Calibration applications Operations Guide

for DMT Gsensor

Introduction

DMT_Calibration Features

How to compile libgsensor.so

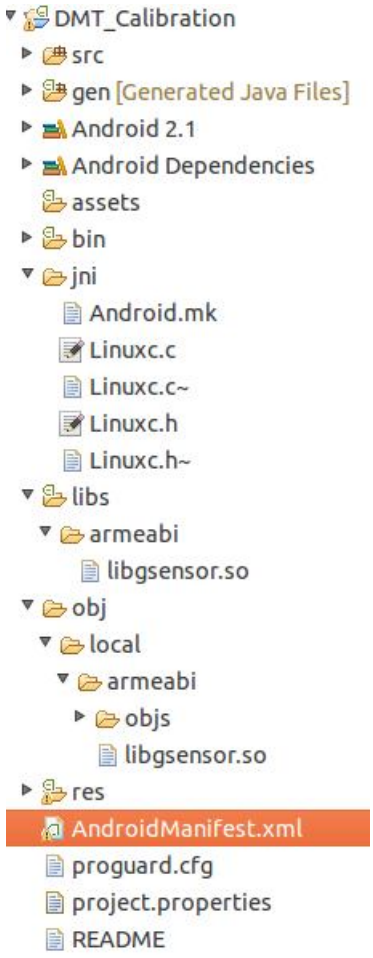
The APK put inside the "setting"

2012/9/11

A. Introduction

Name	DMT_Calibration
Project Name	DMT_Calibration
Application Name	DMT_Calibration
Package Name	tw.com.dmt.gsensor
Create Activity	DMT_GSENSORActivity.java
Min SDK Version	7

1. The picture shows DMT_Calibration.apk source structure.



1.1 src / source directory (source)

src directory contains the required code file in an Android application. These files are packets in the corresponding package subdirectories.

1.2 gen / directory automatically generated (Generate)

gen directory to store all files automatically generated.

the gen directory of the most critical program is the R.java file.

1.3 Android 2.1 / directory

1.4 Assets

1.5 Jni / directory

The Java Native Interface, JNI, is just that; an interface. It's an API that allows Java code to interact with code written in another language, typically C or C++. Because the JNI is an interface, JVM vendors are free to implement the virtual machine as they see fit. As long as the JVM follows the specification of the JNI, all native code written to the specification should work with that JVM.

1.6 Libs / directory

library libgsensor.so

1.7 Obj / directory

library libgsensor.so

1.8 res / resource directory (Resource)

