
Equil

Developer Guide

For OS X

PNF R&D S/W

2015. 05

I. Concept

- Hardware Structure
- Software Structure
- Background knowledge

II. Development

- Project setting
- components of Library
- reference
- Guide

III. Design Guide

IV. Go to App Store

I. Concept

- Hardware Structure
- Software Structure
- Background knowledge



II. Development

- Project setting
- Components of Library
- Reference
- Guide

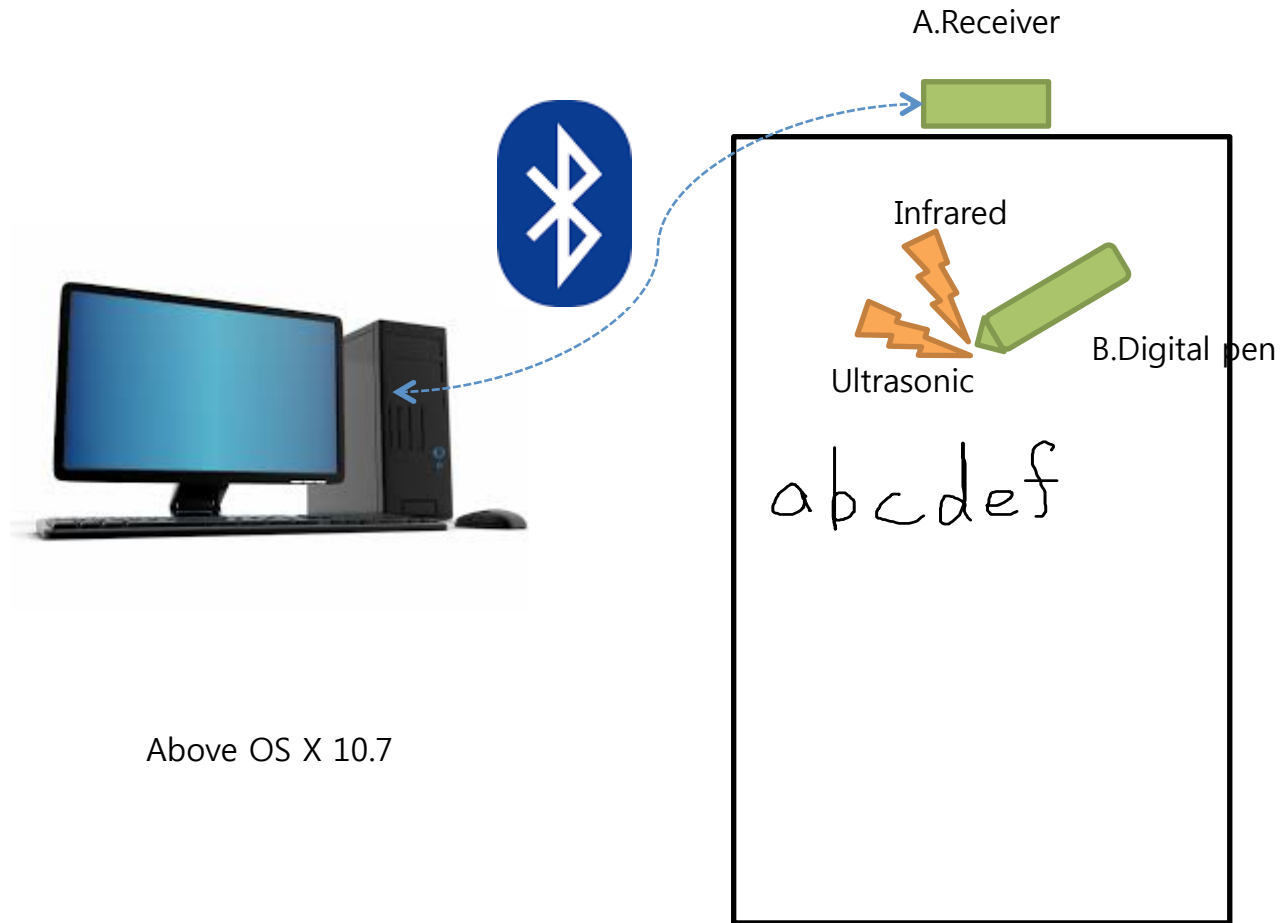
III. Design Guide

IV. Go to App Store

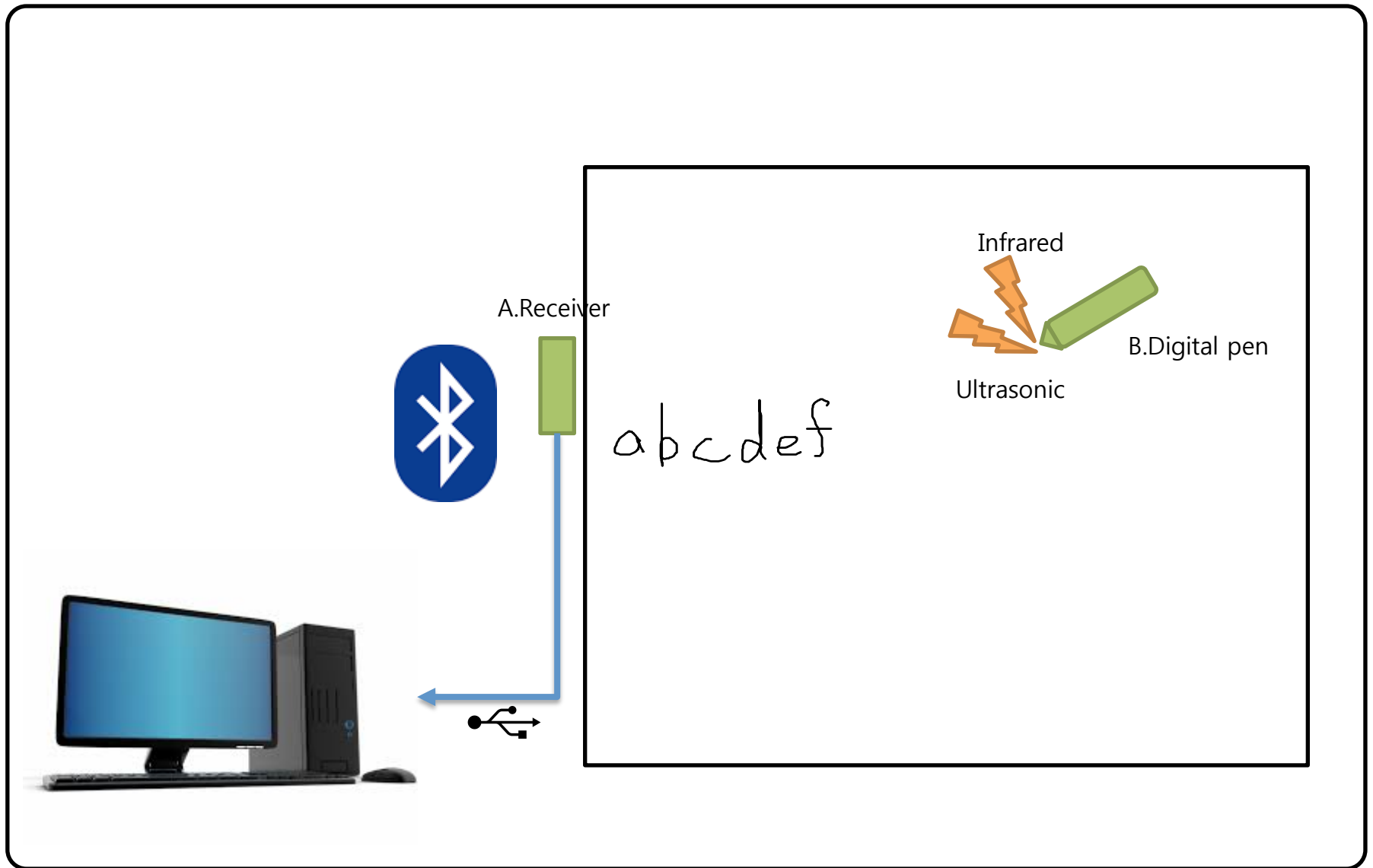
Concept > PNF Hardware

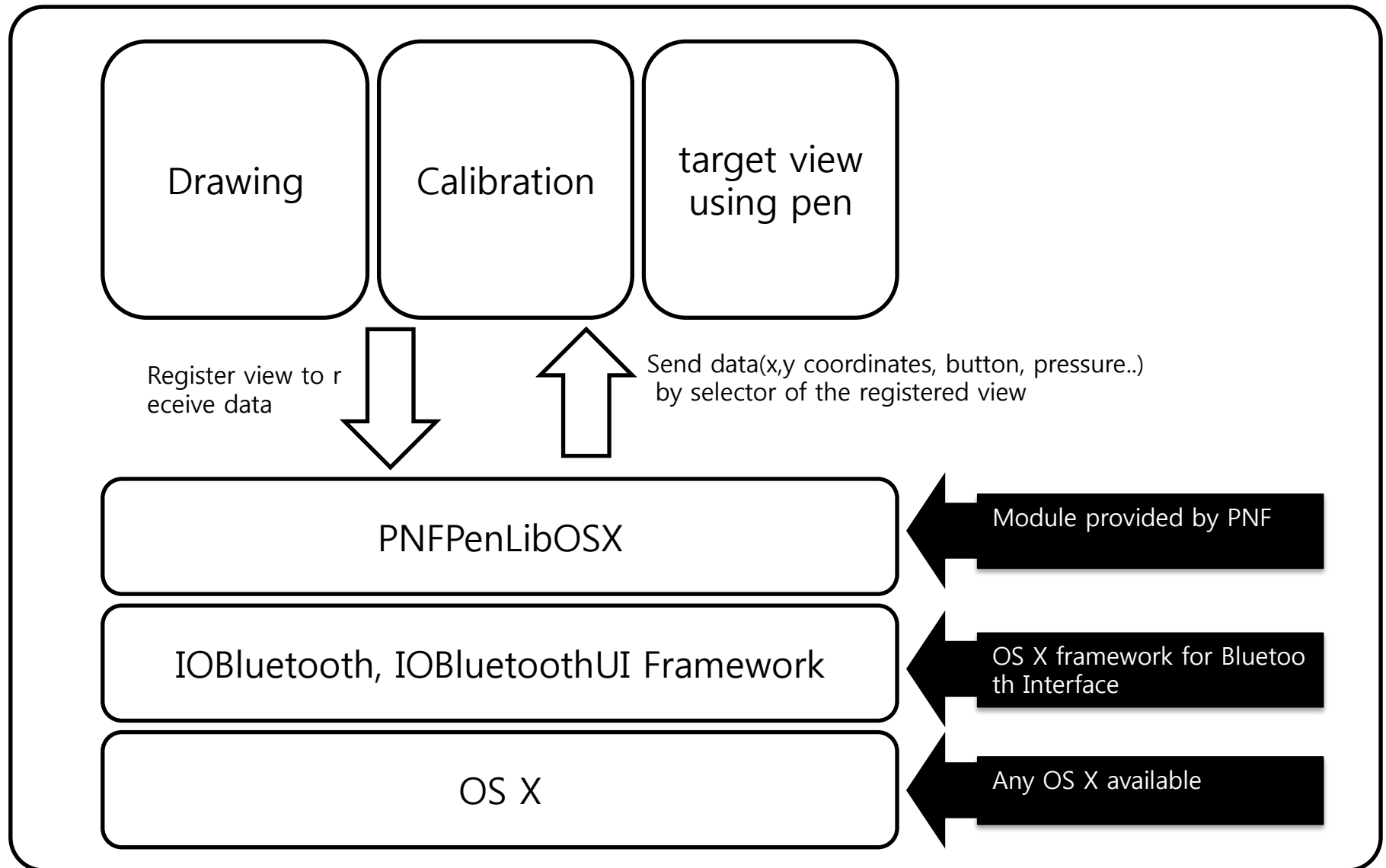
Model	Devices	Connection	Writing	Image
Equil	iPhone,iPod,iPad, Mac,Windows,Andr oid	Wireless(BlueTooth)	On the paper Or desk	
Equil Smartmarker	iPhone,iPod,iPad, Mac,Windows,Andr oid	Wireless(BlueTooth), USB(Windows, OSX)	On the whiteboar d	

