



BXL SDK for iOS_UPOS Compliant API 레퍼런스 가이드

Rev. 1.08

**SPP-R210
SPP-R220
SPP-R310
SPP-R410
SPP-R200III
SRP-350plusIII
SRP-352plusIII
SRP-350III/352III
SRP-275III
SRP-F310II
SRP-F312II
SRP-F313II
SRP-380/382
SRP-330II/332II
SRP-340II/342II
SRP-S300
SRP-Q300/Q302**

■ 목차

1. 매뉴얼 안내.....	6
2. 사용 환경	7
2-1 운영체제	7
2-2 지원 장치 및 인터페이스.....	7
3. 개발 환경	8
3-1 개발 환경 설정.....	8
3-2 iOS 장치 연결.....	8
3-2-1 Bluetooth	8
3-2-2 Network–Infra structure Mode	9
3-2-3 Network–Ad Hoc Mode	10
4. 패키지콘텐츠	12
5. 상수값 (Defines)	13
5-1 Result Code.....	13
5-2 OpenResult Code	13
5-3 State Code.....	14
5-4 Transaction Print.....	14
5-5 Alignment.....	14
5-6 Barcode Type	15
5-7 Barcode Text Position	16
5-8 StatusUpdateEvent	16
6. 각 클래스별 기능 정리	17
6-1 본SDK 에서 지원하는 Class 분류	17
6-2 Support Table	18
6-2-1 Printer Method	18
6-2-2 CashDrawer Method	18
6-2-3 MSR Method.....	19
6-2-4 SCR Method	19
7. [Common] Device Class Reference	20
7-1 Overview	20
7-2 Available Properties	20
7-2-1 modelName	20
7-2-2 Idn (Logical Device Name)	20
7-2-3 InterfaceType.....	20
7-2-4 address.....	21
7-2-5 port	21
7-3 Available Method	21
8. [Common] Device List Class Reference.....	22
8-1 Overview	22
8-2 Available Properties	22
8-3 Available Method	23
8-3-1 getList.....	23
8-3-2 getDeviceIdentity	24
8-3-3 save	24
8-3-4 addDevice.....	25
8-3-5 removeDevice	26

9. [Common] Device Controller Class Reference	27
9-1 Overview	27
9-2 Available Properties	27
9-2-1 CheckHealthText	27
9-2-2 Claimed	27
9-2-3 DeviceEnabled.....	28
9-2-4 OpenResult.....	28
9-2-5 ResultCode	28
9-2-6 ResultCodeExtended	29
9-2-7 OutputID	29
9-2-8 State	29
9-3 Available Method	30
9-3-1 open.....	30
9-3-2 claim	31
9-3-3 releaseDevice	32
9-3-4 close	33
9-3-5 checkHealth	34
10. [Printer] Printer Class Reference	35
10-1 Overview	35
10-2 Available Properties	35
10-3 Available Method	35
11. [Printer] Printer List Class Reference	36
11-1 Overview.....	36
11-2 Available Properties	36
11-3 Available Method.....	37
11-3-1 getList	37
11-3-2 getDeviceIdentity	38
11-3-3 save	39
11-3-4 addDevice	40
11-3-5 removeDevice	41
12. [Printer] Printer Controller Class Reference	42
12-1 Overview	42
12-2 Properties	43
12-2-1 Capability 설정 값	43
12-2-2 프로퍼티 기본 설정 값 및 설정 범위	44
12-2-3 RecLineChars	45
12-2-4 RecLineCharsList.....	46
12-2-5 RecLineSpacing.....	46
12-2-6 RecLineWidth	46
12-2-7 RecEmpty	46
12-2-8 RecNearEnd	47
12-2-9 AsyncMode	47
12-2-10 CharacterSet.....	47
12-2-11 CharacterSetList	47
12-2-12 CoverOpen	48
12-2-13 ErrorLevel	48
12-2-14 ErrorString	48
12-2-15 FlagWhenIdle.....	48
12-3 Available Method	49
12-3-1 open.....	49
12-3-2 claim	50
12-3-3 releaseDevice	51
12-3-4 close	52
12-3-5 cutPaper	53

12-3-6 markFeed.....	54
12-3-7 printBarcode	55
12-3-8 printBitmap (파일 인쇄)	57
12-3-9 printBitmap (UIImage 인쇄)	59
12-3-10 printNormal	61
12-3-11 transactionPrint	62
12-4 Available Delegate	64
12-4-1 StatusUpdateEvent	64
12-4-2 OutputCompleteEvent.....	66
13. [CashDrawer] Cash Drawer Class Reference.....	67
13-1 Overview	67
13-2 Available Properties	67
13-2-1 selectedPrinterName	67
13-2-2 pinNumber	67
13-2-3 pinLevel	67
13-2-4 pulseOnTime	68
13-2-5 pulseOffTime	68
13-3 Available Method	68
14. [CashDrawer] Cash Drawer List Class Reference	69
14-1 Overview	69
14-2 Available Properties	69
14-3 Available Method	70
14-3-1 getList	70
14-3-2 getDeviceIdentity	71
14-3-3 save	72
14-3-4 addDevice	73
14-3-5 removeDevice	74
15. [CashDrawer] Cash Drawer Controller Class Reference.....	75
15-1 Overview	75
15-2 Available Properties	75
15-2-1 DrawerOpened	75
15-3 Available Method	76
15-3-1 open.....	76
15-3-2 claim	77
15-3-3 releaseDevice	78
15-3-4 close	79
15-3-5 OpenDrawer	80
15-4 Available Delegate	81
15-4-1 StatusUpdateEvent	81
16. [MSR] MSR Class Reference	83
16-1 Overview	83
16-2 Available Properties	83
16-3 Available Method	83
17. [MSR] MSR List Class Reference (저장된 MSR 목록).....	84
17-1 Overview	84
17-2 Available Properties	84
17-3 Available Method	85
17-3-1 getList	85
17-3-2 getDeviceIdentity	86
17-3-3 save	87
17-3-4 addDevice	88
17-3-5 removeDevice	89

18. [MSR] MSR Controller Class Reference	90
18-1 Overview	90
18-2 Available Properties	90
18-2-1 Track1Data	90
18-2-2 Track2Data	90
18-2-3 Track3Data	90
18-3 Available Method	91
18-3-1 open	91
18-3-2 claim	92
18-3-3 releaseDevice	93
18-3-4 close	94
18-4 Available Delegate	95
18-4-1 DataEvent	95
19. [SCR] SCR Class Reference.....	96
19-1 Overview	96
19-2 Available Properties	96
19-3 Available Method	96
20. [SCR] SCR List Class Reference(저장된 SCR 목록)	97
20-1 Overview	97
20-2 Available Properties	97
20-3 Available Method	98
20-3-1 getList	98
20-3-2 getDeviceIdentity	99
20-3-3 save	100
20-3-4 addDevice	101
20-3-5 removeDevice	102
21. [SCR] SCR Controller Class Reference	103
21-1 Overview	103
21-2 Available Properties	103
21-3 Available Method	104
21-3-1 open	104
21-3-2 claim	105
21-3-3 releaseDevice	106
21-3-4 close	107
21-4 Available Delegate	108
21-4-1 DataEvent	108
22. [Common] Delegater Class Reference.....	109
22-1 Overview	109
22-2 Available Properties	109
22-3 Available Delegate	110
22-3-1 DataEvent	110
22-3-2 StatusUpdateEvent	110
22-3-3 OutputCompleteEvent.....	110
23. 샘플프로그램.....	111
23-1 프로젝트 설정	111
23-1-1 ExternalAccessory.framework 추가하기	111
23-1-2 Bluetooth Protocol 추가하기	112
23-2 nativeSample	113
23-3 phonegapSample.....	113
23-3-1 환경설정	113
23-3-2 PhoneGap을 사용하는 애플리케이션에서 BXL SDK for iOS_UPOS Compliant를 사용하기위한 설정..	113
24. 에러 정보.....	114
24-1 Error 리스트	114

1. 매뉴얼 안내

BXL SDK for iOS_UPOS Compliant는 UnifiedPOS 버전을 준수합니다.

BXL SDK for iOS_UPOS Compliant는 애플리케이션이 (주)빅솔론의 프린터에 쉽게 접근할 수 있도록 iOS framework을 제공합니다.

이 매뉴얼은 BXL SDK for iOS_UPOS Compliant의 사용방법, 사양, 제약에 대해 설명합니다. 또한 이 매뉴얼은 UPOS 장치를 사용해서 애플리케이션 시스템을 설계하는 개발자를 위해 제공됩니다.

(주)빅솔론의 프린터를 사용하기 전에 BXL SDK for iOS_UPOS Compliant에 포함된 UPOS Setup을 이용하여 장치를 설정해야 합니다.

[참고사이트]

<http://monroeecs.com/unifiedpos.htm>: UnifiedPOS 위원회

<http://www.bixolon.com>: SDK 업데이트

<http://developer.apple.com>: Apple 개발자 사이트

저희 (주)빅솔론은 제품의 기능과 품질 향상을 위하여 지속적인 개선을 하고 있습니다. 이로 인하여 제품의 사양과 매뉴얼의 내용은 사전 통보 없이 변경될 수 있습니다.

2. 사용 환경

2-1 운영체제

이 소프트웨어는 아래 운영체제를 지원합니다.

- iOS 6.0 이상

2-2 지원 장치 및 인터페이스

Models	Interface
SPP-R210	Bluetooth / WLAN
SPP-R220	Bluetooth / WLAN / BLE
SPP-R310	Bluetooth / WLAN
SPP-R410	Bluetooth / WLAN / BLE
SPP-R200III	Bluetooth / WLAN
SRP-350plusIII	Bluetooth / WLAN / Ethernet
SRP-352plusIII	Bluetooth / WLAN / Ethernet
SRP-350III	Ethernet
SRP-352III	Ethernet
SRP-F310II	Bluetooth / WLAN / Ethernet
SRP-F312II	Bluetooth / WLAN / Ethernet
SRP-F313II	Bluetooth / WLAN / Ethernet
SRP-275III	Ethernet
SRP-380	Bluetooth / WLAN / Ethernet
SRP-382	Bluetooth / WLAN / Ethernet
SRP-330II	Ethernet
SRP-332II	Ethernet
SRP-340II	Ethernet
SRP-342II	Ethernet
SRP-S300	Bluetooth / WLAN / Ethernet
SRP-Q300	Bluetooth / WLAN / Ethernet
SRP-Q302	Bluetooth / WLAN / Ethernet

※ BLE : Bluetooth Low Energy

3. 개발 환경

3-1 개발 환경 설정

- Xcode 6.0 이상
- iOS SDK
- 참고: <http://developer.apple.com/devcenter/ios/index.action>

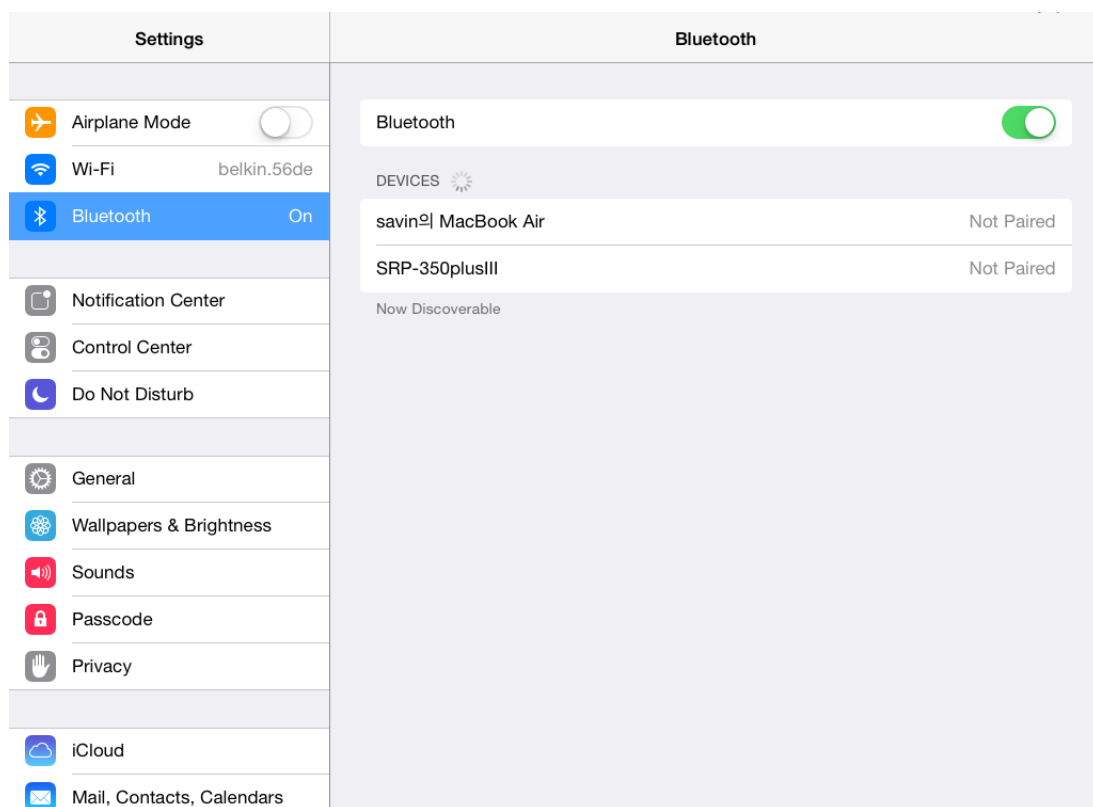
3-2 iOS 장치 연결

아래 스크린샷은 iPad mini 에서 캡처한것입니다.

스크린샷과 항목 이름은 iOS 운영체제 또는 장치에 따라 다를 수 있습니다.

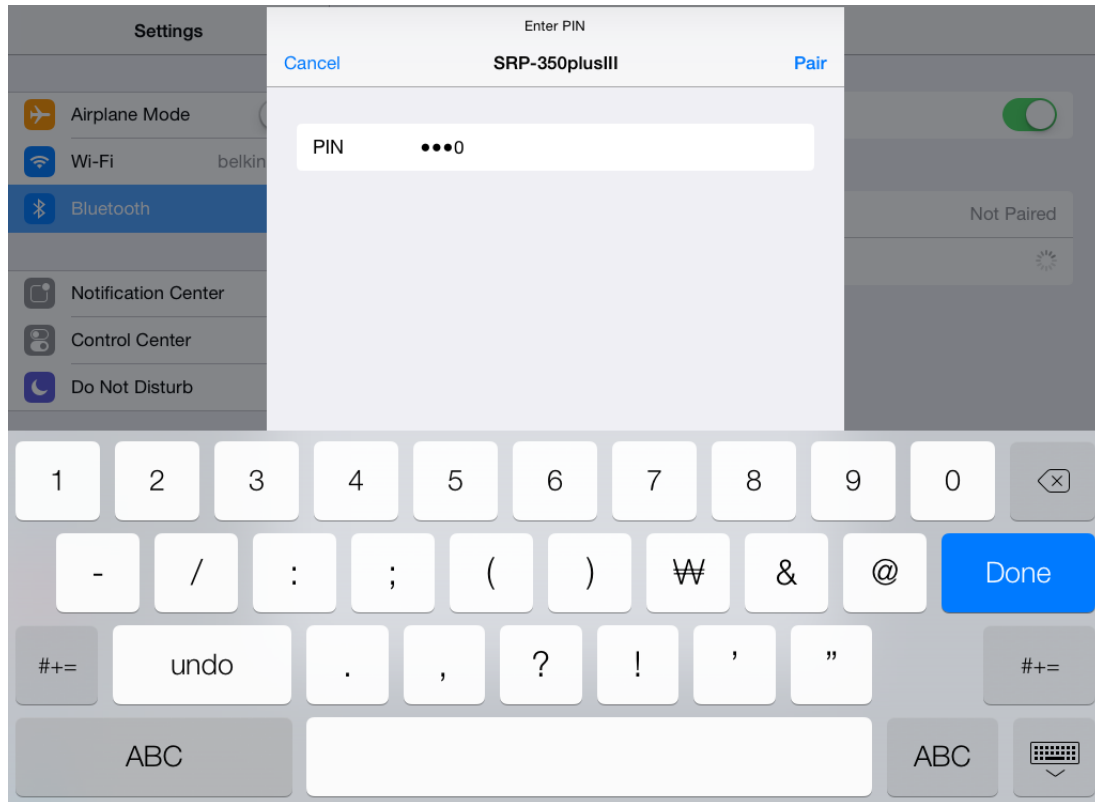
3-2-1 Bluetooth

1. [Settings]를 선택합니다.
2. 반드시 Bluetooth가 켜져있고, 프린터의 전원이 켜져있어야 합니다.
3. 설정을 위해 [Bluetooth]를 선택합니다.



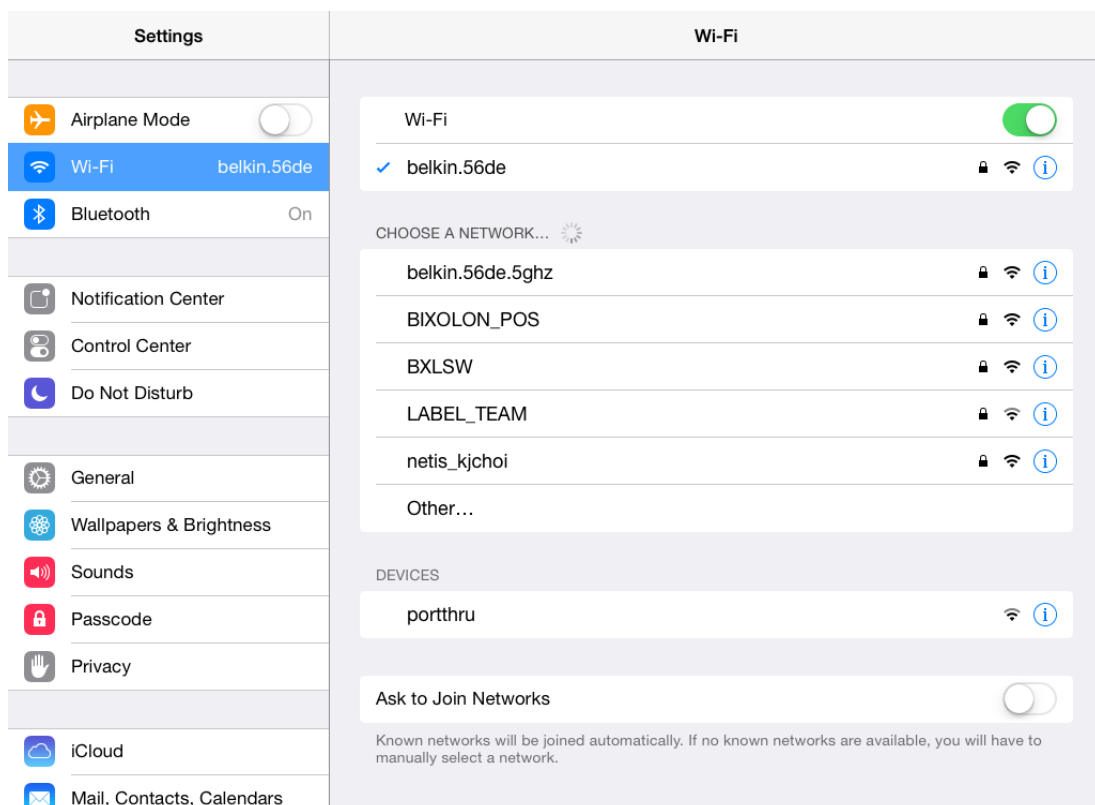
4. 연결하기 위한 프린터를 선택하고 페어링합니다.

