

SenseTek Driver Integrating Notes

Writer: *Patrick Chang*

CONFIDENTIAL

Contents

| | |
|--------------------------------------------------------------|----|
| Contents | 2 |
| Change Log | 3 |
| Linux Kernel Driver Integrating Notes | 5 |
| Android 2.0~2.2 Sensor HAL Integrating Notes | 8 |
| Android 2.3~3.0 Sensor HAL Integrating Notes | 14 |
| Modifications of PowerManagerService (For Reference) | 15 |
| SenseTek Ambient Light Sensor Performance Tuning Notes | 19 |

CONFIDENTIAL

Change Log

| Date | Version | Change log | Sponsor | Remark |
|------------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|--------------------------------------------|
| 8/15/2010 | 1.0 | Initial Release | Patrick Chang | |
| 10/13/2010 | 1.2 | Add G-Sensor Driver | Patrick Chang | Only for Eng Test |
| 12/1/2010 | 1.3 | Added both polling mode and interrupt mode support. Added "Transmittance" for lux-reading calibration. | Patrick Chang | |
| 1/15/2011 | 1.4 | Added STK2107 support. | Patrick Chang | |
| 3/5/2011 | 1.5 | Move driver path to /drivers/i2c/SenseTek. Change driver mode from devfs to sysfs and added input event support. Added STK31xx proximity sensors support. Added several tunable parameters in ENG/DBG mode. (Mount on /sys/devices/platform/stk-xxxx/DBG) Use input event mechanism | Patrick Chang | |
| 3/22/2011 | 1.51 | Modify STK2201 driver: set default R-set to 500KOhm. Add "delay/min delay" for Android | Patrick Chang | |
| 3/30/2011 | 1.55 | Fix STK3128/3101 endianness issue Add support for Android 2.3~3.0 | Patrick Chang | It had NOT been verified under Android 3.0 |
| 4/7/2011 | 1.56 | Fix HAL compatibility to support I-company 3 in 1 drivers Remove extra engineering initializing setting (STK3101/3128) | Patrick Chang | |
| 5/12/2011 | 1.57 | Add driver version support Change binary mode sysfs to text mode sysfs to enable sensor | Patrick Chang | |
| 5/23/2011 | 1.58 | Driver: Support Linux kernel 2.6.35 and above. HAL: Add support Yamaha's sensor | Patrick Chang | |
| 6/23/2011 | 1.59 | Add Proximity Test AP (FakePhone) Driver : fixed typo (Kconfig) HAL : modify makefile naming | Patrick Chang | |

| | | | | |
|------------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|----------------------------------------------------------|
| 7/5/2011 | 1.60 | Add stk220x driver with interrupt | Lex Hsieh | |
| 8/26/2011 | 1.63 | Sync. Stk31xx PS detecting direction with ic | Patrick Chang | |
| 11/9/2011 | 1.64 | Remove instant event report when system enables proximity sensor(STK31xx) Set attribute to 0644 for release mode (don't forget to add "chown system system /sys/device/platform/stk-oss/xxxx" in init.rc) Add referenced modifications of "PowerManagerServer.java" | Patrick Chang | |
| 11/25/2011 | 1.65 | Fixed "ps_code_low_thd" bug Report als data once als is enabled for stk31xx int mode sleep 500 ms of integral time for stk220x_int.c Added check if ALS_gain is larger than 2 times for stk31xx Removed ReadMe in driver. Just keep the pdf document updated Update HAL for error handling | Lex Hsieh | Driver / HAL were compiled under Android 4.0 environment |
| 12/02/2011 | 1.66 | Add wakelock to ensure that system would complete the PS event | Patrick Chang | |

Linux Kernel Driver Integrating Notes

1. Extra SenseTek.tar.gz to /kernel/drivers/i2c/SenseTek
2. Modify /kernel/drivers/i2c/Kconfig (Add **source "drivers/i2c/SenseTek/Kconfig"** at bottom)

```
config I2C_DEBUG_ALGO
    bool "I2C Algorithm debugging messages"
    help
        Say Y here if you want the I2C algorithm drivers to produce a bunch
        of debug messages to the system log. Select this if you are having
        a problem with I2C support and want to see more of what is going
        on.

config I2C_DEBUG_BUS
    bool "I2C Bus debugging messages"
    help
        Say Y here if you want the I2C bus drivers to produce a bunch of
        debug messages to the system log. Select this if you are having
        a problem with I2C support and want to see more of what is going
        on.

config I2C_DEBUG_CHIP
    bool "I2C Chip debugging messages"
    help
        Say Y here if you want the I2C chip drivers to produce a bunch of
        debug messages to the system log. Select this if you are having
        a problem with I2C support and want to see more of what is going
        on.

source "drivers/i2c/SenseTek/Kconfig"
endif # I2C
```

3. Modify /kernel/driver/i2c/Makefile (Add **obj-\$(CONFIG_SENSETEK_I2C_SENSORS) += SenseTek/** at bottom)

```
#
# Makefile for the i2c core.
#

obj-$(CONFIG_I2C_BOARDINFO)      += i2c-boardinfo.o
obj-$(CONFIG_I2C)                += i2c-core.o
obj-$(CONFIG_I2C_CHARDEV)        += i2c-dev.o
obj-y                             += busses/ chips/ algos/
obj-$(CONFIG_SENSETEK_I2C_SENSORS) += SenseTek/

ifeq ($(CONFIG_I2C_DEBUG_CORE),y)
EXTRA_CFLAGS += -DDEBUG
endif
```

