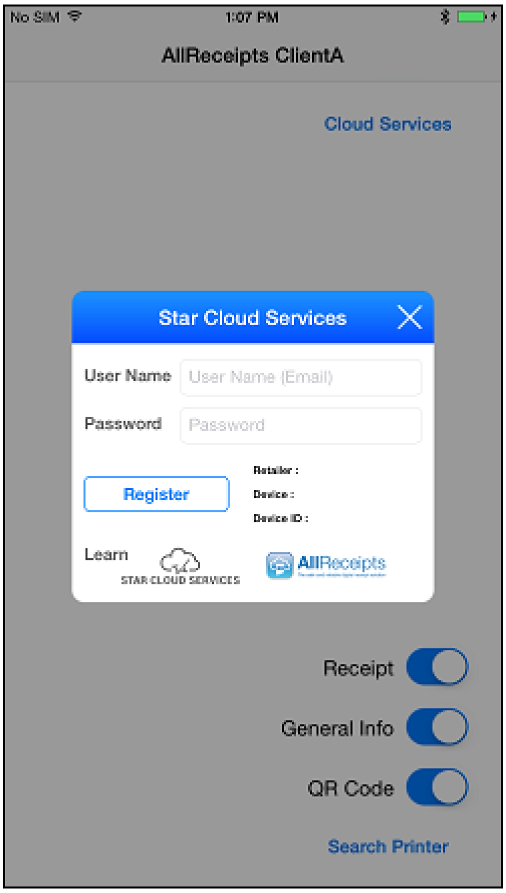
3. Paste the following code:



```
    }
}

- (IBAction)touchUpInsideCloudServicesButton:(id)sender {
    [SMCloudServices showRegistrationView];
}

- (IBAction)touchUpInsidePrintButton:(id)sender {
    BOOL receipt      = _receiptSwitch     .on;
```

```
- (IBAction)touchUpInsideCloudServicesButton:(id)sender {
    [SMCloudServices showRegistrationView];
}
```

4. Tap UIButton to show RegistrationView.



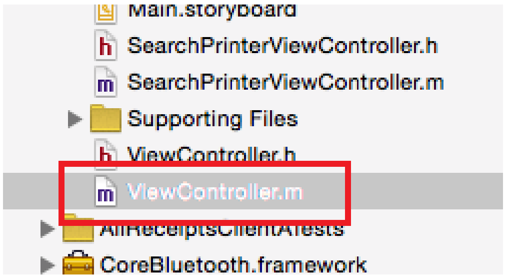
It became possible to register the device with RegistrationView.
Please refer to Star Micronics Cloud Services for more information.

4 How to upload AllReceipts

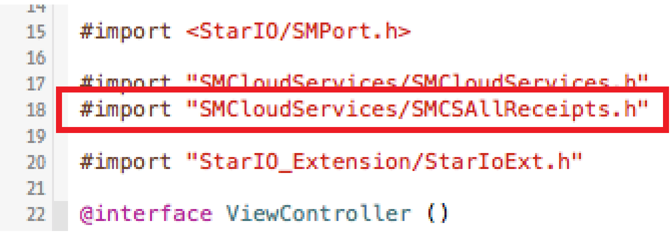
4.1 Edit about upload to Star Micronics Cloud Services

Below is a reference example of the Star AllReceipts iOS SDK.

1. Click on the ViewController.m.



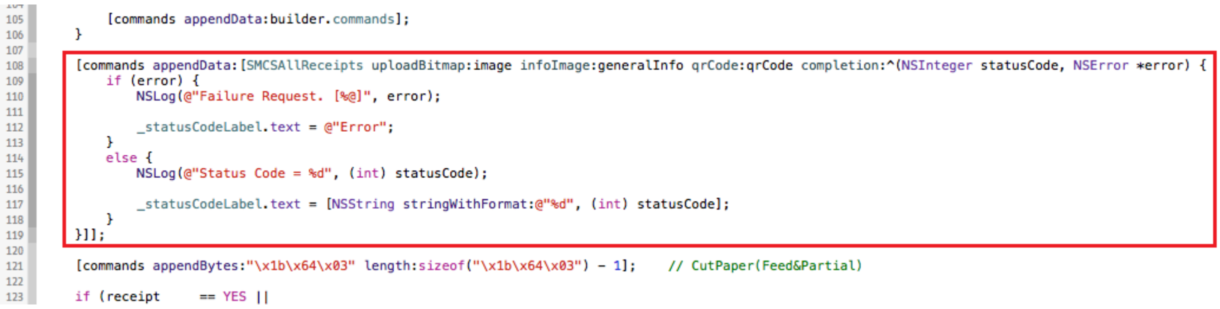
2. Import the SMCloudServicess.h.



```
#import <StarIO/SMPort.h>

#import "SMCloudServices/SMCloudServices.h"
#import "SMCloudServices/SMCSAllReceipts.h"

#import "StarIO_Extension/StarIoExt.h"

@interface ViewController ()
```