5. PIN code를 입력합니다. (쥬빅솔론의 초기 PIN code는 “0000” 입니다.)



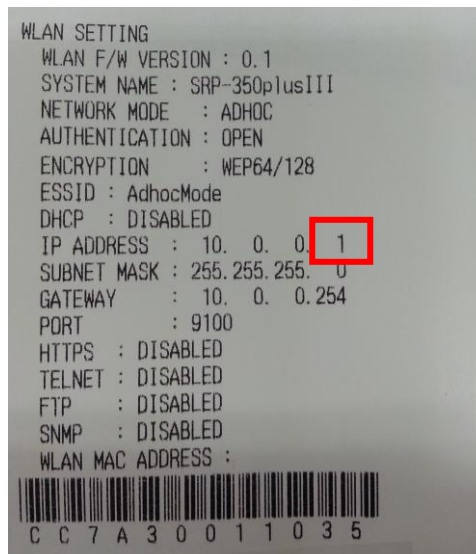
3-2-2 Network–Infra structure Mode


1. [Settings]를 선택합니다.
2. Wi-Fi가 켜져 있어야합니다.
3. (쥬빅솔론의 프린터가 연결되어있는 네트워크와 동일한 네트워크에 연결합니다.

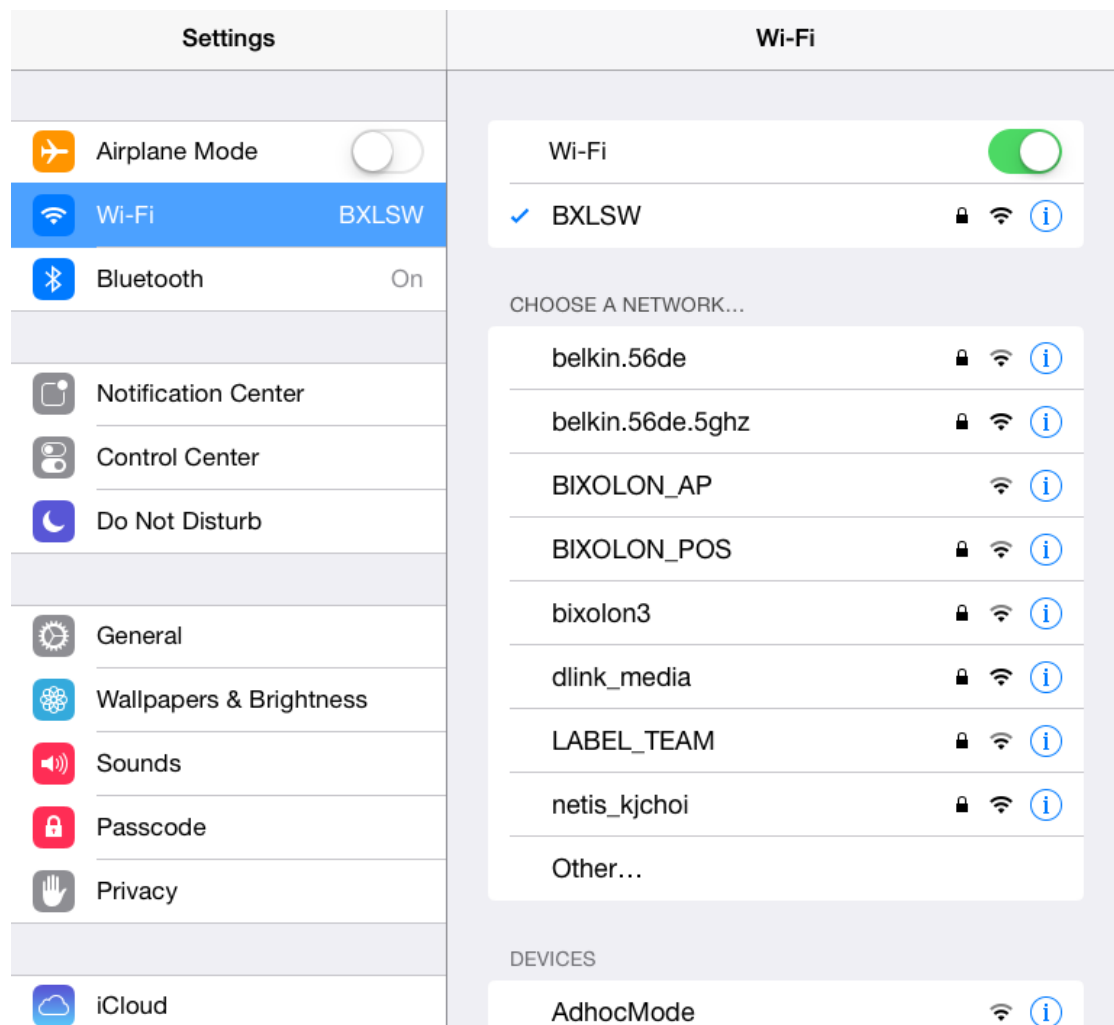


3-2-3 Network-Ad Hoc Mode

1. Printer의 “Network Mode” 가 “ADHOC”으로 표시 되는 경우, 별도의 아이피 설정이 필요합니다.

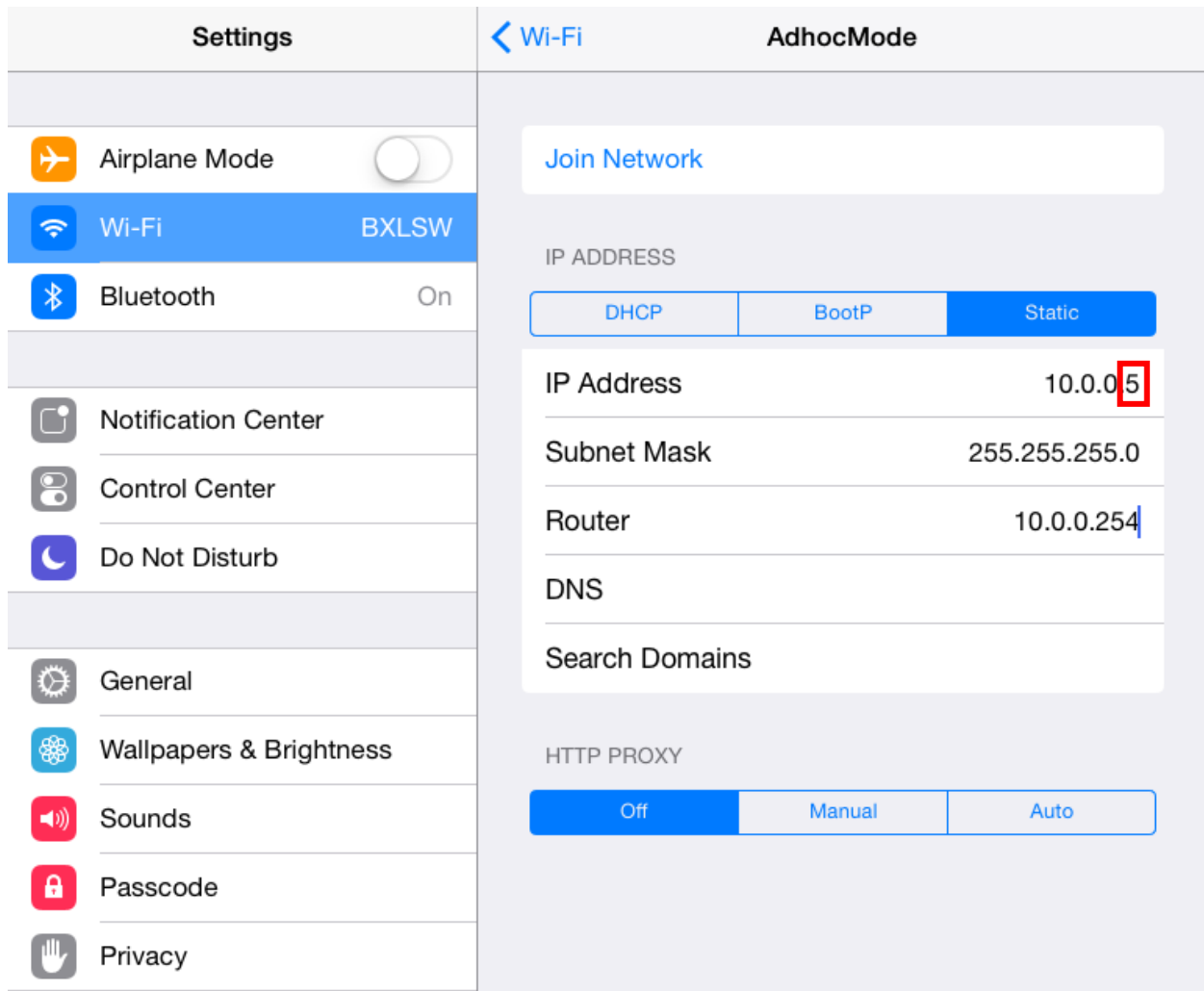


2. 프린터의 SSID 옆에 “”버튼을 누릅니다.



3. 아래와 같이 프린터의 IP를 할당 합니다.

※ 프린터의 IP가 “10.0.0.1” 인경우 앞의 4번째 클래스의 주소만 변경합니다.



4. 패키지콘텐츠

- 한글매뉴얼
Manual_BXL SDK for iOS_UPOS Compliant API Reference Guide_korean_
Rev_x_xx.pdf
- 영문매뉴얼
Manual_BXL SDK for iOS_UPOS Compliant API Reference Guide_english_
Rev_x_xx.pdf
- libs/lib/libBixelonUPOS.a: UPOS SDK – Library type
- libs/framework/frmBixelonUPOS: UPOS SDK – Framework type
- samples/nativeSample: iOS기반의 UPOS 샘플애플리케이션
- samples/phonegapSample: PhoneGap을사용한 Web App 샘플

5. 상수값 (Defines)

5-1 Result Code

- Method에서 특정 기능을 수행 후 반환하는 결과 값을 정의합니다.

Code DEFINE	Description
UPOS_SUCCESS	성공
UPOS_E_CLOSED	Close 된 장치에 접근하였습니다.
UPOS_E_CLAIMED	Claim 메소드 호출이 선행 되어야 합니다.
UPOS_E_NOTCLAIMED	Claim 상태가 아닙니다.
UPOS_E_NOSERVICE	제공하지 않는 기능입니다.
UPOS_E_DISABLED	Enable 상태가 아닙니다.
UPOS_E_ILLEGAL	잘못된 접근이거나 지원하지 않는 기능 입니다.
UPOS_E_NOHARDWARE	Device 가 연결되어있지 않습니다.
UPOS_E_OFFLINE	Device 가 Off-line 상태 입니다.
UPOS_E_NOEXIST	대상이 존재하지 않습니다.
UPOS_E_EXISTS	대상이 이미 존재 합니다.
UPOS_E_FAILURE	요청한 작업에 실패하였습니다.
UPOS_E_TIMEOUT	타임아웃 되었습니다.
UPOS_E_BUSY	먼저 요청한 기능을 수행 중입니다.
UPOS_E_EXTENDED	장치에 에러가 발생하였습니다. 자세한 내용은 ResultCode Extended 를 확인하십시오.
UPOS_E_DEPRECATED	현재는 사용되지 않는 기능입니다.

5-2 OpenResult Code

- Open 메소드에 수행후 발생하는 결과 값에 대한 정의 입니다.

Code DEFINE	Description
UPOS_SUCCESS	Open 성공
UPOS_OR_ALREADYOPEN	이미 Open 상태 입니다.
UPOS_OR_REGBADNAME	저장된 장치 목록에서 지정한 이름을 찾을 수 없습니다.
UPOS_OR_FAILEDOPEN	Open 메소드 호출 후 실패코드를 리턴했으나, 자세한 원인을 알 수 없습니다.

5-3 State Code

- State 프로퍼티에 저장 될 수 있는 값에 대한 정의 입니다.

Code DEFINE	Description
UPOS_S_CLOSED	Closed 된 상태 입니다.
UPOS_S_IDLE	Error 가 없고 대기 상태 입니다.
UPOS_S_BUSY	다른 기능을 수행 중 입니다.
UPOS_S_ERROR	Error 가 발생되었습니다.

5-4 Transaction Print

- Transaction Mode 설정 시 사용 할 수 있는 값에 대한 정의 입니다.

Code DEFINE	Description
PTR_TR_TRANSACTION	Buffer를 비어있는 상태로 초기화 하고 Transaction Mode 를 시작합니다.
PTR_TR_NORMAL	Transaction mode 를 종료하고, Buffer에 쌓여 있는 Data 를 출력합니다.

5-5 Alignment

- 정렬을 지정할 때 필요한 값에 대한 정의 입니다.
(For Barcodes)

Code DEFINE	Description
PTR_BC_LEFT	좌측 정렬
PTR_BC_CENTER	중앙 정렬
PTR_BC_RIGHT	우측 정렬

(For Images)

Code DEFINE	Description
PTR_BM_LEFT	좌측 정렬
PTR_BM_CENTER	중앙 정렬
PTR_BM_RIGHT	우측 정렬

5-6 Barcode Type

- 바코드 출력 시 바코드 지정에 필요한 값에 대한 정의입니다.

Code DEFINE	Description								
PTR_BCS_UPCA	UPCA								
PTR_BCS_UPCE	UPCE								
PTR_BCS_JAN8	JAN8								
PTR_BCS_EAN8	EAN8								
PTR_BCS_JAN13	JAN13								
PTR_BCS_EAN13	EAN13								
PTR_BCS_TF	Standard(ordiscrete) 2 of 5								
PTR_BCS_ITF	Interleaved 2 of 5								
PTR_BCS_Codabar	Codabar								
PTR_BCS_Code39	Code39								
PTR_BCS_Code93	Code93								
PTR_BCS_Code128	Code 128 ※ Code128 의 Special Character <table> <tr> <th>Special Charaters</th><th>Ascii Represntation</th></tr> <tr> <td>Code A</td><td>{A</td></tr> <tr> <td>Code B</td><td>{B</td></tr> <tr> <td>Code C</td><td>{C</td></tr> </table>	Special Charaters	Ascii Represntation	Code A	{A	Code B	{B	Code C	{C
Special Charaters	Ascii Represntation								
Code A	{A								
Code B	{B								
Code C	{C								
PTR_BCS_UPCA_S	UPC-A with supplemental barocde								
PTR_BCS_UPCE_S	UPC-E with supplemental barcode								
PTR_BCS_UPCD1	UPC-D1								
PTR_BCS_UPCD2	UPC-D2								
PTR_BCS_UPCD3	UPC-D3								
PTR_BCS_UPCD4	UPC-D4								
PTR_BCS_UPCD5	UPC-D5								
PTR_BCS_EAN8_S	EAN8 with supplemental barocde								
PTR_BCS_EAN13_S	EAN13 with supplemental barocde								
PTR_BCS_EAN128	EAN128								
PTR_BCS_OCRA	OCR "A"								
PTR_BCS_OCRB	OCR "B"								
PTR_BCS_Code128_Parsed	Code 128 with parsing								
PTR_BCS_GS1DATABAR	GS1 DataBar Omnidirectional								
PTR_BCS_GS1DATABAR_E	GS1 DataBar Stacked Omnidirectional								
PTR_BCS_GS1DATABAR_S	GS1 DataBar Expanded								
PTR_BCS_GS1DATABAR_E_S	GS1 DataBar Expanded Stacked								
PTR_BCS_PDF417	PDF 417								
PTR_BCS_MAXICODE	MAXI Code								
PTR_BCS_DATAMATRIX	Data Matrix								
PTR_BCS_QRCODE	QR Code								
PTR_BCS_UQRCODE	Micro QR Code								
PTR_BCS_AZTEC	Aztec								
PTR_BCS_UPDF417	Micro PDF 417								

5-7 Barcode Text Position

- Text 출력을 지원하는 바코드 일 경우, 바코드의 출력여부 또는 출력 위치를 지정합니다.

Code DEFINE	Description
PTR_BC_TEXT_NONE	Text 를 출력하지 않습니다. 바코드만 출력 합니다.
PTR_BC_TEXT_ABOVE	Text 를 바코드의 상단에 출력 합니다.
PTR_BC_TEXT_BELOW	Text 를 바코드의 하단에 출력합니다.

5-8 StatusUpdateEvent

- StatusUpdateEvent Delegate 발생 시 Parameter로 사용 될 수 있는 값 입니다.

Code DEFINE	Description
PTR_SUE_IDLE	Idle 상태 입니다.
UPOS_SUE_POWER_ONLINE	online 상태 입니다.
UPOS_SUE_POWER_OFF	Off Line 상태 입니다.
UPOS_SUE_POWER_OFFLINE	본 SDK 에서는 별도의 코멘트가 없는 경우 모두 같은 동작으로 처리 됩니다.
UPOS_SUE_POWER_OFF_OFFLINE	
PTR_SUE_COVER_OPEN	프린터의 커버가 열려있습니다.
PTR_SUE_COVER_OK	프린터의 커버가 닫혀있습니다.
PTR_SUE_REC_EMPTY	프린터의 용지가 비어있습니다.
PTR_SUE_REC_NEAREMPTY	프린터의 용지가 비어있는 상태에 가깝습니다.
PTR_SUE_REC_PAPEROK	프린터의 용지가 충분합니다.

6. 각 클래스별 기능 정리

6-1 본SDK 에서 지원하는 Class 분류

Class Type	설명	관련 Class
(1) 단일 장치	각 장치 별 정보를 가지고 있음.	<ul style="list-style-type: none"> - Device Class Reference (Common Device) - [Printer] Printer Class Reference - [CashDrawer] Cash Drawer Class Reference - [MSR] MSR Class Reference
(2) 장치List 관리	“(1) 단일 장치”를 배열 형태로 저장 할 수 있음.	<ul style="list-style-type: none"> - [Common] Device Controller Class Reference - [Printer] Printer List Class Reference - [CashDrawer] Cash Drawer List Class Reference - [MSR] MSR List Class Reference (저장된 MSR 목록)
(3) 장치 컨트롤러	<p>iDevice 에 “(2) 장치List 관리” 형태로 저장되어있는 장치들중 하나를 컨트롤 할 수 있음.</p> <p>Ex) 프린터의 경우 Text 인쇄, 이미지/바코드 인쇄.</p>	<ul style="list-style-type: none"> - Device Controller Class Reference (Common Controller) - [Printer] Printer Controller Class Reference - [CashDrawer] Cash Drawer Controller Class Reference - [MSR] MSR Controller Class Reference
(4) Delegate	각 장치에서 Application Level 로 이벤트를 줄 수 있는데, 이를 이용하려면 이 클래스를 이용해야한다.	<ul style="list-style-type: none"> - [Common] Delegater Class Reference

6-2 Support Table

각 Device Controller 별 지원하는 Methods List 입니다.

