

**Telechips**

---

# TCCxxxx-Android- IceCreamSandwich-V1.1E- Sensor Guide

TCCxxxx-Android-IceCreamSandwich-V1.1E-Sensor Guide

Jun. 21, 2012

***Telechips***

## Telechips

---

### Revision History

Date	Version	Description
2012-01-30	1.00	Initial Release
2012-06-21	1.01	Modification about the i2c channel

## Telechips

---

### TABLE OF CONTENTS

#### Contents

Revision History .....	2
TABLE OF CONTENTS .....	3
Contents .....	3
1. Introduction. ....	4
2. To Use The G-Sensor Driver .....	5
3. To Use the eCompass(AK8975C) Driver .....	8
4. To Add The New G-sensor Driver .....	13
4.1. Predefine Part .....	13
4.2. Function Part .....	13
5. The Sensor Coordinate System & Orientation .....	16
5.1. sensor coordinate system .....	16
5.2. Orientation of the device .....	16
6. The General Issues About The Sensor .....	17
6.1. Issue about the g-sensor coordinate-system .....	18
6.2. Issue about the pitch/roll coordinate-system .....	19
6.3. Issue about the azimuth .....	19

## Telechips

---

### 1. Introduction.

This document describes how to use the sensor and about the general sensor issues.

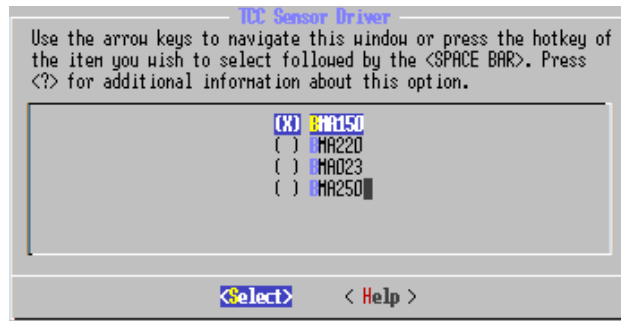
## Telechips

### 2. To Use The G-Sensor Driver

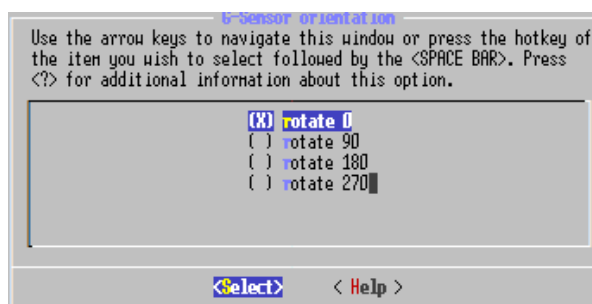
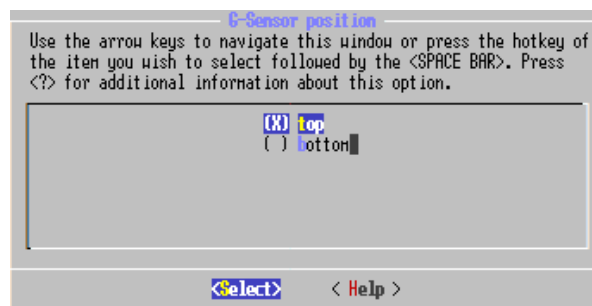
To use the g-sensor, you must change as follows.

1. Please execute "make menuconfig" command from kernel folder and select configurations.  
Select "Device Driver --> Character devices --> TCC Sensor Driver --> Tcc Sensor Driver(BMA150)". You can see below Sensor modules.

You must select the proper driver.



And you must select the proper setting for the position and orientation.  
(refer to next page)



To set the I2C channel for the g-sensor.

