SENSOR SYSTEM

Revision	Date	Description
V1.0	2016/06/30	Initial Draft

Archermind

2016/7/19

Contents

1. Introduction

Sensor is a common and very important device, it is the feeling of a predetermined amount of the various measured according to certain rules to convert it to a device or apparatus useful signal.This paper describes x20 on sensor, supports many kinds of sensor, how to develop a sensor, introduces the kernel layer.

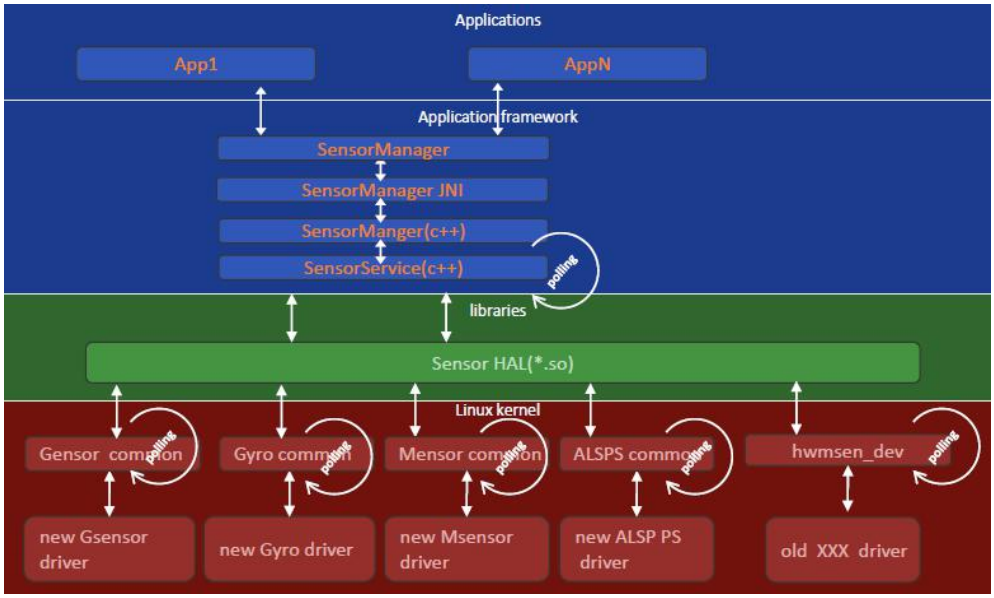
2. Sensor Architecture Description

2.1. Android sensor support types

Now Android support sensor types as follow

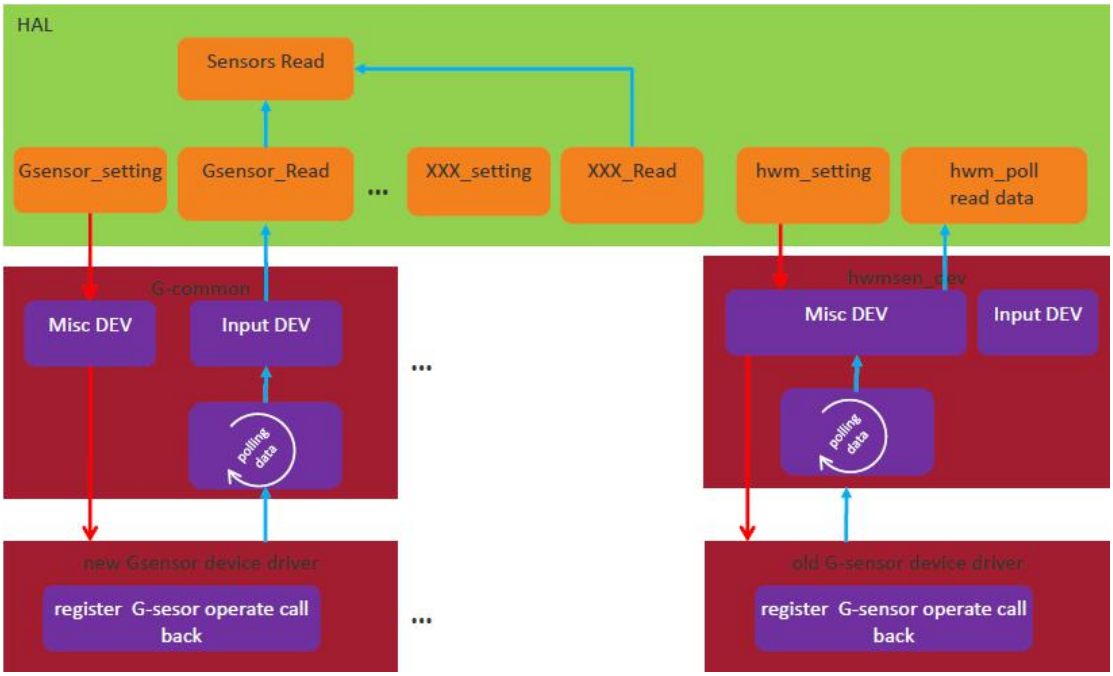
Sensor types		
TYPE_ACCELEROMETER	TYPE_GRAVITY	TYPE_SIGNIFICANT_MOTION
TYPE_MAGNETIC_FIELD	TYPE_LINEAR_ACCELERATION	TYPE_STEP_DETECTOR
TYPE_ORIENTATION	TYPE_ROTATION_VECTOR	TYPE_STEP_COUNTER
TYPE_GYROSCOPE	TYPE_RELATIVE_HUMIDITY	TYPE_GEOMAGNETIC_ROTATION_VECTOR
TYPE_LIGHT	TYPE_AMBIENT_TEMPERATURE	TYPE_HEART_RATE
TYPE_PRESSURE	TYPE_MAGNETIC_FIELD_UNC ALIBRATED	TYPE_TILT_DETECTOR
TYPE_TEMPERATURE	TYPE_GAME_ROTATION_VECT OR	TYPE_TILT_DETECTOR
TYPE_PROXIMITY	TYPE_GAME_ROTATION_VECT OR	TYPE_GLANCE_GESTURE