4. Add the following code for your machine setting. (/kernel/arch/arm/mach-xxxx/xxxxxx.c)

```
/* Added for STK ALS & PS */
#ifdef CONFIG_STK_ALS_GENERIC
    #ifdef CONFIG_STK_ALS_SAHS_FLOAT
        { I2C_BOARD_INFO("stk_als", 0x20>>1), .irq = -1 },
    #endif
    #ifdef CONFIG_STK_ALS_SAHS_PULL_LOW
        { I2C_BOARD_INFO("stk_als", 0x70>>1), .irq = -1 },
    #endif
    #ifdef CONFIG_STK_ALS_SAHS_PULL_HIGH
        { I2C_BOARD_INFO("stk_als", 0x90>>1), .irq = -1 },
    #endif
    #ifdef CONFIG_STK_ALS_AUTO_DETECT
        { I2C_BOARD_INFO("stk_als", 0x20>>1), .irq = -1 },
        { I2C_BOARD_INFO("stk_als", 0x70>>1), .irq = -1 },
        { I2C_BOARD_INFO("stk_als", 0x90>>1), .irq = -1 },
    #endif
#endif
#ifdef CONFIG_STK_ALS_220X_INT_MODE
    { I2C_BOARD_INFO("stk_als", 0x20>>1), .irq = -1 },
#endif
#ifdef CONFIG_STK_ALS22x7
```

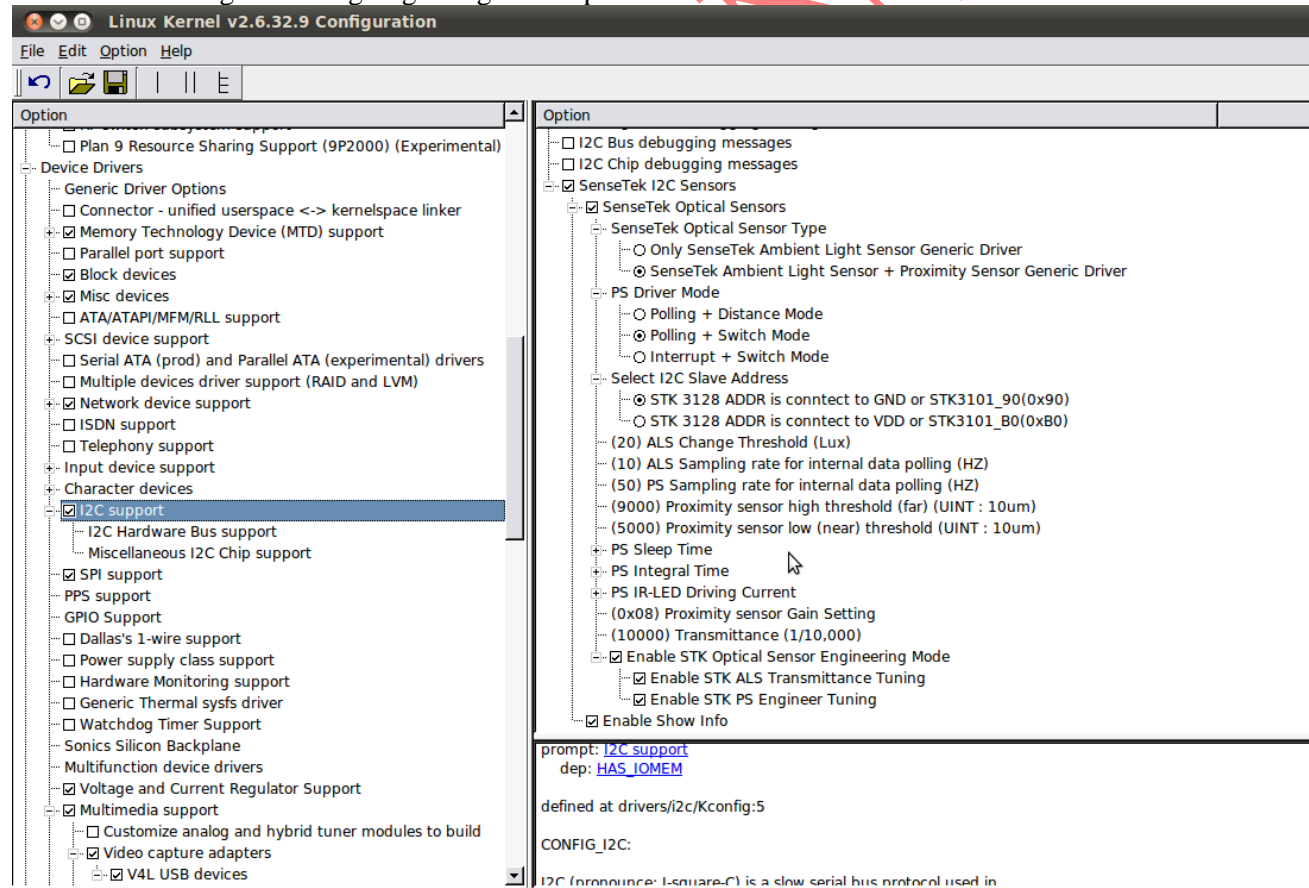
```

        { I2C_BOARD_INFO("stk_als22x7_addr1", 0x20>>1), },
        { I2C_BOARD_INFO("stk_als22x7_addr2", 0x22>>1), },
    #endif
#ifdef CONFIG_STK_PS_0x90
    { I2C_BOARD_INFO("stk_ps", 0x90>>1), .irq = -1},
#endif
#ifdef CONFIG_STK_PS_0xB0
    { I2C_BOARD_INFO("stk_ps", 0xB0>>1), .irq = -1},
#endif
};

static struct i2c_board_info devkit8000_i2c_boardinfo_ext[] __initdata = {
    /* Added for STK ALS & PS */
#ifdef CONFIG_STK_ALS_GENERIC
    { I2C_BOARD_INFO("stk_als", 0x20>>1), .irq = -1 },
#endif
#ifdef CONFIG_STK_ALS_SAHLS_FLOAT
    { I2C_BOARD_INFO("stk_als", 0x70>>1), .irq = -1},
#endif
#ifdef CONFIG_STK_ALS_SAHLS_PULL_LOW
    { I2C_BOARD_INFO("stk_als", 0x70>>1), .irq = -1},
#endif
#ifdef CONFIG_STK_ALS_SAHLS_PULL_HIGH
    { I2C_BOARD_INFO("stk_als", 0x90>>1), .irq = -1},
#endif
#ifdef CONFIG_STK_ALS_AUTO_DETECT
    { I2C_BOARD_INFO("stk_als", 0x20>>1), .irq = -1},
    { I2C_BOARD_INFO("stk_als", 0x70>>1), .irq = -1},
    { I2C_BOARD_INFO("stk_als", 0x90>>1), .irq = -1},
#endif
#ifdef CONFIG_STK_ALS22x7
    { I2C_BOARD_INFO("stk_als22x7_addr1", 0x20>>1), },
    { I2C_BOARD_INFO("stk_als22x7_addr2", 0x22>>1), },
#endif
#ifdef CONFIG_STK_PS_0x90
    { I2C_BOARD_INFO("stk_ps", 0x90>>1), .irq = -1},
#endif
#ifdef CONFIG_STK_PS_0xB0
    { I2C_BOARD_INFO("stk_ps", 0xB0>>1), .irq = -1},
#endif
};