6-2-1 Printer Method

Method	Value	선행 작업
open	O	-
claim	O	open
releaseDevice	O	open – claim
close	O	open
beginInsertion	X	open – claim – deviceEnabled
beginRemoval	X	open – claim – deviceEnabled
changePrintSide	X	open – claim – deviceEnabled
clearPrintArea	O	open – claim – deviceEnabled
cutPaper	O	open – claim – deviceEnabled
drawRuledLine	X	open – claim – deviceEnabled
endInsertion	X	open – claim – deviceEnabled
endRemoval	X	open – claim – deviceEnabled
markFeed	X	open – claim – deviceEnabled
pageModePrint	O	open – claim – deviceEnabled
printBarCode	O	open – claim – deviceEnabled
printBitmap	O	open – claim – deviceEnabled
printImmediate	O	open – claim – deviceEnabled
printMemoryBitmap	X	open – claim – deviceEnabled
printNormal	O	open – claim – deviceEnabled
printTwoNormal	X	open – claim – deviceEnabled
rotatePrint	X	open – claim – deviceEnabled
setBitmap	X	open – claim – deviceEnabled
setLogo	X	open – claim – deviceEnabled
transactionPrint	O	open – claim – deviceEnabled
validateData	X	open – claim – deviceEnabled

O: 지원 X: 미지원

6-2-2 CashDrawer Method

Method	Value	선행작업
open	O	-
claim	O	open
releaseDevice	O	open – claim
close	O	open
openDrawer	O	open – claim – deviceEnabled

O: 지원 X: 미지원

6-2-3 MSR Method

Method	Value	선행작업
open	O	-
claim	O	open
releaseDevice	O	open – claim
close	O	open

O: 지원 X: 미지원

6-2-4 SCR Method

Method	Value	선행작업
open	O	-
claim	O	Open
releaseDevice	O	open – claim
close	O	open
beginInsertion	O	open – claim – deviceEnabled
beginRemoval	O	open – claim – deviceEnabled
endInsertion	O	open – claim – deviceEnabled
endRemoval	O	open – claim – deviceEnabled

O: 지원 X: 미지원

7. [Common] Device Class Reference

Inherits from	NSObject
Framework	libBixolonUPOS.a
[Availability]	iOS 7.0 and later
Class name	UPOSDevice
Declared	UPOSDevices.h

7-1 Overview

UPOSDevice Class 는 각 장치의 Controller 에서 제어 대상이 되는 장치의 공통된 정보를 가지는 객체 입니다.

7-2 Available Properties

7-2-1 modelName

각 Device 의 모델명이 포함되어있습니다.

[Declare]

```
@property (retain) NSString *modelName;
```

[Availability]

SDK 1.0.0 and later

7-2-2 Idn (Logical Device Name)

각 Device 의 Local Name 입니다.
Open 메소드에서 사용할 수 있습니다.

[Declare]

```
@property (retain) NSString *Idn;
```

[Availability]

SDK 1.0.0 and later

7-2-3 InterfaceType

각 장치의 연결 방식을 의미합니다.

[Declare]

```
@property (retain) NSString *interfaceType;
```

[Availability]

SDK 1.0.0 and later

7-2-4 address

각 장치의 주소값 입니다.

Interface Type이 _CONNECT_BLUETOOTH 일 경우 Mac Address
아닌경우 IP 주소 입니다.

[Declare]

@property (retain) NSString	*interfaceType;
-----------------------------	-----------------

[Availability]

SDK 1.0.0 and later

7-2-5 port

Socket(Wifi / Ethernet) 통신 시 사용하는 Network Port 입니다.

Interface Type 이 _CONNECT_BLUETOOTH 일 경우 무시될 수 있습니다.

[Declare]

@property (retain) NSString	*port;
-----------------------------	--------

[Availability]

SDK 1.0.0 and later

7-3 Available Method

- 유효하지 않습니다.

8. [Common] Device List Class Reference

Inherits from	NSObject
Framework	libBixolonUPOS.a
[Availability]	iOS 8.0 and later
Class name	UPOSDevices
Declared	UPOSDevices.h

8-1 Overview

UPOSDevices Class 는 iDevice 에 저장되어있는 각 장치리스트를 가지는 객체 입니다.

8-2 Available Properties

- 유효하지 않습니다.

8-3 Available Method

8-3-1 getList

저장되어있는 장치의 목록을 얻어옵니다.

[See Also]

- Device Class Reference (Common Device)
- Device Controller Class Reference (Common Controller)
- [Printer] Printer Controller Class Reference
- [CashDrawer] Cash Drawer Controller Class Reference
- [MSR] MSR Controller Class Reference
- [Printer] Printer Class Reference
- [CashDrawer] Cash Drawer Class Reference
- [MSR] MSR Class Reference
- [Printer] Printer List Class Reference
- [CashDrawer] Cash Drawer List Class Reference
- [MSR] MSR List Class Reference (저장된 MSR 목록)

[Declare]

```
-(NSMutableArray*)getList;
```

[Parameters]

없음.

[Return Value]

- NSMutableArray*:
저장되어있는 장치의 목록이 Array 형태로 저장 됩니다.
각 객체는 Device Class 형태로 저장 됩니다.

[Example]

이 클래스의 Method 를 직접 호출 할 경우 정상적으로 동작하지 않을 수 있습니다.
필요한 경우 해당 장치에 정의 되어있는 Method 를 호출하여 사용하십시오.

[Availability]

SDK 1.0.0 and later

8-3-2 getDeviceIdentity

현재 장치의 Identity String 을 얻어옵니다.

[Declare]

```
-(NSString*)getDeviceIdentity;
```

[Parameters]

없음.

[Return Value]

- NSString*:
현재 장치의 Identity 를 스트링 형태로 반환합니다.

[Example]

이 클래스의 Method 를 직접 호출 할 경우 정상적으로 동작하지 않을 수 있습니다.
필요한 경우 해당 장치에 정의 되어있는 Method 를 호출하여 사용하십시오.

[Availability]

SDK 1.0.0 and later

8-3-3 save

현재 장치 목록을 저장 합니다.

[Declare]

```
-(BOOL) save;
```

[Parameters]

없음.

[Return Value]

- BOOL:
성공시 YES.

[Example]

이 클래스의 Method 를 직접 호출 할 경우 정상적으로 동작하지 않을 수 있습니다.
필요한 경우 해당 장치에 정의 되어있는 Method 를 호출하여 사용하십시오.

[Availability]

SDK 1.0.0 and later

8-3-4 addDevice

현재 장치 목록에 장치를 추가 합니다.

본 메소드 사용 후 **save** 메소드를 이용하여 저장하지 않으면,
List 새로 고침 될 때 추가된 장치가 포함 되어있지 않을 수 있습니다.

[See Also]

- [Device List Class Reference \(저장된 장치 목록\)](#) :: [save](#) 참조

[Declare]

```
-(BOOL) addDevice:(UPOSDevice*)device;
```

[Parameters]

- (UPOSDevice*) device
추가 할 장치의 정보를 가지고 있는 객체.

[Return Value]

- BOOL:
성공시 YES.

[Example]

이 클래스의 Method 를 직접 호출 할 경우 정상적으로 동작하지 않을 수 있습니다.
필요한 경우 해당 장치에 정의 되어있는 Method 를 호출하여 사용하십시오.

[Availability]

SDK 1.0.0 and later

8-3-5 removeDevice

현재 장치 목록에 장치를 삭제 합니다.

본 메소드 사용 후 **save** 메소드를 이용하여 저장하지 않으면,
List 새로 고침 될 때 장치 삭제가 적용 되어있지 않을 수 있습니다.

[See Also]

- [Device List Class Reference \(저장된 장치 목록\)](#) :: [save](#) 참조

[Declare]

```
-(BOOL) removeDevice:(UPOSDevice*)device;
```

[Parameters]

- (UPOSDevice*) device
추가 할 장치의 정보를 가지고 있는 객체

[Return Value]

- BOOL:
성공시 YES.

[Example]

이 클래스의 **Method** 를 직접 호출 할 경우 정상적으로 동작하지 않을 수 있습니다.
필요한 경우 해당 장치에 정의 되어있는 **Method** 를 호출하여 사용하십시오.

[Availability]

SDK 1.0.0 and later

9. [Common] Device Controller Class Reference

Inherits from	NSObject
Framework	libBixolonUPOS.a
[Availability]	iOS 7.0 and later
Class name	UPOSDeviceController
Declared	UPOSDeviceController.h

9-1 Overview

UPOSDeviceController Class 는 본 SDK 에서 지원하는 장치의 공통된 기능의 제어를 위한 메인 객체 입니다.

9-2 Available Properties

9-2-1 CheckHealthText

각 Device 의 checkHealth 기능을 이용했을 때 출력되는 Text 의 내용입니다.
출력 Device 가 아닐경우 이 값은 의미가 없을 수 있습니다.

[Declare]

```
@property (strong, readonly) NSString* CheckHealthText;
```

[Availability]

SDK 1.0.0 and later

9-2-2 Claimed

Claim 상태를 나타냅니다.

YES 인 경우, 각 Device 의 Claim: API 를 호출하여 성공 했음을 의미합니다.
이 프로퍼티는 Claim 메소드가 호출 될 때 초기화 됩니다.

[See Also]

- Claim

[Declare]

```
@property (readonly) BOOL Claimed;
```

[Availability]

SDK 1.0.0 and later

9-2-3 DeviceEnabled

Device 사용할 지 여부를 포함하고 있습니다.

Claimed 의 상태가 YES 인 경우에도 DeviceEnabled 의 값이 NO 이면 기능 이용이 불가능 할 수 있습니다.

[See Also]

- Claimed

[Declare]

@property	BOOL	DeviceEnabled;
-----------	------	----------------

[Availability]

SDK 1.0.0 and later

9-2-4 OpenResult

Open method 의 수행 결과가 저장 됩니다.

이 프로퍼티는 Open 메소드가 호출 될 때 초기화 됩니다.

[See Also]

- OpenResult Code
- open

[Declare]

@property (readonly)	NSInteger	OpenResult;
----------------------	-----------	-------------

[Availability]

SDK 1.0.0 and later

9-2-5 ResultCode

마지막으로 호출 한 메소드의 호출 결과가 저장 됩니다.

[See Also]

- 상수값_(Defines) :: Result Code

[Declare]

@property (readonly)	NSInteger	ResultCode;
----------------------	-----------	-------------

[Availability]

SDK 1.0.0 and later

9-2-6 ResultCodeExtended

장치에 Error 가 발생했을 때 장치의 자세한 Error 상태를 저장 합니다.
 ResultCode의 결과값으로 UPOS_E_EXTENDED 가 저장 되었을 때 초기화 됩니다.

[See Also]

- 상수값_(Defines) :: Result Code
- [Common] Device Controller Class Reference :: Available Properties :: ResultCode

[Declare]

@property (readonly)	NSInteger	ResultCode;
----------------------	-----------	-------------

[Availability]

SDK 1.0.0 and later

9-2-7 OutputID

Async Mode 일 때 인쇄가 성공 되면
 OutputComplete: Delegate 가 호출 되고, 이 때 1씩 증가 합니다.

[Declare]

@property (readonly)	NSInteger	OutputID;
----------------------	-----------	-----------

[Availability]

SDK 1.0.0 and later

9-2-8 State

Class의 상태값이 저장 됩니다.

[See Also]

- 상수값_(Defines) :: State Code

[Declare]

@property (readonly)	NSInteger	State;
----------------------	-----------	--------

[Availability]

SDK 1.0.0 and later

9-3 Available Method

9-3-1 open

프린터 클래스 사용을 시작하며 메모리 할당 등의 초기화 작업이 포함되어있습니다.
Claim 이상의 Method 호출을 위해서 반드시 선행 되어야 합니다.

[See Also]

- 상수값 (Defines) :: OpenResult Code
- OpenResult
- Idn (Logical Device Name)

[Declare]

```
-(NSInteger) open : (NSString*)logicalDeviceName;
```

[Parameters]

(NSString*) logicalDeviceName

- Open 하려는 Device 의 모델명 또는 저장된 Device Name
[Idn \(Logical Device Name\)](#) 참조

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

이 클래스의 Method 를 직접 호출 할 경우 정상적으로 동작하지 않을 수 있습니다.
필요한 경우 해당 장치에 정의 되어있는 Method 를 호출하여 사용하십시오.

[Availability]

SDK 1.0.0 and later

9-3-2 claim

Device 정보에 포함되어있는 **Port** 를 실제로 **Open** 하기 위해 시도 하며, 메모리 할당 초기화 등의 초기화 작업이 일부 포함되어있습니다.

Device 사용을 활성화 하기 위해서 반드시 선행 되어야 합니다.

[See Also]

- 상수값 (Defines) :: Result Code
- Claimed
- open

[Declare]

```
-(NSInteger) claim : (NSInteger)timeout;
```

[Parameters]

(NSInteger) timeout

- 이 파라미터에서 지정한 시간동안 **Port open** 을 시도합니다.

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

이 클래스의 **Method** 를 직접 호출 할 경우 정상적으로 동작하지 않을 수 있습니다.
필요한 경우 해당 장치에 정의 되어있는 **Method** 를 호출하여 사용하십시오.

[Availability]

SDK 1.0.0 and later

9-3-3 releaseDevice

Claim 된 Device의 Port 사용을 물리적으로 종료 합니다. 메모리 해제 등의 작업이 수행 될 수 있습니다.

[See Also]

- claim
- Claimed

[Declare]

`-(NSInteger) open : (NSString*)logicalDeviceName;`

[Parameters]

(NSString*) logicalDeviceName

- Open 하려는 Device 의 모델명 또는 저장된 Device Name
[Idn \(Logical Device Name\)](#) 참조

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

이 클래스의 Method 를 직접 호출 할 경우 정상적으로 동작하지 않을 수 있습니다.
필요한 경우 해당 장치에 정의 되어있는 Method 를 호출하여 사용하십시오.

[Availability]

SDK 1.0.0 and later

9-3-4 close

Open 된 Device의 사용을 종료 합니다.
메모리 해제 등의 작업이 일부 수행 될 수 있습니다.

[See Also]

- 상수값 (Defines) :: Result Code
- open

[Declare]

```
-(NSInteger) claim : (NSInteger)timeout;
```

[Parameters]

(NSInteger) timeout

- 이 파라미터에서 지정한 시간동안 Port open 을 시도합니다.

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

이 클래스의 Method 를 직접 호출 할 경우 정상적으로 동작하지 않을 수 있습니다.
필요한 경우 해당 장치에 정의 되어있는 Method 를 호출하여 사용하십시오.

[Availability]

SDK 1.0.0 and later

9-3-5 checkHealth

장치가 정상적으로 동작하는 상태인지 확인 합니다.

Open / Claim / DeviceEnabled 가 정상적으로 수행된 상태일 때 사용 가능 합니다.

[See Also]

- 상수값 (Defines) :: Result Code
- open
- claim
- DeviceEnabled

[Declare]

`-(NSInteger) checkHealth : (NSInteger) level;`

[Parameters]

(NSInteger) level

- 고정 값: UPOS_CH_INTERNAL

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

이 클래스의 Method 를 직접 호출 할 경우 정상적으로 동작하지 않을 수 있습니다.
필요한 경우 해당 장치에 정의 되어있는 Method 를 호출하여 사용하십시오.

[Availability]

SDK 1.0.0 and later

10. [Printer] Printer Class Reference

Inherits from	NSObject
Framework	libBixolonUPOS.a
[Availability]	iOS 7.0 and later
Class name	UPOSPrinter
Declared	UPOSDevices.h

10-1 Overview

UPOSPrinter Class 는 UPOSPrinterController 에서 제어 대상이 되는 장치의 정보를 가지는 객체 입니다.

[See Also]

- [Common] Device Class Reference

10-2 Available Properties

- [\[Common\] Device Class Reference](#) :: [Available Properties](#) 참조

10-3 Available Method

- [\[Common\] Device Class Reference](#) :: [Available Method](#) 참조

11. [Printer] Printer List Class Reference

Inherits from	UPOSDevices
Framework	libBixolonUPOS.a
[Availability]	iOS 7.0 and later
Class name	UPOSPrinters
Declared	UPOSDevices.h

11-1 Overview

UPOSPrinters Class 는 iDevice 에 저장되어있는 프린터 리스트를 가지는 객체 입니다. UPOSPrinter 의 집합입니다.

[See Also]

- Printer Controller Class Reference

11-2 Available Properties

- 유효하지 않습니다.

11-3 Available Method

11-3-1 getList

저장되어있는 장치의 목록을 얻어옵니다.

[See Also]

- Device Class Reference (Common Device)
- Device Controller Class Reference (Common Controller)
- Printer Controller Class Reference

[Declare]

- [Device List Class Reference \(저장된 장치 목록\)](#) :: [getList](#) 참조

[Parameters]

없음.

[Return Value]

- NSMutableArray*:
저장되어있는 장치의 목록이 Array 형태로 저장 됩니다.
각 객체는 Device Class 형태로 저장 됩니다.

[Example]

```
UPOSPrinterController* _uposPrinterController = [ UPOSPrinterController new];
UPOSPrinters* printerList = [_uposPrinterController getRegisteredDevice];

NSMutableArray* printerArray = [printerList getList];

if(printerArray == nil)
{
    NSLog(@"저장된 프린터 리스트를 읽어올 수 없습니다.");
}
else
{
    NSLog(@"저장된 프린터 리스트를 읽어왔습니다.");
}
```

[Availability]

SDK 1.0.0 and later

11-3-2 getDeviceIdentity

현재 장치의 Identity String 을 얻어옵니다.

[Declare]

- [Device List Class Reference \(저장된 장치 목록\)](#) :: [getDeviceIdentity](#) 참조

[Parameters]

없음.

[Return Value]

- NSString*:
현재 장치의 Identity 를 스트링 형태로 반환합니다.

[Example]

```
UPOSPrinterController* _uposPrinterController = [ UPOSPrinterController new];
UPOSPrinters* printerList = [_uposPrinterController getRegisteredDevice];

NSString* strIdentity = [printerList getDeviceIdentity];

if(printerArray == nil)
{
    NSLog(@"Identity 값을 읽어올 수 없습니다.");
}
else
{
    NSLog(@" UPOSPrinterController 의 Identity :%@", strIdentity);
}
```

[Availability]

SDK 1.0.0 and later

11-3-3 save

현재 장치 목록을 저장 합니다.

[See Also]

- Printer List Class Reference (저장된 프린터 목록) :: removeDevice
- Printer List Class Reference (저장된 프린터 목록) :: addDevice

[Declare]

- [Device List Class Reference \(저장된 장치 목록\)](#) :: [save](#) 참조

[Parameters]

없음.

[Return Value]

- BOOL:
성공시 YES.