2.2. Sensor system Architecture



2.3. HAL & Driver Work Flow

data flow ——→ control flow ——→



2.4. Config File

android:

device\${COMPANY}\${PROJECT}\ProjectConfig.mk

kernel:

kernel-3.xx\arch\armxx\configs\$(proj)_defconfig and $(proj)_debug_defconfig

kernel-3.xx\arch\armxx\boot\dts\$(proj).dts

Init.rc：

alps\device\mediatek\${PLATFORM}\init.xxx.rc

3. Sensor Customization

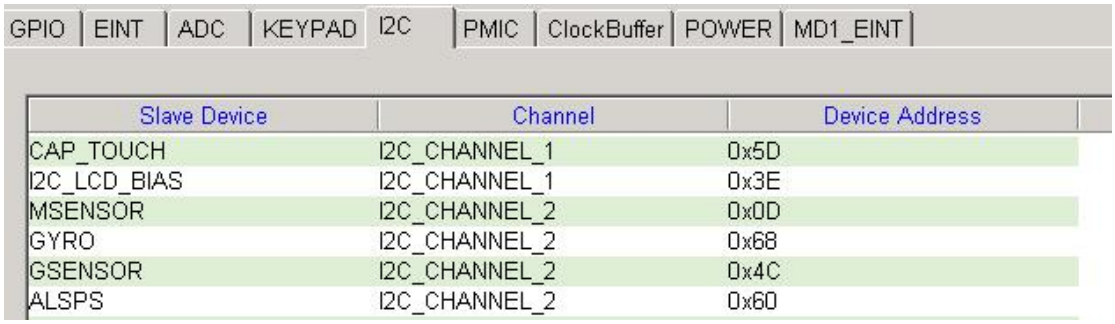
3.1. HAL Customization

No,mtk is no open,now.

4. Driver Customization

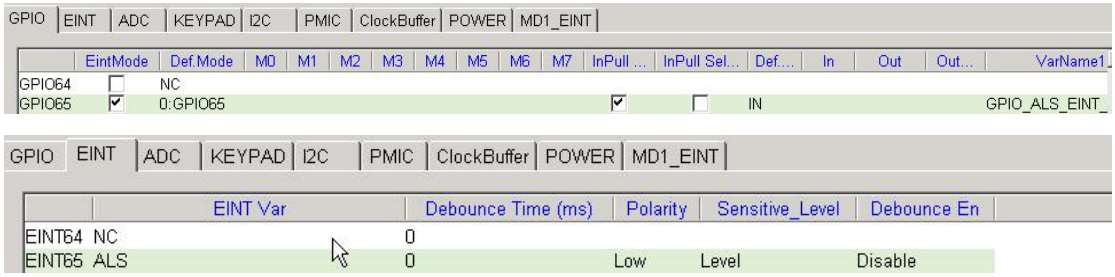
4.1. Config codegen.dws

Config the codegen.dws with DCT tool:
(1)Run the drvgen.exe and open the codegen.dws file
• Drvgen.exe path:
– alps\kernel-3.18\tools\dct\DrvGen.exe
• amt6797_64_open.dws path:
alps\kernel-3.18\drivers\misc\mediatek\dws\$(platform)\$(proj)\amt6797_64_open.dws
(2)config i2c bus and address