```

4. Use menuconfig or xconfig or gconfig to set up.



- Enable "SenseTek Optical Sensors" if it is on your board.
- Choose "SenseTek Ambient Light Sensor +Proximity Sensor" if you use STK PS, otherwise, choose "ONLY ALS".
- Well choose "PS Driver Mode" and "I2C Slave Address" for your configuration if you are using PS.
- Well choose "Sensor Model" for your system configuration if you are using "Only ALS".

- e. Enable “STK Optical Sensor Engineering Mode” for engineer testing/tuning. (See SenseTek Proximity/Ambient Light Sensor Performance Tuning Notes)
- f. Enable “Show Info” for debug information.
- g. For Proximity Sensor, Polling + Switch mode is suggested.
 - I. Polling + Distance Mode → Driver will report “distance data” to input event system. You also need to enter the “Distance” threshold.
 - II. Polling + Switch Mode → Driver will only report 0 (near) or 1 (far) to input event system. You also need to enter the “Distance” threshold.
 - III. Interrupt + Switch Mode → Driver will only report 0 (near) or 1 (far) to input event system. (Driver will use INT-pin to handle) You also need to enter the “PS Code” threshold.
- h. If you use stk220x light sensor, select interrupt mode or polling mode
- i. If you use “Interrupt” mode, you should modify the driver for additional GPIO configuration

```
#define ADDITIONAL_GPIO_CFG 1

/* // Additional GPIO CFG Header */
#if ADDITIONAL_GPIO_CFG
#include <mach/gpio.h>
#endif

in stk_ps_probe()
#if ADDITIONAL_GPIO_CFG // Additional GPIO CFG
    gpio_request(136, "EINT");        /* Request for gpio pin */
    gpio_direction_input(136);
#endif // Additional GPIO CFG
```

*** DON'T forget to add "chown system system /sys/device/platform/stk-oss" in init.rc for "Release" ***

Android 2.0~2.2 Sensor HAL Integrating Notes

SenseTek provides both pre-compiled binary HAL and full source. Please follow the instruction below .

1. Use pre-compiled HAL (ARMv7 compatible) → Directly copy sensors.stk_ef(als+ps).so or sensors.stk_ef(only als).so to /system/lib/hw and rename it according your system setting.
2. Use source and rebuild the whole system
 - a. Copy /libsensors/ to /android source/hardware/xxx/libsensors
 - b. Well modify “Android.mk” according your system and rebuild the system.
 - c. Modify sensors_stk_ef.c if it is necessary.
 - i. Enable “Sensor”

```
/* Enable / Disable Sensor
 * You Must Modify Here
 */
#define AS 0 /*Accelerometer*/
#define MS 0 /*Magnetic Sensor*/
#define OS 0 /*Orientation Sensor*/
#define TS 0 /*Temperature Sensor*/
#define PS 1 /*Proximity Sensor*/
#define LS 1 /*Light Sensor*/
#define GS 0 /*Gyroscope*/
#define SS 0 /*Pressure Sensor, Stress Sensor*/
```

- ii. Enable FD

```
/* Modify here if you want to add another FDs*/
#define AS_FD 0
#define MS_FD 0
#define OS_FD 0
#define TS_FD 0
/* if TS/OS/MS/AS is a group --> only set TS_FD = 1*/
#define PS_FD 1
#define LS_FD 1
/* if LS/PS is a group --> only set LS_FD = 1*/
#define GS_FD 0
#define SS_FD 0
```

- iii. Modify sensors' information if it is necessary

```
static struct sensor_t sSensorList[MAX_NUM_SENSORS] = {
    #if AS
    {
        .name = "SenseTek Accelerometer",
        .vendor = "SenseTek/Sitronix",
        .version = 1,
        .handle = ID_A,
        .type = SENSOR_TYPE_ACCELEROMETER,
        .maxRange = 8.0f,
        .resolution = 8.0f/2048.0f,
        .power = 0.1f,
        .reserved = {}
    },
    #endif
    #if MS
    {
        .name = "SenseTek M-Sensor",
        .vendor = "SenseTek/Sitronix",
        .version = 1,
        .handle = ID_M,
        .type = SENSOR_TYPE_MAGNETIC_FIELD,
        .maxRange = 2000.0f,
        .resolution = 1.0f,
        .power = 0.1f,
        .reserved = {}
    },
    #endif
    #if OS
    {
        .name = "SenseTek Orientation Sensor",
        .vendor = "SenseTek/Sitronix",
        .version = 1,
        .handle = ID_O,
        .type = SENSOR_TYPE_ORIENTATION,
        .maxRange = 360.0f,
        .resolution = 1.0f,
        .power = 0.0f,
        .reserved = {}
    },
    #endif
    #if TS
    {
        .name = "SenseTek Temperature Sensor",
        .vendor = "SenseTek/Sitronix",

