

BIXOLON®

API Reference Guide
BXL SDK for UPOS Compliant
iOS

Rev. 2.01

<http://www.bixon.com>

Table of Contents

1. About This Manual	4
2. Support OS and Interface	4
2-1 Operating System.....	4
2-2 Supported Devices and Interfaces.....	4
3. Development Environment	5
3-1 System Requirements	5
3-1-1 Xcode	5
3-1-2 Project (for Bluetooth connection)	5
3-2 Connecting iOS Device.....	6
3-2-1 Bluetooth	6
3-2-2 Network	7
3-2-3 Network–Ad Hoc Mode.....	8
4. Package Contents	10
4-1 Manual.....	10
4-2 Library.....	10
4-3 Sample source code	10
5. Constant Value (Defines)	11
5-1 Event	11
5-1-1 StatusUpdate Event.....	11
5-1-2 Error Event	11
5-1-3 OutputComplete Event	11
5-1-4 Data Event.....	11
5-2 Result Code.....	12
5-3 OpenResult Code	12
5-4 State Code	12
5-5 Transaction Print.....	13
5-6 Alignment.....	13
5-7 Barcode Type.....	14
5-8 Barcode Text Position	15
5-9 Print Direction in Page Mode	15
6. Functions by Class	16
6-1 UPOSDevice Class.....	16
6-2 UPOSDevices Class.....	16
6-2-1 addDevice()	16
6-2-2 removeDevice()	17
6-2-3 save().....	17
6-2-4 getList().....	17
6-3 UPOSPrinterController Class	18
6-3-1 open()	18
6-3-2 claim().....	18
6-3-3 setDeviceEnabled()	19
6-3-4 releaseDevice ()	19
6-3-5 close().....	19
6-3-6 cutPaper().....	20
6-3-7 printBarcode().....	21

6-3-8 printBitmap()	22
6-3-9 printBitmap()	22
6-3-10 printNormal()	23
6-3-11 printPDF()	24
6-3-12 setPageArea()	25
6-3-13 setLeftPosition()	25
6-3-14 setVerticalPosition()	26
6-3-15 setPageModeDirection()	26
6-3-16 printDataInPageMode()	27
6-3-17 transactionPrint()	27
6-3-18 directIO()	28
6-3-19 displayString()	28
6-3-20 displayStringAtLine()	29
6-3-21 clearScreen()	29
6-3-22 storeImage()	30
6-3-23 storeImage()	30
6-3-24 displayImage()	31
6-3-25 clearImage()	32
6-4 UPOSMSRController Class	33
6-4-1 open()	33
6-4-2 claim()	34
6-4-3 setDeviceEnabled()	34
6-4-4 releaseDevice ()	35
6-4-5 close()	35
6-5 UPOSSCRController Class	36
6-5-1 open()	36
6-5-2 claim()	36
6-5-3 setDeviceEnabled()	37
6-5-4 releaseDevice ()	37
6-5-5 close()	38
6-5-6 beginInsertion()	38
6-5-7 endInsertion()	39
6-5-8 beginRemoval()	39
6-5-9 endRemoval()	40
6-5-10 readData()	40
6-6 UPOSCDController Class	41
6-6-1 open()	41
6-6-2 claim()	42
6-6-3 setDeviceEnabled()	42
6-6-4 releaseDevice()	43
6-6-5 close()	43
6-6-6 openDrawer()	43
7. Samples for Test	44
7-1 Printer Search	44
7-2 Connect / Disconnect	45
7-3 Text print	45
7-4 Image print	45
7-5 PDF file print	46
7-6 Page mode print	46

1. About This Manual

- This SDK manual describes the library required for developing applications for iOS.
- It additionally describes how to use SDK, specifications, and restrictions.

2. Support OS and Interface

2-1 Operating System

- This software supports the following operating systems.
- iOS 8.0 or later is required.

2-2 Supported Devices and Interfaces

Models	Interface
SPP-R200II	Bluetooth / WLAN
SPP-R200III	Bluetooth / WLAN
SPP-R210	Bluetooth / WLAN
SPP-R220	Bluetooth / WLAN / BLE
SPP-R300	Bluetooth / WLAN
SPP-R310	Bluetooth / WLAN
SPP-R400	Bluetooth / WLAN
SPP-R410	Bluetooth / WLAN / BLE
SPP-R418	Bluetooth / WLAN / BLE
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-380	Bluetooth / WLAN / Ethernet
SRP-382	Bluetooth / WLAN / Ethernet
SRP-383	Bluetooth / WLAN / Ethernet
SRP-330II	Ethernet
SRP-332II	Ethernet
SRP-S300	Bluetooth / WLAN / Ethernet
SRP-340II	Ethernet
SRP-342II	Ethernet
SRP-275III	Ethernet
SRP-Q300	Bluetooth / WLAN / Ethernet / BLE
SRP-Q302	Bluetooth / WLAN / Ethernet / BLE
SRP-QE300	Ethernet
SRP-QE302	Ethernet
SRP-E300	Ethernet
SRP-E302	Ethernet
SRP-Q200	Bluetooth / WLAN / Ethernet

※ BLE: Bluetooth Low Energy

3. Development Environment

3-1 System Requirements

3-1-1 Xcode

Reference: <http://developer.apple.com/devcenter/ios/index.action>

3-1-2 Project (for Bluetooth connection)

Key	Type	Value
Information Property List (15 items)		
Localization native development region	String	\$(DEVELOPMENT_LANGUAGE)
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle version	String	1
Application requires iPhone environment	Boolean	YES
Launch screen interface file base name	String	LaunchScreen
Main storyboard file base name	String	Main
Required device capabilities	Array	(1 item)
Supported external accessory protocols	Array	(1 item)
Item 0	String	com.bixolon.protocol
Supported interface orientations	Array	(3 items)
Supported interface orientations (iPad)	Array	(4 items)

