eBeam Smart Marker Developer Guide for iOS

PNF R&D S/W

2018. 05

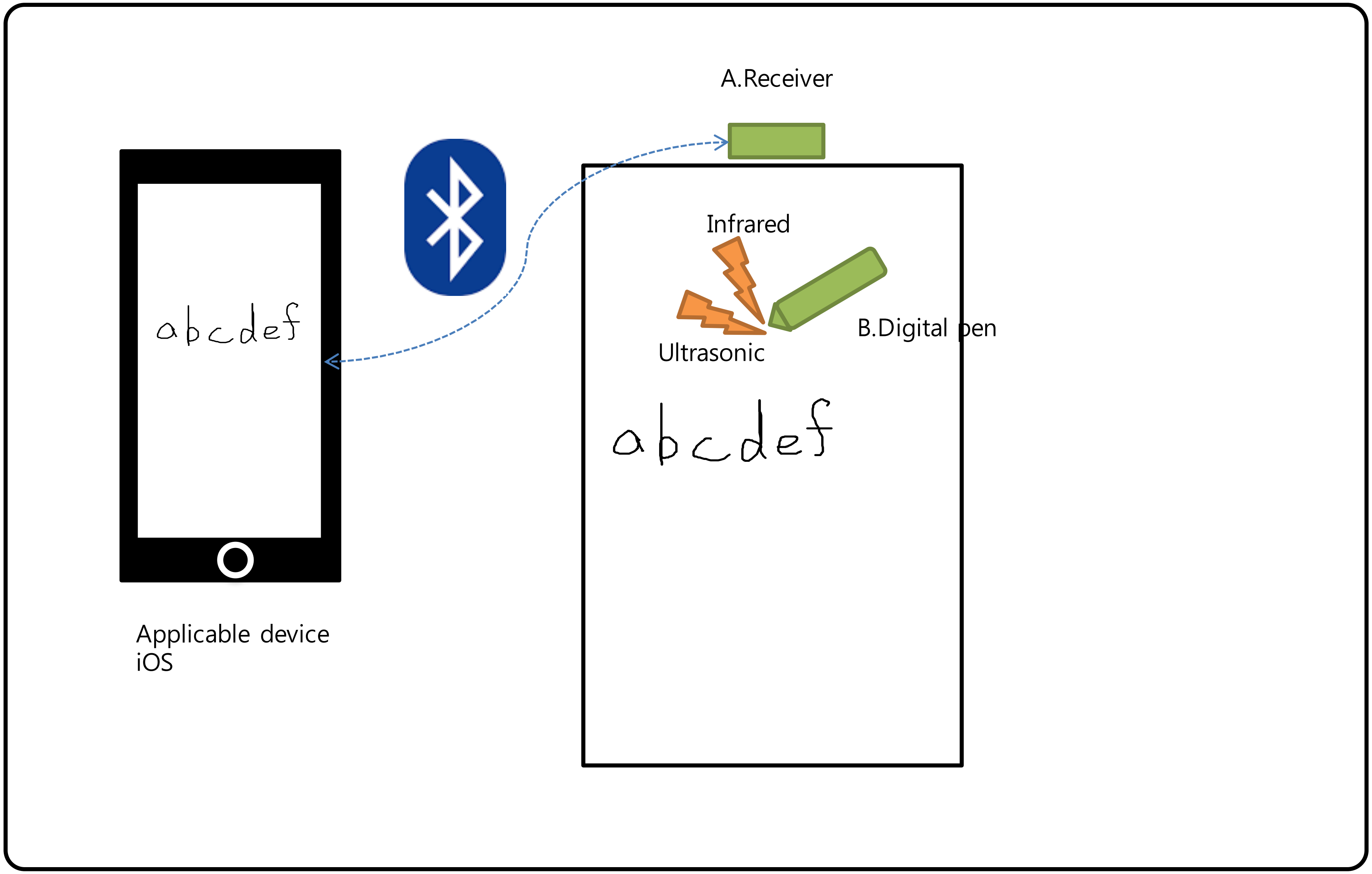
Contents

Contents

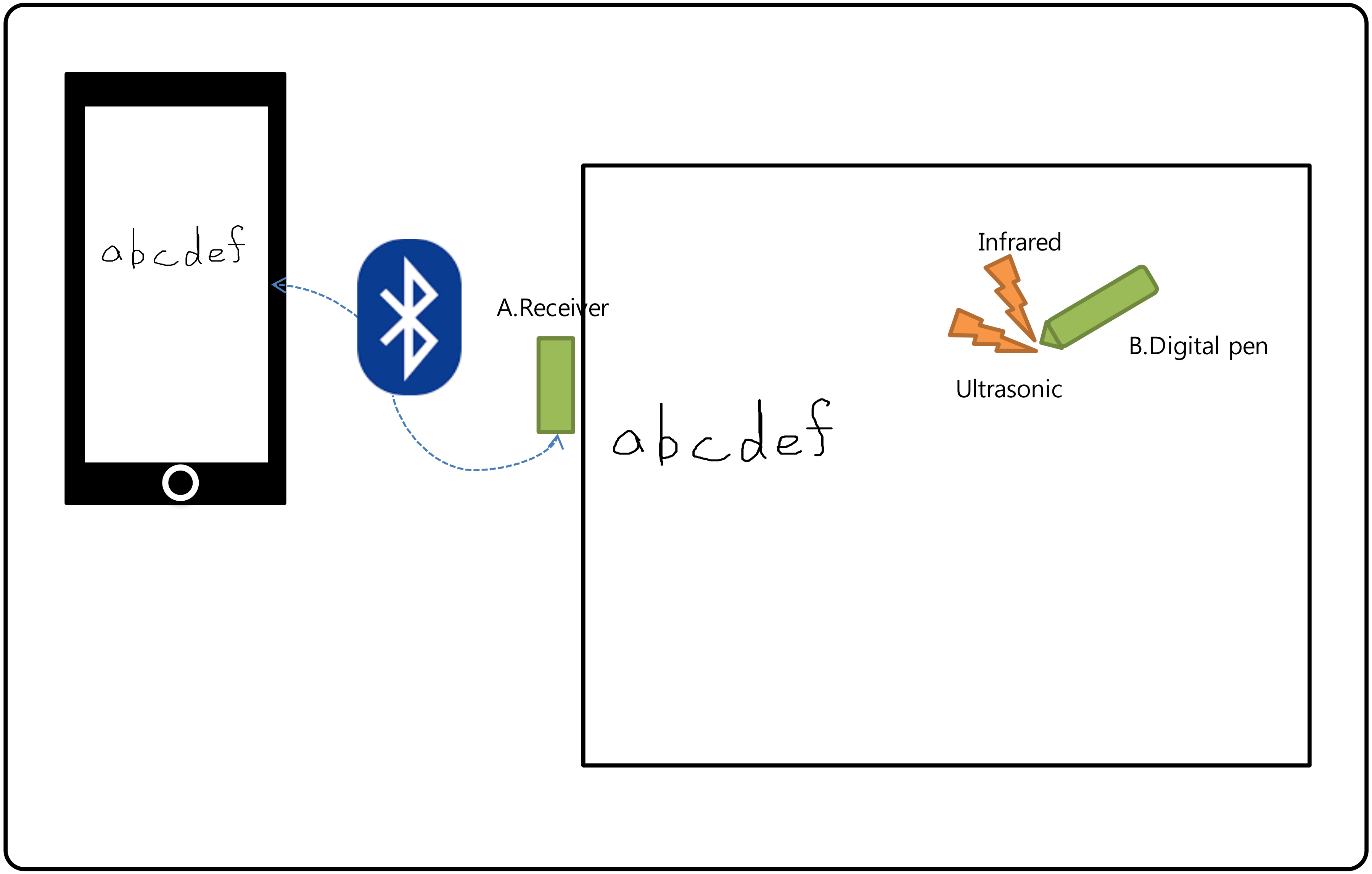
Concept > PNF Hardwares

Model	Devices	Connection	Writing	Image
eBeam Smart Marker	iPhone,iPod,iPad	Wireless(BlueTooth),	On the whiteboard	