File : kernel/drivers/char/sensor\_bmaXXX.c

Function : tcc\_sensor\_i2c\_register()

```
adapter = i2c_get_adapter(I2C Channel Number);
```

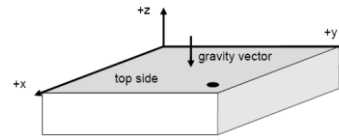
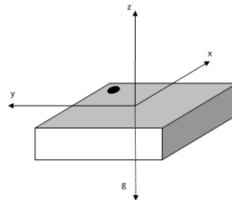
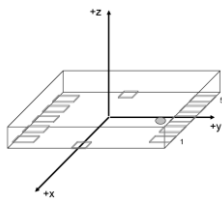
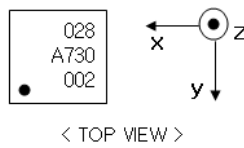
ex) If you want to use the channel '0', set the channel number as below.

```
adapter = i2c_get_adapter(0);
```

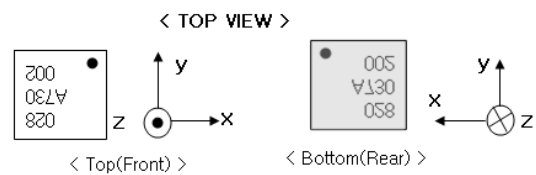
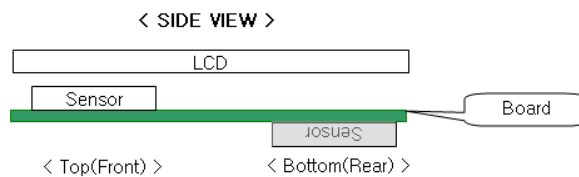
## Telechips

The orientation of sensor can differ with the first pin. It must check the datasheet.

Ex) Orientation of BMA150 ( G-sensor )



In each case, the orientation marks are as below. ( Ex : BMA150 )



Representative eight layout are indicated below. ( TOP VIEW )



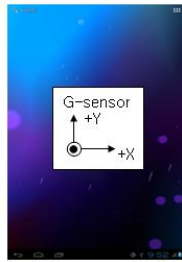
< rotate 0 -> 90 -> 180 -> 270 ( top ) >



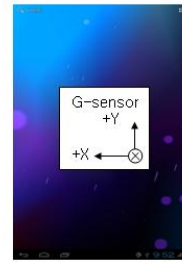
< rotate 0 -> 90 -> 180 -> 270 ( bottom ) >

In handset(portrait) device, the each rotate '0' is as below.

## Telechips



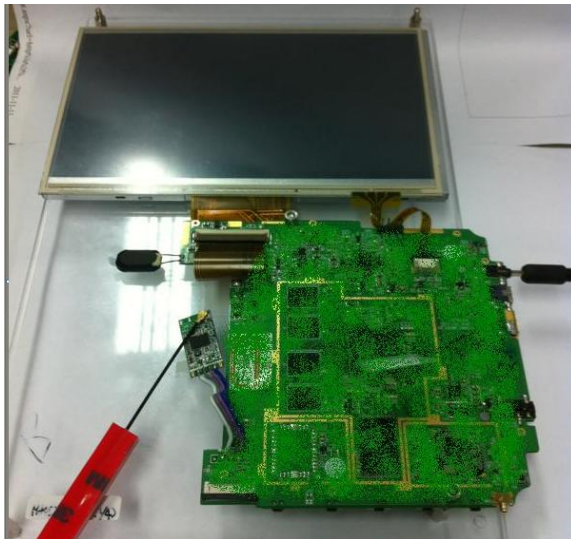
&lt;top(front),rotate 0 &gt;



&lt;bottom(rear), rotate 0 &gt;

In the early stage of development, sometimes the EVM board as below is used.

In this case, you must be careful that there is the difference between the sensor configurations of EVM board and the sensor configurations of real set .