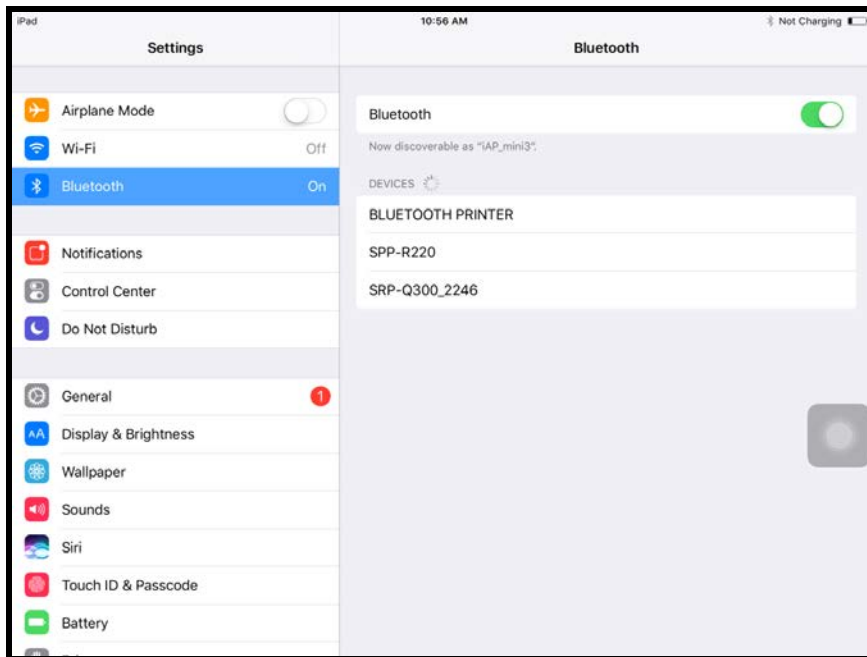
1. Select Info.plist Select.
2. Add Key “Supported external accessory protocols”
3. Save item0’ String in Supported external accessory protocols’key as “com.bixolon.protocol”

3-2 Connecting iOS Device

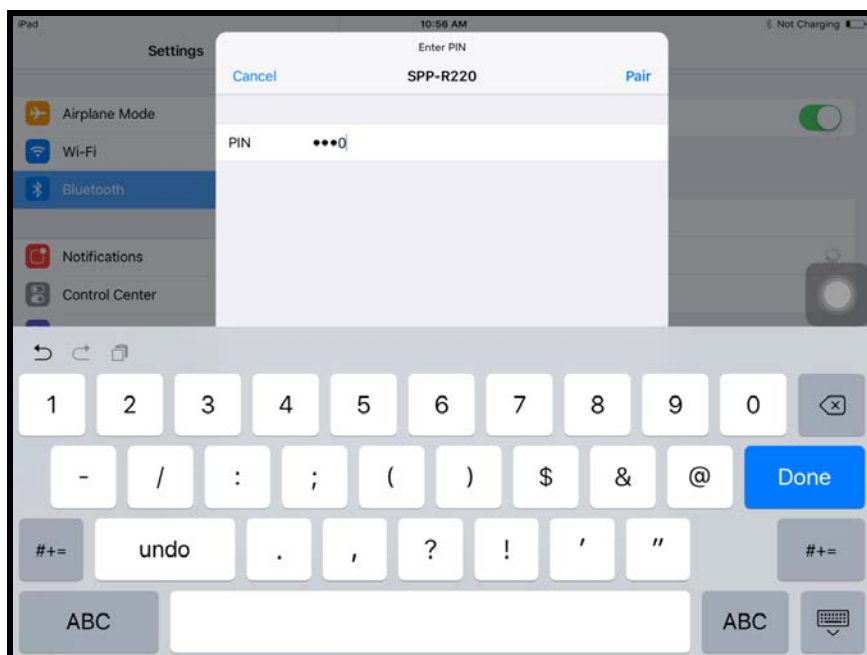
- Some details and names of specific items could be different depending on the iOS version or device.

3-2-1 Bluetooth

1. Select [Settings].
2. Bluetooth should be enabled and the printer power should be on.
3. Select [Bluetooth] for settings.

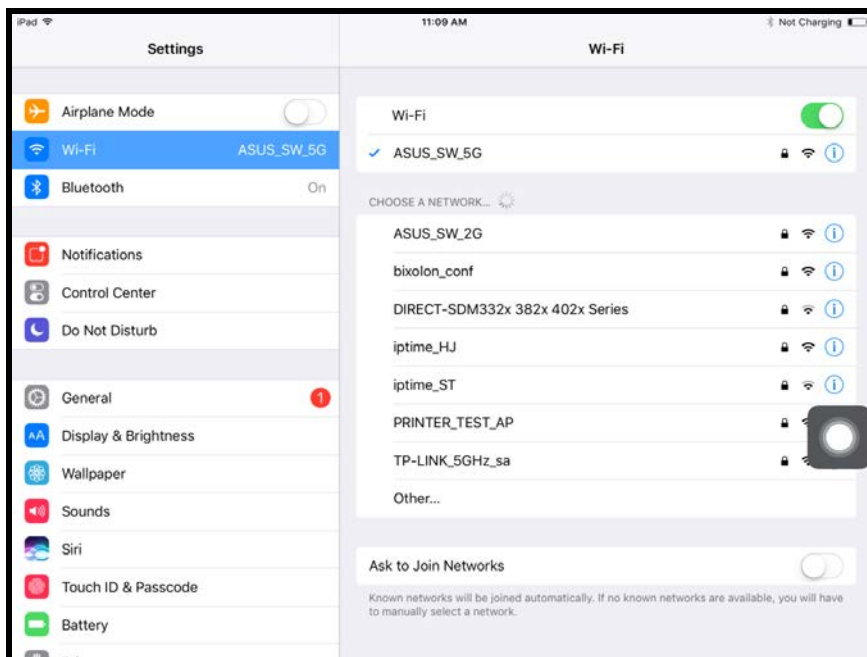


4. Select [Scan] to search the printer to connect and perform pairing..
5. Enter PIN code. Default PIN code is "0000".



3-2-2 Network

1. Connect the printer to the network AP (Access Point) and assign an IP address or obtain one using DHCP. As BIXOLON's printer is initially set to Ad-hoc/SoftAP, it needs to be set up first with our Net Configuration Tool. The Net Configuration Tool can be downloaded from the BIXOLON website.
(Refer to the Net Configuration Tool manual for details on settings)
2. Select [Settings].
3. Wi-Fi should be turned on.
4. Connect the device to the same network that the BIXOLON printer is connected to.