```

- iv. Implements “open data method”


```
/* this must return a file descriptor that will be used to read
 * the sensors data (it is passed to data_data_open() below
 */
static native_handle_t*
control__open_data_source(struct sensors_control_device_t *dev)
{
    sensors_control_context_t*  ctl = (void*)dev;
    native_handle_t* handle;
    int als_fd;
#ifdef PS_FD
    int ps_fd;
#endif
    // TODO
    /* open sensor device */
    if (open_inputs(O_RDONLY,&als_fd,ALS_NAME)<0)
    {
        D("Open ALS INPUT Fail");
        return NULL;
    }
#ifdef PS_FD
    if (open_inputs(O_RDONLY,&ps_fd,PS_NAME)<0)
    {
        D("Open PS INPUT Fail");
        return NULL;
    }
#endif
    handle = native_handle_create(MAX_LINUX_NUM_FDS, 0);
    handle->data[ID_LS_FD] = als_fd;
#ifdef PS_FD
    handle->data[ID_PS_FD] = ps_fd;
#endif

    return handle;
}
```

v. Implements “active method”

```
/* To Do : Modify Here*/
ctl->active_sensors = new_sensors;
if (changed == 0)
    return 0;

switch(handle)
{
#if AS
    case ID_A:
        break;
#endif
#if MS
    case ID_M:
        break;
#endif
#if OS
    case ID_O:
        break;
#endif
#if TS
    case ID_T:
        break;
#endif
#if PS
    case ID_P:
        fd = open(ps_enable_path,O_RDWR);
        if (fd>=0)
        {
            bEnable = enabled?1:0;
            ret = write(fd,&bEnable,1);
            close(fd);
        }
        break;
#endif
#if LS
    case ID_L:
        fd = open(als_enable_path,O_RDWR);
        if (fd>=0)
        {
            bEnable = enabled?1:0;
            ret = write(fd,&bEnable,1);
            close(fd);
        }
        break;
#endif
}
```

vi. Implements “set delay method”

```
static int
control__set_delay(struct sensors_control_device_t *dev, int32_t ms)
{
    sensors_control_context_t*  ctl = (sensors_control_context_t*)dev;
    int fd;

    #if PS
        fd = open(ps_delay_path,O_WRONLY);
        if (fd)
        {
            write(fd,&ms,sizeof(int32_t));
            close(fd);
        }
    #endif

    #if LS
        fd = open(als_delay_path,O_WRONLY);
        if (fd)
        {
            write(fd,&ms,sizeof(int32_t));
            close(fd);
        }
    #endif

    if (ms!=0)
        nDelayMicroSecond = ms*1000;
    else
        nDelayMicroSecond = 2000;
    return 0;
}
```

vii. Implements “poll method”

```

/* read the next event; first, read the ambient light event, then the
   proximity event */
struct input_event event;
int got_syn = 0;
int exit = 0;
int nread;
fd_set rfd_set;
int n;
FD_ZERO(&rfd_set);
FD_SET(als_fd, &rfd_set);

#if PS
FD_SET(ps_fd, &rfd_set);
// timeout.tv_usec = nDelayMicroSecond;
n = select( __MAX(als_fd,ps_fd)+ 1, &rfd_set, NULL, NULL, NULL);
//n = select( __MAX(als_fd,ps_fd)+ 1, &rfd_set, NULL, NULL, &timeout);
#else
n = select(als_fd+ 1, &rfd_set, NULL, NULL, NULL);
#endif

if (FD_ISSET(als_fd, &rfd_set))
{
    nread = read(als_fd, &event, sizeof(event));
    INFO("read(als_fd, &event, sizeof(event))=%d) = %d",nread,sizeof(event));
    if (nread == sizeof(event))
    {
        new_sensors |= data__poll_process_light(dev, als_fd, &event);
        got_syn = event.type == EV_SYN;
        exit = got_syn && event.code == SYN_CONFIG;
        if (got_syn)
        {
            INFO("ps syn %08x", new_sensors);
            data__poll_process_syn(dev, &event, new_sensors);
            new_sensors = 0;
        }
    }
    else E("als read too small %d", nread);
}

#if PS
if (FD_ISSET(ps_fd, &rfd_set))
{
    nread = read(ps_fd, &event, sizeof(event));
    INFO("read(ps_fd, &event, sizeof(event))=%d) = %d",nread,sizeof(event));
    if (nread == sizeof(event))
    {
        new_sensors |= data__poll_process_proximity(dev, ps_fd, &event);

        got_syn = event.type == EV_SYN;
        exit = got_syn && event.code == SYN_CONFIG;
        if (got_syn)
        {
            INFO("ps syn %08x", new_sensors);
            data__poll_process_syn(dev, &event, new_sensors);
            new_sensors = 0;
        }
    }
    else E("ps read too small %d", nread);
}
}

```

if you have >2 FDs (input event handle), you should modify to
 n = select(MAX3(fd1,fd2,fd3)+1,&rfd_set,NULL,NULL,NULL);
 [or MAX4(fd1,fd2,fd3,fd4)+1]
 if the other devices(sensors) do NOT support input event, please use “timeout” to check event
 for other devices.

After you completed the HAL, please also add “Auto-brightness Lookup Table”

1. Use overlay ➡ for your device setting, modify config.xml
2. Or modify source tree directly ➡ /android source/ frameworks/base/core/res/res/values/config.xml

Android 2.3~3.0 Sensor HAL Integrating Notes

SenseTek provides both pre-compiled binary HAL and full source. Please follow the instruction below .