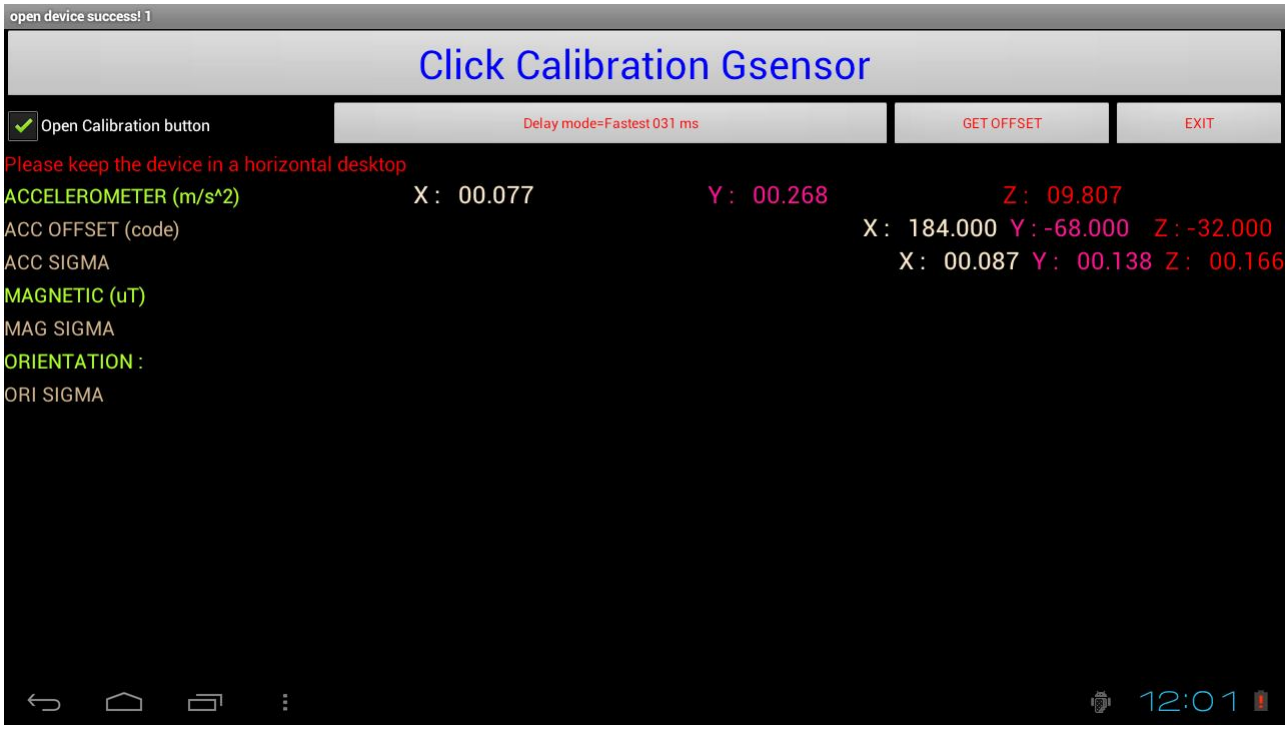
All resource files used in the program is stored in the "res" directory. Resource File data files or compile time will be converted into a program part of the XML description file. Android resources for different subdirectories on the "res" directory, there will be different approach. Therefore, when we write a program, it is best to be able to figure out the contents were placed in each directory.

1.9 AndroidManifest.xml

" AndroidManifest xml "is a list of features of the Android program, application functionality provided by the program are listed here. When the application is open, it will provide services such as content providers (ContentProvider), the type of data processing, the actual running category, cross-application information message. You can specify that your application will be used to service (such as phone features, network features, GPS function). When you add a new page behavior categories (Activity), you also need to be registered in this new Activity category, can be successfully invoked.

B. DMT_Calibration Features

1. Run DMT_Calibration.apk,
   Please keep the device in a horizontal desktop.



Button "Click calibration Gsensor"

When we get the following values, on behalf of Calibration is complete.

ACCELEROMETER(m/s^2) X: 0.0 Y: 0.0 Z: 9.8

Buotton "Delay mode"

The four delay mode can be changed.

Sensor delay fastest

Sensor delay game

Sensor delay ui

Sensor delay normal

Button "Get OFFSET"

This button to get the current offset. 1g = 1024LSB

Button "EXIT"

Exit the application.

2. Check offset.txt whether generated

   $ adb shell

   $ cat "/data/misc/dmt/offset.txt"