Concept > Hardware Structure (Equil Smart Pen)



Concept > Hardware Structure (Equil Smart Marker)

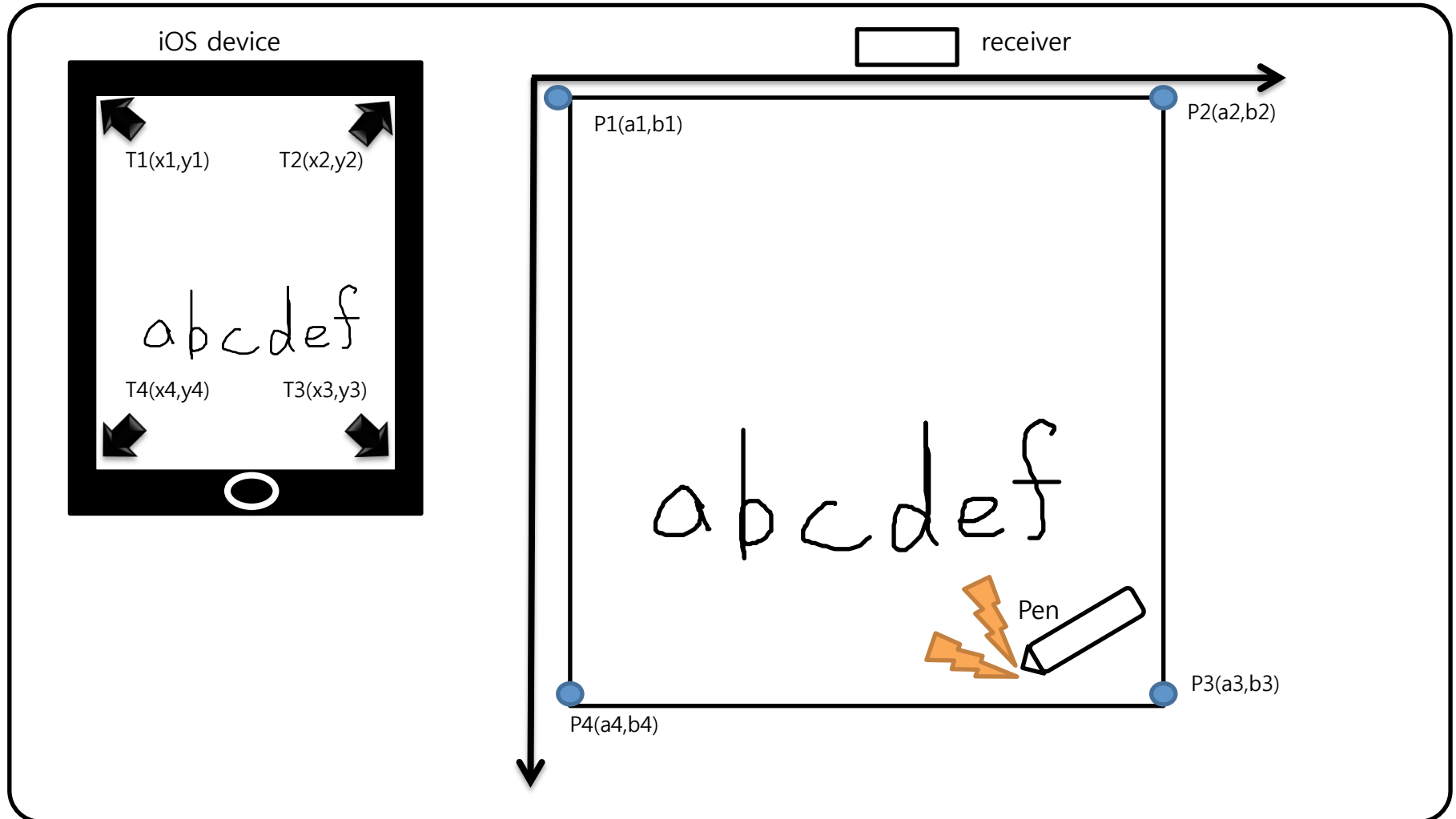




Concept > Background knowledge > Calibration (equil)

Calibration is Mapping the points of paper P1~P4 to coordinates T1~T4 of screen in order to have the image on the screen look the same as the image on the paper.