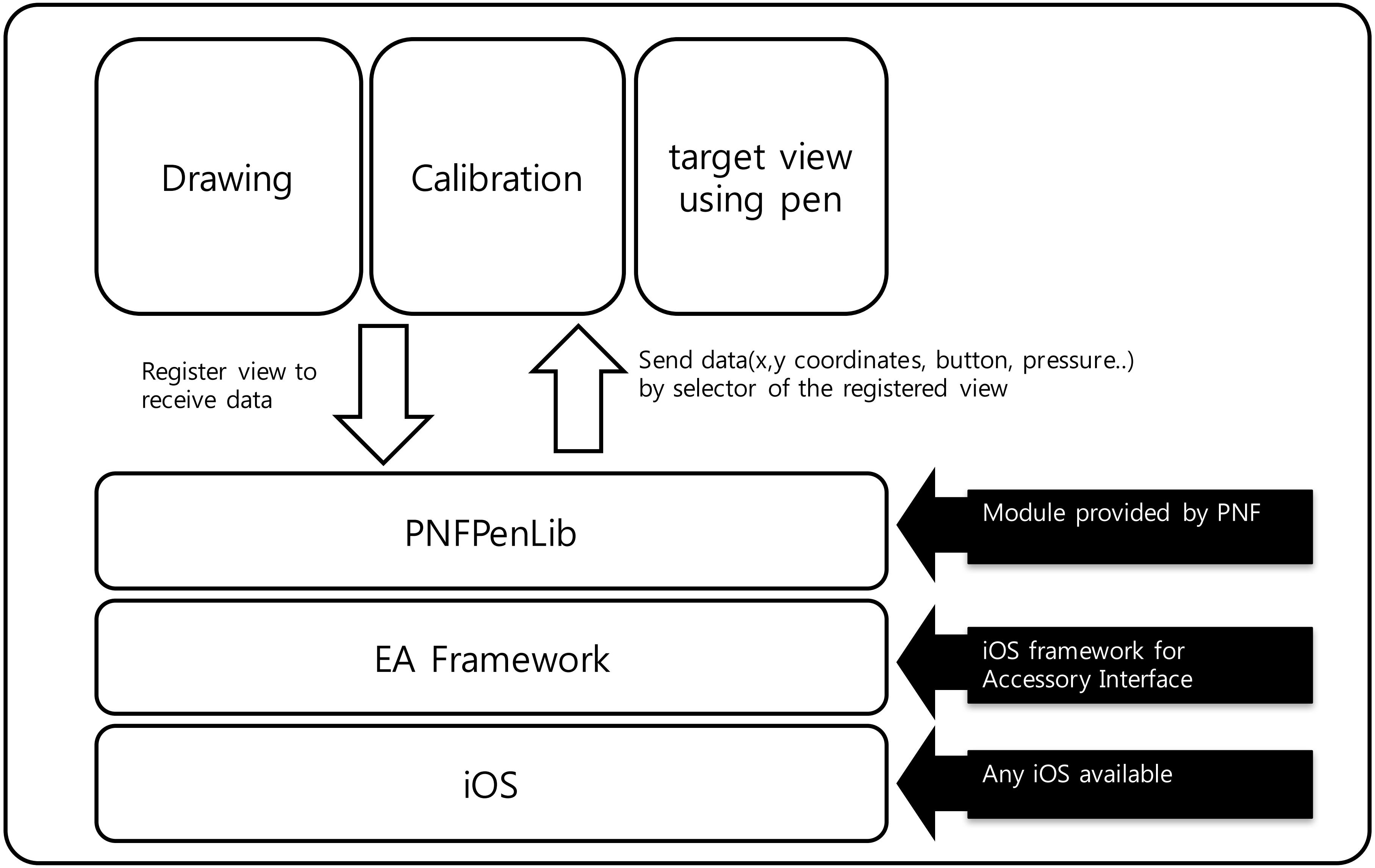
Concept > Hardware Structure (eBeam Smart Marker)



Concept > Hardware Structure (eBeam Smart Marker)



Concept > Software Structure


Drawing
Calibration
target view using pen
Register view to receive data
Send data(x,y coordinates, button, pressure..) by selector of the registered view
PNFPenLib
Module provided by PNF
EA Framework
iOS framework for Accessory Interface
iOS
Any iOS available

Contents

Development > Project setting

- Add PNFModule folder of the sample soures into your project

Development > Components of Test Sample (PenTest)

