Equil
Developer Guide
for Android

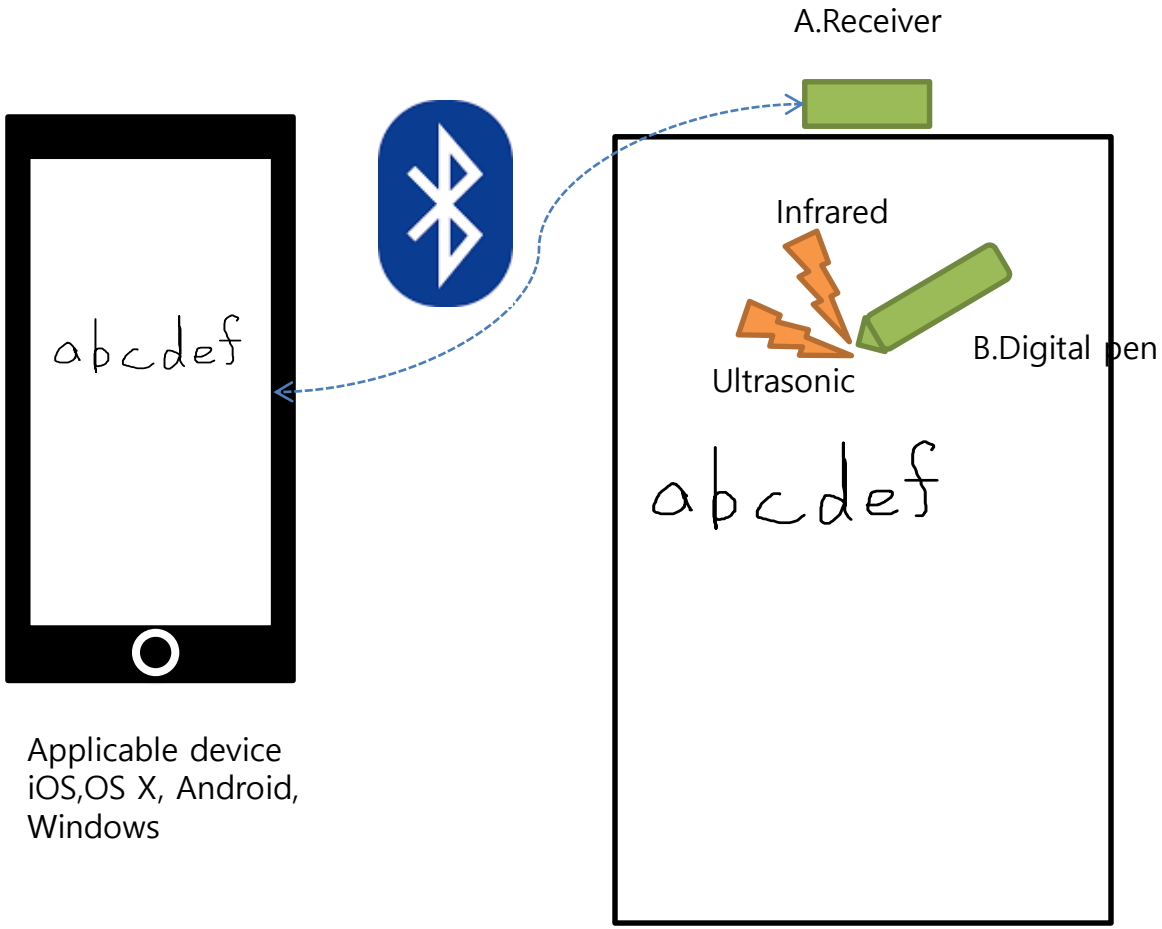
PNF R&D S/W

2015. 05

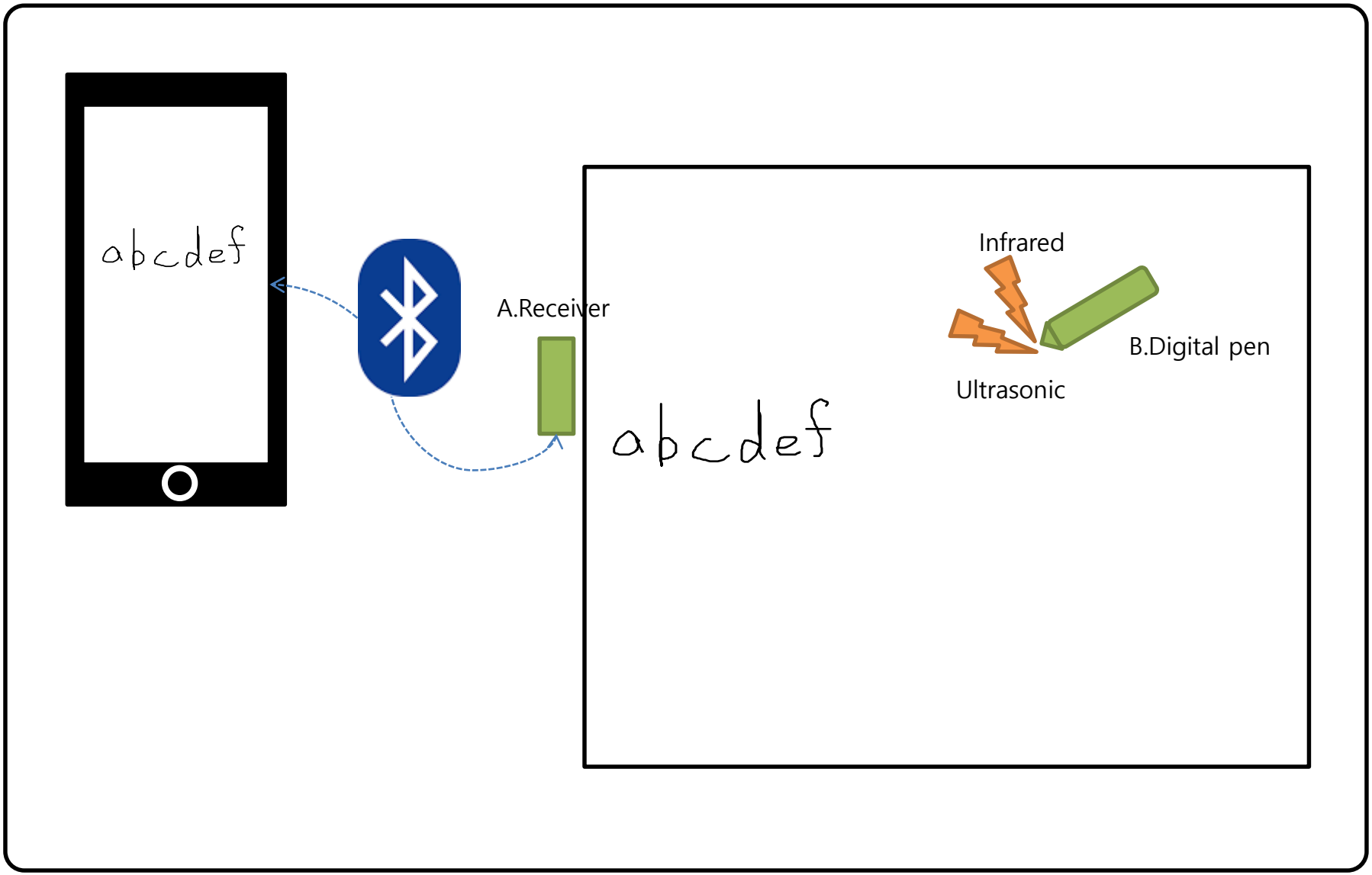
Contents