[Example]

```
UPOSPrinterController* _uposPrinterController = [ UPOSPrinterController new];
UPOSPrinters* printerList = [_uposPrinterController getRegisteredDevice];

UPOSPrinter* newDevice = [UPOSPrinter new];
newDevice.modelName = @"SRP-350plusIII";
newDevice.Idn = @"type1";
newDevice.interfaceType = @"4"; // Bluetooth
newDevice.address = @"74:F0:E1:XX:XX:XX";
newDevice.port = @"";
[printerList addDevice:newDevice]; // Device 추가
[newDevice release];

if([printerList save])
{
    NSLog(@"List 를 저장 했습니다.");
}
else
{
    NSLog(@"List 를 저장 할 수 없습니다.");
}
```

[Availability]

SDK 1.0.0 and later

11-3-4 addDevice

현재 장치 목록에 장치를 추가 합니다.

본 메소드 사용 후 **save** 메소드를 이용하여 저장하지 않으면,
List 새로 고침 될 때 추가된 장치가 포함 되어있지 않을 수 있습니다.

[See Also]

- Printer List Class Reference (저장된 프린터 목록) :: save
- Printer List Class Reference (저장된 프린터 목록) :: removeDevice

[Declare]

- [Device List Class Reference \(저장된 장치 목록\)](#) :: [addDevice](#) 참조

[Parameters]

- (UPOSDevice*) device
추가 할 장치의 정보를 가지고 있는 객체.

[Return Value]

- BOOL:
성공시 YES.

[Example]

```
UPOSPrinterController* _uposPrinterController = [ UPOSPrinterController new];
UPOSPrinters* printerList = [_uposPrinterController getRegisteredDevice];

UPOSPrinter* newDevice = [UPOSPrinter new];
newDevice.modelName = @"SRP-350plusIII";
newDevice.lidn = @"type1";
newDevice.interfaceType = @"4"; // Bluetooth
newDevice.address = @"74:F0:E1:XX:XX:XX";
newDevice.port = @"";

if([printerList addDevice:newDevice]) // Device 추가
{
    NSLog(@"Device 를 추가를 성공했습니다");
}
else
{
    NSLog(@"Device 를 추가를 실패했습니다");
}

[newDevice release];
[printerList save];
```

[Availability]

SDK 1.0.0 and later

11-3-5 removeDevice

현재 장치 목록에 장치를 삭제 합니다.

본 메소드 사용 후 **save** 메소드를 이용하여 저장하지 않으면,
List 새로 고침 될 때 장치 삭제가 적용 되어있지 않을 수 있습니다.

[See Also]

- Printer List Class Reference (저장된 프린터 목록) :: save
- Printer List Class Reference (저장된 프린터 목록) :: addDevice

[Declare]

- [Device List Class Reference \(저장된 장치 목록\)](#) :: [removeDevice](#) 참조

[Parameters]

- (UPOSDevice*) device
추가 할 장치의 정보를 가지고 있는 객체.

[Return Value]

- BOOL:
성공시 YES.

[Example]

```
UPOSPrinterController* _uposPrinterController = [ UPOSPrinterController new];
UPOSPrinters* printerList = [_uposPrinterController getRegisteredDevice];

UPOSPrinter* willRemoveDevice = [[printerList getList] objectAtIndex:0];
if([printerList removeDevice:willRemoveDevice]) // Device 삭제
{
    NSLog(@"0번째 Device 를 목록에서 삭제 했습니다.");
}
else
{
    NSLog(@"0번째 Device 를 목록에서 삭제 할 수 없습니다.");
}

[newDevice release];
[printerList save];
```

[Availability]

SDK 1.0.0 and later

12. [Printer] Printer Controller Class Reference

Inherits from	UPOSDeviceController
Framework	libBixolonUPOS.a
[Availability]	iOS 7.0 and later
Class name	UPOSPrinterController
Declared	UPOSPrinterController.h

12-1 Overview

UPOSPrinterController Class 는 프린터 제어를 위한 메인 객체 입니다.

12-2 Properties

프로퍼티는 아래의 프린터 타입 기준에 따라 프로퍼티가 결정됩니다.
(일부 용지 설정 등에 대한 프로퍼티에 대해서는 프린터 모델에 따라 차이가 발생할 수 있습니다)

12-2-1 Capability 설정 값

Capability Name	1 Color Thermal
CapCompareFirmwareVersion	FALSE
CapPowerReporting	TRUE
CapStatisticsReporting	FALSE
CapUpdateFirmware	FALSE
CapUpdateStatistics	FALSE
CapTransaction	TRUE
CapCoverSensor	TRUE
CapConcurrentRecSlp	FALSE
CapConcurrentJrnSlp	FALSE
CapConcurrentJrnRec	FALSE
CapCharacterSet	TRUE
CapRecUnderline	TRUE
CapRecPageMode	FALSE
CapCuncurrentPageMode	FALSE
CapRecStamp	FALSE
CapRecRotate180	FALSE
CapRecRight90	FALSE
CapRecPapercut	TRUE
CapRecNearEndSensor	TRUE
CapRecMarkFeed	FALSE
CapRecLeft90	FALSE
CapRecItalic	FALSE
CapRecEmptySensor	TRUE
CapRecDwideDhigh	TRUE
CapRecDwide	TRUE
CapRecDhigh	TRUE
CapRecColor	FALSE
CapRecCartridgeSensor	FALSE
CapRecBold	TRUE
CapRecBitmap	TRUE
CapRecBarCode	TRUE
CapRec2Color	FALSE
CapRecPresent	TRUE

* 프린터 모델에 따라 Capability 값이 일부 다를 수 있습니다.

12-2-2 프로퍼티 기본 설정 값 및 설정 범위

- List 프로퍼티

Model	RecLineCharsList	RecBarCode RotationList	FontTypefaceList	RecBitmapList
SPP-R210	"32,42"	"0"	""	"0"
SPP-R220	"32,42"	"0"	""	"0"
SPP-R310	"48,64"	"0"	""	"0"
SPP-R410	"69,92"	"0"	""	"0"
SPP-R200III	"32,42"	"0"	""	"0"
SRP-350plusIII	"42,56"	"0"	""	"0"
SRP-352plusIII	"48,64"	"0"	""	"0"
SRP-350III	"42,56"	"0"	""	"0"
SRP-352III	"48,64"	"0"	""	"0"
SRP-F310II	"42,56"	"0"	""	"0"
SRP-F312II	"48,64"	"0"	""	"0"
SRP-F313II	"50,67"	"0"	""	"0"
SRP-275III	"33,44"	"0"	""	"0"
SRP-380	"42,56"	"0"	""	"0"
SRP-382	"48,64"	"0"	""	"0"
SRP-330II	"42,56"	"0"	""	"0"
SRP-332II	"48,64"	"0"	""	"0"
SRP-340II	"42,56"	"0"	""	"0"
SRP-342II	"48,64"	"0"	""	"0"
SRP-S300	"48,64"	"0"	""	"0"
SRP-Q300	"42,56"	"0"	""	"0"
SRP-Q302	"48,64"	"0"	""	"0"

- CharacterSetList 프로퍼티

Model	Value
SPP-R210 / SPP-R220 / SPP-R310 / SPP-R410 / SPP-R200III / SRP-350plusIII / SRP-352plusIII / SRP-350III / SRP-352III / SRP-275III / SRP-F310II / SRP-F312II / SRP-F313II / SRP-380 / SRP-382 / SRP-330II / SRP-332II / SRP-340II / SRP-342II / SRP-S300 / SRP-Q300 / SRP-Q302	"437,737,775,850,852,855,857,858,860,862,863, 864,865,866,928,Farsi,Thai12,Thai14,Thai16, Thai18,Thai421250,1251,1252,1253,1254,1255, 1256,1257,1258"

- 용지 넓이 및 높이 프로퍼티

Model	RecLineHeight	RecLineWidth	RecLinePaperCut
SPP-R210	24	384	-
SPP-R220	24	384	-
SPP-R310	24	576	-
SPP-R410	24	832	-
SPP-R200III	24	384	-
SRP-350plusIII	24	512	5
SRP-352plusIII	24	576	5
SRP-350III	24	512	5
SRP-352III	24	576	5
SRP-F310II	24	512	5
SRP-F312II	24	576	5
SRP-F313II	24	604	5
SRP-275III	24	400	5
SRP-380	24	512	5
SRP-382	24	576	5
SRP-330II	24	512	5
SRP-332II	24	576	5
SRP-340II	24	512	5
SRP-342II	24	576	5
SRP-S300	24	576	5
SRP-Q300	24	512	5
SRP-Q302	24	576	5

- RecLineSpacing 프로퍼티

Property	설정범위		
	기본 값	최대 값	최소 값
RecLineSpacing	16	127	0

12-2-3 RecLineChars

라인 당 표시 가능한 글자 수 입니다.

RecLineCharsList에 표시되는 숫자 들 중에서 선택 할 수 있습니다.

[Declare]

```
@property                NSInteger                RecLineChars;
```

[Availability]

SDK 1.0.0 and later

12-2-4 RecLineCharsList

라인 당 표시 가능한 글자 수 입니다.

폰트의 폭에 따라 달라질 수 있는 라인당 표시 가능한 글자 수 입니다.

[Declare]

@property (strong, readonly)	NSString*	RecLineCharsList;
------------------------------	-----------	-------------------

[Availability]

SDK 1.0.0 and later

12-2-5 RecLineSpacing

라인과 라인 사이의 줄 간격 입니다.

[Declare]

@property	NSInteger	RecLineSpacing;
-----------	-----------	-----------------

[Availability]

SDK 1.0.0 and later

12-2-6 RecLineWidth

프린터가 기본으로 지원하는 용지의 너비 입니다.

[Declare]

@property (readonly)	NSInteger	RecLineWidth;
----------------------	-----------	---------------

[Availability]

SDK 1.0.0 and later

12-2-7 RecEmpty

용지의 유무를 나타냅니다.

YES 일 경우 용지 없음을 의미합니다.

[Declare]

@property (readonly)	BOOL	RecEmpty;
----------------------	------	-----------

[Availability]

SDK 1.0.0 and later

12-2-8 RecNearEnd

용지의 유무를 나타냅니다.
용지가 없을 경우 이 값에 YES 가 대입 됩니다.

[Declare]

@property (readonly)	BOOL	RecNearEnd;
----------------------	------	-------------

[Availability]

SDK 1.0.0 and later

12-2-9 AsyncMode

YES로 설정 하면 인쇄 관련 메소드가 비동기 모드로 동작 합니다.
NO로 설정하면 인쇄 관련 메소드가 동기 모드로 동작 합니다.

초기 설정은 NO로, Open 할 때마다 초기화 됩니다.
비동기 모드 일 경우 OutputCompleteEvent를 통하여
인쇄 완료를 확인 할 수 있습니다.

[See Also]

- outputCompleteEvent

[Declare]

@property (nonatomic)	BOOL	AsyncMode;
-----------------------	------	------------

[Availability]

SDK 1.0.0 and later

12-2-10 CharacterSet

프린터에 인쇄시 사용할 character set 을 표기합니다.

[Declare]

@property (nonatomic)	NSInteger	CharacterSet;
-----------------------	-----------	---------------

[Availability]

SDK 1.0.0 and later

12-2-11 CharacterSetList

프린터에서 지원하는 character set의 목록 입니다.

[Declare]

@property (strong, readonly)	NSString*	CharacterSetList;
------------------------------	-----------	-------------------

[Availability]

SDK 1.0.0 and later

12-2-12 CoverOpen

프린터의 Cover 상태를 나타냅니다.
YES일 경우 Cover 열림을 의미합니다.

[Declare]

@property	BOOL	CoverOpen;
-----------	------	------------

[Availability]

SDK 1.0.0 and later

12-2-13 ErrorLevel

Error 의 상태를 나타냅니다.

[Declare]

@property (readonly)	NSInteger	ErrorLevel;
----------------------	-----------	-------------

[Availability]

SDK 1.0.0 and later

12-2-14 ErrorString

Error 의 상태를 문자열 형태로 확인 할 수 있습니다.

[Declare]

@property (strong, readonly)	NSString*	ErrorString;
------------------------------	-----------	--------------

[Availability]

SDK 1.0.0 and later

12-2-15 FlagWhenIdle

Error 상태가 해결 되었을 때 `statusUpdateEvent` 를 받을지 여부를 선택합니다.
YES를 설정하면 Error 상태가 해결될 때마다 `statusUpdateEvent` 가 발생합니다.

초기 설정은 NO로, Open 할 때마다 초기화 됩니다.

[Declare]

@property (nonatomic)	BOOL	FlagWhenIdle;
-----------------------	------	---------------

[Availability]

SDK 1.0.0 and later

12-3 Available Method

12-3-1 open

프린터 클래스 사용을 시작하며 메모리 할당 등의 초기화 작업이 포함되어있습니다.
Claim 이상의 Method 호출을 위해서 반드시 선행 되어야 합니다.

[See Also]

- 상수값 (Defines) :: OpenResult Code
- [Common] Device Controller Class Reference :: Available Properties :: OpenResult
- [Common] Device Controller Class Reference :: Available Properties ::
Idn(Logical Device Name)
- [Printer] Printer List Class Reference :: addDevice

[Declare]

- [Common] Device Controller Class Reference :: Available Method :: open

[Parameters]

(NSString*) logicalDeviceName

- Open 하려는 Device 의 모델명 또는 저장된 Device Name
[Idn \(Logical Device Name\)](#) 참조
- 사용하려는 Device 는 [Printer] Printer List Class Reference :: addDevice
메소드를 통해 미리 추가가 되어있어야 합니다.

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

```
UPOSPrinterController* _uposPrinterController = [ UPOSPrinterController new];
UPOSPrinters* printerList = [_uposPrinterController getRegisteredDevice];

if(UPOS_SUCCESS == [_uposPrinterController open:@"type1"])
{
    if(UPOS_SUCCESS == [_uposPrinterController claim:3000])
    {
        _uposDeviceController.deviceEnabled = YES;
        // 프린터 사용 가능
    }
}
```

[Availability]

SDK 1.0.0 and later

12-3-2 claim

Device 정보에 포함되어있는 Port 를 실제로 Open 하기 위해 시도 하며, 메모리 할당 초기화 등의 초기화 작업이 일부 포함되어있습니다.

Device 사용을 활성화 하기 위해서 반드시 선행 되어야 합니다.

[See Also]

- 상수값 (Defines) :: Result Code
- [Common] Device Controller Class Reference :: Available Method :: open
- [Printer] Printer Controller Class Reference :: Available Method :: open
- [Common] Device Controller Class Reference :: Available Method :: claim
- [Common] Device Controller Class Reference :: Available Properties :: Claimed

[Declare]

- [Common] Device Controller Class Reference :: Available Method :: claim

[Parameters]

(NSInteger) timeout

- 이 파라미터에서 지정한 시간동안 Port open 을 시도합니다.

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

```
UPOSPrinterController* _uposPrinterController = [ UPOSPrinterController new];
UPOSPrinters* printerList = [_uposPrinterController getRegisteredDevice];

if(UPOS_SUCCESS == [_uposPrinterController open:@"type1"])
{
    if(UPOS_SUCCESS == [_uposPrinterController claim:3000])
    {
        _uposDeviceController.deviceEnabled = YES;
        // 프린터 사용 가능
    }
}
```

[Availability]

SDK 1.0.0 and later

12-3-3 releaseDevice

Claim 된 Device의 Port 사용을 물리적으로 종료 합니다. 메모리 해제 등의 작업이 수행 될 수 있습니다.

[See Also]

- 상수값 (Defines) :: Result Code
- [Common] Device Controller Class Reference :: Available Method :: open
- [Printer] Printer Controller Class Reference :: Available Method :: open
- [Common] Device Controller Class Reference :: Available Method :: claim
- [Printer] Printer Controller Class Reference :: Available Method :: claim
- [Common] Device Controller Class Reference::Available Properties::DeviceEnabled
- [Common] Device Controller Class Reference :: Available Properties :: Claimed
- [Printer] Printer Controller Class Reference :: Available Method :: releaseDevice

[Declare]

- [Common] Device Controller Class Reference :: Available Method :: releaseDevice

[Parameters]

(NSString*) logicalDeviceName

- Open 하려는 Device 의 모델명 또는 저장된 Device Name
[Idn \(Logical Device Name\)](#) 참조

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

```
UPOSPrinterController* _uposPrinterController = [ UPOSPrinterController new];
UPOSPrinters* printerList = [_uposPrinterController getRegisteredDevice];

/// Printer 사용 후

// Printer 사용 정리 작업
_uposDeviceController.deviceEnabled = NO;
[_uposPrinterController releaseDevice];
[_uposPrinterController close];
```

[Availability]

SDK 1.0.0 and later

12-3-4 close

Open 된 Device의 사용을 종료 합니다.
메모리 해제 등의 작업이 일부 수행 될 수 있습니다.

[See Also]

- 상수값 (Defines) :: Result Code
- [Common] Device Controller Class Reference :: Available Method :: open
- [Printer] Printer Controller Class Reference :: Available Method :: open
- [Common] Device Controller Class Reference :: Available Method :: claim
- [Printer] Printer Controller Class Reference :: Available Method :: claim
- [Common] Device Controller Class Reference::Available Properties::DeviceEnabled
- [Common] Device Controller Class Reference :: Available Properties :: Claimed
- [Common] Device Controller Class Reference :: Available Method :: close

[Declare]

- [Common] Device Controller Class Reference :: Available Method :: close

[Parameters]

(NSInteger) timeout

- 이 파라미터에서 지정한 시간동안 Port open 을 시도합니다.

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

```
UPOSPrinterController* _uposPrinterController = [ UPOSPrinterController new];
UPOSPrinters* printerList = [_uposPrinterController getRegisteredDevice];

/// Printer 사용 후

// Printer 사용 정리 작업
_uposDeviceController.deviceEnabled = NO;
[_uposPrinterController releaseDevice];
[_uposPrinterController close];
```

[Availability]

SDK 1.0.0 and later

12-3-5 cutPaper

Auto Cutter 가 내장된 모델의 경우 종이를 절단합니다.
 open-claim-enable 이후에 사용 가능 합니다.
 AsyncMode가 True 인 경우 비동기모드로 동작 합니다.

[See Also]

- 상수값 (Defines) :: Result Code
- [Common] Device Controller Class Reference :: Available Method :: open
- [Printer] Printer Controller Class Reference :: Available Method :: open
- [Common] Device Controller Class Reference :: Available Method :: claim
- [Printer] Printer Controller Class Reference :: Available Method :: claim
- [Common] Device Controller Class Reference::Available Properties::DeviceEnabled
- [Common] Device Controller Class Reference :: Available Properties :: Claimed

[Declare]

```
-(NSInteger) cutPaper : (NSInteger)percentage;
```

[Parameters]

(NSInteger) percentage

- 종이 절단 Level 을 지정 합니다.
- 숫자가 클수록 많이 자릅니다.