※ $(SrcHome) : [unZipped folder]/

Folder		File	Description
$(SrcHome)/PenTest/	./	main.m	
		PenTest-Info.plist	
		PenTest-Prefix.pch	
		AppDelegate.h .m	
		ViewController.h .m .xib	Main controller
		BTNameChangeViewController.m	Smart marker name change.
	DrawView/	DrawView.h .m	Drawing lines according to the coordinate from pen.
		DrawViewController.h .m .xib	
$(SrcHome)/Common/	Calibration/	MarkerCalibrationViewController.m .xib	2 points calibration view(eBeam Smart marker)
	Common/	Toast+UIView.h .m	Shows error information about Pen.
		UIImage+ImageNamed.m	Load image data
		Common.h	Default Calibration value
	PNFModule/	libPNFPenLib.a	Standard library
		PNFDefine.h	Constants
		PNFPenLib.h	Interfaces
		PNFPenLibExtension.h	Interfaces
	PNFStrokePoint/	PNFStrokePoint.h .m	Objects for drawings
	Resource/		

Development > Reference

● PNFPenLibExtension Class

Inherits from	NSObject
Declared in	PNFPenLibExtension.h

➢ Overview

PNFPenLibExtension is the class of PNFPenLib Library to manage the information of device , make calibrated coordinates and tranfer it to the other classes.

➢ Members

ptRaw			
Type	CGPoint	Property	readonly
Description	Coordinates before calibrating		
Range	0 ~ 6500		
Device	eBeam Smart Marker		
Usage			

Development > Reference

ptConv			
Type	CGPoint	Property	readonly
Description	Calibrated coordinates		
Range	According to the target view size		
Device	eBeam Smart Marker		
Usage			

PenStatus			
Type	int	Property	readonly
Description	Where pentip is pressed or not		
Range	PEN_DOWN : Pentip down PEN_MOVE : Move with Pentip down PEN_UP :Pentip up		
Device	eBeam Smart Marker		
Usage			

Development > Reference

StationPosition			
Type	int	Property	readonly
Description	Current position of eBeam Smart Marker station.		
Range	DIRECTION_LEFT DIRECTION_RIGHT DIRECTION_TOP DIRECTION_BOTTOM DIRECTION_BOTH (defined in PNFDefine.h)		
Device	eBeam Smart Marker		

Usage	`[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(PenCallBackFunc:) name:@"PNF_MSG" object:nil];`
	
	`....`
	`....`
	`-(void) PenCallBackFunc:(NSNotification *)call {`
	`if ([szS isEqualToString:@"CHANGE_DEVECE_POSITION"] || [szS isEqualToString:@"CHANGE_DEVECE_POSITION_FIRST"]) {`
	`        if (self.penController.StationPosition  == DIRECTION_LEFT)`
	`            self.position = @"Left";`
	`        else if (self.penController.StationPosition  == DIRECTION_RIGHT)`
	`            self.position = @"Right";`
	`        else if (self.penController.StationPosition  == DIRECTION_TOP)`
	`            self.position = @"Top";`
	`        else if (self.penController.StationPosition  == DIRECTION_BOTTOM)`
	`           self.position = @"Bottom";`
	`        else`
	`            self.position = @"Both";`
	`}`		

bStopped			
Type	BOOL	Property	readonly
Description	Whether Pause is set or not If it is set, Pen data is not transferred to target view.		
Range	Yes / No		
Device	eBeam Smart Marker		

Usage	[m_PenController stopPen]; // set pause
	NSLog(@"%@", m_PenController.bStopped ? @"YES",@"NO");
	/// display YES
	
	[m_PenController restartPen]; // release pause
	NSLog(@"%@", m_PenController.bStopped ? @"YES",@"NO");
	/// display NO		

Development > Reference

AudioMode			
Type	Int	Property	readonly
Description	Audio Mode of Smart Marker		

Range	YES = beep only
	NO = beep + voice		

Device	eBeam Smart Marker		
Usage			

Volume			
Type	Int	Property	readonly
Description	Audio volume of Smart Marker		
Range	0 ~ 255 0 = loud 255 = slient		
Device	eBeam Smart Marker		
Usage			

Development > Reference

battery_station			
Type	Int	Property	readonly
Description	Battery status of sensor		
Range	0 ~ 100		
Device	eBeam Smart Marker		
Usage			

battery_pen			
Type	Int	Property	readonly
Description	Battery status of pen		
Range	• Smart Marker 0 = High Else = Low		
Device	eBeam Smart Marker		
Usage			

Development > Reference

➢ Methods

BLEInit		
Description	Start to communicate with device	
out		
input	N/A	
Device	eBeam Smart Marker	

Usage	-(void) viewDidLoad
	{
	    .....
	     self.penController = [[[PNFPenLibExtension alloc] init] autorelease];
	     [self.penController setDefaultModelCode:EbeamSmartMarkerBLE];
	     [self.penController setProjectiveLevel:4];
	     [self.penController fixStationPosition:DIRECTION_LEFT];
	     [self.penController BLEInit];
	     [self.penController setRetObj:self];
	     [self.penController setRetObjForEnv:self];
	    ......
	
