GlobalMems Accelerometer Sensor -Proting Guides

Platform: MTK

Rev: V0.1

Date: 6/26/2015

目錄

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

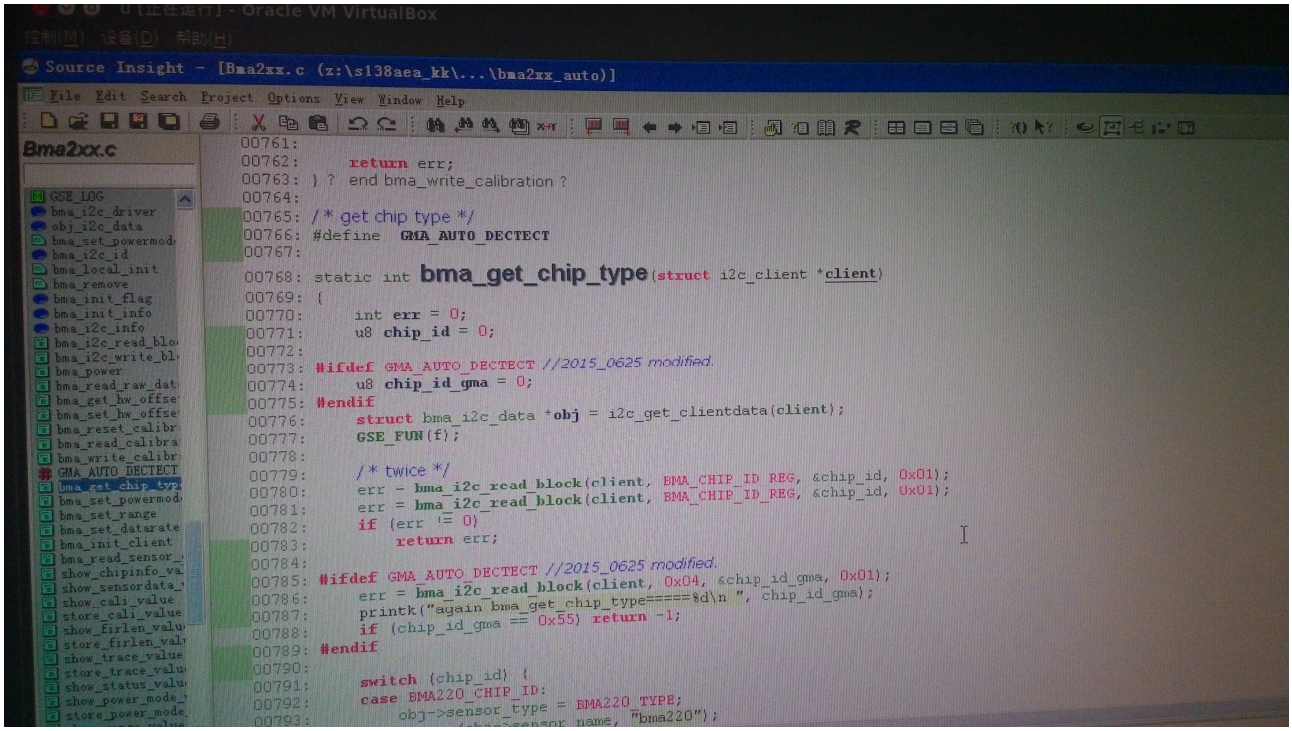
gsensor_init_list[i]->unint() 將其解除初始化

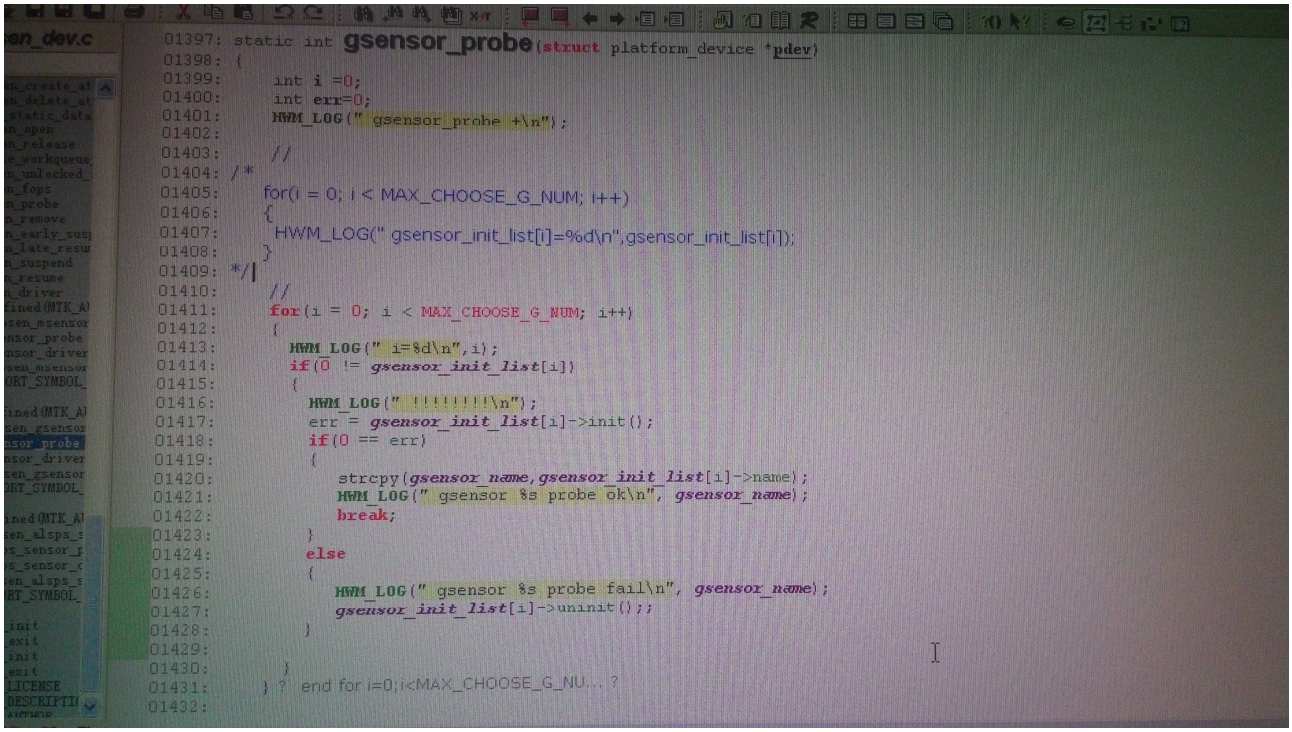
(解除初始化的函式 unint() 是 MTK 原本有寫好的 code,但是並沒有被使用.)

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





```
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: /*
01405:     for(i = 0; i < MAX_CHOOSE_G_NUM; i++)
01406:     {
01407:      HWM_LOG(" gsensor_init_list[i]=%d\n",gsensor_init_list[i]);
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_NU... ?
01432:
```

