Equil
Developer Guide
for Android

PNF R&D S/W

2015. 05

Contents

I. Concept

- Hardware Structure
- Software Structure
- Background knowledge

II. Development

- Project setting
- components of Library
- reference
- Guide

III. Design Guide

Contents

I. Concept

- Hardware Structure
- Software Structure
- Background knowledge

II. Development

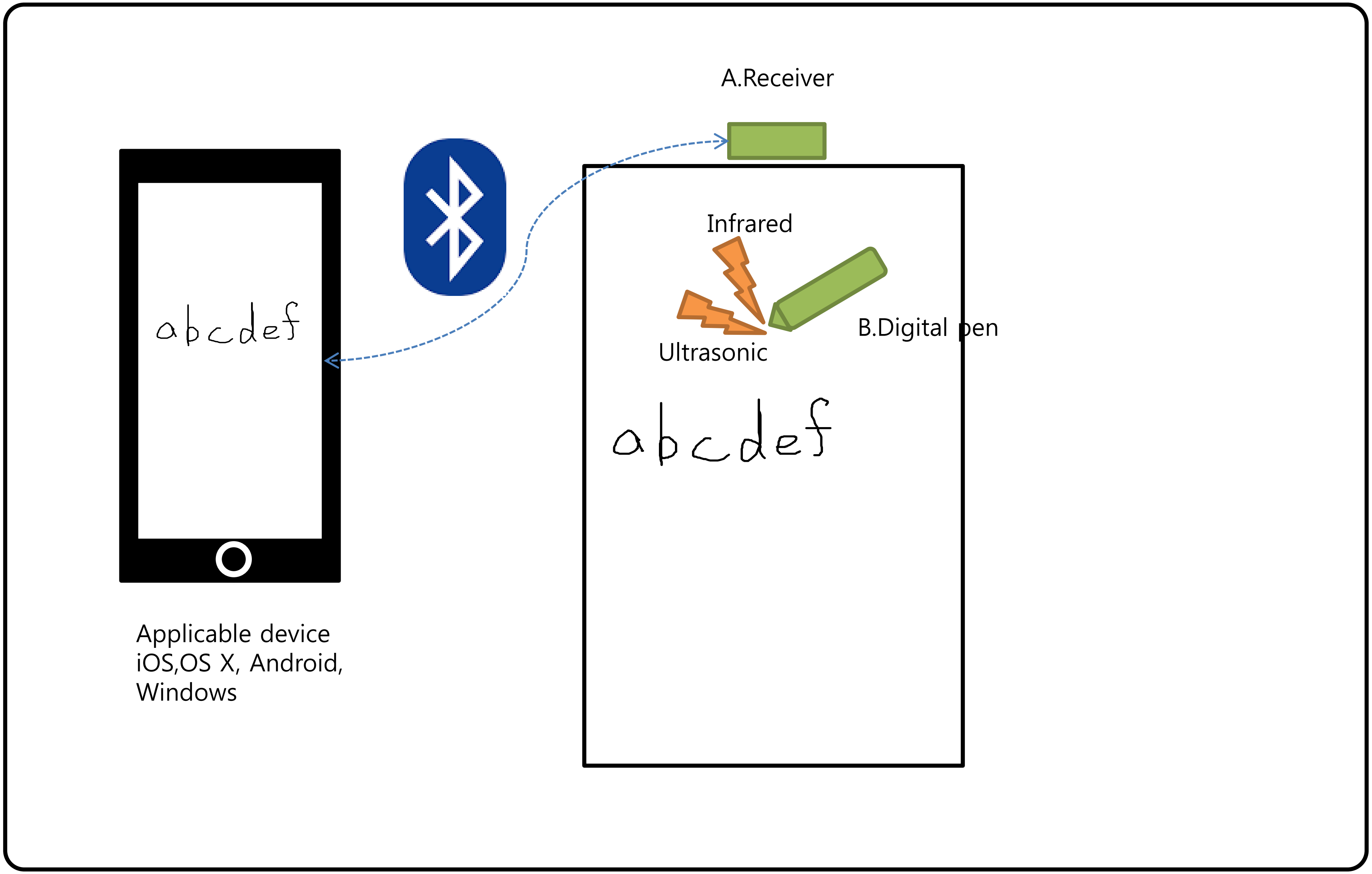
- Project setting
- Components of Library
- Reference
- Guide