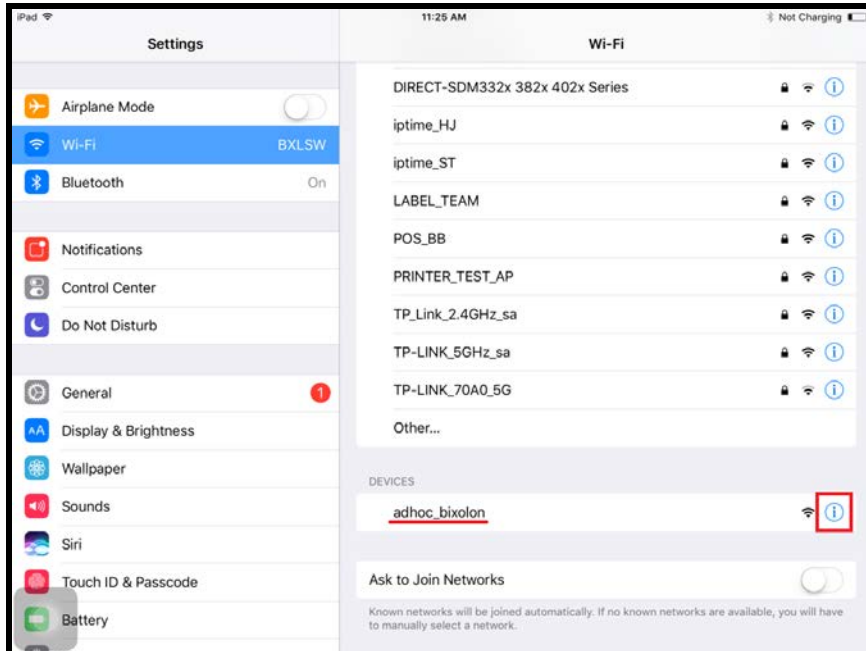
3-2-3 Network–Ad Hoc Mode

1. If the "Network Mode" of the printer is specified as "ADHOC", It is needed to configure IP Address".

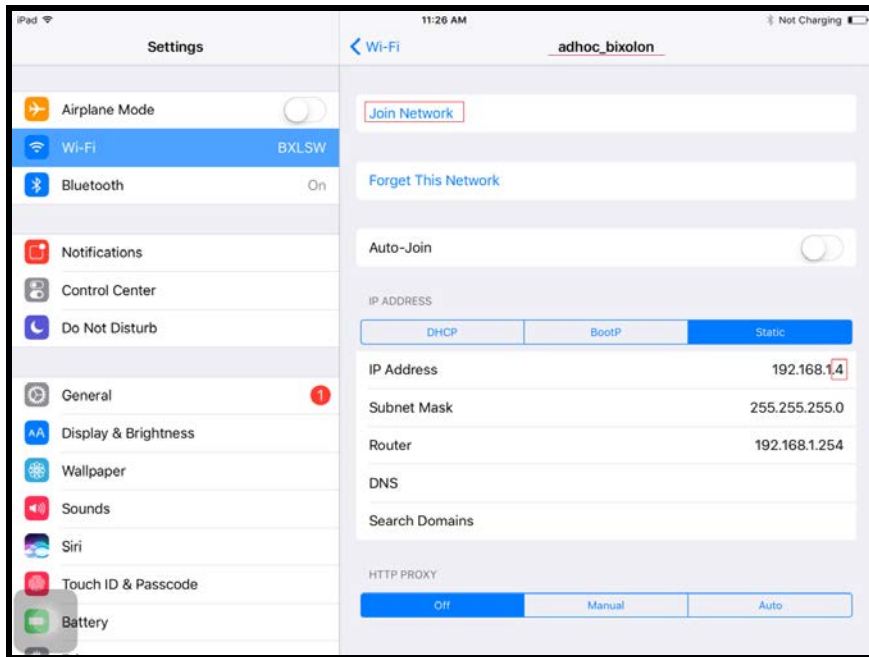


※ self test printout

2. Click the "  " button next to SSID.



3. Assign IP address which consists of four decimal numbers as below.
 - ※ Modify the fourth part if the ip address starts with "192.168.1.x".



4. Package Contents

4-1 Manual

Manual location/ Name	Description
Manual/Manual_BXL SDK for iOS_UPOS compliant API Reference Guide_english_Rev_x_xx	Manual in English
Manual/Manual_BXL SDK for iOS_UPOS compliant API Reference Guide_korean_Rev_x_xx	Manual in Korean

4-2 Library

Library location/ Name	Description
libs/libBixelonUPOS.a	Libaray for controlling devices

4-3 Sample source code

Sample location/ Name	Description
Samples/sample	Printer/MSR/SCR/CashDrawer control sample application

5. Constant Value (Defines)

5-1 Event

- Each Event is defined in the UPOSDeviceControlDelegate protocol.

5-1-1 StatusUpdate Event

A StatusUpdate event occurs whenever the printer status changes.

Code	Value	Description
PTR_SUE_IDLE	1001	Idle state.
UPOS_SUE_POWER_ONLINE	2001	Online state.
UPOS_SUE_POWER_OFF	2002	Offline state.
UPOS_SUE_POWER_OFFLINE	2003	All of them have the same effects in this SDK unless there is a separate comment.
PTR_SUE_COVER_OPEN	11	Cover Open.
PTR_SUE_COVER_OK	12	Cover Closed.
PTR_SUE_REC_EMPTY	24	Paper Empty.
PTR_SUE_REC_NEAREMPTY	25	Paper Near End
PTR_SUE_REC_PAPEROK	26	Paper OK
PTR_SUE_REC_BATTERY_NORMAL	60	Battery Normal
PTR_SUE_REC_BATTERY_LOW	61	Battery Low

5-1-2 Error Event

Code	Value	Description
UPOS_EPTR_COVER_OPEN	201	Cover Open
UPOS_EPTR_REC_EMPTY	203	Paper Empty

5-1-3 OutputComplete Event

Generates a print completion event. However, it must be used in Async mode.