(3)exit and save the amt6797_64_open.dws
(4) Change the name to codegen.dws and copy to the follow path:
lk:
alps\vendor\mediatek\proprietary\bootable\bootloader\lk\target\$(proj)\dct\dct\codegen.dws

(5)If the sensor have used EINT like alsps, it must config gpio and eint.
if the sensor do not use EINT, please ignore this step.



(6)exit and save the codegen.dws
(7) Change the name to codegen.dws and copy to the follow path:
gen.dws
lk:
alps\vendor\mediatek\proprietary\bootable\bootloader\lk\target\$(proj)\dct\dct\codegen.dws

4.2. Driver Customization-parameters

4.2.1. Accelerometer parameters customization

a、 Config the parameters in the path:
alps\kernel-3.18\arch\armxx\boot\dts\$(proj).dts

```
cust_accel@0 {
compatible = "mediatek,mc3410";
i2c_num = <2>;
i2c_addr = <0x4c 0 0 0>;
direction = <0>;
power_id = <0xffff>;
power_vol = <0>;
firlen = <0>;
is_batch_supported = <0>;
};
```

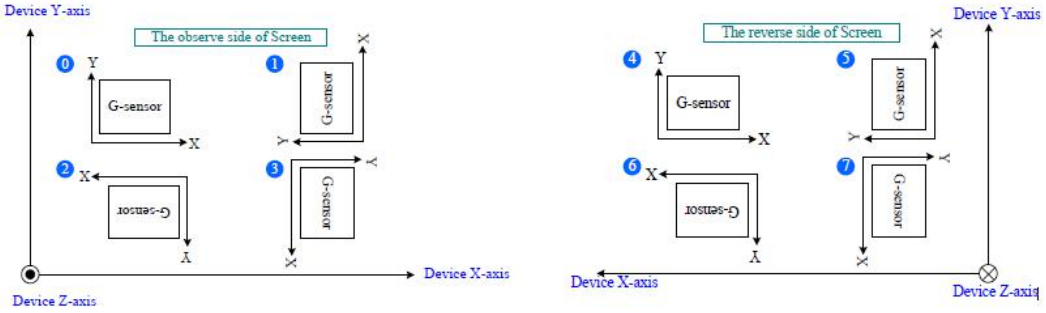
Note:
If the project is 32bits, armxx is arm.
If the project is 64bits, armxx is arm64.
b、Accelerometer parameters description:

compatible ： Identification for driver
i2c_num ： i2c channel ,depend on hardware
direction ： Sensor layout direction
power_id ： LDO id which the sensor has used
power_vol ： LDO voltage which the sensor hse used
is_batch_supported ： support batch mode or not

4.2.2.Notice

Mapping coordinate(accelerometer, gyroscope, magnetometer)

Value	Description
0	{x, y, z} => { x, y, z}
1	{x, y, z} => {-y, x, z}
2	{x, y, z} => {-x, -y, z}
3	{x, y, z} => { y, -x, z}
4	{x, y, z} => {-x, y, -z}
5	{x, y, z} => { y, x, -z}
6	{x, y, z} => { x, -y, -z}
7	{x, y, z} => {-y, -x, -z}



4.2.3. Gyroscope parameters customization

a、Config the parameters in the path:
alps\kernel-3.18\arch\armxx\boot\dts\$(proj).dts

```
cust_gyro@0 {
compatible              = "mediatek,itg1010";
i2c_num                 = <2>;
i2c_addr                = <0x68 0 0 0>;
direction               = <3>;
power_id                = <0xffff>;
power_vol               = <0>;
firlen                  = <0>;
is_batch_supported          = <0>;
};
```

Note:

If the project is 32bits, armxx is arm.