In case of Equil, assuming that receiver is parallel with paper, just clicking two points(P1,P3) is enough.



➤ Refer to Developer.apple.com

https://developer.apple.com/library/mac/documentation/DeviceDrivers/Reference/IOBluetoothUI/index.html#//apple_ref/doc/uid/TP30000813

https://developer.apple.com/library/mac/documentation/DeviceDrivers/Reference/IOBluetooth/index.html#//apple_ref/doc/uid/TP30000814

I. Concept

- Hardware Structure
- Software Structure
- Background knowledge

II. Development

- Project setting
- Components of Library
- Reference
- Guide

III. Design Guide

IV. Go to App Store

- Add PNFModule folder of the sample sources into your project
- Add IOBluetooth framework into your project
- Add IOBluetoothUI framework into your project
- in case of using Sandbox.
 - com.apple.security.device.usb -> TRUE
 - com.apple.security.device.bluetooth -> TRUE

Development > Components of Test Sample (PenTestOSX)

※ \$(SrcHome) : [unZipped folder]/

Folder		File	Description
\$(SrcHome)/PenTestOSX/	./	main.m	
		PenTest-InfoOSX.plist	
		PenTest-PrefixOSX.pch	
		AppDelegate.h .m	
		MainWindowController.h .m .xib	Main controller
	DrawView/	DrawView.h .m	Drawing lines according to the coordinate from pen.
		DrawViewWindowController.h .m .xib	
\$(SrcHome)/Common/	Calibration/	EquilCalibrationViewController.h .m .xib	
		CalibWindowController.h .m .xib	2 points calibration view(Equil)
	PNFModule/	libPNFPenLib.a	Standard library
		PNFDefine.h	Constants
		PNFPenLib.h	Interfaces
	PNFStrokePoint/	PNFStrokePoint.h .m	Objects for drawings
	Resource/		
	TimeTestView/	TimeTestWindowController.h .m .xib	Class checking pen alive time

● PNFPenController Class

Inherits from	NSObject
Declared in	PNFPenLib.h

➤ Overview

PNFPenController 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	Equil Smart Pen / Marker		
Usage			

ptConv			
Type	CGPoint	Property	readonly
Description	Calibrated coordinates		
Range	According to the target view size		
Device	Equil Smart Pen / 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 PEN_HOVER : Move with Pentip up * Equil Smart Pen only PEN_HOVER_DOWN : Pen button down PEN_HOVER_MOVE : Move with Pen button down (defined in PNFDefine.h)		
Device	Equil Smart Pen / Marker		
Usage			

StationPosition			
Type	enum DEVICE_DIRECTION	Property	readonly
Description	When sensor position is changed		
Range	<pre>enum DEVICE_DIRECTION { DIRECTION_TOP=1, DIRECTION_LEFT, DIRECTION_RIGHT, DIRECTION_BOTTOM }; (defined in PNFDefine.h)</pre>		
Device	Equil Smart Marker		
Usage	<pre>[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(PenCallBackFunc :) name:@"PNF_MSG" object:nil]; if ([szS isEqualToString:@"CHANGE_DEVECE_POSITION"] [szS isEqualToString:@"CHANGE_DEVECE_POSITION_FIRST"]) { [directionTextField setStringValue:@"Left"]; if (self.penController.StationPosition == DIRECTION_LEFT) [directionTextField setStringValue:@"Left"]; else if (self.penController.StationPosition == DIRECTION_RIGHT) [directionTextField setStringValue:@"Left"]; else if (self.penController.StationPosition == DIRECTION_TOP) [directionTextField setStringValue:@"Top"]; else if (self.penController.StationPosition == DIRECTION_BOTTOM) [directionTextField setStringValue:@"Bottom"]; }</pre>		

Temperature			
Type	int	Property	readonly
Description	Temperature		
Range	0~60 (Celsius)		
Device	Equil Smart Pen / Marker		
Usage	<code>[txtTemperature setText::[NSString stringWithFormat:@"%d",m_PenController.Temperature]];</code>		

bConnected			
Type	BOOL	Property	readonly
Description	Whether receiver is connected or not		
Range	Yes / No		
Device	Equil Smart Pen / Marker		
Usage	<pre>if(m_PenController.bConnected) { //// if receiver is connected } else { // if receiver is not connected }</pre>		

bConnectedHID			
Type	BOOL	Property	readonly
Description	Check if connected by USB HID		
Range	Yes / No		
Device	Equil Smart Marker		
Usage	<pre>if(m_PenController.bConnectedHID) //// if receiver is connected { } else // if receiver is not connected { }</pre>		

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	Equil Smart Pen / Marker		
Usage	<pre>[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</pre>		

bExistCalibrationInfo			
Type	BOOL	Property	readonly
Description	Where calibration data file exists or not		
Range	Yes / No		
Device	Equil Smart Pen / Marker		
Usage	<pre> if(m_PenController. bExistCalibrationInfo) { /// } else { // } </pre>		

pressure			
Type	Int	Property	readonly
Description	Pressure value of Equil.		
Range	0 ~ 700		
Device	Equil Smart Pen		
Usage	<pre> int pressureValue = [m_PenController pressure]; </pre>		

modelCode		
Description	Connected device	
Out	Int	2 : Equil Smart Pen 3 : Equil Smart Pen2 4 : Equil Smart Marker
Input	N/A	
Device	Equil Smart Pen / Marker	
Usage	<pre> if (![penController bExistCalibrationInfo]) { [self addDebugText:@"calibration data not exist"]; if (penController.modelCode == 2) { [self addDebugText:@"Equil"]; } [self initCalibration]; } else { [self addDebugText:@"calibration data exist"]; if (penController.modelCode == 2) { [self addDebugText:@"Equil"]; } [penController changeScreenSize:CGRectMake(0, 0, screenSize.width, screenSize.height)]; } [modelCodeTextField setStringValue:[NSString stringWithFormat:@"%d", penController.modelCode]]; </pre>	