III. Design Guide

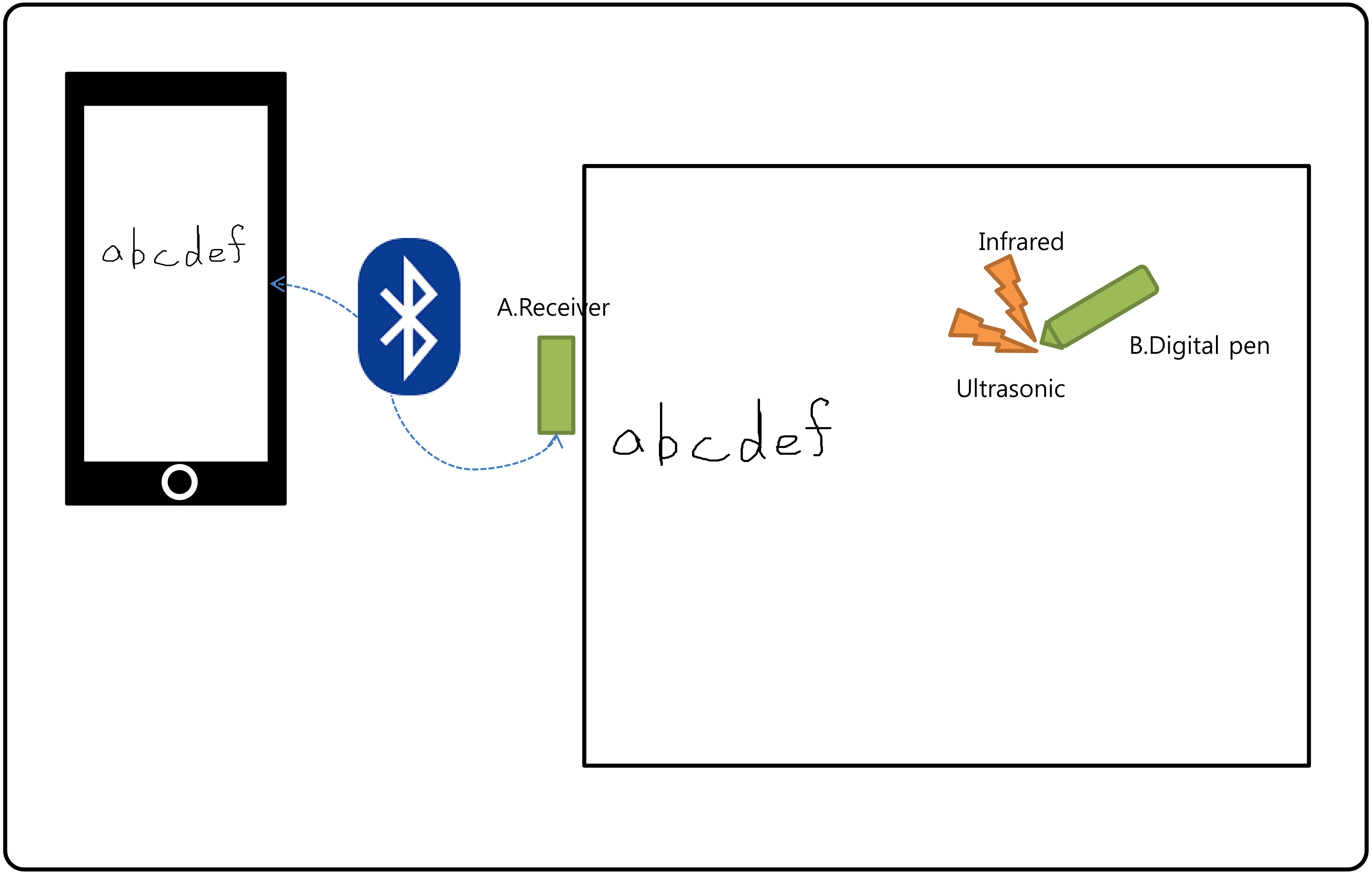
Concept > PNF Hardwares

Model	Devices	Connection	Writing	Image
Equil Smart Pen	iPhone,iPod,iPad, Mac,Windows,Android	Wireless(BlueTooth)	On the paper Or desk	
Equil Smart Marker	iPhone,iPod,iPad, Mac,Windows,Android	Wireless(BlueTooth), USB(Windows, OSX)	On the whiteboard	

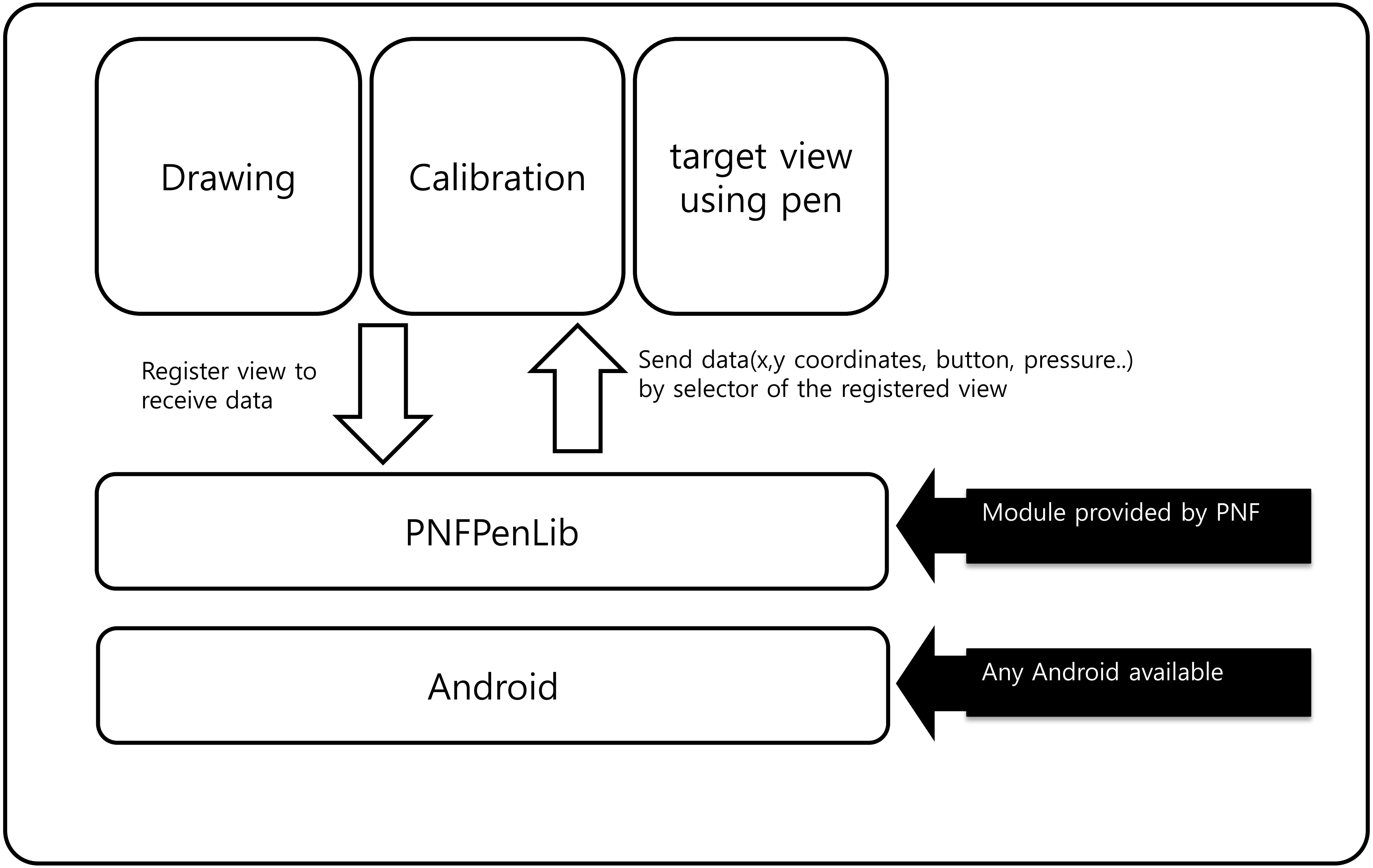
Concept > Hardware Structure (Equil Smart Pen)



Concept > Hardware Structure (Equil 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