1. Use pre-compiled HAL (ARMv7 compatible) → Directly copy sensors.stk_gh(als+ps).so or sensors.stk_gh(only als).so to /system/lib/hw and rename it according your system setting.
2. Use source and rebuild the whole system
 - a. Copy /libsensors/ to /android source/hardware/xxx/libsensors
 - b. Well modify “Android.mk” according your system and rebuild the system.
 - c. Modify enabled_sensors.h to enable your sensors

```
10  #define ENABLED_SENSORS_H
11
12  /* if you are using AKM 3 in 1 sensor, set AKM_3IN1_SENSOR = 1 */
13  #define AKM_3IN1_SENSOR 0
14  #define INVENSENSE_3IN1_SENSOR 0
15
16  #if (INVENSENSE_3IN1_SENSOR&&AKM_3IN1_SENSOR)
17  #error This HAL does NOT support AKM_3IN1_SENSOR and INVENSENSE_3IN1_SENSOR at the same time
18  #endif
19
20  #define AS 0    /*Accelerometer*/
21  #define MS 0    /*Magnetic Sensor*/
22  #define OS 0    /*Orientation Sensor*/
23  #define TS 0    /*Temperature Sensor*/
24  #define PS 1    /*Proximity Sensor*/
25  #define LS 1    /*Light Sensor*/
26  #define GS 0    /*Gyroscope*/
27  #define PR 0    /*Pressure Sensor*/
28  #define GR 0    /*Gravity*/
29  #define LA 0    /*Linear Acceleration*/
30  #define RV 0    /*Rotation Vector*/
```

If you are using AKM 3 in 1 sensor [e-Compass (hw)+Accelerometer (hw)+Orientation(sw)], please set AKM_3IN1_SENSOR = 1

If you are using Invensense 3 in 1 sensor [Gyroscope(hw) + Accelerometer(hw) + e-Compass(hw) + Orientation(sw)+Gravity(sw)+Linear Acceleration(sw)+Rotation Vector(sw)], please set INVENSENSE_3IN1_SENSOR = 1

3. Modify some codes if it is necessary.

After you completed the HAL, please also add “Auto-brightness Lookup Table”. Please reference to “Android 2.0~2.2 Sensor HAL Integrating Notes”.

Besides, you might need some utilities to check that sensor HAL works well. You could use SenseTekSensorDisplay.apk to archive this purpose. Add “SenseTekSensorDisplay” to “PRODUCT_PACKAGES” and add

“device/SenseTek/board_example/SenseTekSensorDisplay.apk:system/app/SenseTekSensorDisplay.apk” to “PRODUCT_COPY_FILES”. Where “device/SenseTek/board_example/SenseTekSensorDisplay.apk” is the path of this APK utilities.

Modifications of PowerManagerService (For Reference)

SenseTek provides modifications of PowerManagerService for customer reference. Customers could use these files directly or use "diff" tool to analyze the auto-brightness-control behavior.

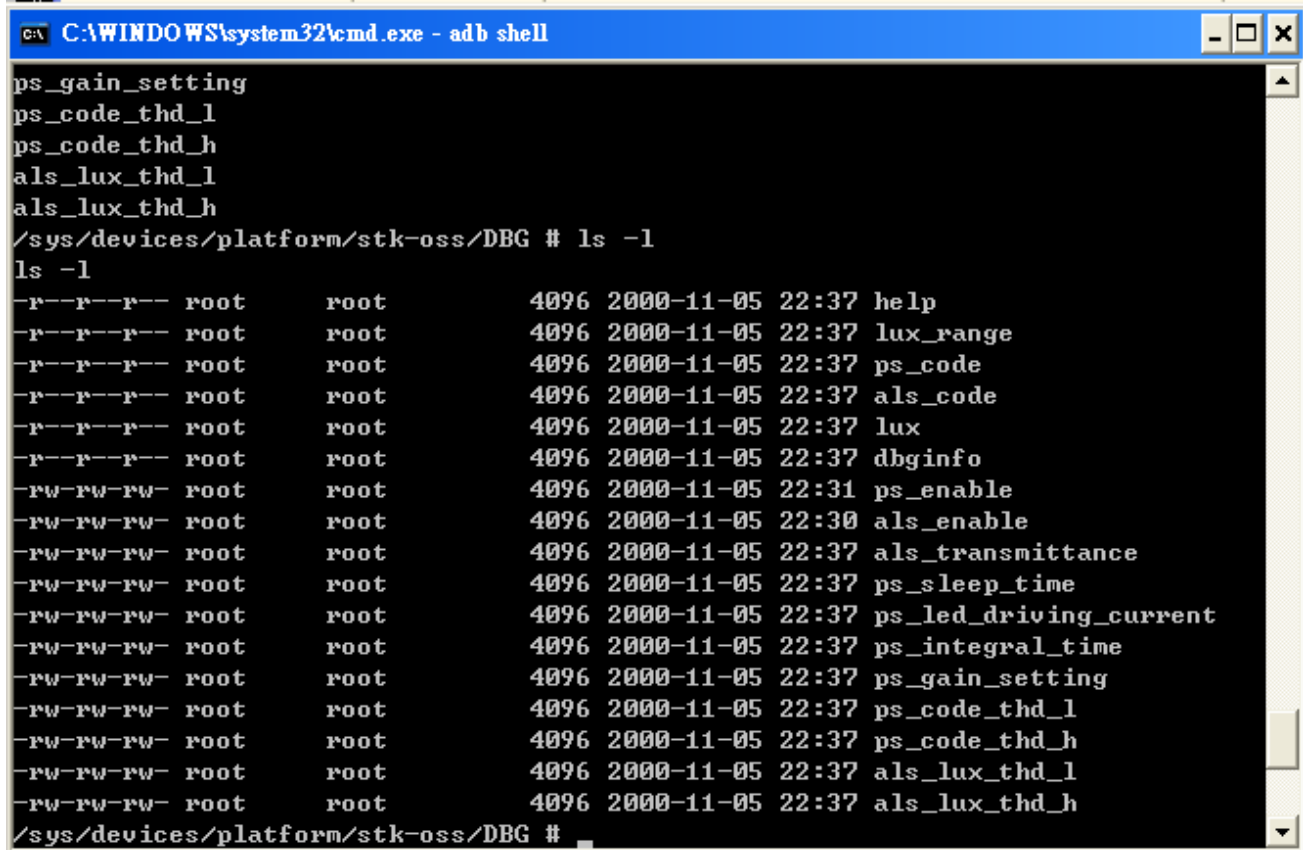
Note that if light sensor reading is correct but back light can't change, maybe this is because of wrong "mIsDocked" parameter setting. Please refer to "PowerManagerService.mod1.java".