5-1-4 Data Event

Receives MSR Track information data.

5-2 Result Code

- These constants are used for the results returned from methods after executing specific functions.

Code	Value	Description
UPOS_SUCCESS	0	Operation success.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_NOTCLAIMED	103	Device is not in Claim state.
UPOS_E_NOSERVICE	104	Function is not supported.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_ILLEGAL	106	Illegal access or unsupported function
UPOS_E_NOHARDWARE	107	Device is not connected.
UPOS_E_OFFLINE	108	Device is off-line.
UPOS_E_NOEXIST	109	Target does not exist.
UPOS_E_EXISTS	110	Target already exists.
UPOS_E_FAILURE	111	The requested operation failed.
UPOS_E_TIMEOUT	112	Timeout
UPOS_E_BUSY	113	Device is busy executing previously requested operation.
UPOS_E_EXTENDED	114	Device error. Refer to the ResultCode Extended section for more details.
UPOS_E_DEPRECATED	113	The function is currently not used.

5-3 OpenResult Code

- These constants are the results returned by the Open method.

Code	Value	Description
UPOS_OR_ALREADYOPEN	301	Device is already open.
UPOS_OR_REGBADNAME	302	The specified device name cannot be found in the stored device list.
UPOS_OR_FAILEDOPEN	305	The execution of the Open method failed, but specific reason is unknown.

5-4 State Code

- These are the constants that are used for the property of State.

Code	Value	Description
UPOS_S_CLOSED	1	Device is closed.
UPOS_S_IDLE	2	Device is in standby state without error.
UPOS_S_BUSY	3	Device is currently busy executing another method.
UPOS_S_ERROR	4	There is an error.

5-5 Transaction Print

- These are the constants for setting the Transaction mode.

Code	Value	Description
PTR_TP_TRANSACTION	11	Initialize the buffer to Empty state and start the Transaction mode.
PTR_TP_NORMAL	12	Terminate the Transaction mode and print the data stored in the buffer.

5-6 Alignment

- These are the constants required for specifying alignment.

(For Barcodes)

Code	Value	Description
PTR_BC_LEFT	-1	Align to left
PTR_BC_CENTER	-2	Align to center
PTR_BC_RIGHT	-3	Align to right

(For Images)

Code	Value	Description
PTR_BM_LEFT	-1	Align to left
PTR_BM_CENTER	-2	Align to center
PTR_BM_RIGHT	-3	Align to right

5-7 Barcode Type

- Definitions of the values required to specify barcode type when barcode is printed.

Code	Value	Description
PTR_BCS_UPCA	101	UPCA
PTR_BCS_UPCE	102	UPCE
PTR_BCS_JAN8	103	JAN8
PTR_BCS_EAN8	103	EAN8
PTR_BCS_JAN13	104	JAN13
PTR_BCS_EAN13	104	EAN13
PTR_BCS_TF	105	Standard(ordiscrete) 2 of 5
PTR_BCS_ITF	106	Interleaved 2 of 5
PTR_BCS_Codabar	107	Codabar
PTR_BCS_Code39	108	Code39
PTR_BCS_Code93	109	Code93
PTR_BCS_Code128	110	Code 128 ※ Code128 Subset
		Subset Identifier
		Subset A Barcode data starts with '{A'
		Subset B Barcode data starts with '{B'
PTR_BCS_UPCA_S	111	UPC-A with supplemental barcode
PTR_BCS_UPCE_S	112	UPC-E with supplemental barcode
PTR_BCS_UPCD1	113	UPC-D1
PTR_BCS_UPCD2	114	UPC-D2
PTR_BCS_UPCD3	115	UPC-D3
PTR_BCS_UPCD4	116	UPC-D4
PTR_BCS_UPCD5	117	UPC-D5
PTR_BCS_EAN8_S	118	EAN8 with supplemental barcode
PTR_BCS_EAN13_S	119	EAN13 with supplemental barcode
PTR_BCS_EAN128	120	EAN128
PTR_BCS_OCRA	121	OCR "A"
PTR_BCS_OCRB	122	OCR "B"
PTR_BCS_Code128_Parsed	123	Code 128 with parsing
PTR_BCS_GS1DATABAR	131	GS1 DataBar Omnidirectional
PTR_BCS_GS1DATABAR_E	132	GS1 DataBar Stacked Omnidirectional
PTR_BCS_GS1DATABAR_S	133	GS1 DataBar Expanded
PTR_BCS_GS1DATABAR_E_S	134	GS1 DataBar Expanded Stacked
PTR_BCS_PDF417	201	PDF 417
PTR_BCS_MAXICODE	202	MAXI Code
PTR_BCS_DATAMATRIX	203	Data Matrix
PTR_BCS_QRCODE	204	QR Code
PTR_BCS_UQRCODE	205	Micro QR Code
PTR_BCS_AZTEC	206	Aztec
PTR_BCS_UPDF417	207	Micro PDF 417

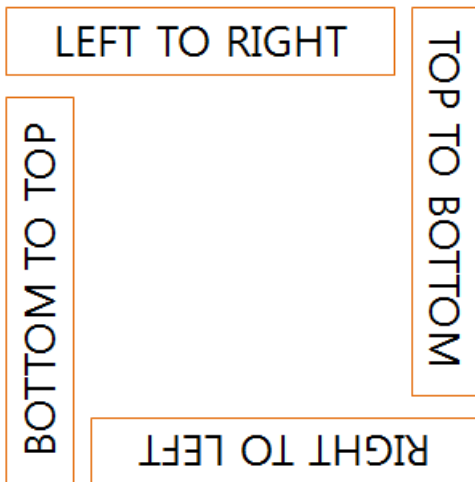
5-8 Barcode Text Position

- These are the constants for specifying the text printing option and printing position when the specific barcode supports text printing..