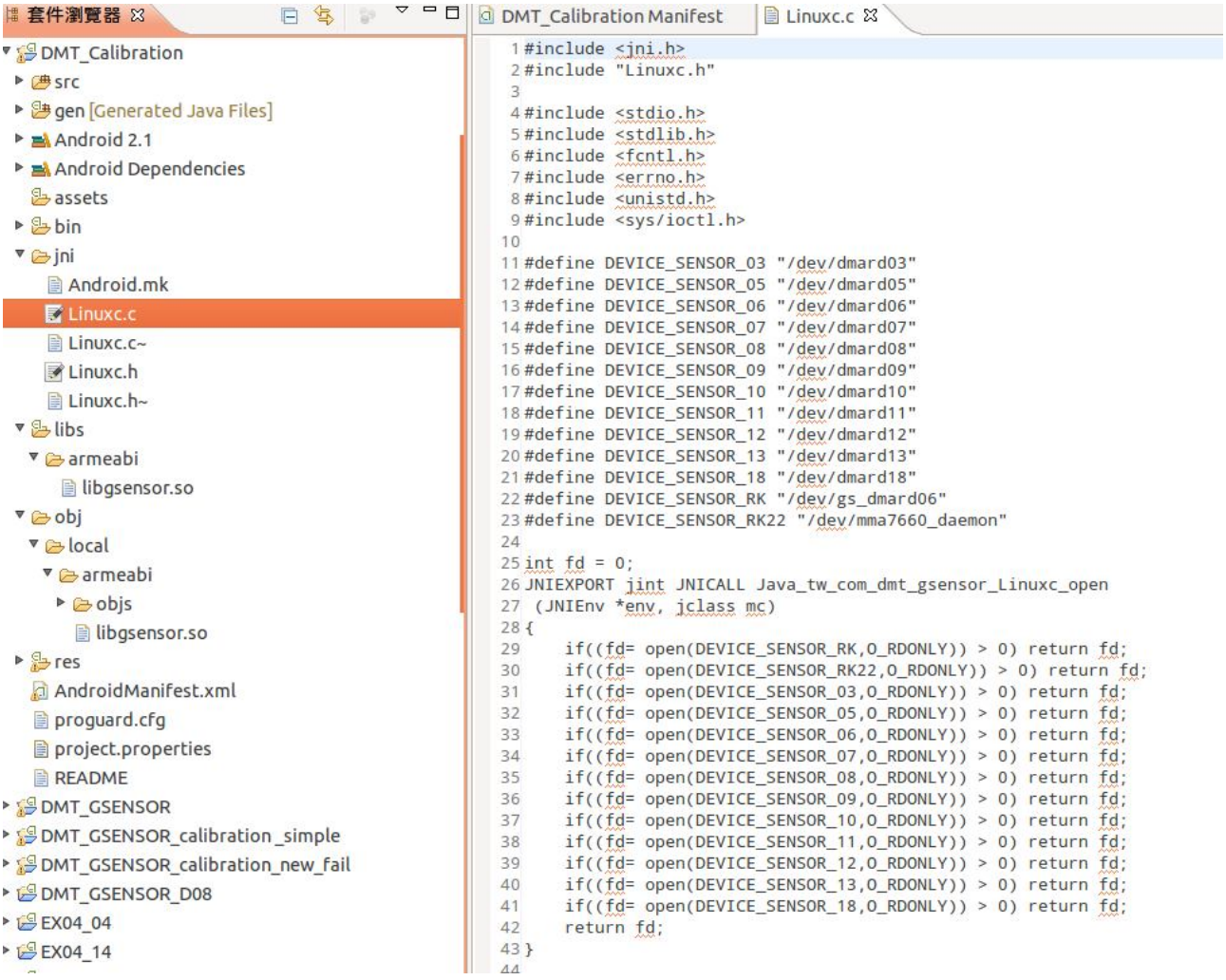
   If not, please refer to Part C.

   The confirm libgsensor.so correct

C. How to compile libgsensor.so

Subject to change “jni/linuxc.c” “jni/linuxc.h” must recompile libgsensor.so.
For example: If you want to open character device “/dev/gsensor”