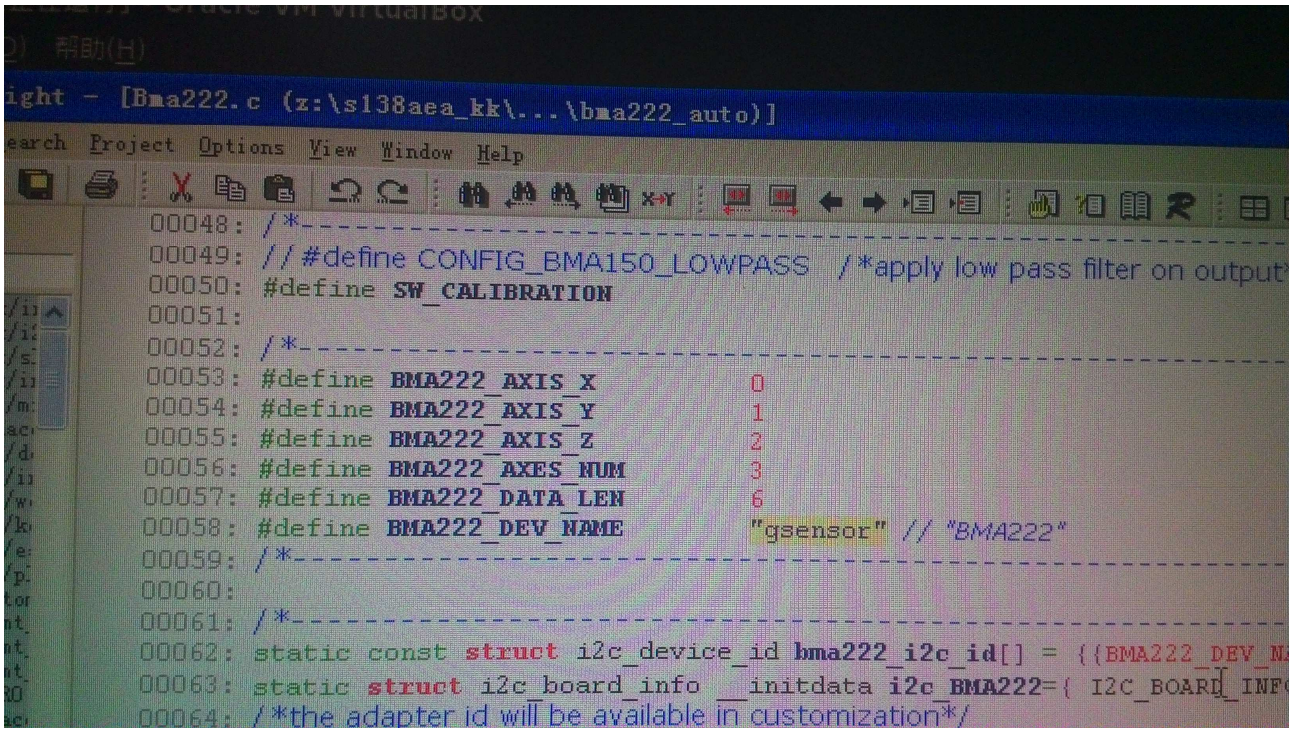
4.3 bma222.c

a. 兼容時 BMA222_DEV_NAME 由”BMA222”改為 “gsensor”