#import "SMCloudServices/SMCSAllReceipts.h"

3. Upload of generated the Bitmap(UIImage object).



```
        [commands appendData:builder.commands];
    }

    [commands appendData:[SMCSAllReceipts uploadBitmap:image infoImage:generalInfo qrCode:qrCode completion:^(NSInteger statusCode, NSError *error) {
        if (error) {
            NSLog(@"Failure Request. [%@]", error);

            _statusCodeLabel.text = @"Error";
        }
        else {
            NSLog(@"Status Code = %d", (int) statusCode);

            _statusCodeLabel.text = [NSString stringWithFormat:@"%d", (int) statusCode];
        }
    }]];

    [commands appendBytes:"\x1b\x64\x03" length:sizeof("\x1b\x64\x03") - 1];    // CutPaper(Feed&Partial)

    if (receipt     == YES ||
```

Example

```
- (IBAction)touchUpInsidePrintButton:(id)sender {
    _statusCodeLabel.text = @"";

    UIImage *image = [ImageFunctions createImageEnglish3inchRasterModeReceipt];     // English!!!

    NSMutableData *commands = [[NSMutableData alloc] init];

    if (receipt == YES) {
        ISCBBuilder *builder = [StarIoExt createCommandBuilder:SCBFactoryEmulationStar];

        [builder appendBitmap:image diffusion:NO width:384 bothScale:NO];     // 2inch(384dots)

        [commands appendData:builder.commands];
    }

    [commands appendData:[SMCSAllReceipts uploadBitmap:image infoImage:YES qrCode:YES
completion:^(NSInteger statusCode, NSError *error) {
        if (error) {
            NSLog(@"Failure Request. [%@]", error);

            _statusCodeLabel.text = @"Error";
        }
        else {
            NSLog(@"Status Code = %d", (int) statusCode);

            _statusCodeLabel.text = [NSString stringWithFormat:@"%d", (int) statusCode];
        }
    }]];

    [commands appendBytes:"\x1b\x64\x03" length:sizeof("\x1b\x64\x03") - 1];    // CutPaper(Feed&Partial)

    self.blind = YES;

    [ViewController sendCommands:commands portName:[AppDelegate getPortName] portSettings:[AppDelegate
getPortSettings] timeout:10000];

    self.blind = NO;
}
```

5 SMCSAllReceipts

Method

Name	Description
uploadBitmap	Upload bitmap (UIImage object) to the Star Micronics Cloud Services.

5.1 uploadBitmap Method

Upload bitmap(UIImage object) to the Star Micronics Cloud Services.

Declaration

+ (NSData *)uploadBitmap:(UIImage *)image infoImage:(BOOL)infoImage qrCode:(BOOL)qrCode completion:(void (^)(NSInteger statusCode, NSError *error))completion;

Parameter

Name	Description	Object type
image	Bitmap (UIImage object).	UIImage
infoImage	Append the AllReceipts general information to the return value. • YES … Need • NO … No need	BOOL
qrCode	Append the QR Code to the return value. • YES … Need • NO … No need	BOOL
completion	A block that is called after the uploaded.	void (^)(NSInteger statusCode, NSError *error)

Return value

Description	Object type
Printed information by the Star Emulation Command about the upload.	NSData

Example

```
- (IBAction)touchUpInsidePrintButton:(id)sender {
    _statusCodeLabel.text = @"";

    UIImage *image = [ImageFunctions createImageEnglish3inchRasterModeReceipt];      // English!!!

    NSMutableData *commands = [[NSMutableData alloc] init];

    if (receipt == YES) {
        ISCBBuilder *builder = [StarIoExt createCommandBuilder:SCBFactoryEmulationStar];

        [builder appendBitmap:image diffusion:NO width:384 bothScale:NO];      // 2inch(384dots)

        [commands appendData:builder.commands];
    }

    [commands appendData:[SMCSAllReceipts uploadBitmap:image infoImage:YES qrCode:YES
completion:^(NSInteger statusCode, NSError *error) {
        if (error) {
            NSLog(@"Failure Request. [%@]", error);

            _statusCodeLabel.text = @"Error";
        }
        else {
            NSLog(@"Status Code = %d", (int) statusCode);

            _statusCodeLabel.text = [NSString stringWithFormat:@"%d", (int) statusCode];
        }
    }]];

    [commands appendBytes:"\x1b\x64\x03" length:sizeof("\x1b\x64\x03") - 1];    // CutPaper(Feed&Partial)

    self.blind = YES;

    [ViewController sendCommands:commands portName:[AppDelegate getPortName] portSettings:[AppDelegate
getPortSettings] timeout:10000];

    self.blind = NO;
}
```

★How to convert the UIImage object to the Star Emulation Command and how to communicate with the printer:

Please download StarIO iOS SDK and Star mPOP iOS SDK, and refer to the document.