Code	Value	Description
PTR_BC_TEXT_NONE	-11	Does not print the text (barcode only)
PTR_BC_TEXT_ABOVE	-12	Prints the text at the top of the barcode.
PTR_BC_TEXT_BELOW	-13	Prints the text at the bottom of the barcode.

5-9 Print Direction in Page Mode

Code	Value	Description
LEFT_TO_RIGHT	0	Prints left to right
BOTTOM_TO_TOP	1	Prints from bottom to top
RIGHT_TO_LEFT	2	Prints right to left
TOP_TO_BOTTOM	3	Prints from top to bottom



6. Functions by Class

6-1 UPOSDevice Class

- UPOSDevice Class is an object that contains the common information about all target control devices controlled by the controller of each device.

[Property]

Type	Name	Description
NSString*	modelName	Device Model name
NSString*	interfaceType	Method of connecting each device.
NSString*	address	Device Address(IP address, Mac Address) value
NSString*	serialNumber	Device Serial Number(Bluetooth only)
NSString*	port	used for the Network port number that is used for socket (Wi-Fi / Ethernet)
BOOL	connectedFlag	Device connection status

6-2 UPOSDevices Class

- UPOSDevices Class is an object that contains the device list in the Device.

6-2-1 addDevice()

This method adds a device to the current device list.

[Syntax]

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

[Parameters]

Type	Name	Description
UPOSDevice*	device	Object that contains the information about the device to add

[Return Values]

Type	Value	Description
BOOL	YES	Returned on success
BOOL	NO	Returned on failure

6-2-2 removeDevice()

This method removes the device in the current device list.

[Syntax]

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

[Parameters]

Type	Name	Description
UPOSDevice*	device	Object that contains the information about the device to delete

[Return Values]

Type	Value	Description
BOOL	YES	Returned on success
BOOL	NO	Returned on failure

6-2-3 save()

This method saves the current list of devices.

[Syntax]

-(BOOL) save;

[Return Values]

Type	Value	Description
BOOL	YES	Returned on success
BOOL	NO	Returned on failure

6-2-4 getList()

This method returns the saved device list.

[Syntax]

-(NSMutableArray*) getList;

[Return Values]

Type	Value	Description
NSMutableArray*	.	The list of devices is returned.

6-3 UPOSPrinterController Class

- UPOSPrinterController Class is the main object to control the common functions of the devices supported by this SDK.

[Property]

Type	Name	Description
BOOL	AsyncMode	Select whether to use asynchronous mode or not.

6-3-1 open()

This method initiates the use of the printer class, and it includes the initialization process such as memory allocation. This method should be called first before calling the claim and other subsequent methods.

[Syntax]

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

[Parameters]

Type	Name	Description
NSString*	logicalDeviceName	the name of the device to be opened

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-2 claim()

This method tries to open the port specified in the device information, and it includes some initialization processes such as memory allocation and initialization.

[Syntax]

-(NSInteger) claim : (NSInteger)timeout;

[Parameters]

Type	Name	Description
NSInteger	timeout	Open the port for the duration specified in this parameter.

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-3 setDeviceEnabled()

This property is an option to use the device.

This function may not be available if the property of DeviceEnabled is NO, even when the state if the property of Claimed is YES.

[Syntax]

-(void) setDeviceEnabled: (BOOL);

[Parameters]

Type	Value	Description
BOOL	YES	Enable
BOOL	NO	Disable

6-3-4 releaseDevice()

This method terminates the use of the port of the claimed device and releases the physical resources.

Some of the memory resources may also be released as a result.

[Syntax]

-(NSInteger) releaseDevice;

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-5 close()

This method terminates the use of the open device.

Some of the memory resources may also be released as a result.

[Syntax]

-(NSInteger) close;

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-6 cutPaper()

Cuts the paper..

[Syntax]

-(NSInteger) cutPaper : (NSInteger)percentage;

[Parameters]

Type	Name	Description	
NSInteger	percentage	Full cut / Partial cut. It works only in auto cutter built in models.	
		Value	Description
		100	Full cut
		90	Partial cut

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-7 printBarcode()

Prints a barcode.

[Syntax]

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

[Parameters]

Type	Name	Description
NSInteger	station	Fixed Value PTR_S_RECEIPT
NSString*	data	The data to be included in the barcode. The data allowed by the barcode type may differ
NSInteger	symbology	Select the type of barcode. (Refer to “5-6 Barcord type”)
NSInteger	height	Specify the height of the barcode.
NSInteger	width	Specify the width of the barcode.
NSInteger	alignment	Select the alignment of the barcode. (Refer to “5-5 Alignment”)
NSInteger	textPosition	Determine the postion of the text to be printed with the barcode. (Refer to “5-7 Barcode Text Location”)

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-8 printBitmap()

Prints image.(file)

[Syntax]

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

[Parameters]

Type	Name	Description
NSInteger	station	Fixed Value : PTR_S_RECEIPT
NSString*	filename	Specify the path to the image file.
NSInteger	width	Specify the image width
NSInteger	alignment	Select the image alignment. (Refer to "5-5 Alignment")

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-9 printBitmap()

Prints image.(image data)

[Syntax]

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

[Parameters]

Type	Name	Description
NSInteger	station	Fixed Value : PTR_S_RECEIPT
UIImage *	image	Type the image data.
NSInteger	width	Specify the image width
NSInteger	alignment	Select the image alignment. (Refer to "5-5 Alignment")

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-10 printNormal()

Prints text.

[Syntax]

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

[Parameters]

Type	Name	Description
NSInteger	station	Fixed Value PTR_S_RECEIPT
NSString *	data	Specify the data to be printed. Printable characters and escape sequences, carriage returns, line feeds Data are allowed..

ESC Sequences

These are commands that begin with the characters ESC(0x1B) + '['(0x7C). Out of the commands below, “#” represents a decimal number value, and a range value exists for each command. “[” indicates commands that can be omitted. If a number value is omitted, it will become “0”. “!” is a command that cancels the relevant configuration. An ESC Sequence command that is unsupported by the printer will be ignored.