MCU1Version			
Type	int	Property	readonly
Description	Version of MCU 1 of receiver and pen 0: 2: the latest F/W version including Pen Alive		
Range	0 , 2		
Device	Equil Smart Pen / Marker		
Usage			

MCU2Version			
Type	Int	Property	readonly
Description	Version of MCU 2 of receiver and pen 0: 2: the latest F/W version including Pen Alive		
Range	0 , 2		
Device	Equil Smart Pen / Marker		
Usage			

HWVersion			
Type	Int	Property	readonly
Description	Version of Hardware of receiver and pen 0: 2: the latest F/W version including Pen Alive		
Range	0 , 2		
Device	Equil Smart Pen / Marker		
Usage			

penAliveSec			
Type	Int	Property	readonly
Description	<ul style="list-style-type: none"> - Remaining time before going to sleep mode (sec) - It is applied only when MCU1Version, MCU2Version, HWVersion All are 2 		
Range	0 ~ 600		
Device	Equil Smart Pen / Marker		
Usage			

AudioMode			
Type	Int	Property	readonly
Description	Audio Mode of Smart Marker		
Range	YES = beep only NO = beep + voice		
Device	Equil Smart Marker		
Usage			

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

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

battery_pen			
Type	Int	Property	readonly
Description	Battery status of pen		
Range	<ul style="list-style-type: none"> Smart Marker 0 = High Else = Low Smart Pen 0 ~ 100 		
Device	Equil Smart Pen / Marker		
Usage			

➤ Methods

startPen		
Description	Start to communicate with device	
out	int	0 : success PNF_E_NOT_CONNECTED : device is not connected PNF_E_INVALID_PROTOCOL: device is invalid PNF_E_FAIL_LISTENING: receiving error (should reconnect the device) (Define in PNFDefine.h)
input	N/A	
Device	Equil Smart Pen / Marker	
Usage	<pre> -(void) windowDidLoad { [penController startPen]; } -(void) FreeLogMsg:(NSNotification *) note { else if ([szS isEqualToString:@"CONNECTED"]) { // pen connected. } else if ([szS isEqualToString:@"FAIL_CONNECTING"]) { [self addDebugText:@"FAIL_CONNECTING"]; [NSTimer scheduledTimerWithTimeInterval:1 target:self selector:@selector(BTErrorProcess) userInfo:nil repeats:NO]; } return; } </pre>	

➤ Methods

startPen	
Usage	<pre>-(void) BTErrProcess { NSAlert* alert = [[[NSAlert alloc] init] autorelease]; [alert addButtonWithTitle:@"Retry"]; [alert addButtonWithTitle:@"Pairing"]; [alert addButtonWithTitle:@"Cancel"]; [alert setMessageText:@"Can not connect pen."]; [alert setAlertStyle:NSInformationalAlertStyle]; [alert beginSheetModalForWindow:self.window modalDelegate:self didEndSelector:@selector(BTErrProcessRet:returnCode:contextInfo:) contextInfo:NULL]; } -(void) BTErrProcessRet:(NSAlert *)alert returnCode:(int)returnCode contextInfo:(void *)contextInfo { NSLog(@"retcode = %d",returnCode); if (returnCode == NSAlertFirstButtonReturn) { NSLog(@"first button try again"); [penController startPen]; } else if(returnCode == NSAlertSecondButtonReturn) { NSLog(@"second button init"); [penController InitBTConnection]; [penController startPen]; } }</pre>

stopPen		
Description	Stop receiving data temporarily Pen data is not transferred to target view.	
out	Void	
input	N/A	
Device	Equil Smart Pen / Marker	
Usage	<pre>[penController stopPen]; // set pause NSLog(@"%@", penController.bStopped ? @"YES",@"NO"); /// display YES [penController restartPen]; // release pause NSLog(@"%@", penController.bStopped ? @"YES",@"NO"); /// display NO</pre>	

Development > Reference

restartPen		
Description	Restart to receive pen data which is stopped by stopPen again	
out	Void	
input	N/A	
Device	Equil Smart Pen / Marker	
Usage	<pre>[penController stopPen]; // set pause NSLog(@"%@", penController.bStopped ? @"YES",@"NO"); /// display YES [penController restartPen]; // release pause NSLog(@"%@", penController.bStopped ? @"YES",@"NO"); /// display NO</pre>	

disconnectPen		
Description	Disconnect bluetooth connection.	
Out	Void	
Input	Void	
Device	Equil Smart Pen / Marker	
Usage	<pre>[penController disconnectPen];</pre>	

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	Equil Smart Pen / Marker	
Usage	<pre>-(void) viewDidLoad { penController = [[PNFPenController alloc] init]; [penController startPen]; [penController setRetObj:self]; }</pre>	

getRetObj		
Description	Return registered object to receive pen data	
Out	NSObject*	
Input	Void	
Device	Equil Smart Pen / Marker	
Usage	<pre>[penController getRetObj];</pre>	