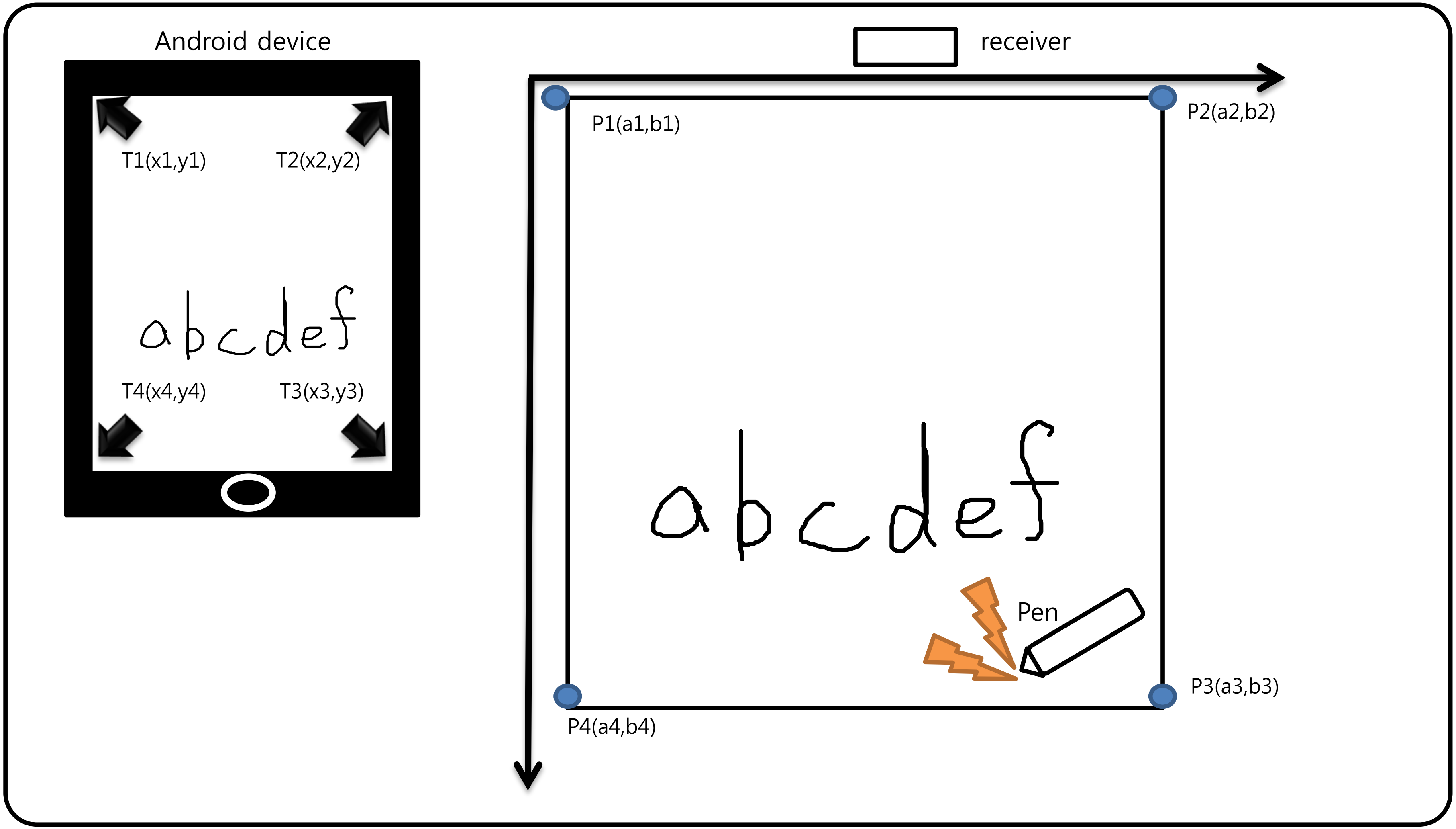
Module provided by PNF

Android

Any Android available

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.



Concept > Background knowledge > External Accessory Framework

- Refer to http://developer.android.com/samples

http://developer.android.com/samples/BluetoothChat/index.html

Contents

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

Development > Components of Test Sample (PenTest)

Folder		File	Description
$(SDKHome)/src/com/pnf/pen	calibration/	CalibrationPointActivity.java	Calibration Activity
	drawingview/	DrawView.java	pen drawing view Class
		DrawViewActivity.java	pen drawing Activity
	dataimport/	DataImportActivity.java	data import Activity
	test/	BaseActivity.java	Base Activity
		MainActivity.java	Main Activity
		MainDefine.java	Define Class
$(SDKHome)/res/layout		activity_main.xml	Main layout
		baseview.xml	Base layout
		calibration.xml	켈리브레이션 설정 layout
		drawview.xml	펜이 그려지는 layout
		dialog_device_alive_view.xml	펜 슬립 모드시 나오는 layout
		di_data_main	Data import layout

※ $(SDKHome) : [unZipped folder]/

PNFPenController Class

Inherits from	
Declared in	PNFPenController.java

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	PointF	Property	readonly
Description	Coordinates before calibrating		
Range	0 ~ 6500		
Device	Equil Smart Pen / Marker		
Usage			

Development > Reference

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 only** PEN_HOVER_DOWN : Pen button down PEN_HOVER_MOVE : Move with Pen button down (defined in PNFDefine.java)		
Device	Equil Smart Pen / Marker		

Usage	PenHandlerWithDictionary(PenDataClass penData)
	{
	    int pressureValue = penData.pressure;		

Development > Reference

Temperature			
Type	int	Property	readonly
Description	Temperature		
Range	0~60 (Celsius)		
Device	Equil Smart Pen / Marker		

Usage	PenHandlerWithDictionary(PenDataClass penData)
	{
	    int temperatureValue = penData. Temperature;		

pressure			
Type	int	Property	readonly
Description	Pressure value of Equil.		
Range	0 ~ 700		
Device	Equil Smart Pen / Marker		

Usage	PenHandlerWithDictionary(PenDataClass penData)
	{
	    int pressureValue = penData.pressure;		

Development > Reference

getCoordinatePostionXY			
Type	PointF	Property	readonly
Description	Calibrated coordinates		
Range	According to the target view size		
Device	Equil Smart Pen / Marker		
Usage	PointF ptConv = MainDefine.penController.GetCoordinatePostionXY(ptRaw.x ,ptRaw.y ,bRight);		

isPenMode()			
Type	boolean	Property	readonly
Description	Whether receiver is connected or not		
Range	Yes / No		
Device	Equil Smart Pen / Marker		