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS ([상수값 \(Defines\)](#) :: [Result Code](#) 참조)

[Example]

```
UPOSPrinterController* _uposPrinterController = [ UPOSPrinterController new];
UPOSPrinters* printerList = [_uposPrinterController getRegisteredDevice];

/// Printer 초기화(open-claim-deviceEnable) 후
[_uposPrinterController cutPaper:100];
// Printer 사용 정리 작업
_uposDeviceController.deviceEnabled = NO;
[_uposPrinterController releaseDevice];
[_uposPrinterController close];
```

[Availability]

SDK 1.0.0 and later

12-3-6 markFeed

다음 인쇄 위치까지 용지를 Feeding 합니다.
open-claim-enable 이후에 사용 가능 합니다.
AsyncMode가 True 인 경우 비동기모드로 동작 합니다.

[See Also]

- 상수값 (Defines) :: Result Code
- [Common] Device Controller Class Reference :: Available Method :: open
- [Printer] Printer Controller Class Reference :: Available Method :: open
- [Common] Device Controller Class Reference :: Available Method :: claim
- [Printer] Printer Controller Class Reference :: Available Method :: claim
- [Common] Device Controller Class Reference::Available Properties::DeviceEnabled
- [Common] Device Controller Class Reference :: Available Properties :: Claimed

[Declare]

```
-(NSInteger) markFeed : (NSInteger)type;
```

[Parameters]

(NSInteger) type

- Mark type 을 지정 합니다.

Value	Description
PTR_MF_TO_TAKEUP	용지의 mark가 프린터의 mark sensor 에 인식 될 때까지 Feeding 합니다.
PTR_MF_TO_CUTTER	인쇄된 내용이 잘리지 않을 만큼 Feeding 합니다.
PTR_MF_TO_CURRENT_TOP	지원하지 않습니다.
PTR_MF_TO_NEXT_TOF	다음 용지의 mark 가 프린터의 mark sensor에 인식 될 때까지 용지를 Feeding 합니다.

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS ([상수값 \(Defines\)](#) :: [Result Code](#) 참조)

[Example]

```
UPOSPrinterController* _uposPrinterController = [ UPOSPrinterController new];
UPOSPrinters* printerList = [_uposPrinterController getRegisteredDevice];

/// Printer 초기화(open-claim-deviceEnable) 후
[_uposPrinterController markeFeed:PTR_MF_TO_TAKEUP];
// Printer 사용 정리 작업
_uposDeviceController.deviceEnabled = NO;
[_uposPrinterController releaseDevice];
[_uposPrinterController close];
```

[Availability]

SDK 1.0.0 and later

12-3-7 printBarcode

바코드를 인쇄 합니다.

[See Also]

- 상수값 (Defines) :: Result Code
- 상수값 (Defines) :: Barcode Type
- 상수값 (Defines) :: Alignment
- [Common] Device Controller Class Reference :: Available Method :: open
- [Printer] Printer Controller Class Reference :: Available Method :: open
- [Common] Device Controller Class Reference :: Available Method :: claim
- [Printer] Printer Controller Class Reference :: Available Method :: claim
- [Common] Device Controller Class Reference::Available Properties::DeviceEnabled
- [Common] Device Controller Class Reference :: Available Properties :: Claimed

[Declare]

```
-(NSInteger) printBarcode : (NSInteger) station
                        data : (NSString*) data
                    symbology : (NSInteger) symbology
                      height : (NSInteger) height
                      width: (NSInteger) width
                    alignment: (NSInteger) alignment
                    textPostion: (NSInteger) textPosition;
```

[Parameters]

(NSInteger) station

- PTR_S_RECEIPT 고정

(NSString) data

- 바코드에 포함할 Data, 바코드 타입 별 허용되는 Data가 다를 수 있습니다.

(NSInteger) symbology

- Barcode 종류를 선택 합니다. ([상수값 \(Defines\)](#) :: [Barcode Type](#) 참조)

(NSInteger) height

- Barcode 의 높이를 지정합니다.

(NSInteger) width

- Barcode 의 너비를 지정합니다.

(NSInteger) alignment

- 바코드 정렬을 선택합니다. ([상수값 \(Defines\)](#) :: [Alignment](#) 참조)

(NSInteger) textPrositon

- 바코드와 함께 출력 될 Text의 Postion 을 결정합니다.
([상수값 \(Defines\)](#) :: [Barcode Text Position](#) 참조)

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS ([상수값 \(Defines\)](#) :: [Result Code](#) 참조)

[Example]

```

UPOSPrinterController* _uposPrinterController = [ UPOSPrinterController new];
UPOSPrinters* printerList = [_uposPrinterController getRegisteredDevice];

/// Printer 초기화(open-claim-deviceEnable) 후
[_uposPrinterController printBarcode :PTR_S_RECEIPT
                                data :@"1234567890123"
                                symbology : PTR_BCS_EAN13
                                height :100
                                width :200
                                alignment : PTR_BC_CENTER
                                textPostion:PTR_BC_TEXT_BELOW];

// Printer 사용 정리 작업
_uposDeviceController.deviceEnabled = NO;
[_uposPrinterController releaseDevice];
[_uposPrinterController close];

```

[Availability]

SDK 1.0.0 and later

12-3-8 printBitmap (파일 인쇄)

이미지를 인쇄 합니다.

[See Also]

- 상수값 (Defines) :: Result Code
- 상수값 (Defines) :: Alignment
- [Common] Device Controller Class Reference :: Available Method :: open
- [Printer] Printer Controller Class Reference :: Available Method :: open
- [Common] Device Controller Class Reference :: Available Method :: claim
- [Printer] Printer Controller Class Reference :: Available Method :: claim
- [Common] Device Controller Class Reference::Available Properties::DeviceEnabled
- [Common] Device Controller Class Reference :: Available Properties :: Claimed

[Declare]

```
-(NSInteger) printBitmap : (NSInteger) station
                      fileName : (NSString*) fileName
                      width : (NSInteger) width
                      alignment: (NSInteger) alignment;
```

[Parameters]

(NSInteger) station

- PTR_S_RECEIPT 고정

(NSString) fileName

- 이미지 파일의 경로를 지정합니다.

(NSInteger) width

- 이미지 너비를 지정합니다.

(이미지 크기에 상관 없이 프린터에서 지원하는 너비 내에 지정한 너비로 출력됩니다.)

(NSInteger) alignment

- 이미지 정렬을 선택합니다. ([상수값 \(Defines\)](#) :: [Alignment](#) 참조)

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS ([상수값 \(Defines\)](#) :: [Result Code](#) 참조)

[Example]

```

UPOSPrinterController* _uposPrinterController = [ UPOSPrinterController new];
UPOSPrinters* printerList = [_uposPrinterController getRegisteredDevice];

/// Printer 초기화(open-claim-deviceEnable) 후
NSString *path = [[NSBundle mainBundle] pathForResource:@"Sample"
ofType:@"png"];

[_uposPrinterController printBitmap:PTR_S_RECEIPT
                        fileName :path
                        width :200
                        alignment : PTR_BM_CENTER];

// Printer 사용 정리 작업
_uposDeviceController.deviceEnabled = NO;
[_uposPrinterController releaseDevice];
[_uposPrinterController close];

```

[Availability]

SDK 1.0.0 and later

12-3-9 printBitmap (UIImage 인쇄)

이미지를 인쇄 합니다.

[See Also]

- 상수값 (Defines) :: Result Code
- 상수값 (Defines) :: Alignment
- [Common] Device Controller Class Reference :: Available Method :: open
- [Printer] Printer Controller Class Reference :: Available Method :: open
- [Common] Device Controller Class Reference :: Available Method :: claim
- [Printer] Printer Controller Class Reference :: Available Method :: claim
- [Common] Device Controller Class Reference::Available Properties::DeviceEnabled
- [Common] Device Controller Class Reference :: Available Properties :: Claimed

[Declare]

```
-(NSInteger) printBitmap : (NSInteger) station
                        image : (NSString*) image
                        width : (NSInteger) width
                        alignment: (NSInteger) alignment;
```

[Parameters]

(NSInteger) station

- PTR_S_RECEIPT 고정

(UIImage) image

- 이미지를 지정합니다.

(NSInteger) width

- 이미지 너비를 지정합니다.

(이미지 크기에 상관 없이 프린터에서 지원하는 너비 내에 지정한 너비로 출력됩니다.)

(NSInteger) alignment

- 이미지 정렬을 선택합니다. ([상수값 \(Defines\)](#) :: [Alignment](#) 참조)

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS ([상수값 \(Defines\)](#) :: [Result Code](#) 참조)

[Example]

```

UPOSPrinterController* _uposPrinterController = [ UPOSPrinterController new];
UPOSPrinters* printerList = [_uposPrinterController getRegisteredDevice];

// Printer 초기화(open-claim-deviceEnable) 후
IBOutlet UIImageView * _imageView; // 리소스와 연결되어있는 뷰
[_uposPrinterController printBitmap:PTR_S_RECEIPT
                        image :_imageView.image
                        width  :200
                        alignment : PTR_BM_CENTER];

// Printer 사용 정리 작업
_uposDeviceController.deviceEnabled = NO;
[_uposPrinterController releaseDevice];
[_uposPrinterController close];

```

[Availability]

SDK 1.0.0 and later

12-3-10 printNormal

Text 를 인쇄 합니다.

[See Also]

- 상수값 (Defines) :: Result Code
- [Common] Device Controller Class Reference :: Available Method :: open
- [Printer] Printer Controller Class Reference :: Available Method :: open
- [Common] Device Controller Class Reference :: Available Method :: claim
- [Printer] Printer Controller Class Reference :: Available Method :: claim
- [Common] Device Controller Class Reference::Available Properties::DeviceEnabled
- [Common] Device Controller Class Reference :: Available Properties :: Claimed

[Declare]

```
-(NSInteger) printNormal : (NSInteger) station
                        data : (NSString*) data;
```

[Parameters]

(NSInteger) station

- PTR_S_RECEIPT 고정

(NSString) data

- 출력할 Data를 지정합니다.
- 출력 가능한 문자와 escape sequences, carriage returns, line feeds Data가 허용됩니다.

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS ([상수값 \(Defines\)](#) :: [Result Code](#) 참조)

[Example]

```
UPOSPrinterController* _uposPrinterController = [ UPOSPrinterController new];
UPOSPrinters* printerList = [_uposPrinterController getRegisteredDevice];

/// Printer 초기화(open-claim-deviceEnable) 후
IBOutlet UIImageView * _imageView; // 리소스와 연결되어있는 뷰
[_uposPrinterController printNormal :PTR_S_RECEIPT
                                data :@"|N|cAPrint
CenterAlienedText\r\n"];

// Printer 사용 정리 작업
_uposDeviceController.deviceEnabled = NO;
[_uposPrinterController releaseDevice];
[_uposPrinterController close];
```

[Availability]

SDK 1.0.0 and later

12-3-11 transactionPrint

Text 를 인쇄 합니다.

[See Also]

- 상수값 (Defines) :: Result Code
- 상수값 (Defines) :: Transaction Print
- [Common] Device Controller Class Reference :: Available Method :: open
- [Printer] Printer Controller Class Reference :: Available Method :: open
- [Common] Device Controller Class Reference :: Available Method :: claim
- [Printer] Printer Controller Class Reference :: Available Method :: claim
- [Common] Device Controller Class Reference::Available Properties::DeviceEnabled
- [Common] Device Controller Class Reference :: Available Properties :: Claimed

[Declare]

```
-(NSInteger) printNormal : (NSInteger) station
                      control : (NSInteger) control;
```

[Parameters]

(NSInteger) station

- PTR_S_RECEIPT 고정

(NSString) control

- Transaction Mode 를 설정 합니다. ([상수값 \(Defines\)](#) :: [Transaction Print](#) 참조)

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS ([상수값 \(Defines\)](#) :: [Result Code](#) 참조)

[Example]

```

UPOSPrinterController* _uposPrinterController = [ UPOSPrinterController new];
UPOSPrinters* printerList = [_uposPrinterController getRegisteredDevice];

/// Printer 초기화(open-claim-deviceEnable) 후
IBOutlet UIImageView * _imageView; // 리소스와 연결되어있는 뷰

[_uposPrinterController transactionPrint :PTR_S_RECEIPT
                        control : PTR_TR_TRANSACTION];

[_uposPrinterController printNormal :PTR_S_RECEIPT
                        data :@"|N|cAPrint CenterAlienedText 1 \r\n"];
[_uposPrinterController printNormal :PTR_S_RECEIPT
                        data :@"|N|cAPrint CenterAlienedText 2 \r\n"];
[_uposPrinterController printNormal :PTR_S_RECEIPT
                        data :@"|N|cAPrint CenterAlienedText 3 \r\n"];

// 버퍼에 있는 내용을 인쇄하고, transaction Mode 를 종료.
[_uposPrinterController transactionPrint :PTR_S_RECEIPT
                        control : PTR_TR_NORMAL];

// Printer 사용 정리 작업
_uposDeviceController.deviceEnabled = NO;
[_uposPrinterController releaseDevice];
[_uposPrinterController close];

```

[Availability]

SDK 1.0.0 and later

12-4 Available Delegate

12-4-1 StatusUpdateEvent

Printer 의 상태가 변경 됐을 때 발생하는 이벤트 입니다.

[See Also]

- [Common] Delegater Class Reference :: Available Delegate :: StatusUpdateEvent
- 상수값 (Defines) :: StatusUpdateEvent

[Declare]

- [\[Common\] Delegater Class Reference](#) :: [Available Delegate](#) :: [StatusUpdateEvent](#)
참조

[Parameters]

(NSNumber*) Status

- 변경 된 상태 값이 포함되어있습니다.
([상수값 \(Defines\)](#) :: [StatusUpdateEvent](#) 참조)

[Return Value]

- void

[Example]

```

-(void) StatusUpdateEvent: (NSNumber*) Status
{
    NSLog(@"!!!!!!!!!!!! StatusUpdateEvent : %ld !!!!!!!!!!", (long)Status.integerValue);
    NSString *message;
    switch([ Status integerValue)
    {
        case PTR_SUE_COVER_OPEN:
            message = [NSString stringWithFormat:@"[StatusUpdateEvent] Cover
Open"];
            break;
        case PTR_SUE_COVER_OK:
            message = [NSString stringWithFormat:@"[StatusUpdateEvent] Cover
OK"];
            break;
        case PTR_SUE_REC_EMPTY:
            message = [NSString stringWithFormat:@"[StatusUpdateEvent] Paper
Empty"];
            break;
        case PTR_SUE_REC_PAPEROK:
            message = [NSString stringWithFormat:@"[StatusUpdateEvent] Paper
OK"];
            break;
        case PTR_SUE_REC_NEAREMPTY:
            message = [NSString stringWithFormat:@"[StatusUpdateEvent] Paper
Near End"];
            break;
        case UPOS_SUE_POWER_OFF:
        case UPOS_SUE_POWER_OFF_OFFLINE:
        case UPOS_SUE_POWER_OFFLINE:
            message = [NSString stringWithFormat:@"[StatusUpdateEvent] Device off
or offLine"];
            break;
        case UPOS_SUE_POWER_ONLINE:
            message = [NSString stringWithFormat:@"[StatusUpdateEvent] Device
OnLine"];
            break;
        default:
            message = [NSString stringWithFormat:@"[StatusUpdateEvent]
UNKNOWN"];
    }

    NSLog(@"%@@", message)
}

```

[Availability]

SDK 1.0.0 and later

12-4-2 OutputCompleteEvent

각 Device 가 받은 output 요청을 완료했을 때 발생합니다.

[See Also]

- [Printer] Printer Controller Class Reference :: Available Method
- [Printer] Printer Controller Class Reference :: Available Method :: cutPaper
- [Printer] Printer Controller Class Reference :: Available Method :: markFeed
- [Printer] Printer Controller Class Reference :: Available Method :: printBarcode
- [Printer] Printer Controller Class Reference :: Available Method :: printBitmap (파일 인쇄)
- [Printer] Printer Controller Class Reference :: Available Method :: printBitmap (UIImage 인쇄)
- [Printer] Printer Controller Class Reference :: Available Method :: printNormal
- [Printer] Printer Controller Class Reference :: Available Method :: transactionPrint

[Declare]

- [\[Common\] Delegater Class Reference](#) :: [Available Delegate](#) :: [DataEvent](#) 참조

[Parameters]

(NSNumber*) Status

- 변경 된 상태 값이 포함되어있습니다.
(상수값 ([Defines](#)) :: [StatusUpdateEvent](#) 참조)

[Return Value]

- void

[Declare]

```
-(void)OutputCompleteEvent:(NSNumber*)OutputID;
```

[Availability]

SDK 1.0.0 and later

13. [CashDrawer] Cash Drawer Class Reference

13-1 Overview

UPOSCashDrawer Class 는 UPOSCDController 에서 제어 대상이 되는 장치의 정보를 가지는 객체 입니다.

[See Also]

- [Common] Device Class Reference

13-2 Available Properties

[See Also]

- [\[Common\] Device Class Reference](#) :: [Available Properties](#) 참조

13-2-1 selectedPrinterName

캐시드로어를 연결 하기로 지정한 프린터의 이름

[Declare]

```
@property (retain) NSString *selectedPrinterName;
```

[Availability]

SDK 1.0.0 and later

13-2-2 pinNumber

캐시드로어를 open 할 때 사용하는 Pin Number.

[Declare]

```
@property (retain) NSNumber *pinNumber;
```

[Availability]

SDK 1.0.0 and later

13-2-3 pinLevel

캐시드로어를 상태의 Open 상태를 감지하는 Level 을 선택 합니다.
캐시드로어의 상태가 반전되어 표시되는 경우 이 옵션을 변경 하십시오.

[Declare]

```
@property (retain) NSNumber *pinLevel;
```

[Availability]

SDK 1.0.0 and later

13-2-4 pulseOnTime

캐시드로어 Open 시 신호를 공급 할 시간을 결정합니다.

[Declare]

@property (retain) NSNumber *pinLevel;
--

[Availability]

SDK 1.0.0 and later

13-2-5 pulseOffTime

캐시드로어 Open 시 신호를 공급 후 휴지 시간을 결정합니다.

[Declare]

@property (retain) NSNumber *pinLevel;
--

[Availability]

SDK 1.0.0 and later

13-3 Available Method

- [\[Common\] Device Class Reference](#) :: [Available Method](#) 참조

14. [CashDrawer] Cash Drawer List Class Reference

Inherits from	UPOSDevices
Framework	libBixolonUPOS.a
[Availability]	iOS 7.0 and later
Class name	UPOSCashDrawers
Declared	UPOSDevices.h

14-1 Overview

UPOSCashDrawers Class 는 iDevice 에 저장되어있는 캐시드로어 리스트를 가지는 객체입니다. (UPOSCashDrawer 의 집합입니다.)

[See Also]

- Cash Drawer Class Reference

14-2 Available Properties

- 유효하지 않습니다.

14-3 Available Method

14-3-1 getList

저장되어있는 장치의 목록을 얻어옵니다.

[See Also]

- Device Class Reference (Common Device)
- Device Controller Class Reference (Common Controller)
- Cash Drawer Class Reference

[Declare]

- [Cash Drawer List Class Reference \(저장된 cash Drawer 목록\)](#) :: [getList](#) 참조

[Parameters]

없음.

[Return Value]

- NSMutableArray*:
저장되어있는 장치의 목록이 Array 형태로 저장 됩니다.
각 객체는 Device Class 형태로 저장 됩니다.