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	Equil Smart Pen / Marker	
Usage	<pre> -(void) viewDidLoad { penController = [[PNFPenController alloc] init]; [penController startPen]; [penController setRetObj: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]; } </pre>	

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	Equil Smart Pen / Marker	
Usage	<pre>// CGPoint m_CaResultPoint[4]; // Equil → 4 points [penController setCalibrationData:[m_calView bounds] // For drawing bounds GuideMargin:0 CalibPoint:m_CalResultPoint]];</pre>	

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

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

ReadQ		
Description	Read on data from read Queue	
Out	NSDictionary	
Input	Void	
Device	Equil Smart Pen / Marker	
Usage	<pre>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];</pre>	

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

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

StartReadQ		
Description	Read Pen mode through Read Queue	
Out	Void	
Input	Void	
Device	Equil Smart Pen / Marker	
Usage	<pre> [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) { } } </pre>	

EndReadQ		
Description	Read Pen mode through Notification	
Out	Void	
Input	Void	
Device	Equil Smart Pen / Marker	
Usage	<pre> [[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]; } </pre>	

InitPenUp		
Description	Enforce Pen-up and initialize library coordinates if there is no pen-up after a certain seconds while drawing.	
Out	Void	
Input	Void	
Device	Equil Smart Marker	
Usage	<pre>-(void) eraseUpTimerCallBack:(NSTimer*)t { [self stopEraseUpTimer]; [m_PenController initPenUp]; [drawView DoMouseUp:di_old_point pressure:100]; }</pre>	

changeScreenSize		
Description	Change the drawing screen size. Calibration coordinates is not change.	
Out	Void	
Input	CGRect	Ex) CGRectMake(0, 0,640, 960)
Device	Equil Smart Pen / Marker	
Usage	[penController changeScreenSize:CGRectMake(0, 0,640, 960)];	

InitBTConnection		
Description	Initialize bluetooth connection info of pen library.	
Out	Void	
Input	Void	
Device	Equil Smart Pen / Marker	
Usage	[penController initBTConnection]; [penController startPen];	

➤ Overview

Create and initialize object PNFPenController

➤ Example

1. Create PNFPenController object
`penController = [[PNFPenController alloc] init];`
2. Set object to receive data
`[penController setRetObj:self];`
3. Connect pen
`[penController startPen];`

➤ Overview

To recognize which device is connected.
After the device is connected and model code is sent after 1.5 sec.

➤ Example

```
-(void) checkPenConnect {
.... // After the device is connected
        if (penController.bConnected) {
            m_penConnctectedStatus = YES;
            [NSTimer scheduledTimerWithTimeInterval:1.5f
                target:self
                selector:@selector(lazyCheckCalibration)
                userInfo:nil
                repeats:NO];
        }
    .....
}

-(void) lazyCheckCalibration {
if (![penController bExistCalibrationInfo]) {
    [self addDebugText:@"calibration data not exist"];
    if (penController.modelCode == 2) {
        [self addDebugText:@"Equil"];
    }
    [self initCalibration];
}
else {
    [self addDebugText:@"calibration data exist"];
    if (penController.modelCode == 2) {
        [self addDebugText:@"Equil"];
    }
    [penController changeScreenSize:CGRectMake(0, 0, screenSize.width, screenSize.height)];
}
[modelCodeTextField stringValue:[NSString stringWithFormat:@"%d", penController.modelCode]];...
}
```

example source: MainWindowController.h MainWindowController.m

➤ 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

```
-(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

```
-(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;
        case PEN_HOVER:
            break;
        default:
            break;
    }
    ptDrawing = m_PenController.ptConv ;
}
```

➤ 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"]) {
    }
    }
    else if ([szS isEqualToString:@"DOUBLE_CLICK"]) {
    }
    else if ([szS isEqualToString:@"CLICK"]) {
    }
    else if ([szS isEqualToString:@"Gesture Circle Clockwise"]) {
    }
    else if ([szS isEqualToString:@"Gesture Circle CounterClockwise"]) {
    }
}
```

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
* Equil Smart Pen only	
DOUBLE_CLICK	Equil pen button double click
CLICK	Equil pen button click
Gesture Circle Clockwise	Equil pen circle clockwise gesture
Gesture Circle CounterClockwise	Equil pen circle counter clockwise gesture

➤ 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"]) {
        self.position = @"Left";
        if (self.penController.StationPosition == UIInterfaceOrientationLandscapeLeft)
            self.position = @"Left";
        else if (self.penController.StationPosition == UIInterfaceOrientationLandscapeRight)
            self.position = @"Left";
        else if (self.penController.StationPosition == UIInterfaceOrientationPortrait)
            self.position = @"Top";
        else if (self.penController.StationPosition == UIDeviceOrientationPortraitUpsideDown)
            self.position = @"Bottom";
        [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