※ To use the temperature operation within the BMA150/BMA250

To use the temperature operation within the g-sensor chips, you must change as follows.

kernel/drivers/char/sensor\_bma150(or 250).c

```
#define SENSOR_TEMPERATURE_SUPPORT SENSOR_SUPPORT
```

## Telechips

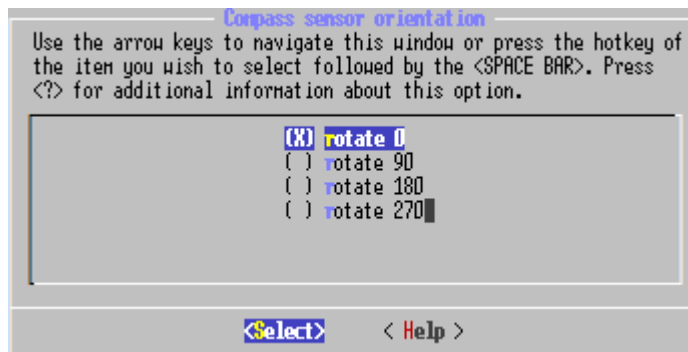
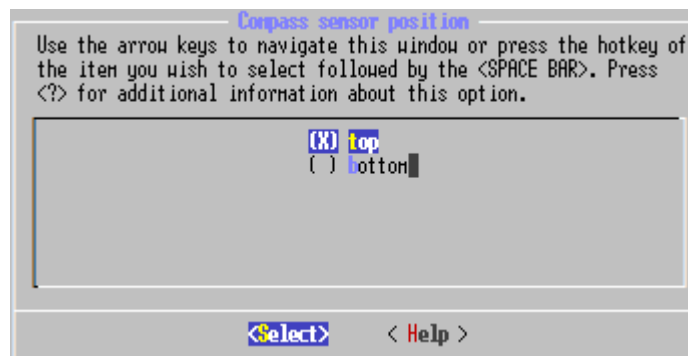
### 3. To Use the eCompass(AK8975C) Driver

To use the eCompass driver, you must change as follows.

And, **enable** "Device Driver --> Misc devices --> AK8975 compass support".  
If not support g-sensor and compass, disable.

```
< >  laos ISL2550 ambient light sensor
< * >  AK8975 compass support
      Compass sensor position (top) --->
      Compass sensor orientation (rotate 0) --->
< >  Dallas DS1682 Total Elapsed Time Recorder with Alarm
```

And you must select the proper setting for position and orientation.  
(refer to next page)

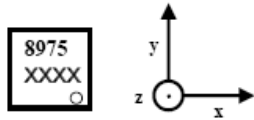




## Telechips

The orientation of sensor can differ with the first pin. It must check the datasheet.

Ex) Orientation of AK8975C ( eCompass )



The representative eight layout are as below. ( TOP VIEW )



Figure 1. rotate 0 -> 90 -> 180 -> 270 ( top )

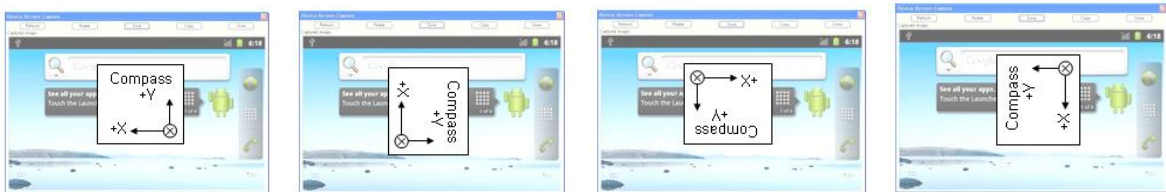
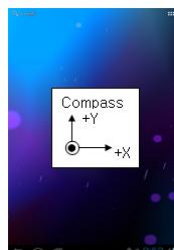
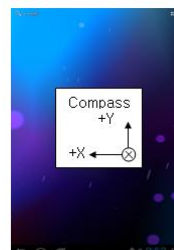


Figure 2. rotate 0 -> 90 -> 180 -> 270 ( bottom )

In handset(portrait) device, the each rotate 0 is as below.