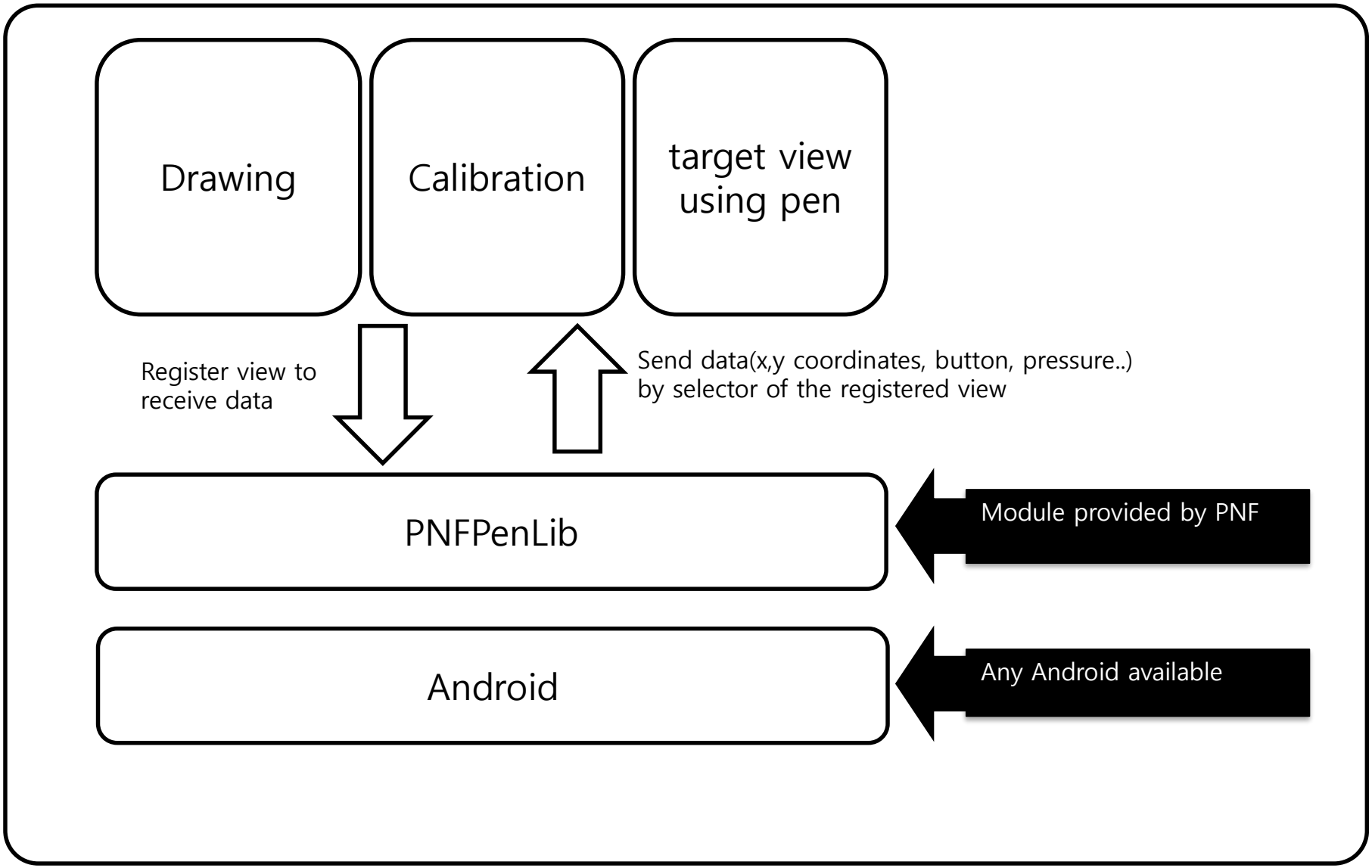
Model	Devices	Connection	Writing	Image
Equil Smart Pen	iPhone,iPod,iPad, Mac,Windows,Android	Wireless(BlueTooth)	On the paper Or desk	
Equil Smart Marker	iPhone,iPod,iPad, Mac,Windows,Android	Wireless(BlueTooth), USB(Windows, OSX)	On the whiteboard	

Concept > Hardware Structure (Equil Smart Pen)

