

# Calibration Operations Guide

*Bubble Level for GlobalMems Gsensor*

Introduction

Bubble Level Features

The APK put inside the "setting"

Two kinds of differences apk in source

## Revision History

Ver.	Date	Updates	Descriptions
1.0	Feb.25th, 2015	Only support MTK Platform	first Release
1.1	Mar.03th, 2015	UI adjustment	Add button(clear offset)
1.2	Jun.24th, 2015	Package rename	Net.androgames.level => com.globalmems.level

## A. Introduction

Application of GlobalMems G-sensor calibration.

Name	Glevel-mtk
Project Name	Glevel-mtk
Application Name	G-sensor calibration
Package Name	com.globalmems.level
Create Activity	Level.java
Min SDK Version	11

## B. Bubble Level Features

**LAUNCHER : The apk(Glevel-mtk\_L.apk) on behalf of the show on the desktop icon.**

1. push Glevel-mtk\_L.apk in /system/priv-app

```
$adb push Glevel-mtk_L.apk /system/priv-app
```

2. First boot, it will trigger GsensorCalibrationBoardCastReceiver..

```
EmSensor.getGsensorCalibration(MTKoffset);
if (MTKoffset[0] != 0 && MTKoffset[1] != 0 && MTKoffset[2] != 0)
    log.e(TAG,"offset existed, Do nothing");
else
    EmSensor.doGsensorCalibration();
```

3. Before Calibration

See Figure 1

Note : Please keep the device in a horizontal desktop.

4. Click Button “Gsensor calibration”

The bubble moves to the center.

See Figure 2

Calibration has been successful at this time.



Figure 1(Before calibration)



Figure 2(after calibration)

X/Y/Z value of close to 0/0/9.81

And the offset value save to nvram.

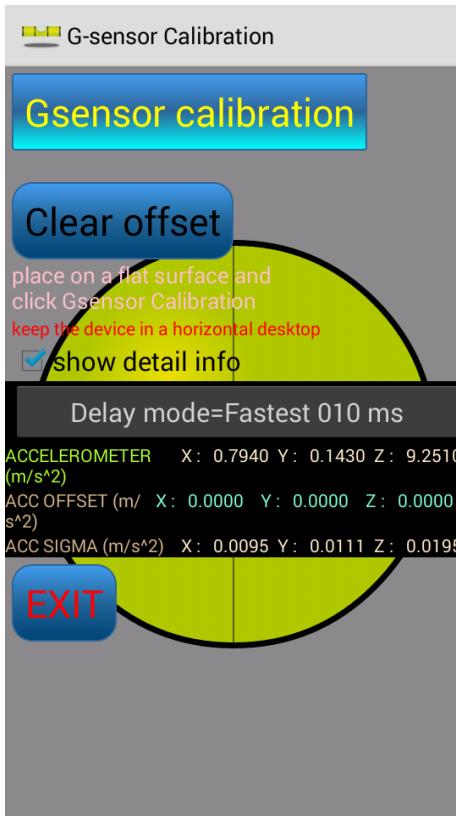


Figure 3(clear offset)

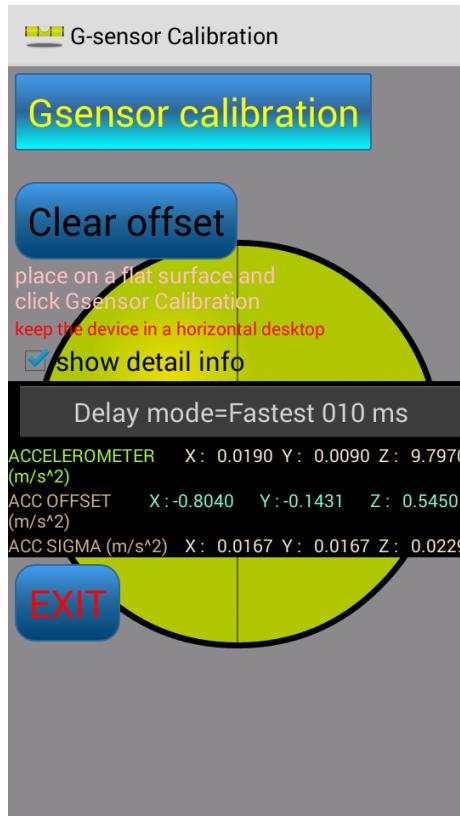


Figure 4(Gsensor calibration)

## C. The APK put inside the "setting > display"

DEFAULT : The apk(Glevel-mtk\_D.apk) on behalf of the show in the setting -> display.