<top,rotate 0 >



<bottom(rear), rotate 0 >

## Telechips

To set the GPIO for the IRQ of eCompass

File : kernel/arch/arm/mach-tccXXXX/board-tccXXXX.c

Function : tcc8920\_init\_machine()

Ex) To use the #29 pin of GPIO E port

```
tcc_gpio_config(TCC_GPE(29), GPIO_FN(0)|GPIO_PULL_DISABLE);
gpio_direction_input(TCC_GPE(29));
tcc_gpio_config_ext_intr(INT_EI1, EXTINT_GPIOE_29);
```

To set the I2C channel for the eCompass

File : kernel/arch/arm/mach-tccXXXX/board-tccXXXX.c

Structure name :

I2C channel number	Structure name
0	i2c_devices0[]
1	i2c_devices1[]
2	i2c_devices2[]
3	i2c_devices3[]

Add the below member at the array of proper above structure.

```
#if defined(CONFIG_SENSORS_AK8975)
{
    I2C_BOARD_INFO("akm8975", 0x0F),
    .irq          = INT_EI1,
    .platform_data = &akm8975_data,
},
#endif
```

Ex) If you want to use the channel '0', add the above member to the i2c\_devices1[].

Function : tcc8920\_init\_machine()

```
i2c_register_board_info(0, i2c_devices0, ARRAY_SIZE(i2c_devices0));
i2c_register_board_info(1, i2c_devices1, ARRAY_SIZE(i2c_devices1));
i2c_register_board_info(2, i2c_devices2, ARRAY_SIZE(i2c_devices2));
i2c_register_board_info(3, i2c_devices3, ARRAY_SIZE(i2c_devices3));
```

## Telechips

---

### ※ To check the inner calibration of eCompass.

If you checked the basic operation of eCompass, you must check the calibration level of eCompass. In this step, the calibration is the inner operation of eCompass module. If there is a wondering or strange something about the calibration of eCompass, telechips can not analysis personally. Because the main library of eCompass module is the confidential material of AKM.

The step for checking the calibration of eCompass(AK8975C) is as below.

1. Low format whole memory before download the firmware.
  - A. For removing the /data/misc/akmd\_set.txt file
2. Download the firmware.
3. Reboot the device
4. Install the application which uses the eCompass data
  - A. ex) Compass application..
5. Run the application
6. Wave your device in a figure 8 pattern and check the ddms log as below.
  - A. In normal case, the level will become '3' within several or dozens seconds

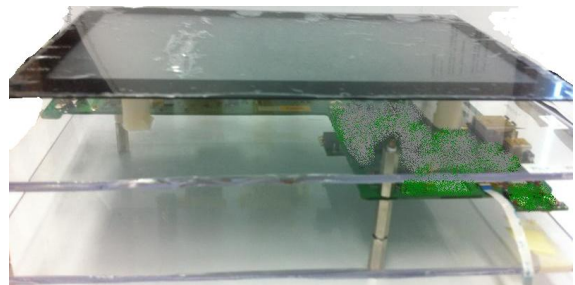
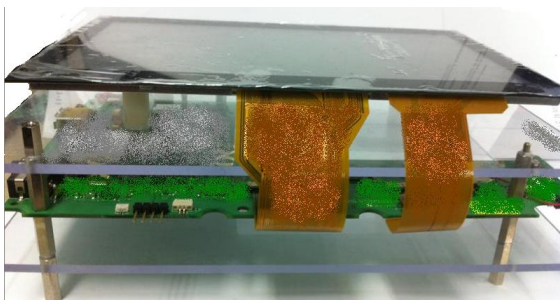
```
01-31 21:41:22,490  D  1096  Sensors  M-Sensor status 1
01-31 21:41:25,170  D  1096  Sensors  M-Sensor status 2
01-31 21:41:26,720  D  1096  Sensors  M-Sensor status 3
```

If the calibration level doesn't become level '3' (the basic operation works normally), the accuracy of eCompass is a bad. In this case, you will have to check the H/W layout with the AKM. When you have a question or issue about the eCompass, contact to Telechips first.

## Telechips

---

In development step, sometimes the EVM board as below is used.



Even if the calibration of eCompass in EVM board works normally, sometimes the calibration of eCompass in the real set(or the set which is similar with the real set) doesn't work normally.



Because the eCompass can be influenced by the around electric element.

