

Drive Proting Note

1. sensor i2c slave address

1) optional i2c slave address

some sensors have optional i2c slave address , please check the hardware schematic design to decide which is the i2c salve address.

2) avoid sensor i2c slave address conflict.

In the same i2c bus, make sure sensor devices i2c slave address do not conflict with other sensor. There are not two sensors with the same i2c slave address.

2. check the linux kernel driver porting

1) check if hardware connect correctly on the i2c bus

If sensor connect to i2c bus error, there will be kernel error log to print , use "**dmesg**" command (enter into debug com, or use "**adb shell**" to enter into your mobile console) to print the kernel log,

```
#dmesg
....
read chip ID 0x.. is not equal to ....
....
```

It shows it check the sensor "**WHO AM I**" (chip id) register data error when driver probes.

2) check if kernel driver probe successfully

If success, the driver will create a input device, Accelerometer sensor will create a input device called "**FreescaleAccelerometer**" , and magnetometer will create a input device "**FreescaleMagnetometer**" and a input device "**eCompass**" to report calibrated data. And each input device has some sysfs interfaces , including "**enable**" , "**position**" , "**poll**" , "**min**" and "**max**" .

Use command "**cat /sys/class/input/*/name**" to show all the input devices name in you mobile. There will be two devices "**FreescaleAccelerometer**" , "**FreescaleMagnetometer**" and "**eCompass**"

```
#cat /sys/class/input/*/name
```

```
gpio-keys
FSL MPR03X Touchkey
eGalax Touch Screen
da9052-onkey
isl29023 light sensor
FreescaleAccelerometer
FreescaleMagnetometer
eCompass
.....
```

FreescaleAccelerometer and **FreescaleMagnetometer** are input5 and input6 respectively .
Check the sysfs interfaces of the input device :

```
# ls -l /sys/class/input/input5
```

```
.....
-rw-rw-rw- root  input  4096 1970-01-01 00:00 enable
-rw-rw-rw- root  input  4096 1970-01-01 00:00 max
-rw-rw-rw- root  input  4096 1970-01-01 00:00 min
-rw-rw---- root  input  4096 1970-01-01 00:00 name
-rw-rw-rw- root  input  4096 1970-01-01 00:00 poll
-rw-r--r-- root  root    4096 1970-01-02 00:14 position
.....
```

Please make sure "**enable**", "**poll**", "**min**" and "**max**" have "read/write" permission . there will be used for Android hal to "enable/disable" and "setDelay" sensor.

3) check kernel driver data report

Firstly , set the sensor enable and report delay time ,use command

```
# echo 1 > /sys/class/input/input5/enable
```

```
# echo 100 > /sys/class/input/input5/poll
```

Then , use command "**getevent**" to print data report information.

```
# getevent /dev/input/event5
```

```
0003 0000 000000e0
0003 0001 00000410
0003 0002 00004190
0000 0000 00000000
0003 0000 000001f0
0003 0001 00000428
0000 0000 00000000
0003 0000 00000128
0003 0001 00000490
0003 0002 00004120
0000 0000 00000000
```

It show there is sensor data to report, if no information to print, the kernel driver loading is error.

Note : the number marked with red color should be different on different mobile device , it depends on how many input devices in your mobile and the order your system load the sensor driver.

3. check magd running

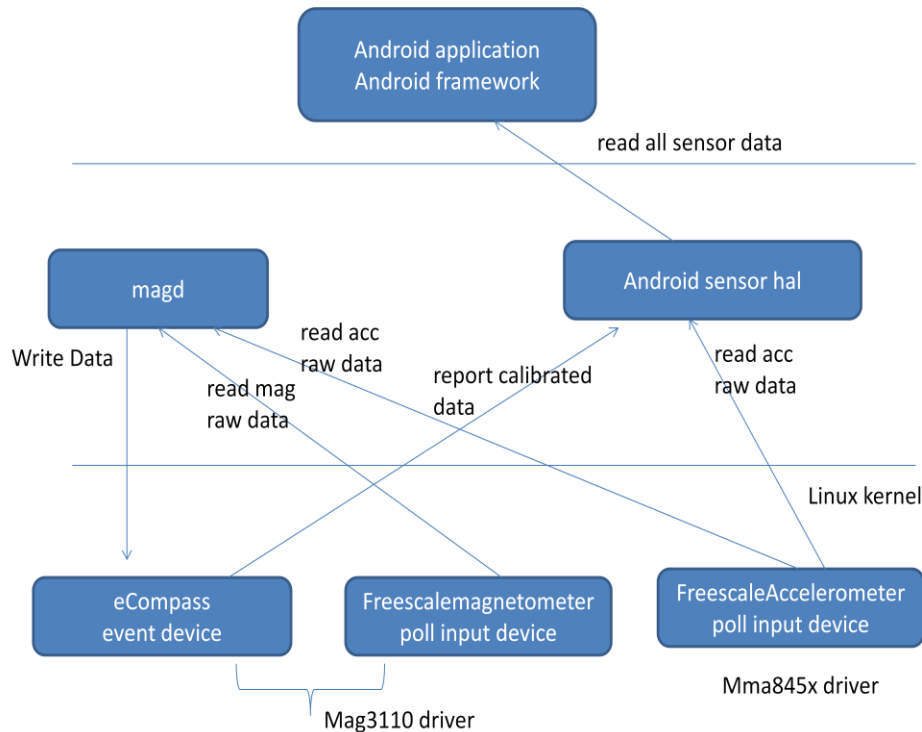
1) check if magd run correctly .

set to start *magd* (Freescale Magnetometer sensor calibration service) in *init.rc* of your system, it must start after the linux kernel run completely and before Android load sensor dynamic library(*sensors.default.so*).

when *magd* start , it will check if there are two input device "*FreescaleAccelerometer*" and "*FreescaleMagnetometer*", if there are the two devices , *magd* will read data "*FreescaleAccelerometer*" and "*FreescaleMagnetometer*", and do calibration and report the calibrated data to "*eCompass*". if there are not the two devices , *magd* will exit.

4. Relationship of kernel , magd and Android HAL

The relationship of kernel , *magd* and Android Sensor Hal is as picture 1 show:



Picture

5. how the Android sensors system work

when mobile power on and boot , the system will load kernel and run it , it will create **“FreescaleAccelerometer”** poll input device and **“FreescaleMagnetometer”** poll input device and **“eCompass”**. when kernel run completely , system will check **init.rc** and start the **magd** service , when **magd** run, it will check if there are **“FreescaleAccelerometer”** and **“FreescaleMagnetometer”** device . if not , **magd** will exit . if existing the devices , **magd** will wait for **“FreescaleAccelerometer”** and **“FreescaleMagnetometer”** device to report raw data. After this, Android system load some dynamic library include **sensor.default.so**(Android sensor HAL) and start Android sensor Service, then it will check if there are **“eCompass”** , if **“eCompass”** exist, HAL will read **“eCompass”** data as **SENSOR_TYPE_MAGNETIC_FIELD** and **SENSOR_TYPE_ORIENTATION** data , which will be reported to Android framework and Application.