Escape Sequence	Supported or unsupported, and configuration value range	Description
[#]P	O (# : 0~100)	Cutting
[#]fP	O (# : 0~100)	Cutting after feed
[#]IF	O (# : 0~50)	Line feeding as much as # number
#fT	O (# : 0~3)	Configures font types according to # value 0: Default font (A), 1 = Font B, 2 = Font C
[!]bC	O	Bold
[!][#]uC	O (# : 1~2)	Underline
[!]rvC	O	Reverse
1C	O	Font size 1 times width, 1 times height
2C	O	Font size 2 times width, 1 times height
3C	O	Font size 1 times width, 2 times height
4C	O	Font size 2 times width, 2 times height
#hC	O	# times font width proportion
#vC	O	# times font height proportion
cA	O	Align center
rA	O	Align right
lA	O	Align left
N	O	Initialize value that is configurable as ESC Sequence Font type A, font size 1 times width, 1 times height Cancel bold, cancel underline, cancel reverse

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-11 printPDF()

Prints an PDF file.

[Syntax]

```

-(NSInteger) printPDF : (NSInteger) station
                    fileName : (NSString*) fileName
                    width : (NSInteger) width
                    alignment : (NSInteger) alignment
                    page : (NSInteger) page
                    brightness : (NSInteger) brightness;
    
```

[Parameters]

Type	Name	Description
NSInteger	Station	Fixed Value PTR_S_RECEIPT
NSString *	fileName	Specify the path to the PDF file
NSInteger	width	Specify the PDF width
NSInteger	alignment	Select image alignment. (Refer to "5-5 Alignment")
NSInteger	page	Specify the page number of the PDF to be printed
NSInteger	brightness	Specify the brightness value. (0 ~ 100)

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-12 setPageArea()

Specifies the page mode area.

[Syntax]

```
- (NSInteger)setPageArea:(NSInteger)startingX
    startingY:(NSInteger)startingY
    width:(NSInteger)width
    height:(NSInteger)height;
```

[Parameters]

Type	Name	Description
NSInteger	startingX	X coordinate of area.
NSInteger	startingY	Y coordinate of area.
NSInteger	width	Width of area.
NSInteger	height	Height of area.

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-13 setLeftPosition()

Specifies the print start position (Horizontal).

[Syntax]

```
- (NSInteger)setLeftPosition:(NSInteger)positionX
```

[Parameters]

Type	Name	Description
NSInteger	positionX	Print start position (Horizontal)

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-14 setVerticalPosition()

Specifies the print start position (Vertical).

[Syntax]

- (NSInteger)setVerticalPosition:(NSInteger)positionY;

[Parameters]

Type	Name	Description
NSInteger	positionY	print start position (Vertical)

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-15 setPageModeDirection()

Specifies the direction of printing in Page Mode.

[Syntax]

- (NSInteger)setPageModeDirection:(PAGE_MODE_DIRECTION)direction;

[Parameters]

Type	Name	Description
PAGE_MODE_DIRECTION	direction	Specify the direction of printing. (Refer to "Print Direction in Page Mode")

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-16 printDataInPageMode()

Starts printing with the data in the buffer for the page mode.

[Syntax]

- (NSInteger)printDataInPageMode;

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-17 transactionPrint()

Prints using the Transaction Mode.

[Syntax]

-(NSInteger) transactionPrint : (NSInteger)station
control : (NSInteger)control;

[Parameters]

Type	Name	Description
NSInteger	station	fixed Value PTR_S_RECEIPT
NSInteger	control	Transaction Mode(Refer to “5-4 Transaction Print”)

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-18 directIO()

User defined data is sent to printer

[Syntax]

-(NSInteger) directIO : (NSInteger)command
 data : (void*)data;

[Parameters]

Type	Name	Description
NSInteger	command	0: one-direction, 1: bi-direction
void*	data	Enters a user defined data

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-19 displayString()

Displays the text on BCD-3000.

This time, BCD-3000 should be connected to SRP-Q300 DK port.

[Syntax]

-(NSInteger) displayString:(NSString*)string;

[Parameters]

Type	Name	Description
NSString*	string	Text data to be printed with BCD-3000

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-20 displayStringAtLine()

Displays the text at assigned line on BCD-3000.
This time, BCD-3000 should be connected to SRP-Q300 DK port.

[Syntax]

```
-(NSInteger) displayStringAtLine:(NSInteger)line
                        data:(NSString*)data;
```

[Parameters]

Type	Name	Description
NSInteger	line	Line texts are displayed
NSString*	data	Text data to be printed with BCD-3000

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-21 clearScreen()

Clear BCD-3000.
This time, BCD-3000 should be connected to SRP-Q300 DK port.

[Syntax]

```
-(NSInteger) clearScreen;
```

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-22 storeImage()

Save the image data in the image buffer of BCD-3000.
This time, BCD-3000 should be connected to SRP-Q300 DK port.

[Syntax]

```
-(NSInteger) storeImage:(UIImage*)image
                    width:(NSInteger)width
                    imageNumber:(NSInteger)imageNumber;
```

[Parameters]

Type	Name	Description
UIImage*	image	Image data
NSInteger	width	Specify the width of image (1 ~ 160)
NSInteger	imageNumber	Specify the number of the image data to be saved (1 ~ 5)

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-23 storeImage()

Save the image data in the image buffer of BCD-3000.
This time, BCD-3000 should be connected to SRP-Q300 DK port.

[Syntax]

```
-(NSInteger) storeImage:(UIImage*)image
                    width:(NSInteger)width
                    imageNumber:(NSInteger)imageNumber;
```

[Parameters]

Type	Name	Description
NSString*	filename	Specify the path of image file.
NSInteger	width	Specify the width of image (1 ~ 160)
NSInteger	imageNumber	Specify the number of the image data to be saved (1 ~ 5)

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-24 displayImage()

Print the image saved in the image buffer of BCD-3000.
This time, BCD-3000 should be connected to SRP-Q300 DK port.