	}	

Development > Reference

BLEConnect		
Description	Connect to communicate with device	
out	int	CONNECTED : success FIRST_DATA_RECV : first data read SESSION_CLOSED: receiving error (should reconnect the device) (Define in PNFDefine.h)
input	N/A	
Device	eBeam Smart Marker	
Usage	[self.penController BLEConnect:peripheral];	

Development > Reference

BLEDisconnectClicked		
Description	Disconnect device	
out	Void	
input	N/A	
Device	eBeam Smart Marker	
Usage	[self.penController BLEDisconnect];	

Development > Reference

setRetObj		
Description	Set an object to receive the pen data The object should have “-(void) PenHandler:(id) sender{}”	
Out	Void	
input	NSObject*	Object pointer to receive the pen data
Device	eBeam Smart Marker	

Usage	-(void) viewDidLoad
	{
	    .....
	    self.penController = [[[PNFPenLibExtension alloc] init] autorelease];
	    [self.penControllersetRetObj:self];
	    ……
	}	

getRetObj		
Description	Return registered object to receive pen data	
Out	NSObject*	
Input	Void	
Device	eBeam Smart Marker	
Usage	[self.penController getRetObj:self];	

Development > Reference

setRetObjForEnv		
Description	Set an object to receive the pen data for environment The object should have "-(void) PenHandlerEnv:(NSArray*)info {}"	
out	Void	
input	NSObject*	Object pointer to receive the pen data for environment
Device	eBeam Smart Marker	

Usage	-(void) viewDidLoad
	{
	    .....
	    self.penController = [[[PNFPenLibExtension alloc] init] autorelease];
	
	    [self.penController setRetObj:self];
	    [self.penController setRetObjForEnv:self];
	    ......
	
	}
	-(void) PenHandlerEnv:(NSArray*)info {
	    // info count = 2
	    // ir = Infrared Gap
	    // us = Sensor distance
	    unsigned short ir = [[info objectAtIndex:0] unsignedShortValue];
	    unsigned short us = [[info objectAtIndex:1] unsignedShortValue];
	}	

setCalibrationDataToDevice		
Description	Set data for calibration with position	
out	Void	
input	CGRect	square which consists of calibrated coordinates
	Float	Margin between displayed point and edge of screen
	CGPoint[]	Original points
Device	eBeam Smart Marker	
Usage	// CGPoint m_CaResultPoint[4]; //4 points …………………….. [m_PenController setCalibrationDataToDevice:DEVICE_DIRECTION CalibPoint:m_CalResultPoint]];	

Development > Reference

setCalibrationData		
Description	Set data for calibration	
out	Void	
input	CGRect	square which consists of calibrated coordinates
	Float	Margin between displayed point and edge of screen
	CGPoint[]	Original points
Device	eBeam Smart Marker	

Usage	`// CGPoint  m_CaResultPoint[4]; //4 points`
	`………………………`
	
	`[m_PenController setCalibrationData:[m_calView bounds]`
	`        GuideMargin:0`
	`        CalibPoint:m_CalResultPoint]];`	

setProjectiveLevel		
Description	Set calibration points	
out	Void	
input	Int	
Device	eBeam Smart Marker	

Usage	-(void) viewDidLoad
	{
	    .....
	    [self.penController setDefaultModelCode:EbeamSmartMarkerBLE];
	    [self.penController setProjectiveLevel:4];
	
	    [self.penController setRetObj:self];
	    ......
	