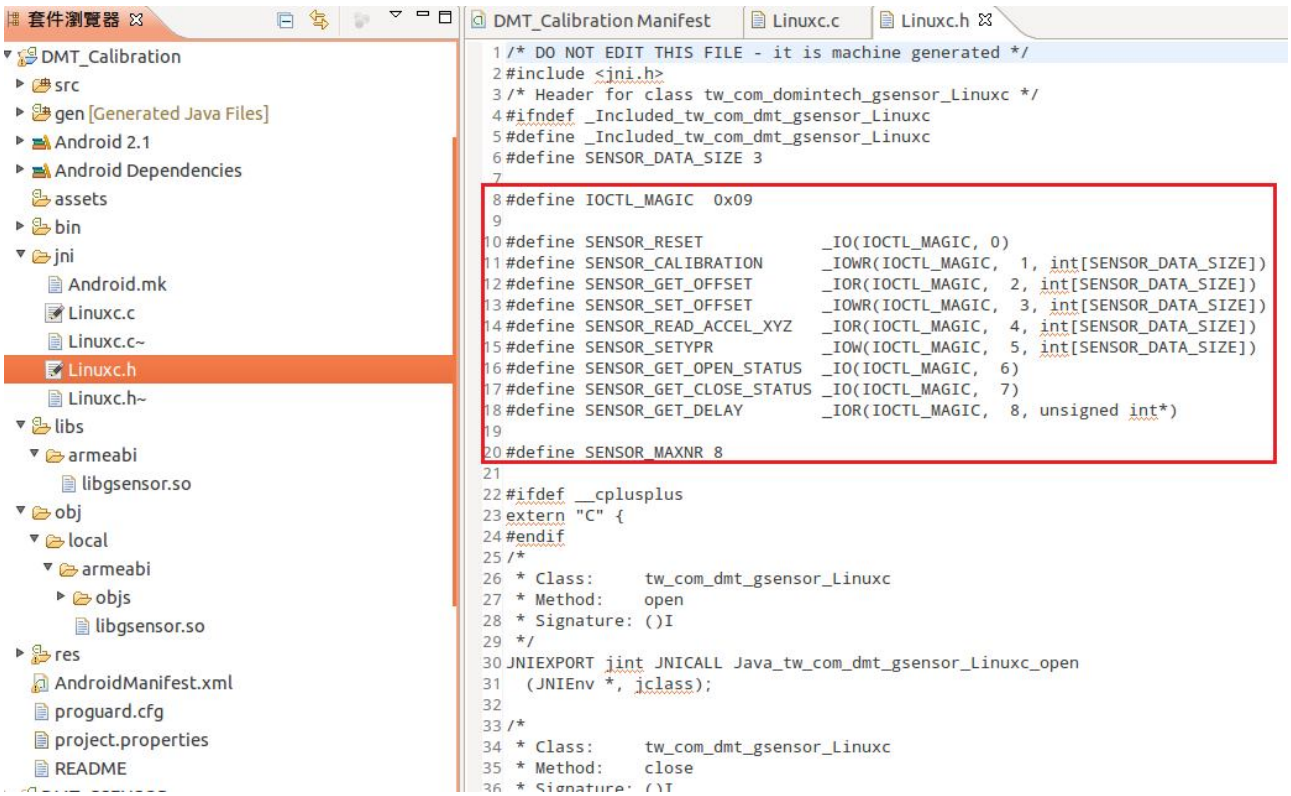
1. Check whether open gsensor character device
   1.1 Open file “jni/Linuxc.c”
   Add the following.
   #define example_dev “/dev/gsensor”
   1.2 In function “Java_tw_com_dmt_gsensor_Linuxc_open”
   Add the following.
   if((fd= open(example_dev,O_RDONLY))>0) return fd;



2. Check the the IOCTL magic number and interface

   Open file "jni/Linuxc.h"

   The following circle part replaced by a declaration in the current driver.



3. Recompile libgsensor.so

   $ cd DMT_Calibration/

   $ ndk-build

```
chiehyang@lab:~/workspace$ cd DMT_Calibration/
chiehyang@lab:~/workspace/DMT_Calibration$ ndk-build
Compile thumb  : gsensor <= Linuxc.c
jni/Linuxc.c: In function 'Java_tw_com_dmt_gsensor_Linuxc_readxyz':
jni/Linuxc.c:108: warning: 'tmp[0]' is used uninitialized in this function
jni/Linuxc.c:109: warning: 'tmp[1]' is used uninitialized in this function
jni/Linuxc.c:110: warning: 'tmp[2]' is used uninitialized in this function
SharedLibrary  : libgsensor.so
Install        : libgsensor.so => libs/armeabi/libgsensor.so
```

4. Back to Eclipse DMT_Calibration project

   Right-click on it and select Refresh.