[Syntax]

```
-(NSInteger) displayImage:(NSInteger)imageNumber  
                    xPos:(NSInteger)xPos  
                    yPos:(NSInteger)yPos;
```

[Parameters]

Type	Name	Description
NSInteger	imageNumber	Specify the number of the image data to be printed (1 ~ 5)
NSInteger	xPos	Input X coordinate to print the image on (0 ~ 159)
NSInteger	yPos	Input Y coordinate to print the image on (0 ~ 31)

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-3-25 clearImage()

Delete the image saved in the image buffer of BCD-3000.
This time, BCD-3000 should be connected to SRP-Q300 DK port.

[Syntax]

```
-(NSInteger) clearImage:(BOOL)isAll
                imageNumber:(NSInteger)imageNumber;
```

[Parameters]

Type	Name	Description	
BOOL	isAll	Image buffer clear mode.	
		Value	Description
		YES	Delete all the images in image buffer.
		NO	Delete only the image specified by 'imageNumber'
NSInteger	imageNumber	Specify the number of the image data to be deleted (1~5)	

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-4 UPOSMSRController Class

- UPOSMSRController Class is the main object to control the common functions of the devices supported by this SDK.

[Property]

Type	Name	Description
NSString*	Track1Data	The most recently obtained Track 1 data of the MSR card
NSString*	Track2Data	The most recently obtained Track 2 data of the MSR card
NSString*	Track3Data	The most recently obtained Track 3 data of the MSR card

6-4-1 open()

This method initiates the use of the MSR class, and it includes the initialization process such as memory allocation. This method should be called first before calling the claim and other subsequent methods.

[Syntax]

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

[Parameters]

Type	Name	Description
NSString *	logicalDeviceName	Enter the name of the device to be opened

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-4-2 claim()

This method tries to open the port specified in the device information, and it includes some initialization processes such as memory allocation and initialization.

[Syntax]

-(NSInteger) claim : (NSInteger)timeout;

[Parameters]

Type	Name	Description
NSInteger	timeout	Open the port for the duration specified in this parameter.

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-4-3 setDeviceEnabled()

This property is an option to use the device.

This function may not be available if the property of DeviceEnabled is NO, even when the state if the property of Claimed is YES.

[Syntax]

-(void) setDeviceEnabled: (BOOL);

[Parameters]

Type	Value	Description
BOOL	YES	Enable
BOOL	NO	Disable

6-4-4 releaseDevice ()

This method terminates the use of the port of the claimed device and releases the physical resources. Some of the memory resources may also be released as a result.

[Syntax]

-(NSInteger) releaseDevice;

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-4-5 close()

This method terminates the use of the open device.
Some of the memory resources may also be released as a result.

[Syntax]

-(NSInteger) close;

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-5 UPOSSCRController Class

- UPOSSCRController Class is the main object to control the common functions of the devices supported by this SDK.

6-5-1 open()

This method initiates the use of the SCR class, and it includes the initialization process such as memory allocation. This method should be called first before calling the claim and other subsequent methods.

[Syntax]

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

[Parameters]

Type	Name	Description
NSString *	logicalDeviceName	Enter the name of the device to be opened

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-5-2 claim()

This method tries to open the port specified in the device information, and it includes some initialization processes such as memory allocation and initialization.

[Syntax]

-(NSInteger) claim : (NSInteger)timeout;

[Parameters]

Type	Name	Description
NSInteger	timeout	Open the port for the duration specified in this parameter.

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-5-3 setDeviceEnabled()

This property is an option to use the device.

This function may not be available if the property of DeviceEnabled is NO, even when the state if the property of Claimed is YES.

[Syntax]

-(void) setDeviceEnabled: (BOOL);

[Parameters]

Type	Value	Description
BOOL	YES	Enable
BOOL	NO	Disable

6-5-4 releaseDevice ()

This method terminates the use of the port of the claimed device and releases the physical resources. Some of the memory resources may also be released as a result.

[Syntax]

-(NSInteger) releaseDevice;

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-5-5 close()

This method terminates the use of the open device.
Some of the memory resources may also be released as a result.

[Syntax]

-(NSInteger) close;

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-5-6 beginInsertion()

Checks if the Smart Card is inserted for the specified time.
Be sure to call the endInsertion() function after calling the function.

[Syntax]

-(NSInteger) beginInsertion : (NSInteger)timeout;

[Parameters]

Type	Name	Description
NSInteger	timeout	Sets card insertion check time

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-5-7 endInsertion()

Applies power to the inserted Smart Card chip.
Be sure to call the beginInsertion function before calling the function.

[Syntax]

-(NSInteger) endInsertion;

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-5-8 beginRemoval()

Terminates the power to the Smart Card chip.
Be sure to call the endRemoval function after calling the function.

[Syntax]

-(NSInteger) beginRemoval : (NSInteger) timeout;

[Parameters]

Type	Name	Description
NSInteger	timeout	Sets power-off time

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-5-9 endRemoval()

Check whether the power supplying to the Smart Card chip has been terminated normally or not. Be sure to call the beginRemoval function before calling the function.

[Syntax]

-(NSInteger) endRemoval;

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-5-10 readData()

Reads and writes to Smart Card chip.

[Syntax]

-(NSInteger) readData : (NSInteger) action
data : (NSData**) data;

[Parameters]

Type	Name	Description
NSInteger	action	Fixed Value SC_READ_DATA
NSData**	data	R/W buffer

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-6 UPOSCDController Class

- UPOSCDController Class is the main object to control the common functions of the devices supported by this SDK.

[Property]

Type	Name	Description	
BOOL	DrawerOpened	Cash drawer status	
		Value	Description
		YES	Cash drawer is opened.
		NO	Cash drawer is closed.

6-6-1 open()

This method initiates the use of the CD(Cash Drawer) class, and it includes the initialization process such as memory allocation. This method should be called first before calling the claim and other subsequent methods.

[Syntax]

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

[Parameters]

Type	Name	Description
NSString *	logicalDeviceName	Enter the name of the device to be opened

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-6-2 claim()

This method tries to open the port specified in the device information, and it includes some initialization processes such as memory allocation and initialization.

[Syntax]

-(NSInteger) claim : (NSInteger)timeout;

[Parameters]

Type	Name	Description
NSInteger	timeout	Open the port for the duration specified in this parameter.

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-6-3 setDeviceEnabled()

This property is an option to use the device.

