AS9888 Android Driver V1.1.0.1214 Porting Guide

Files

Name	Version	Description
AS9888_AD_V1.1.0.1214_PG.pdf	20111214	This document
AS9888_API_V1.0.0.823_Manual.pdf	20111129	API Manual
AS9888_FST_V1.0.0.823_Manual.pdf	20111129	Factory shipment test manual
AS9888_Daemon_V1.0.0.823_Manual.pdf	20111129	Daemon operation manual
src/Kernel_Driver/*	1.1.0.1214	Linux I2C device driver
src/HAL/libsensors/*	1.0.0.823	Android HAL
src/as9888/*	1.0.0.823	Daemon program dmtd
src/obj/STATIC_LIBRARIES/libAS9888_intermediates/libAS9888.a	1.1.0.1214	libAS9888.a (static library)

Android Version

AS9888 Android Driver V1.1 supports Android 4.0 (Ice-Cream-Sandwich) system.

Contents

1.Build Kernel Image

1.1 Copy “src/Kernel_Driver/as9888.c” to “$KERNEL/drivers/misc/”

1.2 Copy “src/Kernel_Driver/as9888.h” to “$KERNEL/include/linux”

1.3 Modify “$KERNEL/driver/misc/Makefile” with addition of the following line

```
obj-$(CONFIG_SENSORS_AS9888)        += as9888.o
```

1.4 Modify “$KERNEL/driver/misc/Kconfig” with addition of the following line

```
config SENSORS_AS9888
        tristate "AS9888 compass support"
        default y
        depends on I2C=y
        ---help---
          If you say yes here you get support for orientation
          sensor AS9888.
```

1.5 Setup interrupt pin 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.

Interrupt line modification

on board interrupt line is connected to GPIO 134 & GPIO 136.

Please change the following if your accelerometer interrupt line is connected to different GPIO.

Pin mux change in board configuration file.

Add following line to board_mux structure to arch/arm/mach-omap2/board-omap4panda.c

```
#include <linux/irq.h>
#if (defined(CONFIG_SENSORS_AS9888) | | \
                                defined(CONFIG_SENSORS_AS9888_MODULE))
#include <linux/as9888.h>
#define AS9888_GPIO_RSTN      0
#define AS9888_GPIO_INT1      134
#define AS9888_GPIO_INT2      136
#endif

/*ADD Modify AS9888 data structure*/
#if (defined(CONFIG_SENSORS_AS9888) | | \
                          defined(CONFIG_SENSORS_AS9888_MODULE))
static struct as9888_platform_data platform_data_as9888 = {
            .gpio_RSTN = AS9888_GPIO_RSTN,
            .gpio_INT1 = AS9888_GPIO_INT1,
            .gpio_INT2 = AS9888_GPIO_INT2,
};
#endif

static struct omap_board_mux board_mux[] __initdata = {
             /* GPIO 134 & GPIO 136 */
             OMAP3_MUX(MCSPI1_CLK, OMAP_PIN_INPUT |OMAP_MUX_MODE3),
             OMAP3_MUX(MCSPI1_SIMO, OMAP_PIN_INPUT|OMAP_MUX_MODE3 ),
             { .reg_offset = OMAP_MUX_TERMINATOR },
 };
```

Note: Please change OMAP_MUX arguments as per your GPIO No.

```
static void __init panda_sensor_init(void) {
            omap_mux_init_gpio(134, OMAP_PIN_INPUT);
            omap_mux_init_gpio(136, OMAP_PIN_INPUT);
            /* INPUT */
            gpio_request(134, "as9888_gpio_int1");
            gpio_direction_input(134);
            irq_set_irq_type(OMAP_GPIO_IRQ(134), IRQ_TYPE_EDGE_RISING);

            gpio_request(136, "as9888_gpio_int2");
            gpio_direction_input(136);
            irq_set_irq_type(OMAP_GPIO_IRQ(136), IRQ_TYPE_EDGE_RISING);
}
```

1.6 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.