   This action, it will reference new libgsensor.so.

D. The APK put inside the "setting"

The following is placed inside the apk: “setting” > “display”

1. File “$Android\packages\apps\Settings\res\xml\display_settings.xml”

Add the following code to add an option to perform link “DMT_GSENSORActivity”

```
<Preference
   Android:key=”accelerometerAdjust”
   Android:title=”@string/tscalibration_title”>
   <intent
        Android:targetPackage=”tw.com.dmt.gsensor”
        Android:targetClass=”tw.com.dmt.gsensor.DMT_GSENSORActivity” />
</Preference>
```

2. File ”$Android\packages\apps\Settings\res\values\strings.xml”

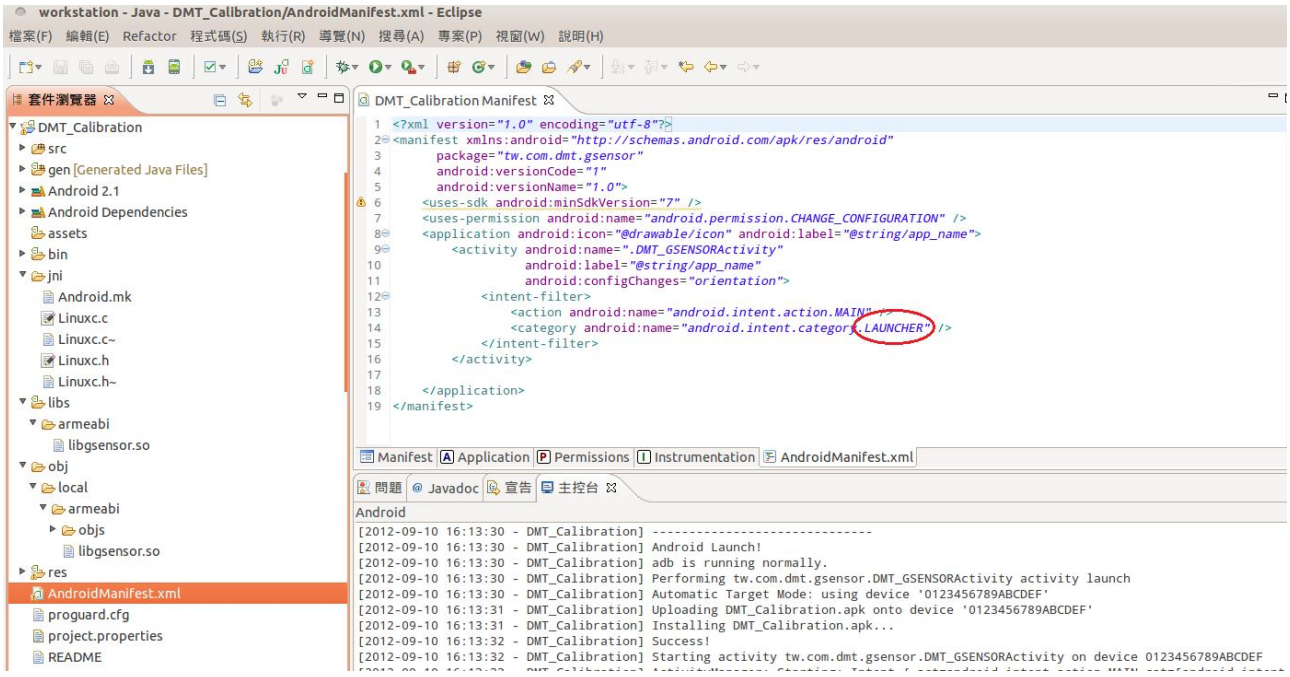
Add `<string name="tscalibration_title">G Sensor Calibration</string>`

3. In DMT_Calibration.apk source code

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

LAUNCHER : The apk on behalf of the show on the desktop icon.

DEFAULT : The apk on behalf of the show in the setting.



4. Rebuilding “DMT_Calibration.apk”,and Install