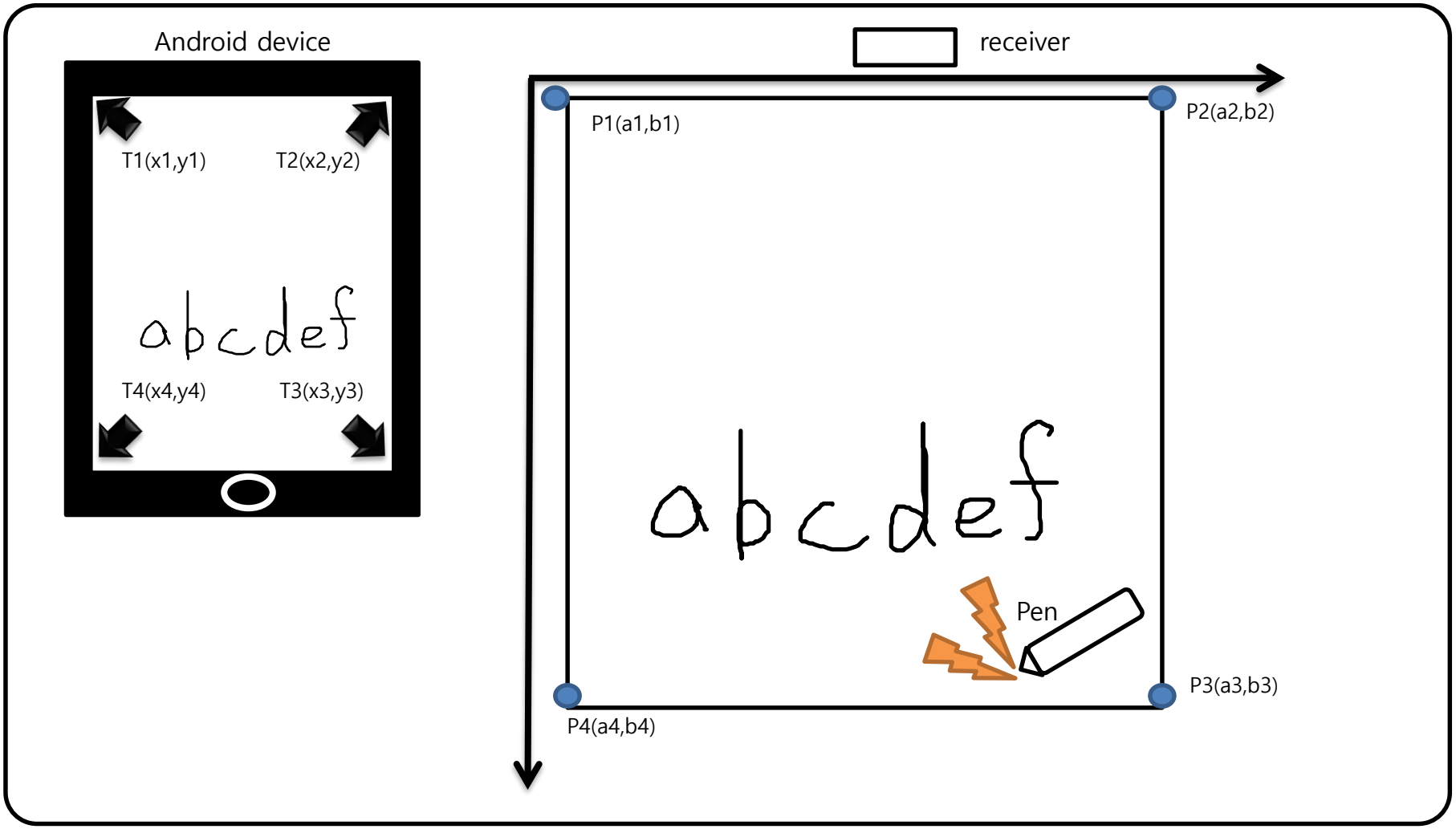


Concept > Hardware Structure (Equil Smart Marker)




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

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 http://developer.android.com/samples

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

Contents

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
		DrawViewActivity.java	pen drawing Activity
	dataimport/	DataImportActivity.java	data import Activity
		DataImportListViewAdapter.java	data import file list view Adapter
		DataImportPreView.java	data import drawing view
		DataImportFigureList.java	data import Figure buffer
	test/	BaseActivity.java	Base Activity
		MainActivity.java	Main Activity
		MainDefine.java	Define Class
$(SDKHome)/res/layout	layer	activity_main.xml	Main layout
		baseview.xml	Base layout
		calibration.xml	Calibration setting layout
		drawview.xml	pen drawing layout
		dialog_device_alive_view.xml	pen sleep popup layout
		dataimportlistview	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			

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");
	}	

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			

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			

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	onPenEvent(int penState, int RawX, int RawY , Object obj) {		

Temperature			
Type	int	Property	readonly
Description	Temperature		
Range	0~60 (Celsius)		
Device	Equil Smart Pen / Marker		
Usage	onPenEvent(int penState, int RawX, int RawY , Object obj) {		

pressure			
Type	int	Property	readonly
Description	Pressure value of Equil.		
Range	0 ~ 700		
Device	Equil Smart Pen / Marker		
Usage	onPenEvent(int penState, int RawX, int RawY ,Object obj) {		