This function may not be available if the property of DeviceEnabled is NO, even when the state if the property of Claimed is YES.

[Syntax]

-(void) setDeviceEnabled: (BOOL);

[Parameters]

Type	Value	Description
BOOL	YES	Enable
BOOL	NO	Disable

6-6-4 releaseDevice()

This method terminates the use of the port of the claimed device and releases the physical resources. Some of the memory resources may also be released as a result.

[Syntax]

-(NSInteger) releaseDevice;

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-6-5 close()

This method terminates the use of the open device. Some of the memory resources may also be released as a result.

[Syntax]

-(NSInteger) close;

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

6-6-6 openDrawer()

Opens Cash Drawer.

[Syntax]

-(NSInteger) OpenDrawer;

[Return Values]

Code	Value	Description
UPOS_SUCCESS	0	Operation is successful.
UPOS_E_CLOSED	101	Device to access is closed.
UPOS_E_CLAIMED	102	Claim method should be called first.
UPOS_E_DISABLED	105	Not enabled.
UPOS_E_FAILURE	111	The requested operation failed.

7. Samples for Test

7-1 Printer Search

```
#import "ViewController.h"
#import "UPOSPrinterController.h"

@interface ViewController () <UPOSDeviceControlDelegate> {
    UPOSPrinterController *printerCon;
    UPOSPrinters *printerList;
}
@end

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];

    [self initUPOS]; // init
    [self btLookup]; // Bluetooth device search
}

-(void) initUPOS {
    printerCon = [[UPOSPrinterController alloc]init];
    printerList = [[UPOSPrinters alloc]init];

    [printerCon setLogLevel: LOG_SHOW_NEVER ];
    printerCon.delegate = self;
    [printerCon setStringEncoding: NSASCIIStringEncoding];

    [[NSNotificationCenter defaultCenter] addObserver:self
                                             selector:@selector(didBTStart:)
                                             name: __NOTIFICATION_NAME_BT_WILL_LOOKUP_
                                             object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self
                                             selector:@selector(didBTDeviceList:)
                                             name: __NOTIFICATION_NAME_BT_FOUND_PRINTER_
                                             object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self
                                             selector:@selector(didBTComplete:)
                                             name: __NOTIFICATION_NAME_BT_LOOKUP_COMPLETE_
                                             object:nil];
}

-(void) btLookup {
    [printerCon refreshBTLookup];
}

//MARK: - lookup notification
- (void) didBTStart:(NSNotification*)notification {}
- (void) didBTComplete:(NSNotification*)notification {}
- (void) didBTDeviceList:(NSNotification*)notification {
    UPOSPrinter* lookupDevice = (UPOSPrinter*)[[notification userInfo]
objectForKey: __NOTIFICATION_NAME_BT_FOUND_PRINTER_];
    if( lookupDevice == nil) return;
    [printerList addDevice:lookupDevice];
    [printerList save];
}

-(void)StatusUpdateEvent:(NSNumber*)Status {}
@end
```

7-2 Connect / Disconnect

It is continuous to example 7-1

```
-(void) connect {
    UPOSPrinter* target = (UPOSPrinter*)[printerList getList].lastObject;

    if([printerCon open:target.modelName] == UPOS_SUCCESS) {
        if([printerCon claim:5000] == UPOS_SUCCESS){
            [NSThread sleepForTimeInterval:0.1f];
            [printerCon setDeviceEnabled:YES];
        }
    }
}

-(void) disconnect {
    printerCon.DeviceEnabled = NO;
    if([printerCon releaseDevice] == UPOS_SUCCESS){
        [NSThread sleepForTimeInterval:0.01f];
        [printerCon close];
    }
}
```

7-3 Text print

It is continuous to example 7-1 and 7-2

```
-(void) printText {
    [printerCon printNormal:PTR_S_RECEIPT
                    data:[NSString stringWithFormat:@"%test print\r\n"]];
}
```

7-4 Image print

It is continuous to example 7-1 and 7-2

```
-(void) printImage {
    UIImage* img = [UIImage imageNamed:@"Sample"];
    [printerCon printBitmap:PTR_S_RECEIPT
                        image:img
                        width:350
                        alignment:PTR_BM_CENTER
                        brightness:10050];
}
```

7-5 PDF file print

It is continuous to example 7-1 and 7-2

```
-(void) printPDF {
    NSString *path = [[NSBundle mainBundle] pathForResource:@"testPDF"
                                                         ofType:@"pdf"];

    [printerCon printPDF:PTR_S_RECEIPT
                    fileName:path
                    width:printerCon.RecLineWidth
                    alignment:PTR_BM_CENTER
                    page:2
                    brightness:10050];
}
```

7-6 Page mode print

It is continuous to example 7-1 and 7-2

```
-(void) printPagemode {
    [printerCon setPageArea : 0
                    startingY : 0
                    width : 512
                    height : 500];

    [printerCon setVerticalPosition:0];
    [printerCon setLeftPosition:0];
    [printerCon setPageModeDirection:TOP_TO_BOTTOM];
    [printerCon printNormal:PTR_S_RECEIPT data:@"0.0\r\n"];

    [printerCon printDataInPageMode];
}
```

Copyright

© BIXOLON Co., Ltd. All rights reserved.

This user manual and all property of the product are protected under copyright law. It is strictly prohibited to copy, store, and transmit the whole or any part of the manual and any property of the product without the prior written approval of BIXOLON Co., Ltd. The information contained herein is designed only for use with this BIXOLON product. BIXOLON is not responsible for any direct or indirect damages, arising from or related to use of this information.

- The BIXOLON logo is the registered trademark of BIXOLON Co., Ltd.
- All other brand or product names are trademarks of their respective companies or organizations.

BIXOLON Co., Ltd. maintains ongoing efforts to enhance and upgrade the functions and quality of all our products.

In the following, product specifications and/or user manual content may be changed without prior notice.

Caution

Some semiconductor devices are easily damaged by static electricity. You should turn the printer "OFF", before you connect or remove the cables on the rear side, in order to guard the printer against the static electricity. If the printer is damaged by the static electricity, you should turn the printer "OFF".

