

# GMA302 Android Driver Porting Guide

## Files

Name	Version	Description
AndroidDriverPortingGuide.pdf	2014_11_11	This document
src/kernel/driver/gma30x.c src/kernel/driver/gma30x.h	version 0.1	gma302/gma303 linux driver
src/libsensors/*	2014_11_11	Android HAL
GlobalmemsShellScriptManual.pdf src/gmad/* src/gmad/gss.sh	2014_11_11	Globalmems Shell Script Manual Calibration Execution files Shell Script tools
LevelCalibrationManual.pdf Level/* GLevel_L.apk GLevel_D.apk	2014_11_11	Level Calibration Manual Calibration APP source LAUNCHER APP DEFAULT APP

## Introduction

This document is target to the program developers who port Android Driver. This document is a guideline helps to port to Android.

This driver was tested under Android version 4.4(KitKat) and 4.2(Jelly Bean). Android 2.3 is also supported (Experimental).

## Index

1.	Revision History .....	2
2.	Linux Driver Porting Guide.....	2
3.	Android Driver Porting Guide.....	5
4.	Mounting Direction of Acceleration Sensor .....	7
5.	Embedding Procedure of Acceleration Sensor Calibration Tool.....	8
6.	Acceleration Sensor Driver Specification .....	9
7.	FACTORY RESET .....	9

## 1. Revision History

Ver.	Date	Updates	Descriptions
0.1	Nov.11 <sup>th</sup> , 2014	4.4 KitKat 4.2 Jelly Bean	GMA302 first Release

## 2. Linux Driver Porting Guide

### Introduction

This document is intended for the program developers who port Linux Driver . This document shows a guideline to port to Linux.

This driver was tested with Linux kernel version 3.0.31. To compile this driver, enable "I2C Support" and "Input device support -> Miscellaneous devices" in Linux kernel configuration.

### 2.1 Copy "src/Driver/gma30x.c" to "\$KERNEL/drivers/misc/"

For example in our system it's in "\$KERNEL/drivers/misc/" and

Copy "src/Driver/gma30x.h" to "\$KERNEL/include/linux"

```
$ cp src/Driver/gma30x.c $KERNEL/drivers/misc/
$ cp src/Driver/gma30x.h $KERNEL/include/linux/
```

### 2.2 Specify the g-sensor layout by defining one of the following macro in "gma30x.h". The example code uses DEFAULT\_POSITION 0. See Figure 1 for other g-sensor layout pattern orientations.

```
/* Default sensor layout parameters */
#define GMS_GMA30x_DEFAULT_POSITION    1

/* Transformation matrix for chip mounting position */
first pin top:Positive  bottom:Negative
1: top/upper-left      (-y,-x,-z)
2: top/upper-right     (-x, y,-z)
3: top/lower-right     ( y,-x,-z)
4: top/lower-left      ( x,-y,-z)
-1: bottom/upper-left  $Equal position 2  (-(-x), y,-(-z))
-2: bottom/upper-right $Equal position 1  (-(-y),-x,-(-z))
-3: bottom/lower-right $Equal position 4  (-(-x), y,-(-z))
-4: bottom/lower-left  $Equal position 3  (-(-x), y,-(-z))
```

### 2.3 Makefile Modification

In drivers/misc/Makefile of the kernel source code tree, add the descriptions of the acceleration sensor driver.

```
obj-$(CONFIG_SENSORS_GMA302) += gma302.o
```

## 2.4 Kconfig Modification

In drivers/misc/Kconfig of the kernel source code tree, add the descriptions of the acceleration sensor driver.

```
config SENSORS_GMA302
    tristate "GMA302 GSENSOR support"
    default y
    depends on I2C=y
    ---help---
        If you say yes here you get support for accelemeter
        sensor GMA302.
```

## 2.5 i2c\_board\_info Registration

In the initialization process of the board, call `i2c_register_board_info()` to register `i2c_board_info`. Add descriptions in files such `arch/arm/mach-xxx/board_xxx.c` (`xxx` is an architecture name). include `<linux/i2c.h>` to call `i2c_register_board_info()`.

In the definition of `i2c_board_info` variable, use `I2C_BOARD_INFO("name", addr)` macro to register each sensor. Substitute "gma302", for "name". Set `addr` to the I2C slave address of the sensor device.