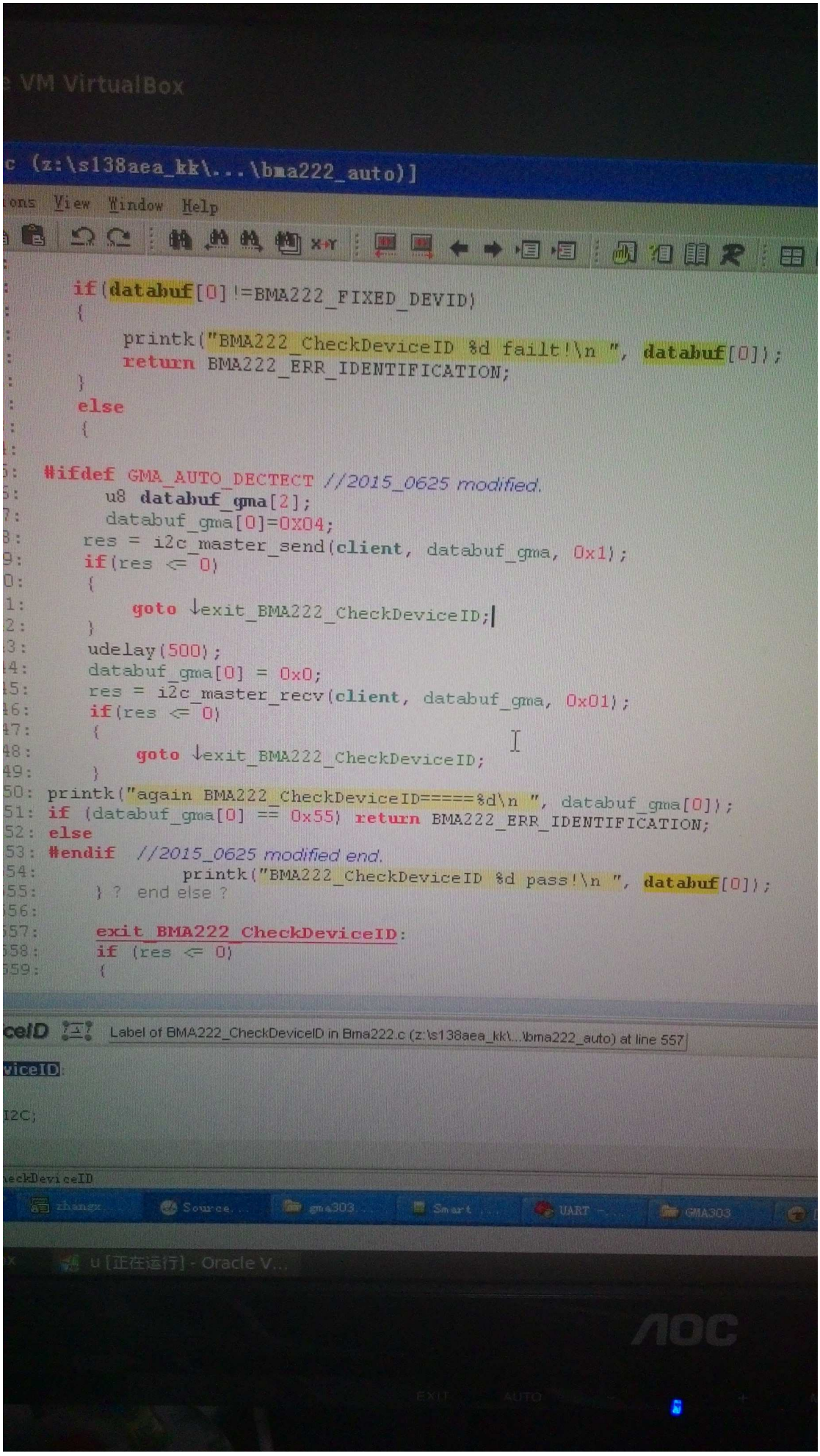
b. 在程式 if(databuf[0]!=BMA222_FIXED_DEVID) 處