If the project is 64bits, armxx is arm64.

b、Gyroscope parameter description：

compatible :Identification for driver
i2c_num :i2c channel ,depend on hardware
direction :Mapping the coordinate
power_id : LDO id which the sensor has used
power_vol : LDO voltage which the sensor hse used
firlen Data :filter length, usually, it is set with 0
is_batch_supported :gyro support batch mode or not

4.2.4.Magnetometer parameters customization

a、alps\kernel-3.18\arch\armxx\boot\dts\$(proj).dts

```
cust_mag@0 {
compatible                  = "mediatek,akm09911";
i2c_num                     = <2>;
i2c_addr                    = <0x0D 0 0 0>;
direction                   = <1>;
power_id                    = <0xffff>;
power_vol                   = <0>;
is_batch_supported              = <0>;
};
```

Note:

If the project is 32bits, armxx is arm.

If the project is 64bits, armxx is arm64.

b、Magnetometer parameter description:

compatible :Identification for driver

i2c_num :i2c channel ,depend on hardware

Direction : Sensor layout direction

power_id : LDO id which the sensor has used

power_vol : LDO voltage which the sensor hse used

is_batch_supported :support batch mode or not

4.2.5.Alsps parameter description

a、Config the parameters with Device Tree

The customization file is :

alps\kernel-3.18\arch\armxx\boot\dts\$(project).dts

```
        cust_alsps@0 {
        compatible              = "mediatek,CM36652";
        i2c_num                 = <2>;
        i2c_addr                = <0x60 0 0 0>;
        polling_mode_ps         = <0>;
        polling_mode_als        = <1>;
        power_id                = <0xffff>;
        power_vol               = <0>;
/* Total has 15 level*/
        als_level               = <0 328 861 1377 3125 7721 7767 12621 23062 28430 33274 47116 57694 57694 65535>;
/*  Total has 16 range*/
        als_value               = <0 133 304 502 1004 2005 3058 5005 8008 10010 12000 16000 20000 20000 20000 20000>;
        ps_threshold_high       =  <26>;
        ps_threshold_low        =  <21>;
        is_batch_supported_ps   = <0>;
        is_batch_supported_als  = <0>;
        };
```

Note:

• If the project is 32bits, armxx is arm.

• If the project is 64bits, armxx is arm64.

b、Alsps parameter description

Compatible : Identification for driver

i2c_num :i2c channel ,depend on hardware
polling_mode_ps 0 :interrupt mode,1:polling mode
polling_mode_als 0:interrupt mode,1:polling mode
power_id : LDO id which the sensor has used
power_vol : LDO voltage which the sensor hse used
als_level : The als data mapping table:
als_value : als_level-->als_value
ps_threshold_high : ps_data > ps_threshold_high : close
ps_threshold_low : ps_data < ps_threshold_low : far away
is_batch_supported_ps : ps support batch mode or not
is_batch_supported_als :als support batch mode or not

4.2.6.Driver Customization-gpio

Config the pin function with Device Tree

The customization file is :alps\kernel-3.18\arch\armxx\boot\dts\$(project).dts

Define the pin state

Note:
(1) If the sensor do not use gpio,please ignore this step.
(2)If you want to know the variables meanings, you can see the documents about GPIO_Usage_Guide .

Correlate the pin state with module

```
/* sensor gpio standization */
&pio {
    alsps_intpin_cfg: alspspincfg {

        pins_cmd_dat {
            pins = <PINMUX_GPIO65__FUNC_GPIO65>;
            slew-rate = <0>;
            bias-pull-up = <00>;
        };
    };

    alsps_intpin_default: alspsdefaultcfg {

    };

    gyro_intpin_cfg: gyropincfg {

        pins_cmd_dat {
            pins = <PINMUX_GPIO67__FUNC_GPIO67>;
            slew-rate = <0>;
            bias-pull-down = <00>;
        };
    };

    gyro_intpin_default: gyrodefaultcfg {

    };
};
&alsps {
    pinctrl-names = "pin_default", "pin_cfg";
    pinctrl-0 = <&alsps_intpin_default>;
    pinctrl-1 = <&alsps_intpin_cfg>;
    status = "okay";
};
```

5. Choose sensor P/N

Choose the special sensor P/N in kernel defconfig file:
arch/armxx/configs/$(proj)_debug_defconfig
arch/armxx/configs/$(proj)_defconfig
Example:

```
CONFIG_MTK_CM36652_NEW=y

CONFIG_MTK_MC3410_NEW=y

CONFIG_MTK_ITG1010_NEW=y

CONFIG_MTK_AKM09911_NEW=y
```

And suggest use “make menuconfig” to do it

6. Sensor Compatible

6.1. Detection principle

– Using i2c connection and device information to detect the different device.
– If using one i2c address to connect device fail, then don't choose this device.
– When i2c connection is success, then read the device information. If the information is right, then choose this device. Otherwise, don't choose it.