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
	}		

<table>
<tr><td colspan="3">SetRetObjForEnv</td></tr>
<tr><td>Description</td><td colspan="2">Set an object to receive the pen data for environment
The object should have "Handler penEnvHandler = new Handler()"</td></tr>
<tr><td>out</td><td>void</td><td></td></tr>
<tr><td>input</td><td>Handler</td><td>PenHandlerEnv</td></tr>
<tr><td>Device</td><td colspan="2">Equil Smart Pen / Marker</td></tr>
<tr><td>Usage</td><td colspan="2">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)</td></tr>
</table>

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)
	{
	onMessageEvent(msg.what ,msg.obj);
	}
	};
	
	void onMessageEvent(int what ,Object obj)	

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());
	.....
	}	

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();	

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	

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	

PenDataClass		
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 float Pen_RawX = 0;
	public float Pen_RawY = 0;
	public int Pen_Temperature = 0;
	public int Pen_Pressure = 0;
	public int Pen_Alive = 0;
	public float Pen_Station_Battery = 0;
	public float Pen_Battery = 0;
	public int MakerPenStatus = 0;
	public int Station_Position = 0;
	public boolean bRight = true;
	}	

➢ Overview

Create and initialize object PNFPenController

➢ Example

1. Create PNFPenController object
   MainDefine.penController = new PNFPenController(getApplicationContext());

2. Set calibration
   MainDefine.penController.setCalibration(getApplicationContext());

3. Start to communicate with device
   MainDefine.penController.startPen();

4. Set object to receive data
   MainDefine.penController.SetRetObjForMsg(messageHandler);

➢ Example

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


void onPenEvent(int penState, int RawX, int RawY ,Object obj)
{
    PenDataClass penData = (PenDataClass)obj;
    switch (penState)
    {
        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(RawX ,RawY ,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

➢ Example

1. Add messageHandler

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

1. Handler for Message

```
void onMessageEvent(int what ,Object obj)
{
    ......
    if(what == PNFDefine.PNF_MSG_FAIL_LISTENING){
    }
    else if(what == PNFDefine.PNF_MSG_CONNECTED){
    }
    else if(what == PNFDefine.PNF_MSG_FIRST_DATA_RECV){
    }
    else if(what == PNFDefine.PNF_MSG_DI_START){
    }
    else if(what == PNFDefine.PNF_MSG_DI_STOP){
    }
    else if(what == PNFDefine.PNF_MSG_DI_OK){
    }
    else if(what == PNFDefine.PNF_MSG_DI_FAIL){
    }
    else if(what == PNFDefine.PNF_MSG_DI_FILE_LIST_COMPLETE){
    }
    else if(what == PNFDefine.PNF_MSG_NEW_PAGE_BTN){
    }
    else if(what == PNFDefine.PNF_MSG_DUPLICATE_BTN){
    }
    else if(what == PNFDefine.PNF_MSG_CHANGE_STATIONPOSITION){
    }
    ......
}
```

Log String Message	Description
PNF_MSG_FAIL_LISTENING	Fail to receive. Need to reconnect.
PNF_MSG_CONNECTED	Device is connected
PNF_MSG_FIRST_DATA_RECV	First data is received after connecting
PNF_MSG_DI_START	Data Import start
PNF_MSG_DI_STOP	Data Import stop
PNF_MSG_DI_OK	Data Import request success
PNF_MSG_DI_FAIL	Data Import request fail
PNF_MSG_DI_FILE_LIST_COMPLETE	Data Import file list request success
PNF_MSG_NEW_PAGE_BTN	New Page button Clicked
* Smart Marker only	
PNF_MSG_DUPLICATE_BTN	Duplicate Page button Clicked
PNF_MSG_CHANGE_STATIONPOSITION	Smart Marker Change Position

example source : MainActivity.java

➢ Example

1. Add PenHandlerEnv

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

1. Handler for Message

```
void onPenDIEvent(int what ,Object obj)
{
        ......
        if(what == PNFDefine.PNF_MSG_ENV_DATA){
        }
        else 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){
        }
        ......
}
```