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

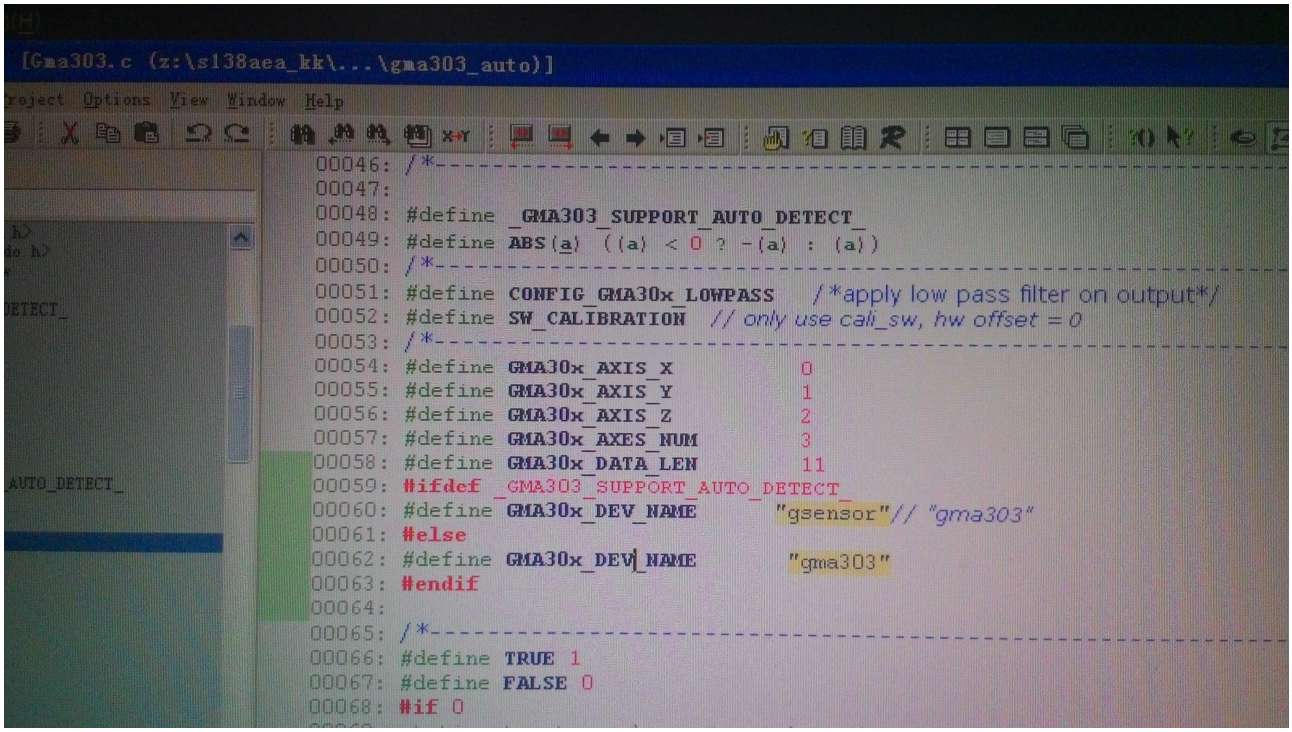


```
ight - [Bma222.c (z:\s138aea_kk\...\bma222_auto)]
earch Project Options View Window Help
00048: /*-----------------------------------------------------------------
00049: //#define CONFIG_BMA150_LOWPASS  /*apply low pass filter on output
00050: #define SW_CALIBRATION
00051:
00052: /*-----------------------------------------------------------------
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_INF
00064: /*the adapter id will be available in customization*/
```


VM VirtualBox
c (z:\s138aea_kk\...\bma222_auto)]
ions View Window Help
if(databuf[0]!=BMA222_FIXED_DEVID)
{
printk("BMA222_CheckDeviceID %d failt!\n ", databuf[0]);
return BMA222_ERR_IDENTIFICATION;
}
else
{
#ifdef GMA_AUTO_DECTECT //2015_0625 modified.
u8 databuf_gma[2];
databuf_gma[0]=0X04;
res = i2c_master_send(client, databuf_gma, 0x1);
if(res <= 0)
{
goto exit_BMA222_CheckDeviceID;
}
udelay(500);
databuf_gma[0] = 0x0;
res = i2c_master_recv(client, databuf_gma, 0x01);
if(res <= 0)
{
goto exit_BMA222_CheckDeviceID;
}
printk("again BMA222_CheckDeviceID=====%d\n ", databuf_gma[0]);
if (databuf_gma[0] == 0x55) return BMA222_ERR_IDENTIFICATION;
else
#endif //2015_0625 modified end.
printk("BMA222_CheckDeviceID %d pass!\n ", databuf[0]);
} ? end else ?
exit_BMA222_CheckDeviceID:
if (res <= 0)
{
Label of BMA222_CheckDeviceID in Bma222.c (z:\s138aea_kk\...\bma222_auto) at line 557
AOC

4.4 gma303.c

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



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。