Usage	if(MainDefine.penController.isPenMode())
	{
	//// if receiver is connected
	}
	else
	{
	// if receiver is not connected
	}		

getModelCode()		
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	if (MainDefine.penController.getModelCode()== 2) {
	    addDebugText(“Equil Smart Pen”);
	 }
	 else if (MainDefine.penController.getModelCode()== 3) {
	    addDebugText(“ Equil Smart Pen2”);
	 }
	else if (MainDefine.penController.getModelCode()== 4) {
	    addDebugText(“Equil Smart Marker”);
	}	

Development > Reference

getMCU1()			
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			

getMCU2()			
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			

getHWVersion()			
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	- 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		
Usage			

getAudioMode()			
Type	int	Property	readonly
Description	Audio Mode of Smart Marker		
Range	0 = beep only 1 = beep + voice		
Device	Equil Smart Marker		
Usage			

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

Development > Reference

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	• Smart Marker 0 = High Else = Low • Smart Pen 0 ~ 100		
Device	Equil Smart Pen / Marker		
Usage			

SetRetObjForMsg		
Description	Set an object to receive the Device data The object should have “Handler messageHandler = new Handler()”	
Out	void	
input	Handler	messageHandler
Device	Equil Smart Pen / Marker	

Usage	public void onResume()
	{
	.....
	MainDefine.penController.SetRetObjForMsg(messageHandler);
	......
	}
	
	Handler messageHandler = new Handler()
	{
	@Override
	public void handleMessage(Message msg)
	{
	FreeLogMsg(msg.what);
	}
	};
	
	void FreeLogMsg(int what)	

Development > Reference

SetRetObjForEnv		
Description	Set an object to receive the pen data for environment The object should have “Handler penEnvHandler = new Handler()”	
out	void	
input	Handler	PenHandlerEnv
Device	Equil Smart Pen / Marker	

Usage	public void onResume()
	{
	    .....
	    MainDefine.penController.SetRetObjForEnv(PenHandlerEnv);
	    ......
	
	}
	
	Handler PenHandlerEnv = new Handler()
	{
	    @Override
	    public void handleMessage(Message msg)
	    {
	        onPenEnvEvent(msg.what ,msg.obj);
	    }
	};
	
	void onPenEnvEvent(int what ,Object obj)	

SetRetObjForDI		
Description	Set an object to receive the pen data for environment The object should have “Handler PenHandlerDI = new Handler()”	
out	void	
input	Handler	PenHandlerDI
Device	Equil Smart Pen / Marker	

Usage	public void onResume()
	{
	    .....
	    MainDefine.penController.SetRetObjForDI(PenHandlerDI);
	    ……
	
	}
	
	Handler PenHandlerDI = new Handler()
	{
	    @Override
	    public void handleMessage(Message msg)
	    {
	        onPenDIEvent(msg.what ,msg.arg1 ,msg.obj);
	    }
	};
	
	void onPenDIEvent(int what ,int diPapersize ,Object obj)	

SetRetObjForDI		
Description	Set an object to receive the pen data for environment The object should have “Handler PenHandlerFunc = new Handler()”	
out	void	
input	Handler	PenHandlerFunc
Device	Equil Smart Pen / Marker	

Usage	public void onResume()
	{
	    .....
	    MainDefine.penController.SetRetObjForFunc(PenHandlerFunc);
	    ......
	}
	
	Handler PenHandlerFunc = new Handler()
	{
	    @Override
	    public void handleMessage(Message msg)
	    {
	        onPenFuncEvent(msg.what ,msg.arg1 ,msg.arg2);
	    }
	};
	
	void onPenFuncEvent(int what ,int batteryStation ,int batteryPen)	

setConnectDelay		
Description	Set if delay time is used or not	
Out	void	
Input	boolean	Yes:/No
Device	Equil Smart Pen / Marker	

Usage	@Override
	public void onCreate(Bundle savedInstanceState)
	{
	    .....
	    MainDefine.penController.setConnectDelay(false);
	    ......
	}	

setCalibration		
Description	Set data for calibration	
out	void	
input	Context	Context to draw
Device	Equil Smart Pen / Marker	

Usage	@Override
	public void onCreate(Bundle savedInstanceState)
	{
	    .....
	    MainDefine.penController.setCalibration(getApplicationContext());
	    .....
	}	

Development > Reference

startPen		
Description	Start to communicate with device	
out	Void	
input	N/A	
Device	Equil Smart Pen / Marker	

Usage	@Override
	public void onCreate(Bundle savedInstanceState)
	{
	    .....
	    MainDefine.penController.startPen();
	    ......
	}	

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	`MainDefine.penController.stopPen();`	

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	MainDefine.penController.restartPen();	

setCalibrationData		
Description	Set data for calibration	
out	void	
input	PointF	square which consists of calibrated coordinates
	float	Margin between displayed point and edge of screen
	PointF	Original points
Device	Equil Smart Pen / Marker	

Usage	PointF[] calScreenPoint = new PointF[4];//screen size
	PointF[] calResultPoint = new PointF[4];//calibration size
	……………………..
	
	MainDefine.penController.setCalibrationData(
	    calScreenPoint , //screen size
	    0 , //margin
	    calResultPoint); //calibration size	

Development > Reference

changeAudioMode		
Description	Change Audio mode of Smart Marker	
Out	void	
Input	boolean	Yes:/No
Device	Equil Smart Marker	
Usage	MainDefine.penController.changeAudioMode(0); -> Change to beep only MainDefine.penController.changeAudioMode(1); -> change to beep and voice	

changeVolume		
Description	Change audio volume	
Out	void	
Input	int	0 ~ 255
Device	Equil Smart Marker	
Usage	MainDefine.penController.changeVolume(0); -> max MainDefine.penController.changeVolume(255); -> min	

ReadQ		
Description	Read one data from read Queue	
Out	PenDataClass	
Input	void	
Device	Equil Smart Pen / Marker	

Usage	public class PenDataClass {
	        public int PenStatus = 0;
	        public PointF ptRaw = new PointF();
	        public int pressure = 0;
	        public int StationPosition = 0;
	        public int Temperature = 0;
	        public int MakerPenStatus = 0;
	        public boolean bRight = true;
	}	

RemoveQ		
Description	Delete one data from read Queue	
Out	void	
Input	void	
Device	Equil Smart Pen / Marker	
Usage	MainDefine.penController.removeQ();	