Add I2C information to the board information. For example in our system it's in "`$KERNEL/arch/arm/mach-omap2/board-omap4panda.c`". Locate your own board information file instead. g-sensor's 7-bit slave address would be 0x18.

```
#if defined(CONFIG_SENSORS_GMA302) || defined(CONFIG_SENSORS_GMA302_MODULE))
#include <linux/gma30x.h>
#define GMA302_GPIO_RSTN 0
// #define GMA302_GPIO_DRDY 134
/* Modify GMA302 data structure*/
static struct gma_platform_data platform_data_gma302 = {
    .layout = 1,
    .gpio_RSTN = GMA302_GPIO_RSTN,
    //.gpio_DRDY = GMA302_GPIO_DRDY,
};
#endif

static struct i2c_board_info __initdata panda_i2c_boardinfo[] = {
#if defined(CONFIG_SENSORS_GMA302) || defined(CONFIG_SENSORS_GMA302_MODULE))
```

```
{
    I2C_BOARD_INFO("gma302", 0x18),
    .flags = I2C_CLIENT_WAKE,
    .platform_data = &platform_data_gma302,
},
#endif
};

static int __init omap4_panda_i2c_init(void)
{
    omap4_pmic_init("twl6030", &omap4_panda_twldata);
    omap_register_i2c_bus(2, 400, NULL, 0);
    omap_register_i2c_bus(3, 100, panda_i2c_eeprom, ARRAY_SIZE(panda_i2c_eeprom));
    omap_register_i2c_bus(4, 400, panda_i2c_boardinfo, ARRAY_SIZE(panda_i2c_boardinfo));
    return 0;
}
```

## 2.6 Make

Run make menuconfig of the kernel, and select Device Drivers -> Miscellaneous devices then select the acceleration sensor.

Mounting Position on Board settings of the acceleration sensor.

Include the acceleration sensor in use into the kernel, then execute make to build the kernel.

Select GMA302 GSENSOR support

### 3. Android Driver Porting Guide

#### Introduction

This document is target to the program developers who port Android Driver. This document is a guideline helps to port to Android.

This driver was tested under Android version 4.2(Jelly Bean) and Android 4.4(KitKat).

#### 3.1 Placement of libsensors Source Code(sensor HAL) and acccalib Source Code(Calibration Execution files)

Copy “src/libsensors” directory to “\$ANDROID/hardware/libhardware/module/libsensors”

Copy “src/gmad” directory to “\$ANDROID/hardware/libhardware/module/libsensors”

```
$ cp -r src/libsensors hardware/libhardware/module/  
$ cp -r src/gmad hardware/libhardware/module/
```

#### 3.2 Modify Android.mk or other board specific makefiles.

For example, add the following descriptions (Red part) to Hardware\_modules in “\$ANDROID/hardware/libhardware/module/Android.mk” as shown below.

```
Hardware_modules := gralloc hwcomposer audio nfc libsensors gmad  
Include $(call all-names-subdir-makefiles,$(hardware_modules))
```

#### 3.3 Modification of device.mk

For example, add the following descriptions (Red part) to “PRODUCT\_COPY\_FILES” section in “\$ANDROID/device/ti/panda/device.mk” as shown below.

```
PRODUCT_COPY_FILES := \  
    $(LOCAL_KERNEL):kernel \  
    device/ti/panda/init.omap4pandaboard.rc:root/init.omap4pandaboard.rc \  
    device/ti/panda/init.omap4pandaboard.usb.rc:root/init.omap4pandaboard.usb.rc \  
    device/ti/panda/fstab.omap4pandaboard:root/fstab.omap4pandaboard \  
    device/ti/panda/ueventd.omap4pandaboard.rc:root/ueventd.omap4pandaboard.rc \  
    ..snip..  
    hardware/libhardware/modules/acccalib/gss.sh:/system/bin/gss.sh
```

#### 3.4 Modification of init.rc

If the kernel driver has completed [gma\\_permission\\_thread](#), you can skip this step.

In “\$ANDROID/system/core/rootdir/init.rc” of Android source code tree,

Modify the permission of sysfs interface. Sensor HAL should be allowed to access to sysfs interface of the acceleration sensor. Add the following descriptions to “on boot” section in init.rc.

```
on boot
# basic network init
    ifup lo
..snip..
    mkdir /data/misc/sensor 01777 system shell
    chown system system /sys/kernel/ipv4/tcp_rmem_def
    chown system system /sys/kernel/ipv4/tcp_rmem_max
    chown root radio /proc/cmdline
    chmod 666 /dev/gma302
    chmod 777 /system/bin/gss.sh
    chmod 755 /system/bin/acccalib
    chown system system /sys/class/input/input0/enable
    chown system system /sys/class/input/input0/delay
```

On the above, input2, input3, input4, input5, and input6 correspond to sysfs interface numbers of each acceleration sensor driver, geomagnetic sensor driver, geomagnetic raw sensor driver, orientation sensor driver, gyro sensor driver. In order to find sysfs interface number, use “getevent” check /sys/class/input/inputX/name (X indicates a number) after embedding each driver to the kernel and starting-up.

### 3.5 Rebuild the Android image

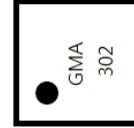
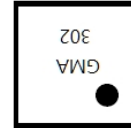
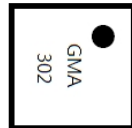
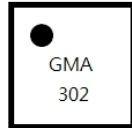
## 4. Mounting Direction of Acceleration Sensor

The mounting position on the board of the acceleration sensor could be changed by modifying the setting in init.rc.