- push Glevel-mtk\_D.apk in /system/priv-app

```
$adb push Glevel-mtk_D.apk /system/priv-app
```

- File "\$Android\packages\apps\Settings\res\xml\display\_settings.xml"

Add the following code to add an option to perform link "Level"

<Preference

    Android:key="accelerometerAdjust"

    Android:title="@string/gma\_calibration">

    <intent

        Android:targetPackage="com.globalmems.level"

        Android:targetClass=" com.globalmems.level.DLevel" />

</Preference>

- File "\$Android\packages\apps\Settings\res\values\strings.xml"

Add <string name="gma\_calibration">G-sensor Calibration</string>

## D. Two kinds of differences apk in source

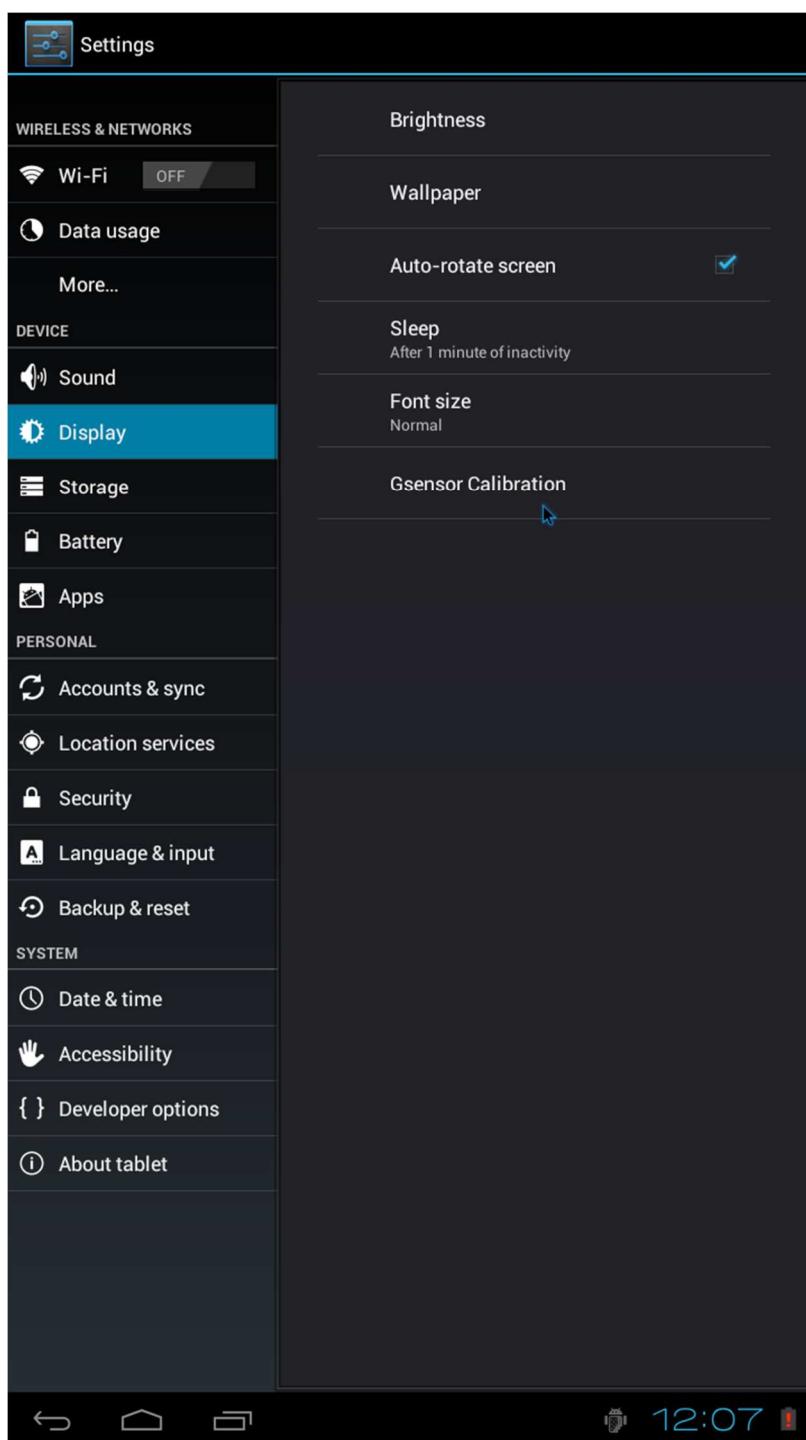
### 1. Modify AndroidManifest.xml

```
<category android:name="android.intent.category.LAUNCHER"/>
<category android:name="android.intent.category.DEFAULT"/>
```

LAUNCHER : The apk(Glevel-mtk\_L.apk) on behalf of the show on the desktop icon.

DEFAULT : The apk(Glevel-mtk\_D.apk) on behalf of the show in the setting -> display.

### 2. Rebuilding “Glevel-mtk.apk”, and Install



## BroadcastReceiver 剛安裝好時沒有辦法收到 Broadcast

---

Starting from Android 3.1, the system's package manager keeps track of applications that are in a stopped state and provides a means of controlling their launch from background processes and other applications.

Note that an application's stopped state is not the same as an Activity's stopped state. The system manages those two stopped states separately.

The platform defines two new intent flags that let a sender specify whether the Intent should be allowed to activate components in stopped application.

- System App

解決方法則是透過把 BroadcastReceiver 的 Application 直接 push 到 system/priv-app 下面

就可以直接讓 Application 成為 Active 狀態了

Ref: <http://fulungchen-blog.logdown.com/posts/261373-broadcast-receiver-active-inactive>