| Filename | Feature |
|-------------------------------|------------------------------------------------------------------------------------------------|
| PowerManagerService.org.java | Original version of AOSP 2.3.4 |
| PowerManagerService.mod1.java | Remove "Docking" check |
| PowerManagerService.mod2.java | Optimize for the more bright environment and avoid to reduplicate calculating brightness level |

CONFIDENTIAL

SenseTek Proximity Sensor Performance Tuning Notes

1. Enable “STK Optical Sensor Engineering Mode” before using this feature.
2. Use “adb shell” or other tty console (ex. UART) to enter your system.



```

C:\WINDOWS\system32\cmd.exe - adb shell
ps_gain_setting
ps_code_thd_l
ps_code_thd_h
als_lux_thd_l
als_lux_thd_h
/sys/devices/platform/stk-oss/DBG # ls -l
ls -l
-r--r--r-- root    root      4096 2000-11-05 22:37 help
-r--r--r-- root    root      4096 2000-11-05 22:37 lux_range
-r--r--r-- root    root      4096 2000-11-05 22:37 ps_code
-r--r--r-- root    root      4096 2000-11-05 22:37 als_code
-r--r--r-- root    root      4096 2000-11-05 22:37 lux
-r--r--r-- root    root      4096 2000-11-05 22:37 dbginfo
-rw-rw-rw- root    root      4096 2000-11-05 22:31 ps_enable
-rw-rw-rw- root    root      4096 2000-11-05 22:30 als_enable
-rw-rw-rw- root    root      4096 2000-11-05 22:37 als_transmittance
-rw-rw-rw- root    root      4096 2000-11-05 22:37 ps_sleep_time
-rw-rw-rw- root    root      4096 2000-11-05 22:37 ps_led_driving_current
-rw-rw-rw- root    root      4096 2000-11-05 22:37 ps_integral_time
-rw-rw-rw- root    root      4096 2000-11-05 22:37 ps_gain_setting
-rw-rw-rw- root    root      4096 2000-11-05 22:37 ps_code_thd_l
-rw-rw-rw- root    root      4096 2000-11-05 22:37 ps_code_thd_h
-rw-rw-rw- root    root      4096 2000-11-05 22:37 als_lux_thd_l
-rw-rw-rw- root    root      4096 2000-11-05 22:37 als_lux_thd_h
/sys/devices/platform/stk-oss/DBG #
  
```

3. Ensure that “ps_enable” = 1 (if not, type echo 1 > ps_enable)
4. Modify ps_gain_setting / ps_integral_time / ps_led_driving_current / ps_sleep_time and threshold for your requirement.

4.1 ps_gain_setting → gain setting (Reserved, use default setting “8”)

Binary Format : G₀₃G₀₂G₀₁G₀₀

G₀₃ and G₀₂ are a group ; G₀₁ and G₀₀ are another group

| G ₀₃ | G ₀₂ | Gain A | G ₀₁ | G ₀₀ | Gain B |
|-----------------|-----------------|---------------|-----------------|-----------------|---------------|
| 0 | 0 | x 1 | 0 | 0 | x 1 (default) |
| 0 | 1 | x 2 | 0 | 1 | x 2 |
| 1 | 0 | x 4 (default) | 1 | 0 | x 4 |
| 1 | 1 | x 8 | 1 | 1 | x 8 |

Total Gain = Gain A x Gain B (Max x64, Min x1)

ps_gain_setting is 8(X4) by default, if one wants longer detection range, one may set ps_gain_setting as 0x9(X8, echo 9 > ps_gain_setting) -> 0xa(X16, echo a > ps_gain_setting) -> 0xb(X32, echo b > ps_gain_setting).....etc.

4.2 ps_sleep_time → sleep time (0~3, bigger for more power saving, but slower response)

| ps_sleep_time | Sleep time (ms) |
|---------------|-----------------|
| 0 | 10 |
| 1 | 30 (default) |
| 2 | 90 |
| 3 | 270 |

4.3 ps_integral_time → integral time and LED on duration (0~3, bigger to get larger signal, but it might be easier to saturate and more power consumption)

| ps_integral_time | Integral time (ms) |
|------------------|--------------------|
| 0 | 0.2 (default) |
| 1 | 0.3 |
| 2 | 0.4 |
| 3 | 0.5 |

4.4 ps_led_driving_current → driving current (0~1)

| ps_led_driving_current | Driving current(mA) |
|------------------------|---------------------|
| 0 | 100 (default) |
| 1 | 200 |

5. Fine tune PS parameters according to your specification. For example, if the specification is

| Distance (gray card) | state |
|----------------------|---------------------|
| <= 3 cm | near |
| 3~5 cm | Keep previous state |
| >= 5 cm | far |

Please assemble the overall phone with cover glass before tuning so that the tuning procedure is meaningful.

5.1 If PS Interrupt + Switch Mode or Polling + Switch Mode was selected in meunconfig, user have to set ps_code_thd_h and ps_code_thd_l to determine how “far” and “near” state is triggered.