	}	

Development > Reference

changeAudioMode		
Description	Change Audio mode of Smart Marker	
Out	Void	
Input	BOOL	Yes:/No
Device	eBeam Smart Marker	
Usage	[penController changeAudioMode:YES]; -> Change to beep only [penController changeAudioMode:NO]; -> change to beep and voice	

changeVolume		
Description	Change audio volume	
Out	Void	
Input	int	0 ~ 255
Device	eBeam Smart Marker	
Usage	[penController changeVolume:0]; -> max [penController changeVolume:255]; -> min	

ReadQ		
Description	Read one data from read Queue	
Out	NSDictionary	
Input	Void	
Device	eBeam Smart Marker	

Usage	NSDictionary* dic = [penController ReadQ];
	CGPoint ptRaw = [[dic objectForKey:@"ptRaw"] CGPointValue];
	CGPoint ptConv = [[dic objectForKey:@"ptConv"] CGPointValue];
	int PenStatus =[[dic objectForKey:@"PenStatus"] intValue];
	int Temperature = [[dic objectForKey:@"Temperature"] intValue];
	int modelCode = [[dic objectForKey:@"modelCode"] intValue];
	int SMPenFlag = [[dic objectForKey:@"SMPenFlag"] intValue];
	int SMPenState = [[dic objectForKey:@"SMPenState"] intValue];
	int pressure = [[dic objectForKey:@"pressure"] intValue];	

Development > Reference

RemoveQ		
Description	Delete one data from read Queue	
Out	Void	
Input	Void	
Device	eBeam Smart Marker	
Usage	[penController removeQ];	

ClearQ		
Description	Clear all data from read Queue	
Out	Void	
Input	Void	
Device	eBeam Smart Marker	
Usage	[penController ClearQ];	

Development > Reference

StartReadQ		
Description	Read Pen mode through Read Queue	
Out	Void	
Input	Void	
Device	eBeam Smart Marker	
Usage	(see code below)	

```
[penController StartReadQ];
.....
-(void) runReadThread {
    @autoreleasepool {
        while (1) {
            if (readThreadStop) {
                break;
            }

            if ([[UIApplication sharedApplication] isIgnoringInteractionEvents]) {
                [NSThread sleepForTimeInterval:0.02];
                continue;
            }

            NSDictionary* dic = [self.penController ReadQ];
            if(dic) {
                [self performSelectorOnMainThread:@selector(PenHandlerWithDictionary:) withObject:dic waitUntilDone:YES];
                [self.penController RemoveQ];
            }
            else {
                [NSThread sleepForTimeInterval:0.02];
            }
        } // while (1) {
    }
}
```

Development > Reference

EndReadQ		
Description	Read Pen mode through Notification	
Out	Void	
Input	Void	
Device	eBeam Smart Marker	
Usage	see below	

```
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(PenHandlerWithMsg:) name:@"PNF_PEN_READ
_DATA" object:nil];
……
-(void) PenHandlerWithMsg:(NSNotification*) note {
    NSDictionary* dic = [note object];
    if ([self.penController getRetObj] != self)
        return;
    [self PenHandlerWithDictionary:dic];
}
-(void) PenHandlerWithDictionary:(NSDictionary*) dic {
    int PenStatus  = [[dic objectForKey:@"PenStatus"] intValue];
    CGPoint ptRaw = [[dic objectForKey:@"ptRaw"] CGPointValue];
    CGPoint ptConv = [[dic objectForKey:@"ptConv"] CGPointValue];
    int Temperature = [[dic objectForKey:@"Temperature"] intValue];
    int modelCode  = [[dic objectForKey:@"modelCode"] intValue];
    int SMPenFlag  = [[dic objectForKey:@"SMPenFlag"] intValue];
    int SMPenState = [[dic objectForKey:@"SMPenState"] intValue];
    int press = [[dic objectForKey:@"pressure"] intValue];
    [self PenHandlerWithArgs:ptRaw
                     ptConv:ptConv
                  PenStatus:PenStatus
                Temperature:Temperature
                  ModelCode:modelCode
                  SMPenFlag:SMPenFlag
                 SMPenState:SMPenState
                   Pressure:press];
}
```

➢ Overview

Create and initialize object PNFPenLibExtension

➢ Example

1. Create PNFPenLibExtension object
```
m_PenController = [[PNFPenLibExtension alloc] init];
```

2. Appoint the calibration points
```
[m_PenController setDefaultModelCode:EbeamSmartMarkerBLE];
[m_PenController setProjectiveLevel:4];      //4 points
[m_PenController fixStationPosition:DIRECTION_LEFT];
[m_PenController BLEInit];
```

3. Set object to receive data
```
[m_PenController setRetObj:self];
[m_PenController setRetObjForEnv:self];
```

example source: ViewController.h ViewController.m

Development > Guide > Receive data from library

➢ Overview

Internally PNFPenController is supposed to call selector named as "PenHandler" of object set by "setRetObj" whenever the pen moves.