[Example]

```
UPOSCDController* _uposCDController = [UPOSCDController new];
UPOSCashDrawers* cdList = [_uposCDController getRegisteredDevice];

NSMutableArray* cdArray = [cdList getList];

if(cdArray == nil)
{
    NSLog(@"저장된 캐시드로우 리스트를 읽어올 수 없습니다.");
}
else
{
    NSLog(@"저장된 캐시드로우 리스트를 읽어왔습니다.");
}
```

[Availability]

SDK 1.0.0 and later

14-3-2 getDeviceIdentity

현재 장치의 Identity String 을 얻어옵니다.

[Declare]

- [Device List Class Reference \(저장된 장치 목록\)](#) :: [getDeviceIdentity](#) 참조

[Parameters]

없음.

[Return Value]

- NSString*:
현재 장치의 Identity 를 스트링 형태로 반환합니다.

[Example]

```
UPOSCDController* _uposCDController = [UPOSCDController new];
UPOSCashDrawers* cdList = [_ uposCDController getRegisteredDevice];

NSString* strIdentity = [cdList getDeviceIdentity];

if(cdArray == nil)
{
    NSLog(@"Identity 값을 읽어올 수 없습니다.");
}
else
{
    NSLog(@" UPOSCDController 의 Identity :%@", strIdentity);
}
```

[Availability]

SDK 1.0.0 and later

14-3-3 save

현재 장치 목록을 저장 합니다.

[See Also]

- Cash Drawer List Class Reference (저장된 cash Drawer 목록) :: removeDevice
- Cash Drawer List Class Reference (저장된 cash Drawer 목록) :: addDevice

[Declare]

- [Device List Class Reference \(저장된 장치 목록\)](#) :: [save](#) 참조

[Parameters]

없음.

[Return Value]

- BOOL:
성공시 YES.

[Example]

```
UPOSCDController* _uposCDController = [UPOSCDController new];
UPOSCashDrawers* cdList = [_uposCDController getRegisteredDevice];

UPOSCashDrawer* newDevice = [UPOSCashDrawer new];
newDevice.modelName = @"SRP-CDW";
newDevice.lidn = @"type1"; newDevice.selectedPrinterName = @"SRP-350plusIII";
newDevice.pinNumber = 2;
newDevice.pinLevel = 0;
newDevice.pulseOnTime = 100;
newDevice.pulseOffTime = 400;
[cdList addDevice:newDevice]; // Device 추가
[newDevice release];

if([printerList save])
{
    NSLog(@"List 를 저장 했습니다.");
}
else
{
    NSLog(@"List 를 저장 할 수 없습니다.");
}
```

[Availability]

SDK 1.0.0 and later

14-3-4 addDevice

현재 장치 목록에 장치를 추가 합니다.

본 메소드 사용 후 **save** 메소드를 이용하여 저장하지 않으면,
List 새로 고침 될 때 추가된 장치가 포함 되어있지 않을 수 있습니다.

[See Also]

- Cash Drawer List Class Reference (저장된 cash Drawer 목록) :: save
- Cash Drawer List Class Reference (저장된 cash Drawer 목록) :: removeDevice

[Declare]

- [Device List Class Reference \(저장된 장치 목록\)](#) :: [addDevice](#) 참조

[Parameters]

- (UPOSDevice*) device
추가 할 장치의 정보를 가지고 있는 객체.

[Return Value]

- BOOL:
성공시 YES.

[Example]

```
UPOSCDController* _uposCDController = [UPOSCDController new];
UPOSCashDrawers* cdList = [_uposCDController getRegisteredDevice];

UPOSCashDrawer* newDevice = [UPOSCashDrawer new];
newDevice.modelName = @"SRP-CDW";
newDevice.lidn = @"type1"; newDevice.selectedPrinterName = @"SRP-350plusIII";
newDevice.pinNumber = 2;
newDevice.pinLevel = 0;
newDevice.pulseOnTime = 100;
newDevice.pulseOffTime = 400;

if([cdList addDevice:newDevice]) // Device 추가
{
    NSLog(@"Device 를 추가를 성공했습니다");
}
else
{
    NSLog(@"Device 를 추가를 실패했습니다");
}

[newDevice release];
[cdList save];
```

[Availability]

SDK 1.0.0 and later

14-3-5 removeDevice

현재 장치 목록에 장치를 삭제 합니다.

본 메소드 사용 후 **save** 메소드를 이용하여 저장하지 않으면,
List 새로 고침 될 때 장치 삭제가 적용 되어있지 않을 수 있습니다.

[See Also]

- Cash Drawer List Class Reference (저장된 cash Drawer 목록) :: save
- Cash Drawer List Class Reference (저장된 cash Drawer 목록) :: addDevice

[Declare]

- [Device List Class Reference \(저장된 장치 목록\)](#) :: [removeDevice](#) 참조

[Parameters]

- (UPOSDevice*) device
추가 할 장치의 정보를 가지고 있는 객체.

[Return Value]

- BOOL:
성공시 YES.

[Example]

```
UPOSCDController* _uposCDController = [UPOSCDController new];
UPOSCashDrawers* cdList = [_uposCDController getRegisteredDevice];

UPOSCashDrawer* * willRemoveDevice = [[printerList getList] objectAtIndex:0];
if([cdList removeDevice:willRemoveDevice]) // Device 삭제
{
    NSLog(@"0번째 Device 를 목록에서 삭제 했습니다.");
}
else
{
    NSLog(@"0번째 Device 를 목록에서 삭제 할 수 없습니다.");
}

[newDevice release];
[cdList save];
```

[Availability]

SDK 1.0.0 and later

15. [CashDrawer] Cash Drawer Controller Class Reference

Inherits from	UPOSDeviceController
Framework	libBixolonUPOS.a
[Availability]	iOS 7.0 and later
Class name	UPOSCDController
Declared	UPOSCDController.h

15-1 Overview

UPOSCDController Class 는 본 SDK 에서 지원하는 장치의 공통된 기능의 제어를 위한 메인 객체 입니다.

15-2 Available Properties

15-2-1 DrawerOpened

CashDrawer 상태를 나타냅니다.

YES 일 경우 Drawer 가 열려있음을 의미 합니다.

[Declare]

@property (readonly)	BOOL	DrawerOpened;
----------------------	------	---------------

[Availability]

SDK 1.0.0 and later

15-3 Available Method

15-3-1 open

캐시드로어 클래스 사용을 시작하며 메모리 할당 등의 초기화 작업이 포함되어있습니다.

Claim 이상의 Method 호출을 위해서 반드시 선행 되어야 합니다.

[See Also]

- 상수값 (Defines) :: OpenResult Code
- [Common] Device Controller Class Reference :: Available Properties :: OpenResult
- [Common] Device Controller Class Reference :: Available Properties :: Idn (Logical Device Name)
- [CashDrawer] Cash Drawer List Class Reference :: Available Method :: addDevice

[Declare]

- [Common] Device Controller Class Reference :: Available Method :: open

[Parameters]

(NSString*) logicalDeviceName

- Open 하려는 Device 의 모델명 또는 저장된 Device Name
[Idn \(Logical Device Name\)](#) 참조
- [CashDrawer] Cash Drawer List Class Reference :: Available Method :: addDevice
사용하려는 Device 는 [CashDrawer] Cash Drawer List Class Reference :: addDevice 메소드를 통해 미리 추가가 되어있어야 합니다.

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

```
UPOSCDController* _uposCDController = [UPOSCDController new];
UPOSCashDrawers* cdList = [_uposCDController getRegisteredDevice];

if(UPOS_SUCCESS == [_uposCDController open:@"type1"])
{
    if(UPOS_SUCCESS == [_uposCDController claim:3000])
    {
        _uposCDController.deviceEnabled = YES;
        // 캐시드로어 사용 가능
    }
}
```

[Availability]

SDK 1.0.0 and later

15-3-2 claim

Device 정보에 포함되어있는 Port 를 실제로 Open 하기 위해 시도 하며, 메모리 할당 초기화 등의 초기화 작업이 일부 포함되어있습니다.

Device 사용을 활성화 하기 위해서 반드시 선행 되어야 합니다.

[See Also]

- 상수값 (Defines) :: Result Code
- [Common] Device Controller Class Reference :: Available Method :: open
- [CashDrawer] Cash Drawer Controller Class Reference :: Available Method :: open
- [Common] Device Controller Class Reference :: Available Method :: claim
- [Common] Device Controller Class Reference :: Available Properties :: Claimed

[Declare]

- [Common] Device Controller Class Reference :: Available Method :: claim

[Parameters]

(NSInteger) timeout

- 이 파라미터에서 지정한 시간동안 Port open 을 시도합니다.

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

```
UPOSCDController* _uposCDController = [UPOSCDController new];
UPOSCashDrawers* cdList = [_ uposCDController getRegisteredDevice];

if(UPOS_SUCCESS == [_ uposCDController open:@"type1"])
{
    if(UPOS_SUCCESS == [_ uposCDController claim:3000])
    {
        _ uposCDController.deviceEnabled = YES;
        // 캐시드로어 사용 가능
    }
}
```

[Availability]

SDK 1.0.0 and later

15-3-3 releaseDevice

Claim 된 Device의 Port 사용을 물리적으로 종료 합니다. 메모리 해제 등의 작업이 수행 될 수 있습니다.

[See Also]

- 상수값 (Defines) :: Result Code
- [Common] Device Controller Class Reference :: Available Method :: open
- [CashDrawer] Cash Drawer Controller Class Reference :: Available Method :: open
- [Common] Device Controller Class Reference :: Available Method :: claim
- [CashDrawer] Cash Drawer Controller Class Reference :: Available Method :: claim
- [Common] Device Controller Class Reference::Available Properties::DeviceEnabled
- [Common] Device Controller Class Reference :: Available Properties :: Claimed
- [CashDrawer]CashDrawerControllerClassReference :: AvailableMethod :: releaseDevice

[Declare]

- [Common] Device Controller Class Reference :: Available Method :: releaseDevice

[Parameters]

(NSString*) logicalDeviceName

- Open 하려는 Device 의 모델명 또는 저장된 Device Name
[Idn \(Logical Device Name\)](#) 참조

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

```
UPOSCDController* _uposCDController = [UPOSCDController new];
UPOSCashDrawers* cdList = [_ uposCDController getRegisteredDevice];

/// CashDrawer 사용 후

// CashDrawer 사용 정리 작업
_ uposCDController deviceEnabled = NO;
[_ uposCDController releaseDevice];
[_ uposCDController close];
```

[Availability]

SDK 1.0.0 and later

15-3-4 close

Open 된 Device의 사용을 종료 합니다.
메모리 해제 등의 작업이 일부 수행 될 수 있습니다.

[See Also]

- 상수값 (Defines) :: Result Code
- [Common] Device Controller Class Reference :: Available Method :: open
- [CashDrawer] Cash Drawer Controller Class Reference :: Available Method :: open
- [Common] Device Controller Class Reference :: Available Method :: claim
- [CashDrawer] Cash Drawer Controller Class Reference :: Available Method :: claim
- [Common] Device Controller Class Reference::Available Properties::DeviceEnabled
- [Common] Device Controller Class Reference :: Available Properties :: Claimed
- [Common] Device Controller Class Reference :: Available Method :: close

[Declare]

- [Common] Device Controller Class Reference :: Available Method :: close

[Parameters]

(NSInteger) timeout

- 이 파라미터에서 지정한 시간동안 Port open 을 시도합니다.

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

```
UPOSCDController* _uposCDController = [UPOSCDController new];
UPOSCashDrawers* cdList = [_ uposCDController getRegisteredDevice];

/// CashDrawer 사용 후

// CashDrawer 사용 정리 작업
_ uposCDController.deviceEnabled = NO;
[_ uposCDController releaseDevice];
[_ uposCDController close];
```

[Availability]

SDK 1.0.0 and later

15-3-5 OpenDrawer

캐시드로어를 open 합니다.

[See Also]

- 상수값 (Defines) :: Result Code
- [Common] Device Controller Class Reference :: Available Method :: open
- [CashDrawer] Cash Drawer Controller Class Reference :: Available Method:: open
- [Common] Device Controller Class Reference :: Available Method :: claim
- [CashDrawer] Cash Drawer Controller Class Reference :: Available Method :: claim
- [Common] Device Controller Class Reference::Available Properties::DeviceEnabled
- [Common] Device Controller Class Reference :: Available Properties :: Claimed

[Declare]

```
-(NSInteger) OpenDrawer;
```

[Parameters]

없음.

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

```
UPOSCDController* _uposCDController = [UPOSCDController new];
UPOSCashDrawers* cdList = [_ uposCDController getRegisteredDevice];

/// CashDrawer 초기화(open-claim-deviceEnable) 후
[_uposCDController openDrawer];

// 캐시드로어 사용 정리 작업
_uposCDController.deviceEnabled = NO;
[_ uposCDController releaseDevice];
[_ uposCDController close];
```

[Availability]

SDK 1.0.0 and later

15-4 Available Delegate

15-4-1 StatusUpdateEvent

CashDrawer 의 상태가 변경 됐을 때 발생하는 이벤트 입니다.

[See Also]

- [Common] Delegater Class Reference :: Available Delegate :: StatusUpdateEvent
- 상수값 (Defines) :: StatusUpdateEvent

[Declare]

- [\[Common\] Delegater Class Reference](#) :: [Available Delegate](#) :: [StatusUpdateEvent](#)
참조

[Parameters]

(NSNumber*) Status

- 변경 된 상태 값이 포함되어있습니다.
([상수값 \(Defines\)](#) :: [StatusUpdateEvent](#) 참조)

[Return Value]

- void

[Example]

```

-(void) StatusUpdateEvent: (NSNumber*) Status
{
    NSLog(@"!!!!!!!!!!!! StatusUpdateEvent : %ld !!!!!!!!!!", (long)Status.integerValue);
    NSString *message;
    switch([ Status integerValue])
    {
        case CASH_SUE_DRAWEROPEN:
            message = [NSString stringWithFormat:@"[StatusUpdateEvent] Cash
Drawer Opened"];
            break;
        case CASH_SUE_DRAWERCLOSED:
            message = [NSString stringWithFormat:@"[StatusUpdateEvent] Cash
Drawer Closed"];
            break;
        case UPOS_SUE_POWER_OFF:
        case UPOS_SUE_POWER_OFF_OFFLINE:
        case UPOS_SUE_POWER_OFFLINE:
            message = [NSString stringWithFormat:@"[StatusUpdateEvent] Device off
or offLine"];
            break;
        case UPOS_SUE_POWER_ONLINE:
            message = [NSString stringWithFormat:@"[StatusUpdateEvent] Device
OnLine"];
            break;
        default:
            message = [NSString stringWithFormat:@"[StatusUpdateEvent]
UNKNOWN"];
    }

    NSLog(@"%@%@", message)
}

```

[Availability]

SDK 1.0.0 and later

16. [MSR] MSR Class Reference

16-1 Overview

UPOSMSR Class 는 UPOSMSRController 에서 제어 대상이 되는 장치의 정보를 가지는 객체 입니다.

16-2 Available Properties

- [\[Common\] Device Class Reference](#) :: [Available Properties](#) 참조

16-3 Available Method

- [\[Common\] Device Class Reference](#) :: [Available Method](#) 참조

17. [MSR] MSR List Class Reference (저장된 MSR 목록)

Inherits from	UPOSDevices
Framework	libBixolonUPOS.a
[Availability]	iOS 7.0 and later
Class name	UPOSMSRs
Declared	UPOSDevices.h

17-1 Overview

UPOSMSRs Class 는 iDevice 에 저장되어있는 MSR 리스트를 가지는 객체 입니다.
(UPOSMSR 의 집합입니다.)

[See Also]

- MSR Class Reference

17-2 Available Properties

- 유효하지 않습니다.

17-3 Available Method

17-3-1 getList

저장되어있는 장치의 목록을 얻어옵니다.

[See Also]

- Device Class Reference (Common Device)
- Device Controller Class Reference (Common Controller)
- MSR Class Reference

[Declare]

- [MSR List Class Reference \(저장된 MSR 목록\)](#) :: [getList](#) 참조

[Parameters]

없음.

[Return Value]

- NSMutableArray*:
저장되어있는 장치의 목록이 Array 형태로 저장 됩니다.
각 객체는 Device Class 형태로 저장 됩니다.

[Example]

```
UPOSMSRController* _uposMSRController = [UPOSMSRController new];
UPOSMSRs* msrList = [_uposMSRController getRegisteredDevice];

NSMutableArray* msrArray = [ msrList getList;

if( msrArray == nil)
{
    NSLog(@"저장된 MSR 리스트를 읽어올 수 없습니다.");
}
else
{
    NSLog(@"저장된 MSR 리스트를 읽어왔습니다.");
}
```

[Availability]

SDK 1.0.0 and later

17-3-2 getDeviceIdentity

현재 장치의 Identity String 을 얻어옵니다.

[Declare]

- [Device List Class Reference \(저장된 장치 목록\)](#) :: [getDeviceIdentity](#) 참조

[Parameters]

없음.

[Return Value]

- NSString*:
현재 장치의 Identity 를 스트링 형태로 반환합니다.

[Example]

```
UPOSMSRController* _uposMSRController = [UPOSMSRController new];
UPOSMSRs* msrList = [_uposMSRController getRegisteredDevice];

NSString* strIdentity = [msrList getDeviceIdentity];

if( strIdentity == nil)
{
    NSLog(@"Identity 값을 읽어올 수 없습니다.");
}
else
{
    NSLog(@" UPOSMSRController 의 Identity :%@", strIdentity);
}
```

[Availability]

SDK 1.0.0 and later

17-3-3 save

현재 장치 목록을 저장 합니다.

[See Also]

- MSR List Class Reference (저장된 MSR 목록) :: removeDevice
- MSR List Class Reference (저장된 MSR 목록) :: addDevice
- Device List Class Reference (저장된 장치 목록) :: save

[Declare]

- [Device List Class Reference \(저장된 장치 목록\)](#) :: [save](#) 참조

[Parameters]

없음.

[Return Value]

- BOOL:
성공시 YES.

[Example]

```
UPOSMSRController* _uposMSRController = [UPOSMSRController new];
UPOSMSRs * msrList = [_uposMSRController getRegisteredDevice];

UPOSMSR* newDevice = [UPOSMSR new];
newDevice.modelName = @"SRP-350plusIII";
newDevice.lidn = @"type1";
newDevice.interfaceType = @"4"; // Bluetooth
newDevice.address = @"74:F0:E1:XX:XX:XX";
newDevice.port = @"";

[printerList addDevice:newDevice]; // Device 추가
[newDevice release];

if([ msrList save])
{
    NSLog(@"List 를 저장 했습니다.");
}
else
{
    NSLog(@"List 를 저장 할 수 없습니다.");
}
```

[Availability]

SDK 1.0.0 and later

17-3-4 addDevice

현재 장치 목록에 장치를 추가 합니다.

본 메소드 사용 후 **save** 메소드를 이용하여 저장하지 않으면,
List 새로 고침 될 때 추가된 장치가 포함 되어있지 않을 수 있습니다.