➤ Overview

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

➤ Example

```
- (IBAction)calibrationClicked:(id)sender{
    .....
    self.calibController = [[[CalibWindowController alloc] initWithWindowNibName:@"CalibWindowController"] autorelease];
    self.calibController.calibViewController.delegate = self;
    self.calibController.calibViewController.curScreenSize = screenSize;
    [self.calibController.calibViewController SetPenController:penController];
    [penController setRetObj:self.calibController.calibViewController];
    [[NSApplication sharedApplication] beginSheet:calibController.window
                                     modalForWindow:self.window
                                     modalDelegate:nil
                                     didEndSelector:nil
                                     contextInfo:nil];
}

-(void) closeCalibViewController {
    [penController setRetObj:self];
    [penController setRetObjForEnv:self];
    [NSTimer scheduledTimerWithTimeInterval:0.1
                                     target:self
                                     selector:@selector(closeCalibViewControllerImpl)
                                     userInfo:nil
                                     repeats:NO];
}

-(void) closeCalibViewControllerImpl {
    [NSApp endSheet:self.calibController.window];
    [self.calibController.window orderOut:nil];
    self.calibController = nil;
}
```

example source : CalibViewController.h CalibViewController.m

➤ 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
   [penController SetClibrationData:[m_calView bounds] GuideMargin:0
                                   CalibPoint:m_CalResultPoint];
```

➤ Overview

Pen goes to sleep mode, when pentip is not clicked and pen button is not pressed for 10 minutes. To wake up pen, user must press pen button. In this case, apps need to show the message which inform users to press the pen button.

➤ Example

1. Set object to receive environment data

```
-(void) windowDidLoad {  
    .....  
    penController = [[PNFPenController alloc] init];  
    [penController setRetObjForEnv:self];  
    isRecvEnvDataFirst = YES;  
    isFirstPenSleepOldDevice = NO;  
}
```

2. Environment data handler implementation

```
-(void) PenHandlerEnv:(NSArray*)info {  
    if (penController.MCU1Version >= 2 && penController.MCU2Version >= 2 && penController.HWVersion >= 2) {  
        if (penController.penAliveSec > 0) {  
            [self closePenSleepView];  
        }  
    }  
    if (isRecvEnvDataFirst) {  
        isRecvEnvDataFirst = NO;  
        [self penIdleTimerStop];  
        penSleepCheckTimer = [[NSTimer scheduledTimerWithTimeInterval:5  
                                                                    target:self  
                                                                    selector:@selector(onTimerForPenAlive:)  
                                                                    userInfo:nil  
                                                                    repeats:YES] retain];  
        savePenSleepRemainingTime = [[NSDate date] timeIntervalSince1970] + 600;  
        savePenAliveSec = 600;  
    }  
}
```

➤ Example

```
-(void) PenHandler:(id)sender {
    isRecvEnvDataFirst = YES;
    [self penIdleTimerStop];
    [self closePenSleepView];
    .....
}
-(void) onTimerForPenAlive:(NSTimer *)timer {
    long curTime = [[NSDate date] timeIntervalSince1970];
    if (penController.MCU1Version >= 2 && penController.MCU2Version >= 2 && penController.HWVersion >= 2) {
        if (penController.penAliveSec == 0) {
            [self showPenSleepView];
            [self penIdleTimerStop];
            return;
        }
        if (penController.penAliveSec != 0 && savePenAliveSec != penController.penAliveSec) {
            savePenAliveSec = penController.penAliveSec;
            savePenSleepRemainingTime = curTime + penController.penAliveSec;
        }
    }
    else {
        if (!isFirstPenSleepOldDevice) {
            isFirstPenSleepOldDevice = YES;
            savePenSleepRemainingTime = curTime - 10;
        }
    }
    if (savePenSleepRemainingTime < curTime) {
        [self showPenSleepView];
        [self penIdleTimerStop];
    }
}
```


➤ Example

```
-(void) FreeLogMsg:(NSNotification *) note {
    NSString * szS = (NSString *) [note object];
    if ([szS isEqualToString:@"CONNECTED"]) {
        .....
        isFirstPenSleepOldDevice = NO;
    }
    else if ([szS isEqualToString:@"SESSION_CLOSED"]) {
        [self penIdleTimerStop];
        isFirstPenSleepOldDevice = NO;
        [self closePenSleepView];
    }
    else if ([szS isEqualToString:@"CLICK"] ||
             [szS isEqualToString:@"DOUBLE_CLICK"] ||
             [szS isEqualToString:@"Gesture Circle Clockwise"] ||
             [szS isEqualToString:@"Gesture Circle CounterClockwise"]) {
        [self penIdleTimerStop];
        [self closePenSleepView];
        return;
    }
    .....
}