Development > Reference

ClearQ		
Description	Clear all data from read Queue	
Out	void	
Input	void	
Device	Equil Smart Pen / Marker	
Usage	MainDefine.penController.ClearQ();	

EndReadQ		
Description	라이브러리로 부터 데이터를 Notification으로 받음.	
Out	void	
Input	void	
Device	Equil Smart Pen / Marker	
Usage	MainDefine.penController.EndReadQ()	

StartReadQ		
Description	Read Pen mode through Read Queue	
Out	void	
Input	void	
Device	Equil Smart Pen / Marker	

Usage

```
MainDefine.penController.StartReadQ();
.....
private class runReadThread implements Runnable
{
    final int RUNNING = 0;
    final int STOPPED = 1;
    private int state = RUNNING;
    Thread thread;
    public runReadThread() {
        thread = new Thread(this);
        state = RUNNING;
    }
    public void run() {
        while (state != STOPPED) {
            try {
                synchronized (MainDefine.penHandler) {
                    PenDataClass penDataClass = MainDefine.penController.ReadQ();
                    if(penDataClass != null) {
                        MainDefine.sendActivityPen(penDataClass);
                    }
                    Thread.sleep(10);
                }
            } catch (Exception e) { }
        }
    }
    public void cancel() {
        state = STOPPED;
    }
    public void start() {
        thread.start();
    }
}
```

Development > Guide > Connect and Initialize

Overview

Create and initialize object PNFPenController

Example

1. Create PNFPenController object

```
MainDefine.penController = new PNFPenController(getApplicationContext());
```

2. Set delay time

```
MainDefine.penController.setConnectDelay(false);
```

3. Set calibration

```
MainDefine.penController.setCalibration(getApplicationContext());
```

4. Start to communicate with device

```
MainDefine.penController.startPen();
```

5. Set object to receive data

```
MainDefine.penController.SetRetObjForMsg(messageHandler);
```

example source: ViewController.h ViewController.m

Development > Guide > Device recognize

➢ Overview

To recognize which device is connected.
After the device is connected and model code is sent after 1 sec.

➢ Example

```
new Handler().postDelayed(new Runnable() {
    public void run() {
        lazyCheckCalibration();
        ReadThreadStart();
    }
    }, 1500);

void lazyCheckCalibration() {
    int modelCode = MainDefine.penController.getModelCode();
    PointF[] calScreenPoint = new PointF[4];
    PointF[] calResultPoint = new PointF[4];
    if(modelCode < 4){
        calResultPoint[0] = new PointF(MainDefine.caliSP_A4.left, MainDefine.caliSP_A4.top);
        calResultPoint[1] = new PointF(MainDefine.caliSP_A4.right, MainDefine.caliSP_A4.top);
        calResultPoint[2] = new PointF(MainDefine.caliSP_A4.right ,MainDefine.caliSP_A4.bottom);
        calResultPoint[3] = new PointF(MainDefine.caliSP_A4.left ,MainDefine.caliSP_A4.bottom);
    }else{
        int stationPostion = MainDefine.penController.getStationPostion();
           if(stationPostion == PNFDefine.DIRECTION_TOP){
            calResultPoint[0] = new PointF(MainDefine.caliSM_TOP_4X6.left, MainDefine.caliSM_TOP_4X6.top);
            calResultPoint[1] = new PointF(MainDefine.caliSM_TOP_4X6.right, MainDefine.caliSM_TOP_4X6.top);
            calResultPoint[2] = new PointF(MainDefine.caliSM_TOP_4X6.right ,MainDefine.caliSM_TOP_4X6.bottom);
            calResultPoint[3] = new PointF(MainDefine.caliSM_TOP_4X6.left ,MainDefine.caliSM_TOP_4X6.bottom);
        }else if(stationPostion == PNFDefine.DIRECTION_BOTTOM){
            calResultPoint[0] = new PointF(MainDefine.caliSM_BOTTOM_4X6.left, MainDefine.caliSM_BOTTOM_4X6.top);
            calResultPoint[1] = new PointF(MainDefine.caliSM_BOTTOM_4X6.right, MainDefine.caliSM_BOTTOM_4X6.top);
            calResultPoint[2] = new PointF(MainDefine.caliSM_BOTTOM_4X6.right ,MainDefine.caliSM_BOTTOM_4X6.bottom);
            calResultPoint[3] = new PointF(MainDefine.caliSM_BOTTOM_4X6.left ,MainDefine.caliSM_BOTTOM_4X6.bottom);
        }else{
            calResultPoint[0] = new PointF(MainDefine.caliSM_LEFT_8X5.left, MainDefine.caliSM_LEFT_8X5.top);
            calResultPoint[1] = new PointF(MainDefine.caliSM_LEFT_8X5.right, MainDefine.caliSM_LEFT_8X5.top);
            calResultPoint[2] = new PointF(MainDefine.caliSM_LEFT_8X5.right ,MainDefine.caliSM_LEFT_8X5.bottom);
            calResultPoint[3] = new PointF(MainDefine.caliSM_LEFT_8X5.left ,MainDefine.caliSM_LEFT_8X5.bottom);
        }
    }
    calScreenPoint[0] = new PointF(0.0f ,0.0f);
    calScreenPoint[1] = new PointF(MainDefine.iDisGetWidth ,0.0f);
    calScreenPoint[2] = new PointF(MainDefine.iDisGetWidth ,MainDefine.iDisGetHeight);
    calScreenPoint[3] = new PointF(0.0f ,MainDefine.iDisGetHeight);
    MainDefine.penController.setCalibrationData(calScreenPoint, 0, calResultPoint);
}
```

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 ReadThreadOff() {
    if(penReadThread != null){
        penReadThread.cancel();
        penReadThread = null;
    }
    if (MainDefine.penController != null) {
        MainDefine.penController.EndReadQ();
    }
}
void ReadThreadStart() {
    if (penReadThread == null) {
        penReadThread = new runReadThread();
        penReadThread.start();
    }
    if (MainDefine.penController != null) {
        MainDefine.penController.StartReadQ();
    }
}
```

example source: ViewController.h ViewController.m

Development > Guide > Receive data from library

➢ Example