[See Also]

- MSR List Class Reference (저장된 MSR 목록) :: save
- MSR List Class Reference (저장된 MSR 목록) :: removeDevice
- Device List Class Reference (저장된 장치 목록) :: addDevice

[Declare]

- [Device List Class Reference \(저장된 장치 목록\)](#) :: [addDevice](#) 참조

[Parameters]

- (UPOSDevice*) device
추가 할 장치의 정보를 가지고 있는 객체.

[Return Value]

- BOOL:
성공시 YES.

[Example]

```
UPOSMSRController* _uposMSRController = [UPOSMSRController new];
UPOSMSRs * msrList = [_uposMSRController getRegisteredDevice];

UPOSMSR* newDevice = [UPOSMSR new];
newDevice.modelName = @"SRP-350plusIII";
newDevice.idn = @"type1";
newDevice.interfaceType = @"4"; // Bluetooth
newDevice.address = @"74:F0:E1:XX:XX:XX";
newDevice.port = @"";
[printerList addDevice:newDevice]; // Device 추가
[newDevice release];

if([ msrList addDevice:newDevice]) // Device 추가
{
    NSLog(@"Device 를 추가를 성공했습니다");
}
else
{
    NSLog(@"Device 를 추가를 실패했습니다");
}

[newDevice release];
[msrList save];
```

[Availability]

SDK 1.0.0 and later

17-3-5 removeDevice

현재 장치 목록에 장치를 삭제 합니다.

본 메소드 사용 후 **save** 메소드를 이용하여 저장하지 않으면,
List 새로 고침 될 때 장치 삭제가 적용 되어있지 않을 수 있습니다.

[See Also]

- MSR List Class Reference (저장된 MSR 목록) :: save
- MSR List Class Reference (저장된 MSR 목록) :: addDevice
- Device List Class Reference (저장된 장치 목록) :: removeDevice

[Declare]

- [Device List Class Reference \(저장된 장치 목록\)](#) :: [removeDevice](#) 참조

[Parameters]

- (UPOSDevice*) device
추가 할 장치의 정보를 가지고 있는 객체.

[Return Value]

- BOOL:
성공시 YES.

[Example]

```
UPOSMSRController* _uposMSRController = [UPOSMSRController new];
UPOSMSRs * msrList = [_uposMSRController getRegisteredDevice];

UPOSMSR * willRemoveDevice = [[printerList getList] objectAtIndex:0];
if([ msrList removeDevice:willRemoveDevice]) // Device 삭제
{
    NSLog(@"0번째 Device 를 목록에서 삭제 했습니다.");
}
else
{
    NSLog(@"0번째 Device 를 목록에서 삭제 할 수 없습니다.");
}

[newDevice release];
[ msrList save];
```

[Availability]

SDK 1.0.0 and later

18. [MSR] MSR Controller Class Reference

Inherits from	NSObject
Framework	libBixolonUPOS.a
[Availability]	iOS 7.0 and later
Class name	UPOSMSRController
Declared	UPOSMSRController.h

18-1 Overview

UPOSMSRController Class 는 본 SDK 에서 지원하는 장치의 공통된 기능의 제어를 위한 메인 객체 입니다.

18-2 Available Properties

18-2-1 Track1Data

가장 최근에 읽은 MSR Card의 Track1 Data 를 저장 하고 있습니다.

[Declare]

@property (strong, readonly)	NSString*	Track1Data;
------------------------------	-----------	-------------

[Availability]

SDK 1.0.0 and later

18-2-2 Track2Data

가장 최근에 읽은 MSR Card의 Track2 Data 를 저장 하고 있습니다.

[Declare]

@property (strong, readonly)	NSString*	Track2Data;
------------------------------	-----------	-------------

[Availability]

SDK 1.0.0 and later

18-2-3 Track3Data

가장 최근에 읽은 MSR Card의 Track3 Data 를 저장 하고 있습니다.

[Declare]

@property (strong, readonly)	NSString*	Track3Data;
------------------------------	-----------	-------------

[Availability]

SDK 1.0.0 and later

18-3 Available Method

18-3-1 open

MSR 클래스 사용을 시작하며 메모리 할당 등의 초기화 작업이 포함되어있습니다.
Claim 이상의 Method 호출을 위해서 반드시 선행 되어야 합니다.

[See Also]

- 상수값 (Defines) :: OpenResult Code
- [Common] Device Controller Class Reference :: Available Properties :: OpenResult
- [Common] Device Controller Class Reference :: Available Properties ::
Idn (Logical Device Name)
- [MSR] MSR Controller Class Reference :: Available Method :: addDevice

[Declare]

- [Common] Device Controller Class Reference :: Available Method :: open

[Parameters]

(NSString*) logicalDeviceName

- Open 하려는 Device 의 모델명 또는 저장된 Device Name
[Idn \(Logical Device Name\)](#) 참조
- 사용하려는 Device 는 [MSR] MSR Controller Class Reference ::
Available Method :: addDevice 메소드를 통해 미리 추가가 되어있어야 합니다.

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

```
UPOMSRController* _uposMSRController = [UPOSMSRController new];
UPOSMSRs* msrList = [_uposMSRController getRegisteredDevice];

if(UPOS_SUCCESS == _uposMSRController open:@"type1"]
{
    if(UPOS_SUCCESS == [_uposMSRController claim:3000])
    {
        _uposMSRController.deviceEnabled = YES;
        // MSR 사용 가능
    }
}
```

[Availability]

SDK 1.0.0 and later

18-3-2 claim

Device 정보에 포함되어있는 Port 를 실제로 Open 하기 위해 시도 하며, 메모리 할당 초기화 등의 초기화 작업이 일부 포함되어있습니다.

Device 사용을 활성화 하기 위해서 반드시 선행 되어야 합니다.

[See Also]

- 상수값 (Defines) :: Result Code
- [Common] Device Controller Class Reference :: Available Method :: open
- [MSR] MSR Controller Class Reference :: Available Method :: open
- [Common] Device Controller Class Reference :: Available Method :: claim
- [Common] Device Controller Class Reference :: Available Properties :: Claimed

[Declare]

- [Common] Device Controller Class Reference :: Available Method :: claim

[Parameters]

(NSInteger) timeout

- 이 파라미터에서 지정한 시간동안 Port open 을 시도합니다.

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

```
UPOMSRController* _uposMSRController = [UPOSMSRController new];
UPOSMSRs* msrList = [_uposMSRController getRegisteredDevice];

if(UPOS_SUCCESS == [_uposMSRController open:@"type1"])
{
    if(UPOS_SUCCESS == [_uposMSRController claim:3000])
    {
        _uposMSRController.deviceEnabled = YES;
        // MSR 사용 가능
    }
}
```

[Availability]

SDK 1.0.0 and later

18-3-3 releaseDevice

Claim 된 Device의 Port 사용을 물리적으로 종료 합니다. 메모리 해제 등의 작업이 수행 될 수 있습니다.

[See Also]

- 상수값 (Defines) :: Result Code
- [Common] Device Controller Class Reference :: Available Method :: open
- [MSR] MSR Controller Class Reference :: Available Method :: open
- [Common] Device Controller Class Reference :: Available Method :: claim
- [MSR] MSR Controller Class Reference :: Available Method :: claim
- [Common] Device Controller Class Reference::Available Properties::DeviceEnabled
- [Common] Device Controller Class Reference :: Available Properties :: Claimed
- [MSR] MSR Controller Class Reference::AvailableMethod::releaseDevice

[Declare]

- [Common] Device Controller Class Reference :: Available Method :: releaseDevice

[Parameters]

(NSString*) logicalDeviceName

- Open 하려는 Device 의 모델명 또는 저장된 Device Name
[Idn \(Logical Device Name\)](#) 참조

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

```
UPOMSRController* _uposMSRController = [UPOSMSRController new];
UPOSMSRs* msrList = [_uposMSRController getRegisteredDevice];

/// MSR 사용 후

// MSR 사용 정리 작업
_uposMSRController deviceEnabled = NO;
[_uposMSRController releaseDevice];
[_uposMSRController close];
```

[Availability]

SDK 1.0.0 and later

18-3-4 close

Open 된 Device의 사용을 종료 합니다.
메모리 해제 등의 작업이 일부 수행 될 수 있습니다.

[See Also]

- 상수값 (Defines) :: Result Code
- [Common] Device Controller Class Reference :: Available Method :: open
- [MSR] MSR Controller Class Reference :: Available Method :: open
- [Common] Device Controller Class Reference :: Available Method :: claim
- [MSR] MSR Controller Class Reference :: Available Method :: claim
- [Common] Device Controller Class Reference::Available Properties::DeviceEnabled
- [Common] Device Controller Class Reference :: Available Properties :: Claimed
- [Common] Device Controller Class Reference :: Available Method :: close

[Declare]

- [Common] Device Controller Class Reference :: Available Method :: close

[Parameters]

(NSInteger) timeout

- 이 파라미터에서 지정한 시간동안 Port open 을 시도합니다.

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

```
UPOMSRController* _uposMSRController = [UPOSMSRController new];
UPOSMSRs* msrList = [_uposMSRController getRegisteredDevice];

/// MSR 사용 후

// MSR 사용 정리 작업
_uposMSRController deviceEnabled = NO;
[_uposMSRController releaseDevice];
[_uposMSRController close];
```

[Availability]

SDK 1.0.0 and later

18-4 Available Delegate

18-4-1 DataEvent

MSR Reader 에 Card를 통과 시켜 MSR Data 가 발생했을 때 전달되는 delegate 입니다.

[See Also]

- [Common] Delegater Class Reference :: Available Delegate :: DataEvent

[Declare]

- [\[Common\] Delegater Class Reference](#) :: [Available Delegate](#) :: [DataEvent](#) 참조

[Parameters]

(NSNumber*) Status

- 인식 된 MSR Data 에 포함된 Track 정보가 전달 됩니다.

[Return Value]

- void

[Example]

```
- (void) DataEvent:(NSNumber*) Status
{
    NSLog(@"!!!!!!!!!!!! (MSR) Data Event : %ld !!!!!!!!!!", (long) Status
integerValue);

    [self updateStringEventHistory:[NSString stringWithFormat:@"=====
Data Event ====="];
    [self updateStringEventHistory:[NSString stringWithFormat:@"track 1: %@",
_uposMSRController.Track1Data]];
    [self updateStringEventHistory:[NSString stringWithFormat:@"track 2: %@",
_uposMSRController.Track2Data]];
    [self updateStringEventHistory:[NSString stringWithFormat:@"track
3: %@\r\n\r\n\r\n", _uposMSRController.Track3Data]];
}
```

[Availability]

SDK 1.0.0 and later

19. [SCR] SCR Class Reference

Inherits from	NSObject
Framework	libBixolonUPOS.a
[Availability]	iOS 7.0 and later
Class name	UPOSScr
Declared	UPOSDevices.h

19-1 Overview

UPOSSCR Class 는 UPOSSCRController 에서 제어 대상이 되는 장치의 정보를 가지는 객체 입니다.

[See Also]

- [Common] Device Class Reference

19-2 Available Properties

- [\[Common\] Device Class Reference](#) :: [Available Properties](#) 참조

19-3 Available Method

- [\[Common\] Device Class Reference](#) :: [Available Method](#) 참조

20. [SCR] SCR List Class Reference(저장된 SCR 목록)

Inherits from	UPOSDevices
Framework	libBixolonUPOS.a
[Availability]	iOS 7.0 and later
Class name	UPOSSCRs
Declared	UPOSDevices.h

20-1 Overview

UPOSSCRs Class 는 iDevice 에 저장되어있는 SCR Device의 리스트를 가지는 객체입니다.

UPOSSCR 의 집합입니다.

[See Also]

- [SCR Controller Class Reference](#)

20-2 Available Properties

- 유효하지 않습니다.

20-3 Available Method

20-3-1 getList

저장되어있는 장치의 목록을 얻어옵니다.

[See Also]

- Device Class Reference (Common Device)
- Device Controller Class Reference (Common Controller)
- [SCR Class Reference](#)

[Declare]

- [Device List Class Reference \(저장된 장치 목록\)](#) :: [getList](#) 참조

[Parameters]

없음.

[Return Value]

- NSMutableArray*:
저장되어있는 장치의 목록이 Array 형태로 저장 됩니다.
각 객체는 Device Class 형태로 저장 됩니다.

[Example]

```
UPOSSCRController* _uposSCRController = [UPOSSCRController new];
UPOSSCRs* scrList = [_uposSCRController getRegisteredDevice];

NSMutableArray* scrArray = [scrList getList];

if( scrArray == nil)
{
    NSLog(@"저장된 SCR 리스트를 읽어올 수 없습니다.");
}
else
{
    NSLog(@"저장된 SCR 리스트를 읽어왔습니다.");
}
```

[Availability]

SDK 1.0.4 and later

20-3-2 getDeviceIdentity

현재 장치의 Identity String 을 얻어옵니다.

[Declare]

- [Device List Class Reference \(저장된 장치 목록\)](#) :: [getDeviceIdentity](#) 참조

[Parameters]

없음.

[Return Value]

- NSString*:
현재 장치의 Identity 를 스트링 형태로 반환합니다.

[Example]

```
UPOSSCRController* _uposSCRController = [UPOSSCRController new];
UPOSSCRs* msrList = [_uposSCRController getRegisteredDevice];

NSString* strIdentity = [msrList getDeviceIdentity];

if( strIdentity == nil)
{
    NSLog(@"Identity 값을 읽어올 수 없습니다.");
}
else
{
    NSLog(@"UPOSMSRController 의 Identity :%@", strIdentity);
}
```

[Availability]

SDK 1.0.4 and later

20-3-3 save

현재 장치 목록을 저장 합니다.

[See Also]

- [SCR List Class Reference \(저장된 SCR 목록\) :: Available Method :: addDevice](#)
- [SCR List Class Reference \(저장된 SCR 목록\) :: Available Method :: removeDevice](#)
- Device List Class Reference (저장된 장치 목록) :: Available Method :: save

[Declare]

- [Device List Class Reference \(저장된 장치 목록\) :: Available Method :: save](#) 참조

[Parameters]

없음.

[Return Value]

- BOOL:
성공시 YES.

[Example]

```
UPOSSCRController* _uposSCRController = [UPOSSCRController new];
UPOSSCRs * scrList = [_uposSCRController getRegisteredDevice];

UPOSSCR* newDevice = [UPOSSCR new];
newDevice.modelName = @"SPP-R210S";
newDevice.idn = @"type1";
newDevice.interfaceType = @"4"; // Bluetooth
newDevice.address = @"74:F0:E1:XX:XX:XX";
newDevice.port = @"";

[scrList addDevice:newDevice]; // Device 추가
[newDevice release];

if([ scrList save])
{
    NSLog(@"List 를 저장 했습니다.");
}
else
{
    NSLog(@"List 를 저장 할 수 없습니다.");
}
```

[Availability]

SDK 1.0.0 and later

20-3-4 addDevice

현재 장치 목록에 장치를 추가 합니다.

본 메소드 사용 후 **save** 메소드를 이용하여 저장하지 않으면,
List 새로 고침 될 때 추가된 장치가 포함 되어있지 않을 수 있습니다.

[See Also]

- Device List Class Reference (저장된 장치 목록) :: Available Method :: save
- [SCR List Class Reference \(저장된 SCR 목록\) :: Available Method :: removeDevice](#)
- Device List Class Reference (저장된 장치 목록) :: Available Method :: addDevice

[Declare]

- [Device List Class Reference \(저장된 장치 목록\)](#) :: Available Method :: [addDevice](#)
참조

[Parameters]

- (UPOSSCR*) device
추가 할 장치의 정보를 가지고 있는 객체.

[Return Value]

- BOOL:
성공시 YES.

[Example]

```
UPOSSCRController* _uposSCRController = [UPOSSCRController new];
UPOSSCRs * scrList = [_uposSCRController getRegisteredDevice];

UPOSSCR* newDevice = [UPOSSCR new];
newDevice.modelName = @"SPP-R210";
newDevice.Idn = @"type1";
newDevice.interfaceType = @"4"; // Bluetooth
newDevice.address = @"74:F0:E1:XX:XX:XX";
newDevice.port = @"";
[printerList addDevice:newDevice]; // Printer 추가
[newDevice release];

if([ scrList addDevice:newDevice]) // SCR 추가
{
    NSLog(@"Device 를 추가를 성공했습니다");
}
else
{
    NSLog(@"Device 를 추가를 실패했습니다");
}

[newDevice release];
[scrList save];
```

[Availability]

SDK 1.0.0 and later

20-3-5 removeDevice

현재 장치 목록에 장치를 삭제 합니다.

본 메소드 사용 후 **save** 메소드를 이용하여 저장하지 않으면,
List 새로 고침 될 때 장치 삭제가 적용 되어있지 않을 수 있습니다.

[See Also]

- [SCR List Class Reference \(저장된 SCR 목록\) :: Available Method :: save](#)
- [SCR List Class Reference \(저장된 SCR 목록\) :: Available Method :: addDevice](#)
- Device List Class Reference (저장된 장치 목록) :: Available Method :: removeDevice

[Declare]

- [Device List Class Reference \(저장된 장치 목록\) :: Available Method :: removeDevice](#) 참조

[Parameters]

- (UPOSDDevice*) device
추가 할 장치의 정보를 가지고 있는 객체.

[Return Value]

- BOOL:
성공시 YES.

[Example]

```
UPOSSCRController* _uposSCRController = [UPOSSCRController new];
UPOSSCRs * scrList = [_uposSCRController getRegisteredDevice];

UPOSSCR * willRemoveDevice = [[scrList getList] objectAtIndex:0];
if([scrList removeDevice:willRemoveDevice]) //remove from scrList
{
    NSLog(@"0번째 Device 를 목록에서 삭제 했습니다.");
}
else
{
    NSLog(@"0번째 Device 를 목록에서 삭제 할 수 없습니다.");
}

[newDevice release];
[scrList save];
```

[Availability]

SDK 1.0.4 and later

21. [SCR] SCR Controller Class Reference

Inherits from	NSObject
Framework	libBixolonUPOS.a
[Availability]	iOS 7.0 and later
Class name	UPOSSCRController
Declared	UPOSSCRController.h

21-1 Overview

UPOSSCRController Class 는 본 SDK 에서 지원하는 SCR 장치의 기능 제어를 위한 메인 객체 입니다.

21-2 Available Properties

- 유효하지 않습니다.

21-3 Available Method

21-3-1 open

SCR 클래스 사용을 시작하며 메모리 할당 등의 초기화 작업이 포함되어있습니다.
Claim 이상의 Method 호출을 위해서 반드시 선행 되어야 합니다.

[See Also]

- 상수값 (Defines) :: OpenResult Code
- [Common] Device Controller Class Reference :: Available Properties :: OpenResult
- [Common] Device Controller Class Reference :: Available Properties ::
Idn (Logical Device Name)
- [SCR List Class Reference \(저장된 SCR 목록\) :: Available Method :: addDevice](#)

[Declare]

- [Common] Device Controller Class Reference :: Available Method :: open

[Parameters]

(NSString*) logicalDeviceName

- Open 하려는 Device 의 모델명 또는 저장된 Device Name
[Idn \(Logical Device Name\)](#) 참조
- 사용하려는 Device 는 [MSR] MSR Controller Class Reference ::
Available Method :: addDevice 메소드를 통해 미리 추가가 되어있어야 합니다.