ADD I2C board info structure

On Pandaboard AS9888 is connected to I2C4. If your sensor is interfaced on different I2C bus then please do the following changes to your board file.

```
static struct i2c_board_info __initdata panda_i2c_boardinfo[] = {
#if (defined(CONFIG_SENSORS_AS9888) | | \
                defined(CONFIG_SENSORS_AS9888_MODULE))
        {
                I2C_BOARD_INFO("as9888", 0x1C),
                .flags = I2C_CLIENT_WAKE,
                .platform_data = &platform_data_as9888,
                .irq = OMAP_GPIO_IRQ(AS9888_GPIO_INT1),
        },
#endif
};
```

Register I2C bus

Register I2C bus with the i2c sub-system.

```
static int __init omap3_beagle_i2c_init(void)
{
        .
        .
        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;
}
```

1.7 Rebuild the kernel image

2.Build Android Image

2.1 Copy src/HAL/libsensors directory to “$ANDROID/hardware/ti/omap4xxx/libsensors”
For example in our system it’s in “$ANDROID/hardware/ti/omap4xxx”

```
$ cp –r src/HAL/libsensors hardware/ti/omap4xxx/
```

2.2 Copy src/as9888 directory to “$ANDROID/external/as9888”

```
$ cp –r src/as9888 externel/as9888
```

2.3 Rebuild the Android image

2.4 Rebuild the libsensors
For example in our system it’s in “$ANDROID/hardware/ti/omap4xxx”

```
$ mmm “$ANDROID/hardware/ti/omap4xxx/libsensors”
```

Copy “$ANDROID/out/target/product/<TARGET_PRODUCT>/system/lib/hw/sensors.default.so” to the “<android_rootfs>/system/lib/hw/sensors.default.so”

2.5 Copy static library (libAS9888.a) to “$ANDROID/out/target/product/<TARGET_PRODUCT>”

```
$cp –r src/obj
“$ANDROID/outtarget/product/<TARGET_PRODUCT>/”
```

2.6 Setting up

Correct file permission should be set to device files, so that Sensor Control Program and Sensor HAL Module can access to a sensor data. From Gingerbread, permission is set according to the resource file “ueventd.rc”. Please modify the file.

```
$vi “$ANDROID/out/target/product/<TARGET_PRODUCT>/root/ueventd.rc”
#Add following line
/dev/as9888_dev 0777 system system
```

Add following lines to “init.rc” to set permission for attributes of sysfs.

```
$vi “$ANDROID/out/target/product/<TARGET_PRODUCT>/root/init.rc”
#Add following line
    chown system system /sys/class/compass/as9888/enable_acc
    chown system system /sys/class/compass/as9888/enable_mag
    chown system system /sys/class/compass/as9888/enable_ori
    chown system system /sys/class/compass/as9888/delay_acc
    chown system system /sys/class/compass/as9888/delay_mag
    chown system system /sys/class/compass/as9888/delay_ori
```

2.7 Register the Sensor Control Program as a Service

To start AS9888 automatically on boot-up, init.rc should be modifyied according to the Android specification.

```
$vi “$ANDROID/out/target/product/<TARGET_PRODUCT>/root/init.rc”
#Add following line
    service dmtd /system/bin/dmtd –m 1
    class main
    user system
    group system
```

2.8 Rebuild the Root File System

3.Initialization

Generate as9888.img file (only the first time)

Daemon program dmtd needs the image file as9888.img to start up. If this file is absent, daemon program will quit immediately. This file contains the value, which is calculated from EEPROM value and current posture. Therefore, this file has to be created by program at first. To create this file, launch dmtd program with “-f” option. When this option is specified, FST is executed and generate as9888.img file. Please notice that FST (specified with -f) should be executed while the device is keeping horizontal and stable (i.e. don’t move the device).

```
# dmtd -m 1 -s -f
```

The image file will be saved as /data/misc/as9888.img.