Log String Message	Description
PNF_MSG_ENV_DATA	Battery information , smart pen alive time
PEN_DI_DATA	Data import file datas
PEN_DI_TEMPLETE	
PEN_DI_ACC_DATA	Data import file acc datas
PEN_DI_DELETE	

example source : MainActivity.java

➢ 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:
                        PenDataClass penData = (PenDataClass)obj;

                        curPenAliveSec = penData.Pen_Alive;
//                      int Pen_Station_Battery = (int) penData.Pen_Station_Battery;
//                      int Pen_Battery = (int) penData.Pen_Battery;

                        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

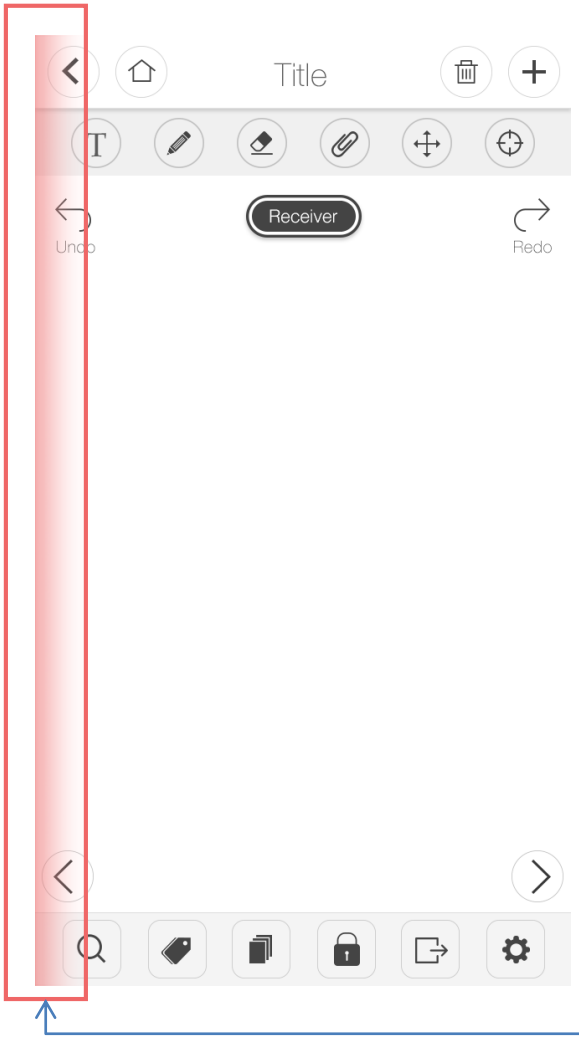
➢ Example

```
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

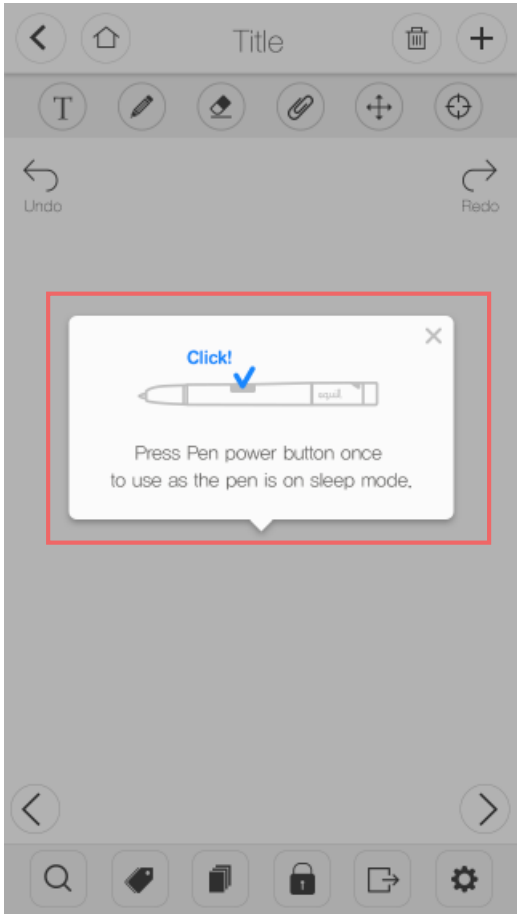
Contents

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)