```
private class runReadThread implements Runnable
{
    final int RUNNING = 0;
    final int STOPPED = 1;
    private int state = RUNNING;
    Thread thread;
    public runReadThread() {
        thread = new Thread(this);
        state = RUNNING;
    }
    public void run() {
        while (state != STOPPED) {
            try {
                if(MainDefine.penHandler != null){
                    synchronized (MainDefine.penHandler) {
                        PenDataClass penDataClass = MainDefine.penController.ReadQ();
                        if(penDataClass != null) {
                            MainDefine.sendActivityPen(penDataClass);
                        }
                        Thread.sleep(10);
                    }
                 }
            } catch (Exception e) { }
        }
    }
    public void cancel() {
        state = STOPPED;
    }
    public void start() {
        thread.start();
    }
}
```

example source: ViewController.h ViewController.m

Development > Guide > Receive pen data from library

➢ Example

```
Handler penHandler = new Handler()
{
    @Override
    public void handleMessage(Message msg) {
        PenDataClass penData = (PenDataClass)msg.obj;
        if(penData != null){
            PenHandlerWithDictionary(penData);
        }
    }
};


void PenHandlerWithDictionary(PenDataClass penData)
{
    switch (penData.PenStatus)
    {
        case PNFDefine.PEN_DOWN:
            break;
        case PNFDefine.PEN_MOVE:
            break;
        case PNFDefine.PEN_UP:
            break;
        case PNFDefine.PEN_HOVER:
            break;
        case PNFDefine.PEN_HOVER_DOWN:
            break;
        case PNFDefine.PEN_HOVER_MOVE:
            break;
    }
    PointF ptConv = MainDefine.penController.getCoordinatePostionXY(penData.ptRaw.x ,penData.ptRaw.y ,penData.bRight);
}
```

Log String Message	Description
PEN_DOWN	
PEN_MOVE	
PEN_UP	
* Equil only	
PEN_HOVER	
PEN_HOVER_DOWN	
PEN_HOVER_MOVE	

example source: ViewController.h ViewController.m

Development > Guide > Receive message from library

➢ Example

1. Add messageHandler

```
Handler messageHandler = new Handler(){
      @Override
      public void handleMessage(Message msg) {
          FreeLogMsg(msg.what);
      }
};
```

1. Handler for Message

```
void FreeLogMsg(int what)
{
        ......
        if(what == PNFDefine.PNF_MSG_FAIL_LISTENING){
        }else if(what == PNFDefine.PNF_MSG_CONNECTED){
        }
        else if(what == PNFDefine.PNF_MSG_INVALID_PROTOCOL){
        }
        else if(what == PNFDefine.PNF_MSG_SESSION_CLOSED){
        }
        else if(what == PNFDefine.PNF_MSG_PEN_RMD_ERROR){
        }
        else if(what == PNFDefine.PNF_MSG_FIRST_DATA_RECV){
        }
        else if(what == PNFDefine.GESTURE_CIRCLE_CLOCKWISE){
        }
        else if(what == PNFDefine.GESTURE_CIRCLE_COUNTERCLOCKWISE){
        }
        else if(what == PNFDefine.GESTURE_CLICK){
        }
        else if(what == PNFDefine.GESTURE_DOUBLECLICK){
        }
        ......
}
```

Log String Message	Description
PNF_MSG_CONNECTED	Device is connected
PNF_MSG_FAIL_LISTENING	Fail to receive. Need to reconnect.
PNF_MSG_INVALID_PROTOCOL	Invalid hardware
PNF_MSG_SESSION_CLOSED	Session is disconnected
PNF_MSG_FIRST_DATA_RECV	First data is received after connecting
PNF_MSG_PEN_RMD_ERROR	Abnormal drawing data
* Equil only	
GESTURE_DOUBLECLICK	Equil pen button double click
GESTURE_CLICK	Equil pen button click
GESTURE_CIRCLE_CLOCKWISE	Equil pen circle clockwise gesture
GESTURE_CIRCLE_COUNTERCLOCKWISE	Equil pen circle counter clockwise gesture

example source : MainActivity.java

Development > Guide >Receive pen alive time data from library

➢ Example

1. Set object to receive environment data

```
    public void onResume() {
        .........
        MainDefine.penController.SetRetObjForEnv(PenHandlerEnv);
}
```

2. Environment data handler implementation

```
 void onPenEnvEvent(int what ,Object obj)
     switch(what)
    {
        case PNFDefine.PNF_MSG_ENV_DATA:
            PenEnvDataClass penEnvData = (PenEnvDataClass)obj;
            curPenAliveSec = penEnvData.penAliveSec;
            int Pen_Station_Battery = (int) penEnvData.battery_station;
            int Pen_Battery = (int) penEnvData.battery_pen;
            if(MainDefine.penController.getMCU1() >= 2 &&
                     MainDefine.penController.getMCU2() >= 2 &&
                     MainDefine.penController.getHWVersion() >= 2){
                         if(curPenAliveSec > 0){
                             if(penSleepView.getVisibility() == View.VISIBLE){
                                 penSleepView.setVisibility(View.GONE);
                             }
                          }
              }
              if(isCheckSleepView){
                  if(penAliveTimer == null) {
                      penAliveTimer = new Timer();
                      TimerTask penAliveTask = new TimerTask() {
                          @Override
                           public void run() {
                               onTimerForPenAlive();
                           }
                       };
                       penAliveTimer.schedule(penAliveTask, 1000 ,1000);
                       savePenSleepRemainingTime = (int) MainDefine.GetCurrentSec() + penSleepDelay;
                       savePenAliveSec = penSleepDelay;
                       curPenAliveSec = penSleepDelay;
                   }
               }
            break;
    }
}
```

example source : ViewController.h .m

Development > Guide >Receive pen alive time data from library

➢ Example

```
void onTimerForPenAlive(){
    int curTime = (int) MainDefine.GetCurrentSec();
    boolean check = false;
    if(MainDefine.penController.getModelCode() == 2){
        if(MainDefine.penController.getMCU1() >= 2 && MainDefine.penController.getMCU2() >= 2 && MainDefine.penController.getHWVersion() >= 2){
            check = true;
        }
    }else if(MainDefine.penController.getModelCode() == 3){
        if(MainDefine.penController.getMCU1() >= 1 && MainDefine.penController.getMCU2() >= 1 && MainDefine.penController.getHWVersion() >= 1){
            check = true;
        }
    }else{
        return;
    }
    if(check){
        if(curPenAliveSec <= 0) {
            penPopupHandler.sendEmptyMessage(SLEEPVIEW_SHOWPOPUP);
            return;
        }else{
            penCheckAliveCnt = 0;
        }
        if(curPenAliveSec != 0){
            if(savePenAliveSec != curPenAliveSec){
                savePenAliveSec = curPenAliveSec;
                savePenSleepRemainingTime = (int) curTime+curPenAliveSec;
            }
         }
     }
    if(savePenSleepRemainingTime - curTime < 0) {
        penPopupHandler.sendEmptyMessage(SLEEPVIEW_SHOWPOPUP);
    }else{
        penCheckAliveCnt = 0;
    }
}
```

example source : MainActivity.java

Development > Guide > Receive datainput from library

➢ Example

1. Add PenHandlerDI