Please place a grey card (recommended) that is 3 cm apart from the cover glass of phone, and issue the command “cat ps_code” to get the PS reading, assume this value is ps_code_a.

Please place a grey card that is 5 cm apart from the cover glass of phone, and issue the command “cat ps_code” to get the PS reading, assume this value is ps_code_b.

Please place phone in a clear place that no obstacle will affect PS reading, and issue the command “cat ps_code” to get the PS reading, assume this value is ps_code_c.

- ps_code_c is influenced by cross talk, typically it is smaller than 50 for 2-in-1 sensor, if the ps_code_c is too large, please refer to “SenseTek PS Mechanical Design Guide” or “SenseTek 3 in 1 PS Module Mechanical Design Guide” and review the mechanical design.
- ps_code_a and ps_code_b are influenced by PS parameters. Please tune ps_gain (first, larger ps_gain induces larger ps_code) -> ps_sleep_time (response speed) -> ps_integral_time (larger integral time induces slarger ps_code) -> ps_led_driving_current (last, consume more power, larger ps_led_driving_current induces larger ps_code).
- Make sure that the difference between ps_code_a and ps_code_b is large enough so that different phone can share the same parameter. Besides, please make the difference larger if ps_gain is larger.
- Make sure that the difference between ps_code_b and ps_code_c is large enough so that sensor can identify the difference between 5cm condition and no obstacle condition.
- Make sure that ps_code_a is not too large and cause ps_code_a overflow. ps_code_a is smaller than 255 due to the limitation of ADC.

If conditions above are met, please set ps_code_thd_h (echo ps_code_a > ps_code_thd_h) and ps_code_thd_l (echo ps_code_b > ps_code_thd_l) and then make sure that performance meets the specification.

5.2 If Polling + Distance Mode or Polling + Switch Mode was chosen, user have to set STK_PS_DISTANCE_HIGH_THRESHOLD, STK_PS_DISTANCE_HIGH_THRESHOLD, STK_PS_CHANGE_THRESHOLD_NEAR, STK_PS_CHANGE_THRESHOLD_MID, and STK_PS_CHANGE_THRESHOLD_FAR in menuconfig to determine when distance will be reported.

Please calibrate “stk_i2c_ps_lookup_table.h” to get accurate distance in advance.

6. Use “FakePhone” to verify the performance of parameter.

6.1 Install “FakePhone.apk” (cmd line : adb install FakePhone.apk)

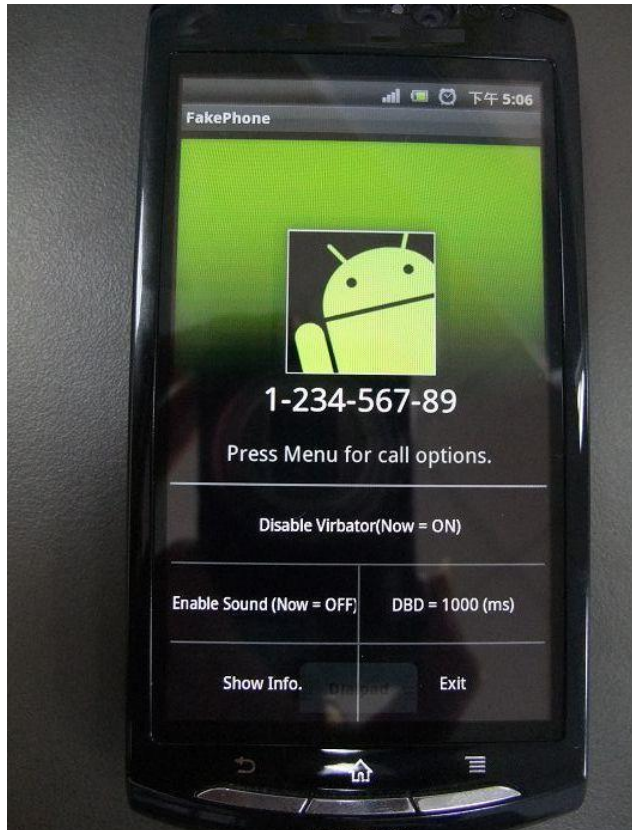
6.2 Launch FakePhone AP and press “Menu” key to ensure “Virbator = ON”.

6.3 Try to put object near or far to observe the behavior of phone.

6.4 Change DBD (Debounce Delay Time) for verification.

6.5 Use terminal to change STK PS parameter to fine tune performance.

Note : The behavior of this AP is almost equivalent to what PowerManagerService does. But it is easier to observe the exact timing (not only screen off) if we enable “Sound”/“Vibrator”.