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

```
UPOSCRCController* _uposSCRController = [UPOSSCRController new];
UPOSSCRs* msrList = [_uposSCRController getRegisteredDevice];

if(UPOS_SUCCESS == _uposSCRController open:@"type1")
{
    if(UPOS_SUCCESS == [_uposSCRController claim:3000])
    {
        _uposSCRController.deviceEnabled = YES;
        // SCR 사용 가능
    }
}
```

[Availability]

SDK 1.0.4 and later

21-3-2 claim

Device 정보에 포함되어있는 Port 를 실제로 Open 하기 위해 시도 하며, 메모리 할당 초기화 등의 초기화 작업이 일부 포함되어있습니다.

Device 사용을 활성화 하기 위해서 반드시 선행 되어야 합니다.

[See Also]

- 상수값 (Defines) :: Result Code
- [Common] Device Controller Class Reference :: Available Method :: open
- [\[SCR\] SCR Controller Class Reference :: Available Method :: open](#)
- [Common] Device Controller Class Reference :: Available Method :: claim
- [Common] Device Controller Class Reference :: Available Properties :: Claimed

[Declare]

- [Common] Device Controller Class Reference :: Available Method :: claim

[Parameters]

(NSInteger) timeout

- 이 파라미터에서 지정한 시간동안 Port open 을 시도합니다.

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

```
UPOSCRCController* _uposSCRCController = [UPOSSCRCController new];
UPOSSCRs* scrList = [_uposSCRCController getRegisteredDevice];

if(UPOS_SUCCESS == [_uposSCRCController open:@"type1"])
{
    if(UPOS_SUCCESS == [_uposSCRCController claim:3000])
    {
        _uposSCRCController.deviceEnabled = YES;
        // SCR 사용 가능
    }
}
```

[Availability]

SDK 1.0.4 and later

21-3-3 releaseDevice

Claim 된 Device의 Port 사용을 물리적으로 종료 합니다. 메모리 해제 등의 작업이 수행 될 수 있습니다.

[See Also]

- 상수값 (Defines) :: Result Code
- [Common] Device Controller Class Reference :: Available Method :: open
- [\[SCR\] SCR Controller Class Reference :: Available Method :: open](#)
- [Common] Device Controller Class Reference :: Available Method :: claim
- [\[SCR\] SCR Controller Class Reference :: Available Method :: claim](#)
- [Common] Device Controller Class Reference::Available Properties::DeviceEnabled
- [Common] Device Controller Class Reference :: Available Properties :: Claimed

[Declare]

- [Common] Device Controller Class Reference :: Available Method :: releaseDevice

[Parameters]

(NSString*) logicalDeviceName

- Open 하려는 Device 의 모델명 또는 저장된 Device Name
[Idn \(Logical Device Name\)](#) 참조

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

```
UPOSSCRController* _uposSCRController = [UPOSSCRController new];
UPOSSCRs* msrList = [_ uposMSRController getRegisteredDevice];

/// SCR 사용 후

// SCR 사용 정리 작업
_uposSCRController deviceEnabled = NO;
[_uposSCRController releaseDevice];
[_uposSCRController close];
```

[Availability]

SDK 1.0.4 and later

21-3-4 close

Open 된 Device의 사용을 종료 합니다.
메모리 해제 등의 작업이 일부 수행 될 수 있습니다.

[See Also]

- 상수값 (Defines) :: Result Code
- [Common] Device Controller Class Reference :: Available Method :: open
- [\[SCR\] SCR Controller Class Reference :: Available Method :: open](#)
- [Common] Device Controller Class Reference :: Available Method :: claim
- [\[SCR\] SCR Controller Class Reference :: Available Method :: claim](#)
- [Common] Device Controller Class Reference::Available Properties::DeviceEnabled
- [Common] Device Controller Class Reference :: Available Properties :: Claimed
- [Common] Device Controller Class Reference :: Available Method :: close

[Declare]

- [Common] Device Controller Class Reference :: Available Method :: close

[Parameters]

(NSInteger) timeout

- 이 파라미터에서 지정한 시간동안 Port open 을 시도합니다.

[Return Value]

NSInteger

- 성공시 UPOS_SUCCESS

[Example]

```
UPOSSCRController* _uposSCRController = [UPOSSCRController new];
UPOSSCRs* msrList = [_ uposSCRController getRegisteredDevice];

/// SCR 사용 후

// SCR 사용 정리 작업
_uposSCRController deviceEnabled = NO;
[_ uposSCRController releaseDevice];
[_ uposSCRController close];
```

[Availability]

SDK 1.0.4 and later

21-4 Available Delegate

21-4-1 DataEvent

SCR 에 Card를 통과 시켜 SCR Data 가 발생했을 때 전달되는 delegate 입니다.

[See Also]

- [Common] Delegater Class Reference :: Available Delegate :: DataEvent

[Declare]

- [\[Common\] Delegater Class Reference](#) :: [Available Delegate](#) :: [DataEvent](#) 참조

[Parameters]

(NSNumber*) Status

- 'SC_READ_DATA' 로 고정되어있습니다.

[Return Value]

- void

[Example]

```
- (void) DataEvent:(NSNumber*) Status
{
    NSLog(@"!!!!!!!!!!!!!! (SCR) Data Event : %ld !!!!!!!!!!!!!", (long) Status
integerValue);

    NSData* data = nil;
    [_uposSCRController readData:SC_READ_DATA data:&data];
    NSLog(@"read SCR Data : %@", data);
}
```

[Availability]

SDK 1.0.4 and later

22. [Common] Delegater Class Reference

Inherits from	NSObject
Framework	libBixolonUPOS.a
[Availability]	iOS 7.0 and later
Class name	UPOSDeviceControlDelegate
Declared	UPOSDeviceControlDelegate.h

22-1 Overview

UPOSDeviceControlDelegate Class 는 본 SDK 에서 전달하는 이벤트를 전달 하는 메인 객체 입니다.

22-2 Available Properties

- 유효하지 않습니다.

22-3 Available Delegate

22-3-1 DataEvent

각 Device 로부터 Data 가 발생했을 때 전달되는 delegate 입니다.

[Declare]

```
-(void)DataEvent:(NSNumber*) Status;
```

[Availability]

SDK 1.0.0 and later

22-3-2 StatusUpdateEvent

각 Device 의 상태가 변경 됐을 때 발생하는 이벤트 입니다.

[Declare]

```
-(void)StatusUpdateEvent:(NSNumber*) Status;
```

[Availability]

SDK 1.0.0 and later

22-3-3 OutputCompleteEvent

각 Device 가 받은 output 요청을 완료했을 때 발생합니다.

[Declare]

```
-(void)OutputCompleteEvent:(NSNumber*)OutputID;
```

[Availability]

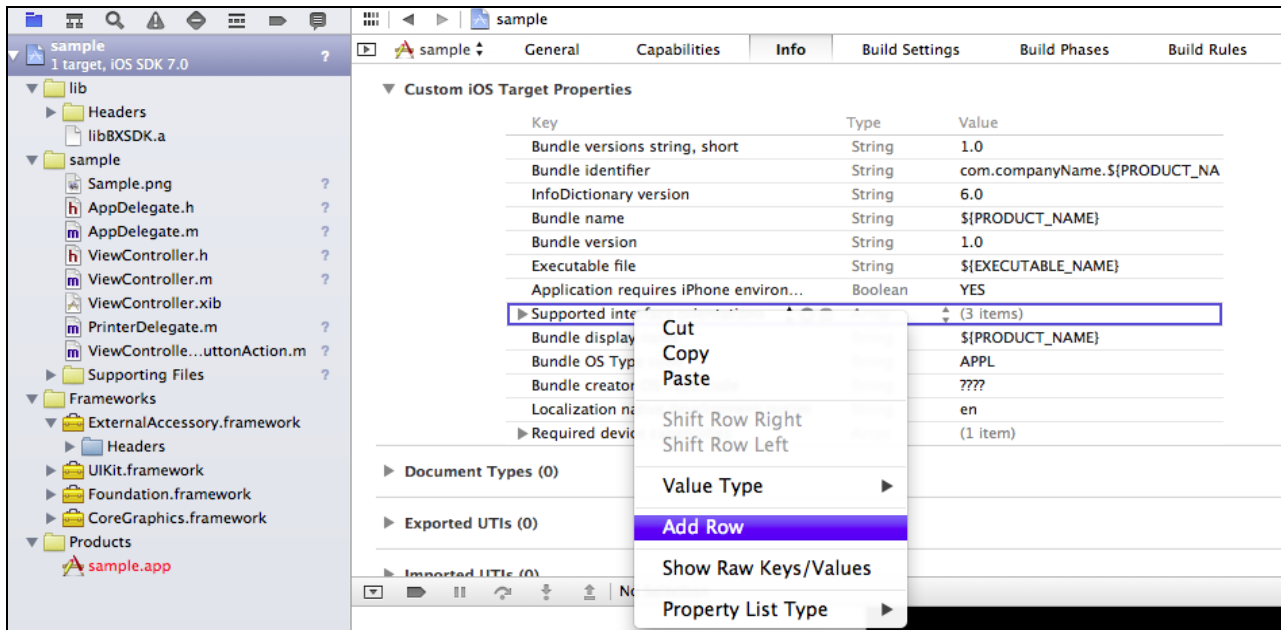
SDK 1.0.0 and later

23. 샘플 프로그램

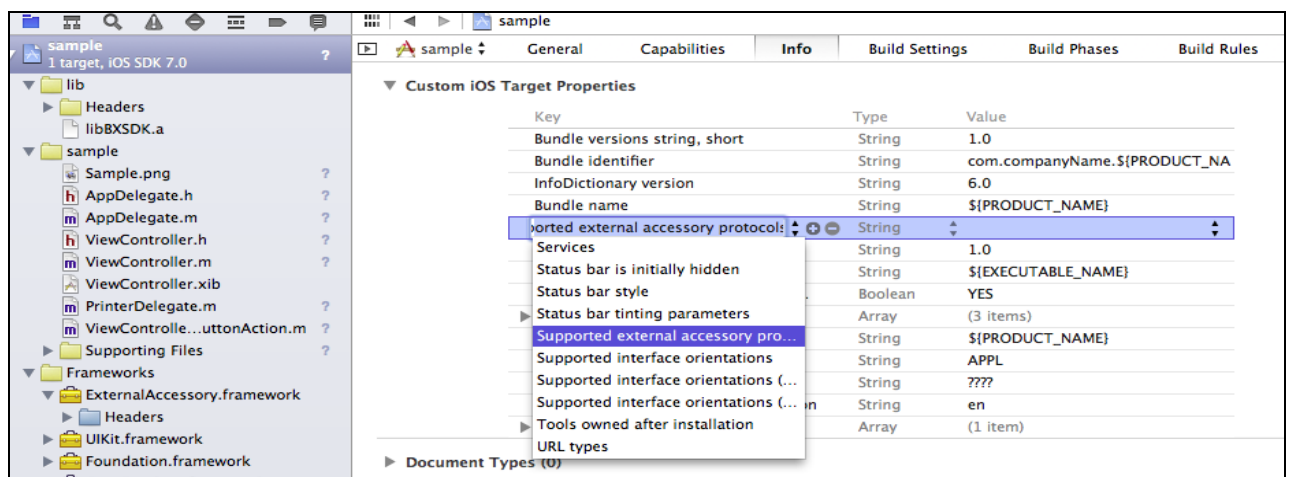
23-1 프로젝트 설정

[참고] iOS 개발을 위해 애플 개발자 프로그램이 등록되어있어야 합니다. 자세한 내용은 애플 개발자 사이트(<http://developer.apple.com/devcenter/ios>)를 참조 하십시오.

23-1-1 ExternalAccessory.framework 추가하기



1. Project 파일을 선택
2. [Info] Tab을 선택
3. 3번의 영역에서 ctrl + click
4. 팝업 메뉴가 보이면 [Add Row] 선택



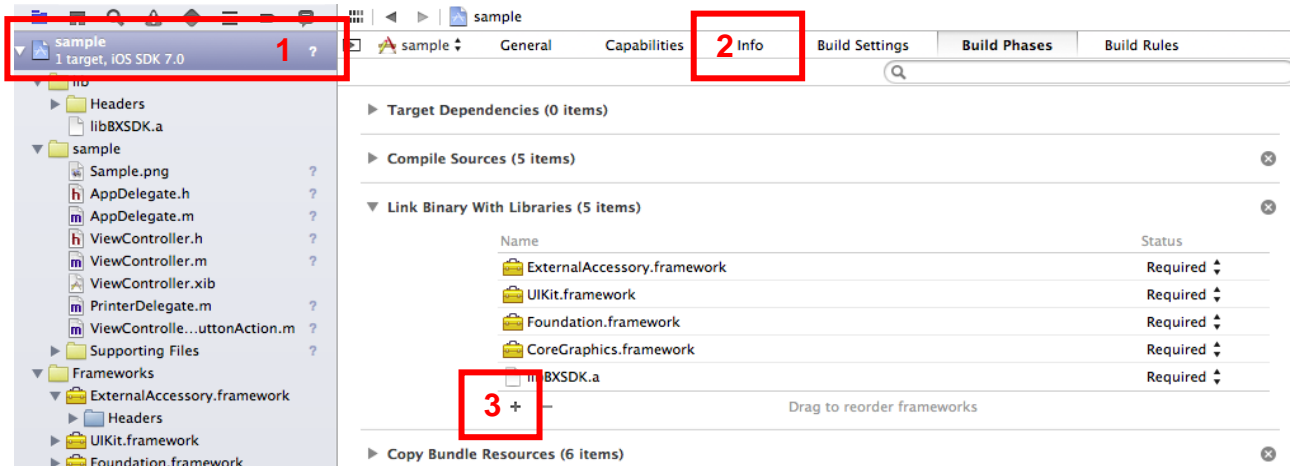
5. "Supported external accessory protocols" 를 입력

6. Item 항목에 “com.bixolon.protocol” 을 입력

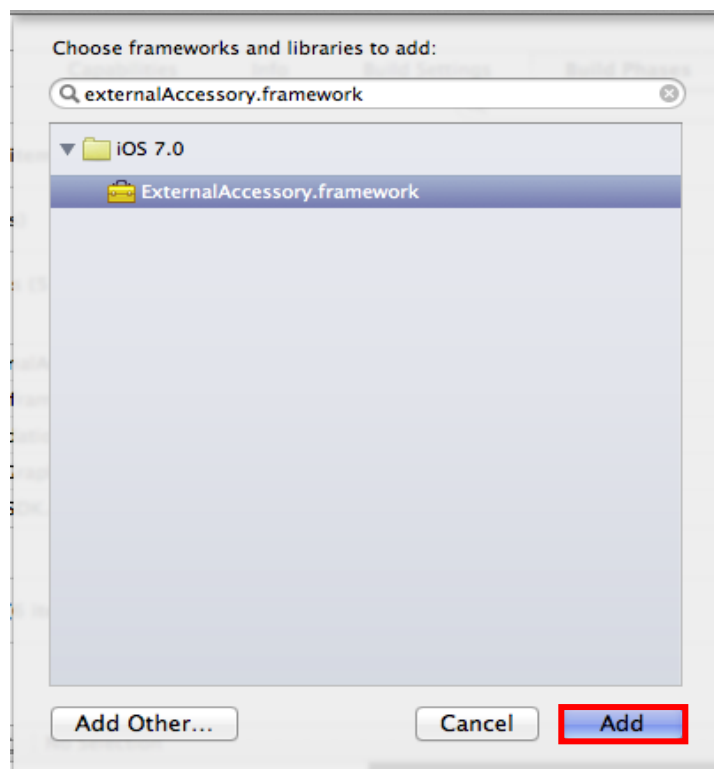
▼ Supported external accessory protocols	Array	(1 item)
Item 0	String	com.bixolon.protocol
Bundle version	String	1.0
Executable file	String	\${EXECUTABLE_NAME}
Application requires iPhone environ...	Boolean	YES

23-1-2 Bluetooth Protocol 추가하기

1. Project – Info - + 를 차례로 선택



2. “ExternalAccessory.framework” 를 검색 후 [Add] 버튼을 눌러 추가



23-2 nativeSample

BXL SDK for iOS_UPOS Compliant를 사용해서 iOS 애플리케이션을 개발하기 위한 레퍼런스입니다.

23-3 phonegapSample

BXL SDK for iOS_UPOS Compliant를 사용해서 iOS 웹 애플리케이션을 개발하기 위한 레퍼런스입니다. PhoneGap버전3.1.0-0.15.0를 사용해서 빌드되었습니다.

23-3-1 환경설정

- Node.js 설치 (<http://nodejs.org> 참고)
- PhoneGap설치 (<http://docs.phonegap.com/en/edge/index.html> 참고)
- 필요한 플러그인추가 및 빌드
(http://docs.phonegap.com/en/edge/guide_cli_index.md.html#The%20Command-Line%20Interface 참고)

23-3-2 PhoneGap을 사용하는 애플리케이션에서 BXL SDK for iOS_UPOS Compliant를 사용하기위한 설정

- CordovaLib/Classes/UPOSService.h, Classes/UPOSService.m 을 추가합니다.
- plugins/com.bixolon.upos.service/www/upos_service.js를 추가합니다.
- www/cordova_plugin.js에 아래 코드를 추가합니다.

```
cordova.define('cordova/plugin_list', function(require, exports, module) {
  module.exports = [
    {
      "file": "plugins/com.bixolon.upos.service/www/upos_service.js",
      "id": "com.bixolon.upos.service.upos_service",
      "clobbers": [
        "upos_service"
      ]
    }
  ]
});
```

- /config.xml에 아래 코드를 추가합니다.

```
<feature name="UPOSService">
  <param name="ios-package" value="UPOSService" />
</feature>
```

24. 에러 정보

이 장은 Priner 메소드를 사용할 때 리턴되는 에러 정보에 대해 설명합니다.
보다 자세한 설명은 UPOS 규약을 참고 하십시오.

24-1 Error 리스트

- claim

Method	Errors
claim	UPOS_E_ILLEGAL
	UPOS_E_TIMEOUT

- checkHealth

Method	Errors
checkHealth	UPOS_E_ILLEGAL

- printNormal

Method	Errors
printNormal	UPOS_E_ILLEGAL
	UPOS_E_BUSY

- printImmediate

Method	Errors
printImmediate	UPOS_E_ILLEGAL

- cutPaper

Method	Errors
cutPaper	UPOS_E_BUSY
	UPOS_E_ILLEGAL

- printBitmap

Method	Errors
printBitmap	UPOS_E_BUSY
	UPOS_E_ILLEGAL
	UPOS_E_NOEXIST

- transactionPrint

Method	Errors
transactionPrint	UPOS_E_BUSY
	UPOS_E_ILLEGAL