```
Handler PenHandlerDI = new Handler()
{
    @Override
    public void handleMessage(Message msg)
    {
        onPenDIEvent(msg.what ,msg.arg1 ,msg.obj);
    }
};
```

1. Handler for Message

```
void onPenDIEvent(int what ,int diPapersize ,Object obj)
{
        ……
        if(what == PNFDefine. PEN_DI_DATA){
        }
       else if(what == PNFDefine.PEN_DI_TEMPLETE){
        }
        else if(what == PNFDefine.PEN_DI_ACC_DATA){
        }
        else if(what == PNFDefine.PEN_DI_DELETE){
        }
        else if(what == PNFDefine.PNF_DI_START){
        }
        else if(what == PNFDefine.PNF_DI_STOP){
        }
        else if(what == PNFDefine.PNF_DI_OK){
        }
        else if(what == PNFDefine.PNF_DI_FAIL){
        }
        else if(what == PNFDefine.PNF_DI_TEMP_EXIST){
        }
        else if(what == PNFDefine.PNF_DI_TEMP_FILE_COMPLETE){
        }
         else if(what == PNFDefine.PNF_DI_FILE_LIST_COMPLETE){
        }
        ……
}
```

Log String Message	Description
PEN_DI_DATA	
PEN_DI_TEMPLETE	
PEN_DI_ACC_DATA	
PEN_DI_DELETE	
PNF_DI_START	
PNF_DI_STOP	
PNF_DI_OK	
PNF_DI_FAIL	
PNF_DI_TEMP_EXIST	
PNF_DI_TEMP_FILE_COMPLETE	
PNF_DI_FILE_LIST_COMPLETE	

example source : MainActivity.java

Development > Guide > Receive pen function from library

➢ Example

1. Add messageHandler

```
Handler PenHandlerFunc = new Handler()
{
    @Override
    public void handleMessage(Message msg)
    {
        onPenFuncEvent(msg.what ,msg.arg1 ,msg.arg2);
    }
};
```

1. Handler for Message