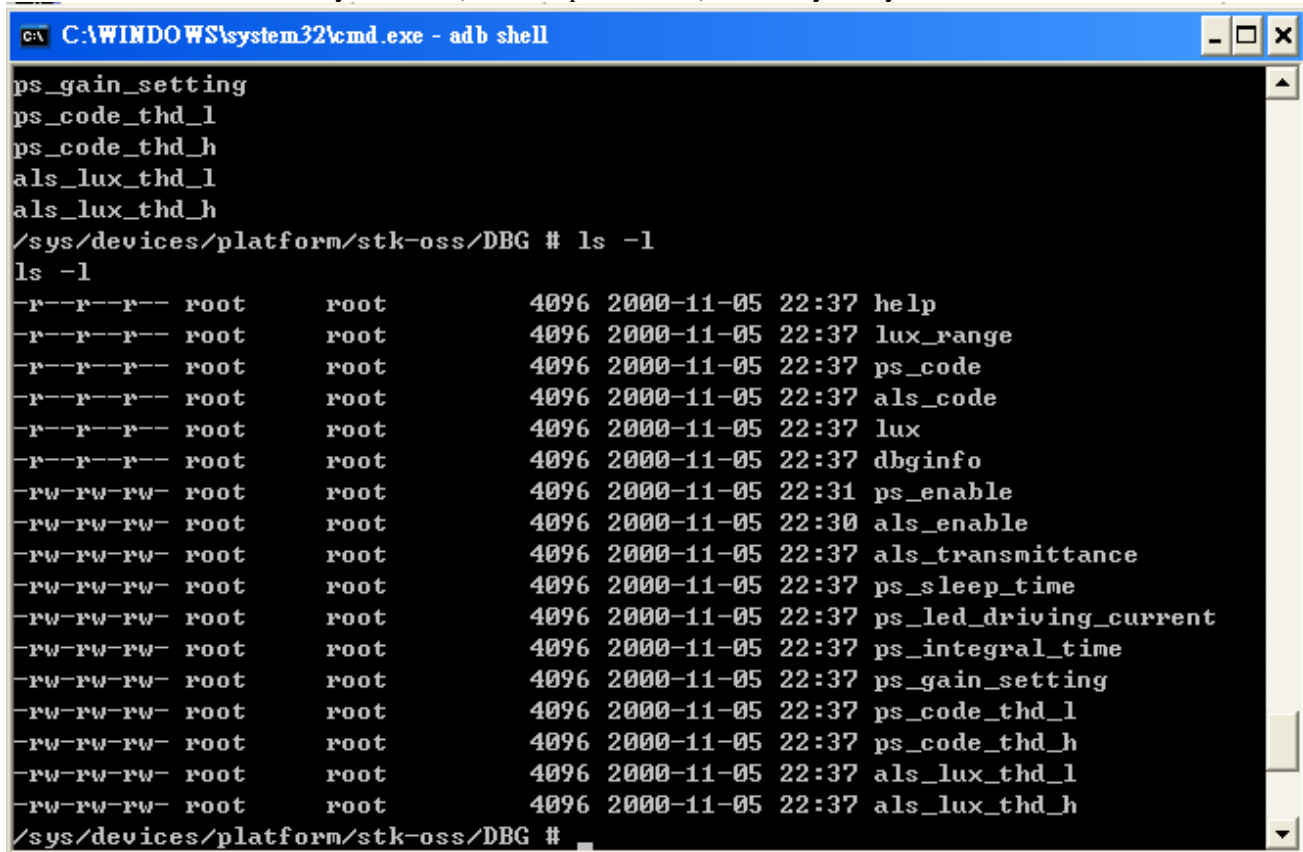
1. Press “Menu”



2. Screen off (and vibrate) if it is near

SenseTek Ambient Light Sensor Performance Tuning Notes

- 1.Enable “STK Optical Sensor Engineering Mode” in menuconfig before using this feature.
- 2.Use “adb shell” or other tty console (for example, UART) to enter your system.



```
C:\WINDOWS\system32\cmd.exe - adb shell

ps_gain_setting
ps_code_thd_l
ps_code_thd_h
als_lux_thd_l
als_lux_thd_h
/sys/devices/platform/stk-oss/DBG # ls -l
ls -l
-r--r--r-- root    root      4096 2000-11-05 22:37 help
-r--r--r-- root    root      4096 2000-11-05 22:37 lux_range
-r--r--r-- root    root      4096 2000-11-05 22:37 ps_code
-r--r--r-- root    root      4096 2000-11-05 22:37 als_code
-r--r--r-- root    root      4096 2000-11-05 22:37 lux
-r--r--r-- root    root      4096 2000-11-05 22:37 dbginfo
-rw-rw-rw- root    root      4096 2000-11-05 22:31 ps_enable
-rw-rw-rw- root    root      4096 2000-11-05 22:30 als_enable
-rw-rw-rw- root    root      4096 2000-11-05 22:37 als_transmittance
-rw-rw-rw- root    root      4096 2000-11-05 22:37 ps_sleep_time
-rw-rw-rw- root    root      4096 2000-11-05 22:37 ps_led_driving_current
-rw-rw-rw- root    root      4096 2000-11-05 22:37 ps_integral_time
-rw-rw-rw- root    root      4096 2000-11-05 22:37 ps_gain_setting
-rw-rw-rw- root    root      4096 2000-11-05 22:37 ps_code_thd_l
-rw-rw-rw- root    root      4096 2000-11-05 22:37 ps_code_thd_h
-rw-rw-rw- root    root      4096 2000-11-05 22:37 als_lux_thd_l
-rw-rw-rw- root    root      4096 2000-11-05 22:37 als_lux_thd_h
/sys/devices/platform/stk-oss/DBG #
```

- 3.Ensure that “als_enable” = 1 (if not, enter “echo 1 > als_enable”)
- 4.Enter “cat lux” to get illuminance reading and modify als_transmittance for calibration (echo 2000 > als_transmittance). For normal indoor condition, the lux value should be about 400. The range of als_transmittance is between 1(0.01%) and 10000(100%). If the reading is larger than 400, try to set larger als_transmittance (ex: echo 4000 > als_transmittance). If the reading is smaller than 400, try to set smaller transmittance (echo 1000 > als_transmittance).

Please note that “als_transmittance” value may not be the true transmittance of cover glass because “als_transmittance” may be influenced by R-set, als_gain, als_it settings.

5. For interrupt mode, ALS might use a lookup-table to trigger the input event (modify “stk_ps3lxx_lux_threshold_table.h” if need)

Notice that, if you are using STK2201, there is another variable “R-set”. SenseTek recommends you to use 500K-Ohm for general purpose. This driver is designed for 500K-Ohm, if you are not using 500K-Ohm, just modify the “als_transmittance” to calibrate the lux-reading. For example, if there is no “cover glass/window”, and the R-set is 100K-Ohm, you should set transmittance to 2000. (1/5 of 10000)

*** DON'T forget to add "chown system system /sys/device/platform/stk-oss" in init.rc for "Release" ***