-(void) penIdleTimerStop {
    // TODO:: timer release
}

-(void) closePenSleepView {
    // TODO:: pen sleep view release
}

-(void) showPenSleepView {
    // TODO:: pen sleep view show
}
```

I. Concept

- Hardware Structure
- Software Structure
- Background knowledge

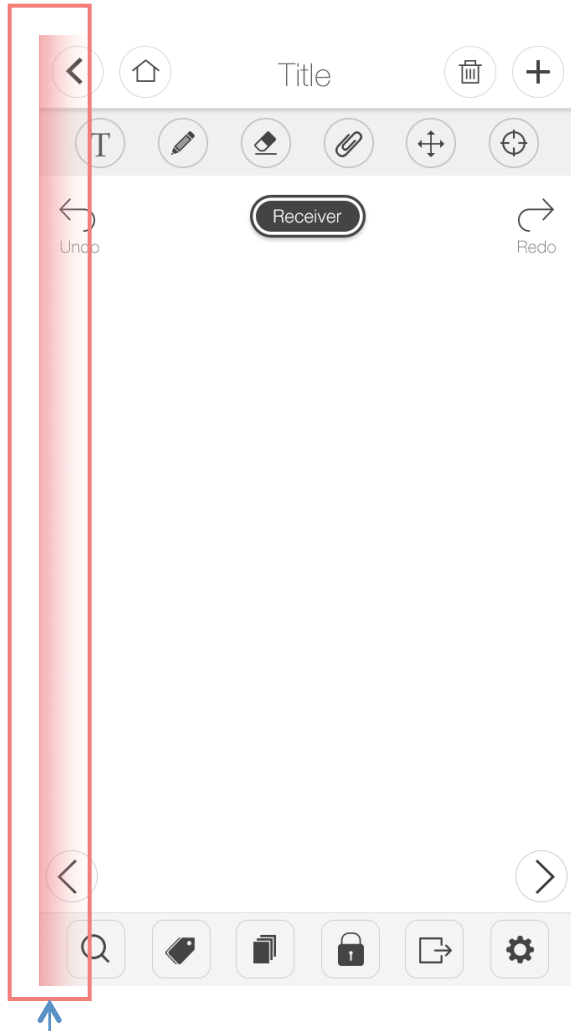
II. Development

- Project setting
- Components of Library
- Reference
- Guide

III.Design Guide

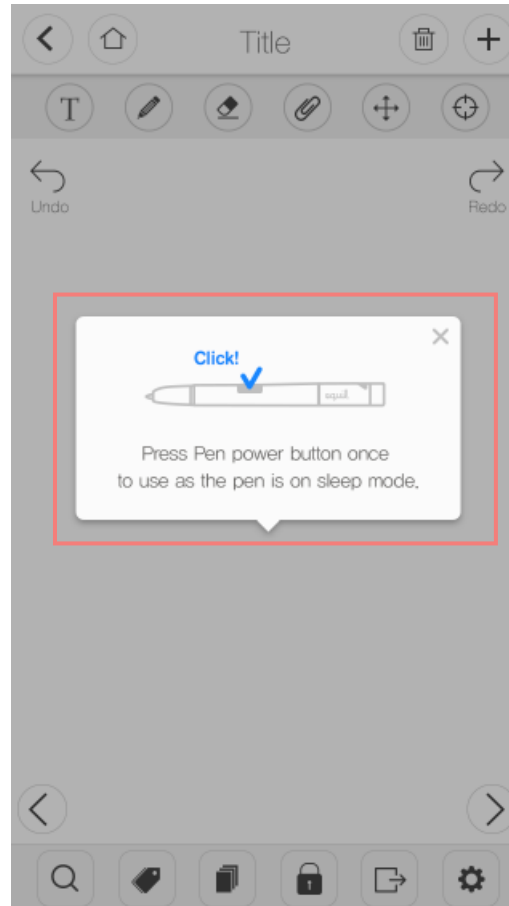
IV. Go to App Store

1. Screen Mode when it is out of the motion area



When the pen is out of range, it shows in red color. The shape of color is changeable, it lets the users know it is deviated

2. Show message when pen goes to sleep mode (Smart Pen only)




3. Tutorial- related to Hardware

The information below must be included in the manual


We can provide source files as .psd format in 9 languages(English, Spanish, French, German, Italian, Japanese, Chinese–Simplified, Chinese-Traditional, Korean)

Please refer to Tutorial_source (Attachment)

Do not use this device right after it was moved from cold place to warm place or vice versa.



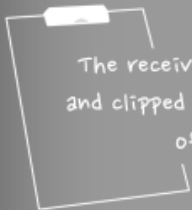
External disturbance like sharp metal noises, winds from heater or air conditioner, PDP TV, and/or external infrared rays may cause product malfunction.
(You can correct mistakes using undo/eraser)



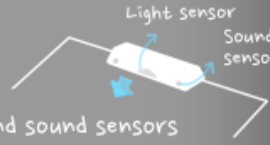
Pen refill is replaceable .

Notice!


The receiver must be balanced and clipped onto the top center of the paper.



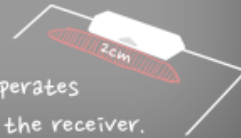
The light and sound sensors should be facing towards the workspace.



The receiver must not move while in use.
clip the receiver on more than 2 pages of papers is recommended.




It may not operates within 2 cm from the receiver.




Pen instructions


- Using a magnetic clip on the bottom of the receiver, attach the receiver to the top center of the paper.



Turn off the receiver first. Press and hold the receiver power button until the blue LED light is turned on and blinks fast.



- Go to 'Setting' > Bluetooth, find the device Equil-xxxxxx and connect to it. (After that, it will connect automatically)



※ iPhone 5S and iPhone 5c with iOS7.0.3 and later, automatic connection for Bluetooth is not available at the moment.
users should manually connect Equil Smartpen to your device whenever turning on the receiver.

- Run the application to use the pen.

I. Concept

- Hardware Structure
- Software Structure
- Background knowledge

II. Development

- Project setting
- Components of Library
- Reference
- Guide

III. Design Guide

IV. Go to App Store

➤ register to MFI

This is Apple certified accessory, we need to register all apps to use this device.

The below points are the information in order to register in Apple, so please fill it in English

Name of app as it will appear in App Store:

:

App version

:

Bundle Identifier

(The unique CFBundleIdentifier that specifies this application.)

:

Does your company own the brand name which will appear on this app?(Yes or No)

:

Is this a new or existing app on the App Store? (New or Existing)

:

Will this app be used independently of this accessory hardware? (Yes or No)

:

Please describe the iOS app for this accessory, including a general functional overview of the software and its key features

:

After done, please send it with your name and contact number to support@penandfree.com

➤ register to App Store

When registering in App Store, record precisely to Mfi Certification ID#:109189-0006 at Review notes section
Equil Smartmarker is Mfi Certification ID#:109189-0006
Equil is Mfi Certification ID#:109189-0005
(Please refers to the example picture below)

➤ Example

Notes ?

MFi Certification ID#:109189-0005
MFi Certification ID#:109189-0006