



Smart MEMS technologies link the globe

GlobalMems Accelerometer Sensor -Proting Guides

Platform: MTK

Rev: V0.1

Date: 6/26/2015

目錄

目錄	2
1. REVISION HISTORY	2
2. PREFACE.....	3
3. RELEASE FOLDERS.....	3
4. GSENSOR 兼容摘要.....	4
5. FACTORY MODE	7

1. Revision History

Version	Revision Date	Descriptions
0.1	06/26/2015	First release of document applicable to gma30x products.

2. Preface

This document highlights the fundamental porting procedures to comply with gma30x driver to MTK platform.

3. Release folders

The software release packages contain the following gma30x device specific folders.

KK : for Android 4.x

L : for Android 5.x

Level APK : gsensor auto calibration

A. Kernel Driver

Android 4.x(Kitkat)

\gma30x_mtk\KK\alps\mediatek\custom\common\kernel\accelerometer\gma302

\gma30x_mtk\KK\alps\mediatek\custom\common\kernel\accelerometer\gma303

\gma30x_mtk\KK\alps\mediatek\custom\common\kernel\accelerometer\gma305

Android 5.x(Lollipop)

\gma30x_mtk\L\alps\device\mediatek\common\sepolicy\untrusted_app.te

\gma30x_mtk\L\alps\kernel-3.10\drivers\misc\mediatek\accelerometer\gma302

\gma30x_mtk\L\alps\kernel-3.10\drivers\misc\mediatek\accelerometer\gma303

\gma30x_mtk\L\alps\kernel-3.10\drivers\misc\mediatek\accelerometer\gma305

B. Java Application

\Glevel\Glevel-mtk_D.apk

\Glevel\Glevel-mtk_L.apk

\Glevel\Calibration_Bubble_Level_mtk.pdf

Commit. L版本untrusted_app.te說明：

(1) 請將以下的內容增加到codebase中相應路徑下的文檔中；

allow untrusted_app em_svr:unix_stream_socket connectto;

(2) alps\device\mediatek\common\sepolicy\untrusted_app.te 此文檔中的內容是為vendor的校準apk增加的，若不使用vendor的校準apk，可以不增加此文檔的內容；

(3) 客制化部分mach\mt6735\k35v1_64是以：

platform : mt6735

project : k35v1_64

為例寫的路徑，實際使用時請換成實際使用的platform和project。

4. Gsensor 兼容摘要

修改方式摘要如下:

4.1 hwmsen_dev.c

在函式 gsensor_probe 處,當加載 g sensor 的初始化函式

gsensor_init_list[i]->init() 失敗時,須增加

gsensor_init_list[i]->unint() 將其解除初始化

(解除初始化的函式 unint() 是 MTK 原本有寫好的 code,但是並沒有被使用.)

4.2 bma2xx.c

在函式 bma_get_chip_type 處須強制驗證 sensor 暫存器 0x04 的值是否為 0x55

若是 0x55,表示這不是 boach 的 g sensor, 須強制退出

```

00761:         return err;
00762:     } ? end_bma_write_calibration ?
00764:
00765: /* get chip type */
00766: #define GMA_AUTO_DETECT
00767:
00768: static int bma_get_chip_type(struct i2c_client *client)
00769: {
00770:     int err = 0;
00771:     u8 chip_id = 0;
00772:
00773: #ifdef GMA_AUTO_DETECT //2015_0625 modified.
00774:     u8 chip_id_gma = 0;
00775: #endif
00776:     struct bma_i2c_data *obj = i2c_get_clientdata(client);
00777:     GSE_FUN(f);
00778:
00779: /* twice */
00780:     err = bma_i2c_read_block(client, BMA_CHIP_ID_REG, &chip_id, 0x01);
00781:     err = bma_i2c_read_block(client, BMA_CHIP_ID_REG, &chip_id, 0x01);
00782:     if (err != 0)
00783:         return err;
00784:
00785: #ifdef GMA_AUTO_DETECT //2015_0625 modified.
00786:     err = bma_i2c_read_block(client, 0x04, &chip_id_gma, 0x01);
00787:     printk("again bma_get_chip_type=====%d\n", chip_id_gma);
00788:     if (chip_id_gma == 0x55) return -1;
00789: #endif
00790:
00791:     switch (chip_id) {
00792:     case BMA220_CHIP_ID:
00793:         obj->sensor_type = BMA220_TYPE;
00794:         break;
00795:     default:
00796:         obj->sensor_name = "bma220";
00797:     }

```

```

sen_dev.c
01397: static int gsensor_probe(struct platform_device *pdev)
01398: {
01399:     int i = 0;
01400:     int err = 0;
01401:     HWM_LOG(" gsensor_probe +\n");
01402:
01403:     /*
01404:      for(i = 0; i < MAX_CHOOSE_G_NUM; i++)
01405:      {
01406:          HWM_LOG(" gsensor_init_list[%d]=%d\n",gsensor_init_list[i]);
01407:
01408:      }
01409: */
01410: /*
01411: for(i = 0; i < MAX_CHOOSE_G_NUM; i++)
01412: {
01413:     HWM_LOG(" i=%d\n",i);
01414:     if(0 != gsensor_init_list[i])
01415:     {
01416:         HWM_LOG(" !!!!!!!\n");
01417:         err = gsensor_init_list[i]->init();
01418:         if(0 == err)
01419:         {
01420:             strcpy(gsensor_name,gsensor_init_list[i]->name);
01421:             HWM_LOG(" gsensor %s probe ok\n", gsensor_name);
01422:             break;
01423:         }
01424:     else
01425:     {
01426:         HWM_LOG(" gsensor %s probe fail\n", gsensor_name);
01427:         gsensor_init_list[i]->uninit();
01428:     }
01429:
01430: }
01431: } ? end for i=0;i<MAX_CHOOSE_G_NUM ?
01432:

```

4.3 bma222.c

- 兼容時 BMA222_DEV_NAME 由“BMA222”改為 “gsensor”
- 在程式 if(databuf[0]!=BMA222_FIXED_DEVID) 處

強制驗證 sensor 暫存器 0x04 的值是否為 0x55

若是 0x55,表示這不是 boach 的 g sensor, 須強制退出

```

right - [Bma222.c (z:\s138aea_kk\...\bma222_auto)]
Search Project Options View Window Help
00048: /*
00049: // #define CONFIG_BMA150_LOWPASS /*apply low pass filter on output
00050: #define SW_CALIBRATION
00051:
00052: /*
00053: #define BMA222_AXIS_X 0
00054: #define BMA222_AXIS_Y 1
00055: #define BMA222_AXIS_Z 2
00056: #define BMA222_AXES_NUM 3
00057: #define BMA222_DATA_LEN 6
00058: #define BMA222_DEV_NAME "gsensor" // "BMA222"
00059: /*
00060:
00061: /*
00062: static const struct i2c_device_id bma222_i2c_id[] = {{BMA222_DEV_N
00063: static struct i2c_board_info __initdata i2c_BMA222= { I2C_BOARD_INFO
00064: /*the adapter id will be available in customization*/

```

VM VirtualBox

```

c (z:\s138aea_kk\...\bma222_auto)]
```

File Insert View Window Help

File Edit View Insert Run Tools Window Help

```

1: if (databuf[0] != BMA222_FIXED_DEVID)
2: {
3:     printk("BMA222_CheckDeviceID %d failt!\n", databuf[0]);
4:     return BMA222_ERR_IDENTIFICATION;
5: }
6: else
7: {
8:
9: #ifdef GMA_AUTO_DETECT //2015_0625 modified.
10:    u8 databuf_gma[2];
11:    databuf_gma[0]=0X04;
12:    res = i2c_master_send(client, databuf_gma, 0x1);
13:    if(res <= 0)
14:    {
15:        goto ↓exit_BMA222_CheckDeviceID;
16:    }
17:    udelay(500);
18:    databuf_gma[0] = 0x0;
19:    res = i2c_master_recv(client, databuf_gma, 0x01);
20:    if(res <= 0)
21:    {
22:        goto ↓exit_BMA222_CheckDeviceID;
23:    }
24:    printk("again BMA222_CheckDeviceID===== %d\n", databuf_gma[0]);
25:    if (databuf_gma[0] == 0x55) return BMA222_ERR_IDENTIFICATION;
26:    else
27: #endif //2015_0625 modified end.
28:         printk("BMA222_CheckDeviceID %d pass!\n", databuf[0]);
29: } ? end else ?
30:
31: exit_BMA222_CheckDeviceID:
32: if (res <= 0)
33: {
34:
35: deviceID: Label of BMA222_CheckDeviceID in Bma222.c (z:\s138aea_kk\...\bma222_auto) at line 557
36:
37: deviceID:
38:
39: I2C:
40:
41: checkDeviceID
42:
43: Zhangx... Source... gma303... Smart... UART... GMA303...
44:
45: u [正在运行] - Oracle V...
46:
47: AOC
48:
49: EXIT AUTO +
```

4.4 gma303.c

兼容時統一將 GMA30x_DEV_NAME 由“gma303”改為“gsensor”

The screenshot shows a software development environment with a menu bar (File, Project, Options, View, Window, Help) and a toolbar with various icons. The main window displays the code for Gma303.c. The code includes several defines for sensor parameters like GMA30x_AXIS_X through GMA30x_AXES_NUM, and defines for TRUE and FALSE. A search bar at the top right contains the text "gma303".

```
[Gma303.c (z:\s138aea_kk\...\gma303_auto)]
Project Options View Window Help
File Edit View Insert Run Tools Help
00046: /*-
00047:
00048: #define _GMA303_SUPPORT_AUTO_DETECT_
00049: #define ABS(a) ((a) < 0 ? -(a) : (a))
00050: /*-
00051: #define CONFIG_GMA30x_LOWPASS /*apply low pass filter on output*/
00052: #define SW_CALIBRATION // only use cali_sw, hw offset = 0
00053: /*-
00054: #define GMA30x_AXIS_X 0
00055: #define GMA30x_AXIS_Y 1
00056: #define GMA30x_AXIS_Z 2
00057: #define GMA30x_AXES_NUM 3
00058: #define GMA30x_DATA_LEN 11
00059: #ifdef _GMA303_SUPPORT_AUTO_DETECT_
00060: #define GMA30x_DEV_NAME "gsensor"// "gma303"
00061: #else
00062: #define GMA30x_DEV_NAME "gma303"
00063: #endif
00064:
00065: /*-
00066: #define TRUE 1
00067: #define FALSE 0
00068: #if 0
00069:
```

5. Factory Mode

(1) Factory mode 自动测试项修改地方如下：

文件 : item.cpp

`ftm_auto_test_items` : 这里面的 item 是自动测试项

ftm test items : 这里面的 item 是测试项

可以尝试将 `ftm_test_items` 中的 `gsensor` 校准项 `copy` 到 `ftm_auto_test_items` 中看是否可以。

文件路径：

KK: alps\mediatek\factory\src\factory.c

L : alps\vendor\mediatek\proprietary\factory\src\item.cpp

(2) engineer mode 路径：

KK: alps\mediatek\packages\apps\EngineerMode*

L: vendor\mediatek\proprietary\packages\apps\EngineerMode*

备注：

上面的文件路径都是在客户能够看到源码的前提下才有用。