➢ Example

```
-(void) PenHandler:(id)sender {
    // deprecated
}
-(void) ReadThreadStart { // if [penController StartReadQ];
    [self addDebugText:@"ReadThreadStart"];
    if (readThread == nil) {
        readThread = [[NSThread alloc] initWithTarget:self
                                    selector:@selector(runReadThread) object:self];
        readThreadStop=NO;
        readThreadPause=NO;
        [readThread start];

    }
    if (self.penController) {
        [self.penController StartReadQ];
    }
}
-(void) PenHandlerWithMsg:(NSNotification*) note {// if [penController EndReadQ];
    NSDictionary* dic = [note object];
    if ([self.penController getRetObj] != self)
        return;
    [self PenHandlerWithDictionary:dic];
}
```

example source: ViewController.h ViewController.m

➢ Example

```
-(void) runReadThread {// if [penController StartReadQ];
    @autoreleasepool {
        while (1) {
            if (readThreadStop) {
                break;
            }

            if ([[UIApplication sharedApplication] isIgnoringInteractionEvents]) {
                [NSThread sleepForTimeInterval:0.02];
                continue;
            }

            NSDictionary* dic = [self.penController ReadQ];
            if(dic) {
                [self performSelectorOnMainThread:@selector(PenHandlerWithDictionary:) withObject:dic waitUntilDone:YES];
                [self.penController RemoveQ];
            }
            else {
                [NSThread sleepForTimeInterval:0.02];
            }
        } // while (1) {
    }
}
-(void) ReadThreadOff {// if [penController StartReadQ];
    [self addDebugText:@"ReadThreadOff"];
    readThreadStop = YES;
    [NSThread sleepForTimeInterval:0.2];
    if (readThread) {
        [readThread cancel];
        [readThread release];
        readThread = nil;
    }
    if (self.penController) {
        [self.penController EndReadQ];
    }
}
```

example source: ViewController.h ViewController.m

Development > Guide > Receive data from library

➢ Example

```
-(void) PenHandlerWithDictionary:(NSDictionary*)  dic {
    int PenStatus   = [[dic objectForKey:@"PenStatus"] intValue];
    CGPoint ptRaw = [[dic objectForKey:@"ptRaw"]  CGPointValue];
    CGPoint ptConv = [[dic objectForKey:@"ptConv"]  CGPointValue];
    int Temperature = [[dic objectForKey:@"Temperature"]  intValue];
    int modelCode  = [[dic objectForKey:@"modelCode"]  intValue];
    int SMPenFlag  = [[dic objectForKey:@"SMPenFlag"]  intValue];
    int SMPenState = [[dic objectForKey:@"SMPenState"]  intValue];
    int press  = [[dic objectForKey:@"pressure"]  intValue];
    [self PenHandlerWithArgs:ptRaw
                     ptConv:ptConv
                  PenStatus:PenStatus
                Temperature:Temperature
                  ModelCode:modelCode
                  SMPenFlag:SMPenFlag
                 SMPenState:SMPenState
                   Pressure:press];
}
-(void) PenHandlerWithArgs:(CGPoint) Arg_ptRaw  ptConv:(CGPoint)  Arg_ptConv  PenStatus:(int) Arg_PenStatus
              Temperature:(int) Arg_Temperature ModelCode:(int)  Arg_modelCode
               SMPenFlag :(int)  Arg_SMPenFlag SMPenState:(int) Arg_SMPenState
                 Pressure:(int)  Arg_pressure {
           CGPoint  ptDrawing;
            switch  (Arg_PenStatus) {
                  case PEN_DOWN:
                      break;
                  case PEN_MOVE:
                      break;
                  case PEN_UP:
                      break;
              }
           }
           ptDrawing  =  m_PenController.ptConv  ;
}
```

example source: ViewController.h ViewController.m

Development > Guide > Receive message from library

➢ Overview

Information of device status is sent by notification named as “PNF_LOG_MSG”.