```
void onPenFuncEvent(int what ,int batteryStation ,int batteryPen)
{
        ……
        if(what == PNFDefine.BATTERY_INFO){
        }
       else if(what == PNFDefine.NEW_PAGE){
        }
        else if(what == PNFDefine.DUPLICATE_PAGE){
        }
        else if(what == PNFDefine.CHANGE_DEVECE_POSITION){
        }
        else if(what == PNFDefine.CHANGE_DEVECE_POSITION_FIRST){
        }
        ……
}
```

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 : MainActivity.java

Contents

I. Concept

- Hardware Structure
- Software Structure
- Background knowledge

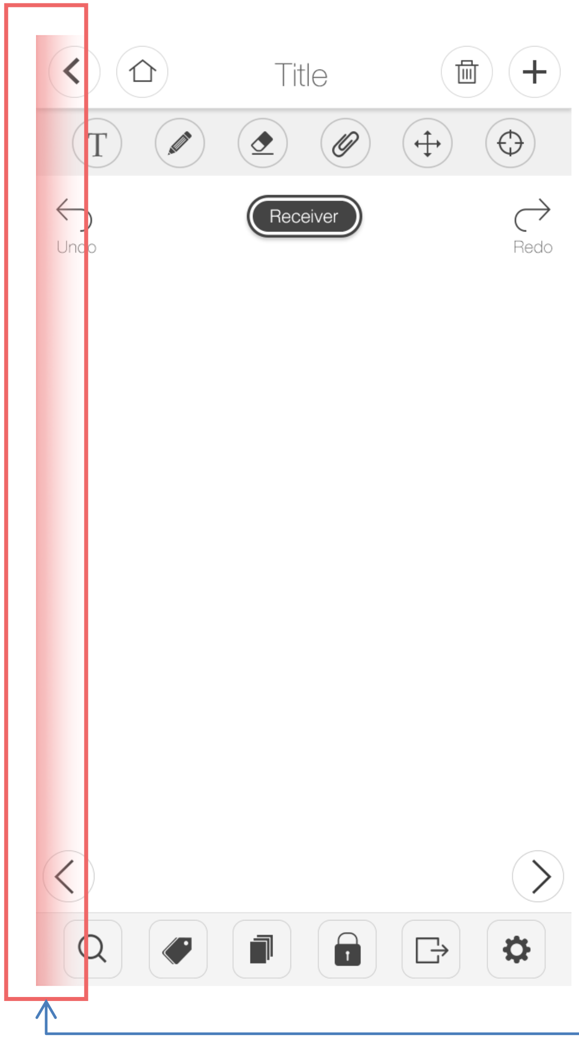
II. Development

- Project setting
- Components of Library
- Reference
- Guide

III. Design Guide

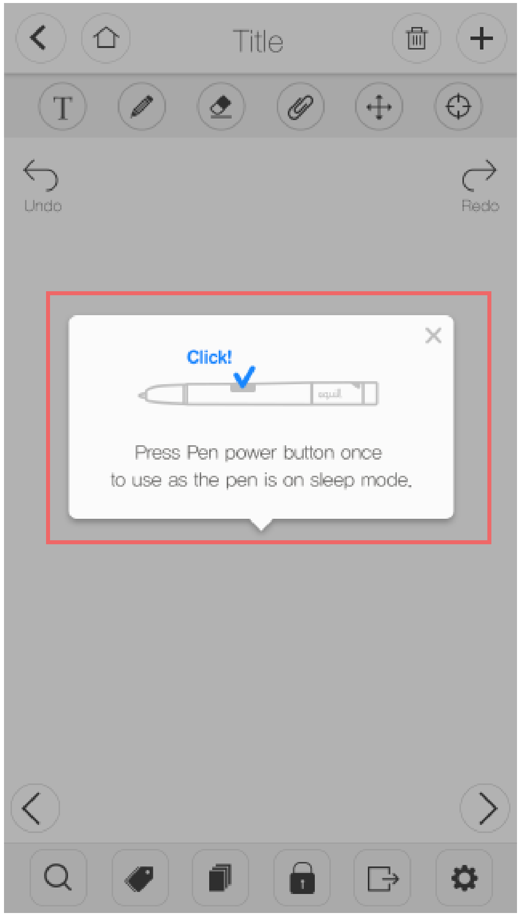
IV. Go to App Store

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

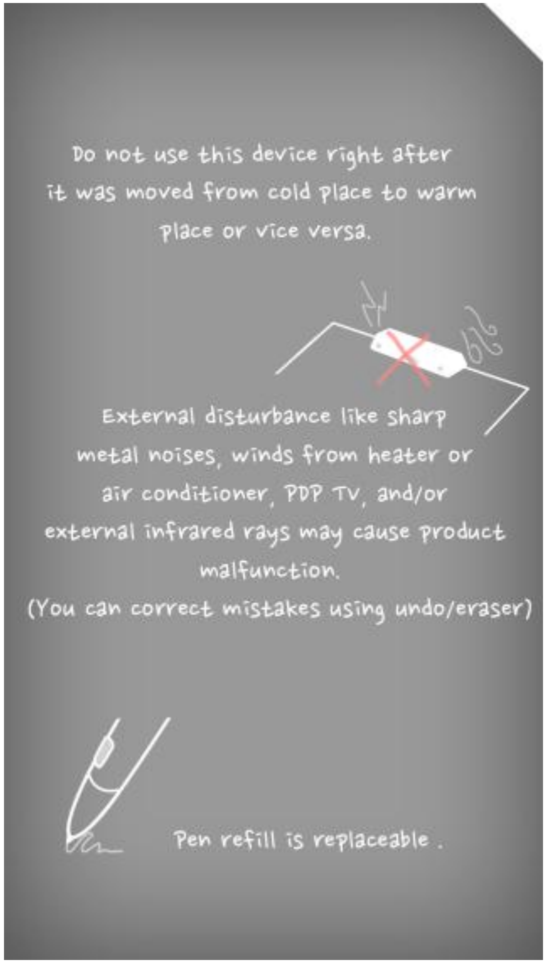
2. Show message when pen goes to sleep mode (Smart Pen only)



Press Pen power button once
to use as the pen is on sleep mode.

3. Tutorial- related to Hardware

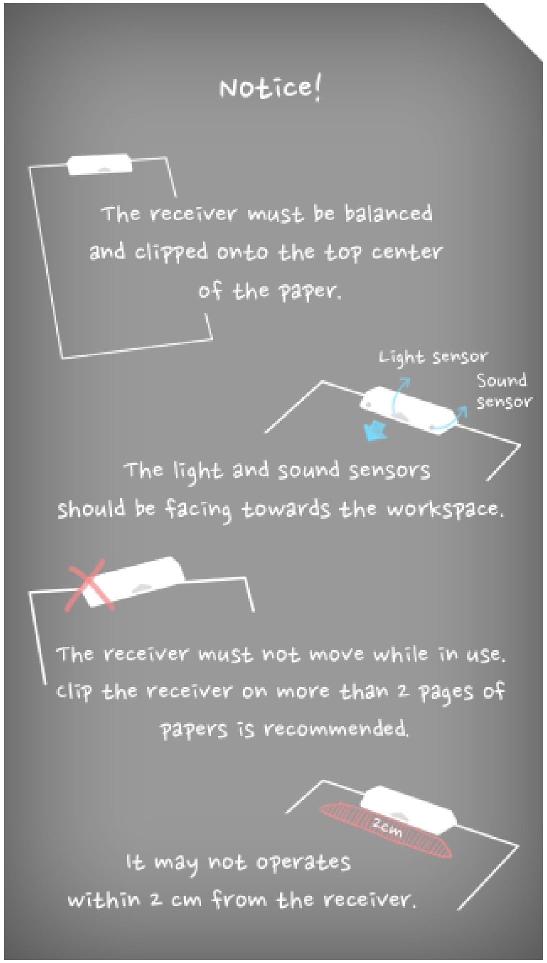
The information below must be included in the manual
We can provide source files as .psd format in 9 languages(English, Spanish, French, German, Italian, Japanese, Chinese–Simplified, Chinese-Traditional, Korean)
Please refer to Tutorial_source (Attachment)



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 .



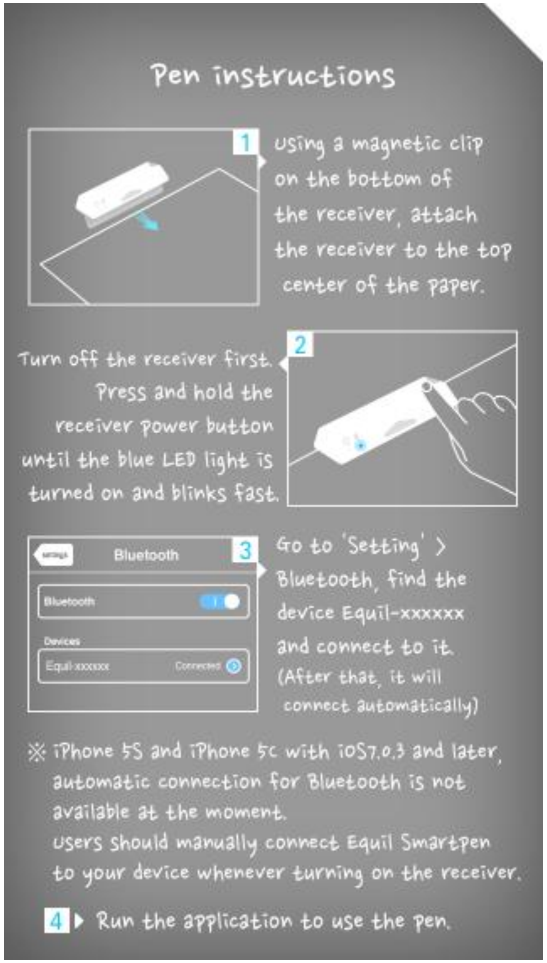
Notice!

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

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

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

3 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.

4 ▶ Run the application to use the pen.