The mounting position on the board and the setting value correspond as below. The position of ● shows No. 1 pin of the acceleration sensor.

### Top View

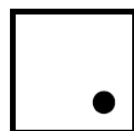
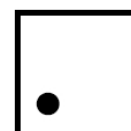
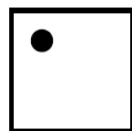
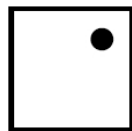
Position of the sensors



Setting Value	1	2	3	4
---------------	---	---	---	---

### Bottom View

Position of the sensors



Setting Value	-1	-2	-3	-4
---------------	----	----	----	----

### Sensor Mounting Direction Setting

In order to set the mounting direction of the acceleration sensor, add the setting value on sysfs interface “position” of the drivers. For example, in the case of setting “-1” as the mounting direction, add the following in “on boot” section of init.rc.

```
on boot
# basic network init
    ifup lo
..snip..

    chown system system /sys/kernel/ipv4/tcp_rmem_def
    chown system system /sys/kernel/ipv4/tcp_rmem_max
    chown root radio /proc/cmdline
write /sys/class/input/input0/position -1
```

## 5. Embedding Procedure of Acceleration Sensor Calibration Tool

In this driver, the acceleration sensor calibration tool is provided for getting the offsets of an acceleration sensor. If necessary, embed this calibration tool into the system.

### 5.1 Copy “src/gmad” directory to “\$ANDROID/hardware/libhardware/module/”

```
$ cp -r src/gmad hardware/libhardware/module/
```

### 5.2 Modify Android.mk or other board specific makefiles to embed the Acceleration Sensor Calibration Tool on Android userland.

For example, add acccalib to Hardware\_modules in

“\$ANDROID/hardware/libhardware/module/Android.mk” as shown below.

```
Hardware_modules := gralloc hwcomposer audio nfc libsensors gmad
Include $(call all-names-subdir-makefiles,$(hardware_modules))
```

### 5.3 Make

Run make command to compile under the top directory of Android source code tree.

Acceleration Sensor Calibration Tool is stored under the following path.

Name	Path
Acceleration Sensor Calibration Tool	/system/bin/gmad

### 5.4 Start-up of Acceleration Sensor Calibration Tool

To operate calibration of the acceleration sensor, place your phone horizontally and set still. Then start-up acceleration sensor calibration tool with root authority. When the calibration is successfully operated, the offsets of the acceleration sensor will be shown on the command line in LSB unit as shown below. (1g = 1024 LSB)

```
$ /system/bin/gmad -c 9
Accelerometer Offset [108] [32] [-64]
```

### 5.5 Acceleration Sensor Calibration Offset Setting

In order to set the calibration offset of the acceleration sensor, add the setting value on sysfs interface “offset” of the acceleration sensor driver. For example, in the case of setting “108 32 -64” as the calibration offsets, add following in “on boot” section of init.rc. In this case “gma302” corresponds to the sysfs interface number of the acceleration sensor driver.

```
on boot
# basic network init
    ifup lo
..snip..
    chown system system /sys/kernel/ipv4/tcp_rmem_def
    chown system system /sys/kernel/ipv4/tcp_rmem_max
    chown root radio /proc/cmdline
    write /sys/class/input/input0/offset “ 108 32 -64”
```



## 6. Acceleration Sensor Driver Specification

Acceleration Sensor Driver is registered for input class. In /sys/class/input/inputX/ on sysfs. The following are sysfs interface of the acceleration sensor.

Name	Range	rw	Default	Descriptions
enable	0,1	rw	0	Write "1" to enable the sensor output Write "0" to disable the sensor output
delay	0-200	rw	100	Set the output interval of sensor in msec unit
data	INT_MIN-INT_MAX	ro	0 0 0	Read the latest data of the sensor (output in order of X,Y,Z value)
offset	INT_MIN-INT_MAX	rw	0 0 0	Set the accelerometer offset (output in order of X,Y,Z value)
position	1,2,3,4,-1,-2,-3,-4	rw	0	Set the sensor mounting direction on board
chipinfo	char	ro	gma302	Gsensor identification
reg	char	ro		Lists the REG value of the current

## 7. FACTORY RESET

If user selects "factory reset" , main system reboots into recovery.

Will erase\_volume 0 reformats /data.

We must save gsensor offset,

Recovery.cpp must be modified so that offset is preserved until the next boot.

Refer reference setup in the "src/recovery.cpp"

```
#define GMA_BACKUP_FILE
#ifdef GMA_BACKUP_FILE
static const char *OFFSET_FILE_PATH = "/data/misc/sensor/offset.txt";
static const char *TEMPORARY_BACKUP_PATH = "/tmp/offset.txt";
static const char *DEST_FILE_PATH="/data/misc/sensor/offset.txt";
#endif
```