➢ Example

1. Add Notification

```
 [[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(FreeLogMsg:)
name:@“PNF_LOG_MSG" object:nil];
```

1. Handler for Message

```
-(void) FreeLogMsg:(NSNotification *) note
{
        NSString * szS = (NSString *) [note object];
    if ([szS isEqualToString :@"FAIL_LISTENING"] ) {
     }
    else if ([szS isEqualToString:@"CONNECTED"]) {
     }
    else if ([szS isEqualToString:@"INVALID_PROTOCOL"]) {
        return;
    }
    else if ([szS isEqualToString:@"SESSION_CLOSED"]) {
    }
    else if ([szS isEqualToString:@"PEN_RMD_ERROR"]) {

    }
    else if ([szS isEqualToString:@"FIRST_DATA_RECV"]) {
    }
}
```

Log String Message	Description
CONNECTED	Device is connected
NOT_CONNECTED	Device is disconnected
FAIL_LISTENING	Fail to receive. Need to reconnect.
INVALID_PROTOCOL	Invalid hardware
SESSION_CLOSED	Session is disconnected
FIRST_DATA_RECV	First data is received after connecting
PEN_RMD_ERROR	Abnormal drawing data

example source : ViewController.h ViewController.m

Development > Guide > Receive message from library

➢ Overview

Information of device status is sent by notification named as “PNF_MSG”.

➢ Example

1. Add Notification
   [[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(PenCallBackFunc:) name:@“PNF_MSG" object:nil];

1. Handler for Message

```
-(void) PenCallBackFunc:(NSNotification *)call {
    NSString * szS = (NSString *) [call object];
    if([szS isEqualToString:@"BATTERY_INFO"]) {
        battery[0] = self.penController.battery_station;
        battery[1] = self.penController.battery_pen;
        [mTableView reloadData];
    }
    else if([szS isEqualToString:@"NEW_PAGE"] || [szS isEqualToString:@"DUPLICATE_PAGE"]) {
        [self addDebugText:szS];
    }
    else if ([szS isEqualToString:@"CHANGE_DEVECE_POSITION"] ||
             [szS isEqualToString:@"CHANGE_DEVECE_POSITION_FIRST"]) {
        if (self.penController.StationPosition == DIRECTION_LEFT)
            self.position = @"Left";
        else if (self.penController.StationPosition == DIRECTION_RIGHT)
            self.position = @"Right";
        else if (self.penController.StationPosition == DIRECTION_TOP)
            self.position = @"Top";
        else if (self.penController.StationPosition == DIRECTION_BOTTOM)
            self.position = @"Bottom";
        else
            self.position = @"Both";

        [mTableView reloadData];
    }
}
```

Log String Message	Description
BATTERY_INFO	Battery information
NEW_PAGE	Button smart marker
DUPLICATE_PAGE	Long press button smart marker
CHANGE_DEVECE_POSITION	Change device position
CHANGE_DEVECE_POSITION_FIRST	Change device position first

example source : ViewController.h ViewController.m

Development > Guide > Calibration

➢ Overview

Pen coordinates is converted to screen coordinates by projective matrix which is set in the calibration view.

➢ Example

1. create calibration controller
```
MarkerCalibrationViewController* cVController = [[MarkerCalibrationViewController alloc]
initWithNibName:@"MarkerCalibrationViewController" bundle:nil];
```

2. connect Pen controller and calibration controller
```
[cVController SetPenController:self.penController];
```

3. set calibration controller as target view
```
[m_PenController setRetObj:cVController];
```

4. show calibration view
```
[self presentModalViewController:cVController animated:YES];
```

example source : CalibViewController.h CalibViewController.m

Development > Guide > Calibration

➢ Overview

Calibration data is saved automatically by this library.
App need not save the data.

➢ Example

5. Save calibration data

```
/// after click the last calibration point
[m_PenController setCalibrationData:[m_calView bounds]
  GuideMargin:0
  CalibPoint:m_CalResultPoint]];
```

example source : CalibViewController.h CalibViewController.m

Development > Guide > Calibration

➢ Overview

Calibration data is saved automatically by this library.
App need not save the data.

➢ Example

5. Save calibration data

```
/// after click the last calibration point
[m_PenController setCalibrationDataToDevice : DEVICE_DIRECTION
                                  CalibPoint:m_CalResultPoint];
```

example source : CalibViewController.h CalibViewController.m