6.2. There use accelerometer as example

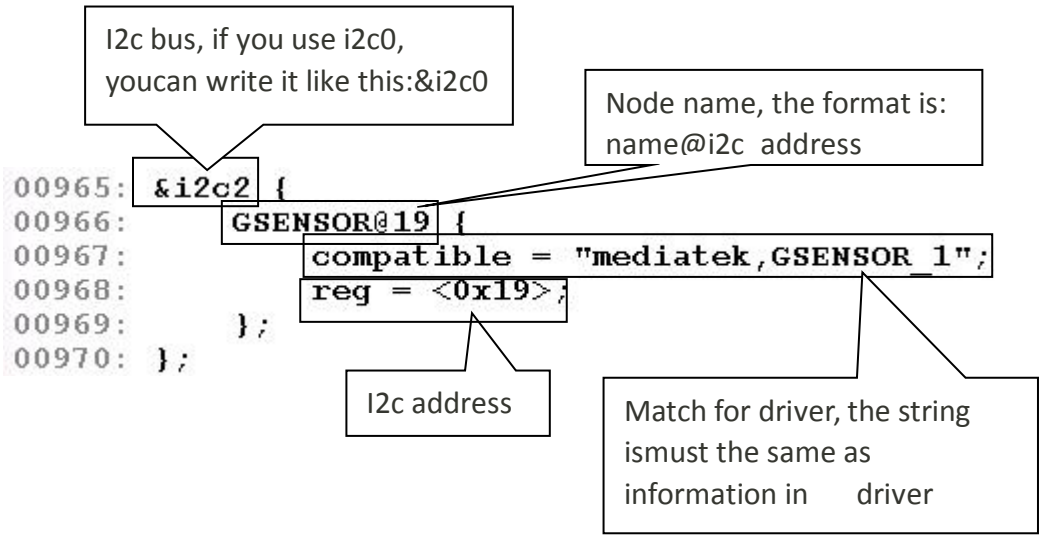
Step 1:
– The first sensor driver do not need change.
– It still use DCT to configure the first sensor's i2c address.

Step 2:
– Modify the second sensor driver.

```
#if defined(CONFIG_OF)
static const struct of_device_id gma303_match_table[] =
{
    {.compatible = "mediatek,GSENSOR_1"},
    {}
};
#endif
static struct i2c_driver gma30x_i2c_driver = {
    .driver = {
//    .owner      = THIS_MODULE,
        .name           = GMA30x_DEV_NAME,
    #if defined(CONFIG_OF)
        .of_match_table = gma303_match_table,
    #endif
    },
    .probe              = gma30x_i2c_probe,
    .remove             = gma30x_i2c_remove,
    .suspend            = gma30x_suspend,
    .resume             = gma30x_resume,
    .id_table = gma303_i2c_id,
};
```

This string is must the same as the information in DTS

Step 3:
– Configure the second sensor's i2c address in $(project).dts



Step 4:

– Configure the customization parameter for second sensor in $(project).dts

```
cust_accel@0 {
        compatible              = "mediatek,icm20645g";
        i2c_num                 = <1>;
        i2c_addr                = <0x68 0 0 0>;
        direction               = <3>;
        power_id                = <0xffff>;
        power_vol               = <0>;
        firlen                  = <0>;
        is_batch_supported      = <0>;
    };

    cust_alsps@0 {
        compatible              = "mediatek,cm36558";
        i2c_num                 = <1>;
        i2c_addr                = <0x51 0 0 0>;
        polling_mode_ps         = <0>;
        polling_mode_als        = <1>;
        power_id                = <0xffff>;
        power_vol               = <0>;
        als_level = <0 328 861 1377 3125 7721 7767 12621 23062 28430 33274 47116 57694
57694 65535>;
        als_value = <0 133 304 502 1004 2005 3058 5005 8008 10010 12000 16000 20000
20000 20000 20000>;
        ps_threshold_high       = <26>;
```

```
        ps_threshold_low        = <21>;
        is_batch_supported_ps   = <0>;
        is_batch_supported_als  = <0>;
    };
```