***If the new EVM board is made or the H/W layout is changed, you must check the calibration of eCompass using the real set (or the set which is similar with the real set).***

## Telechips

### 4. To Add The New G-sensor Driver

#### 4.1. Predefine Part

※ These numerical predefines is declared as an integer type

Predefine Name	Description
BMA_CHIP_ID	The return value which reads the chip id registry. This is used for checking whether the chip is a valid or not.
BMA_ORG_DATA_BIT_WIDTH	The width of acceleration data.
BMA_1G_RESOLUTION	Resolution // ex) Acceleration measurement range $\pm 2g$ // sensitivity = 3.91mg/LSB // resolution = 1 / 0.00391 = 256
SENSOR_I2C_ADDRESS	The unique address of I2C which is used for the g-sensor. check the datasheet of g-sensor.
SENSOR_DEFAULT_RANGE	The default value which is used at the function for setting the range. <b><i>This should be set the range as <math>\pm 2G</math></i></b> (refer to tcc_sensor_set_range() function )
SENSOR_DEFAULT_BW	The default value which is used at the function for setting the band width. (refer to tcc_sensor_set_bandwidth () function )
SENSOR_NAME	"tcc-accel" // fixed
SENSOR_HAL_NAME	Not used
SENSOR_CALIBRATION_MODE	CALIBRATION_SOFT // fixed
SENSOR_DEF_DEVICE_TYPE	DEVICE_TYPE_CHAR // fixed
SENSOR_TEMPERATURE_SUPPORT	SENSOR_NOT_SUPPORT // default SENSOR_SUPPORT // If there is the temperature sensor within the g-sensor ( ex, bma150, bma250)
SENSOR_TEMPERATURE_RESOLUTION	Temperature resolution // 1/sensitivity, // ex) bma250 // sensitivity = 0.5 °C , resolution = 1/sensitivity = 2
SENSOR_TEMPERATURE_CENTER	Temperature offset // ex) bma250 , Temperature offset = 24 °C // Center temperature = 24 °C -> temp = 00000000b

#### 4.2. Function Part

##### A. Functions that must create the whole operation

- tcc\_sensor\_set\_range // fill the operation for setting the range
- tcc\_sensor\_set\_bandwidth // fill the operation for setting the bandwidth
- tcc\_sensor\_read\_accel\_xyz // fill the operation for reading acceleration data from sensor
- tcc\_sensor\_chip\_init // fill the operation for initializing the sensor

##### B. Functions that should create the whole operation if you need

- tcc\_sensor\_read\_temperature

## Telechips

---

### C. Functions that should modify the part of operation

- `tcc_sensor_i2c_register`  
 // set the I2C channel number which is used by the g-sensor  
 -> `i2c_get_adapter( I2C channel number)`
- `tcc_sensor_init`  
 // If the interrupt is used, set the configuration of GPIO which is used for the interrupt  
 Ex) To use the #25 pin of GPIO F port

```
tcc_gpio_config(TCC_GPF(25), GPIO_FN(0)|GPIO_PULL_DISABLE);
gpio_direction_input(TCC_GPF(25));
tcc_gpio_config_ext_intr(INT_EI1, EXTINT_GPIOF_25);
```

### D. Functions that doesn't need to modify

- `tcc_sensor_attr_autocalibration_show`
- `tcc_sensor_attr_autocalibration_store`
- `tcc_sensor_attr_calibration_show`
- `tcc_sensor_attr_calibration_store`
- `tcc_sensor_attr_resolution_show`
- `tcc_sensor_attr_resolution_store`
- `tcc_sensor_attr_inputdevice_show`
- `tcc_sensor_attr_inputdevice_store`
- `tcc_sensor_attr_delay_show`
- `tcc_sensor_attr_delay_store`
- `tcc_sensor_attr_enable_show`
- `tcc_sensor_set_enable_by_client_for_calibration`
- `tcc_sensor_attr_set_enable_by_client`
- `tcc_sensor_attr_set_enable`
- `tcc_sensor_attr_enable_store`
- `tcc_sensor_attr_halname_show`
- `tcc_sensor_attr_halname_store`
- `tcc_sensor_attr_temperature_show`
- `tcc_sensor_attr_temperature_store`
- `tcc_sensor_convertCoordination`
- `tcc_sensor_compensation`
- `tcc_sensor_set_data`
- `tcc_sensor_avg_data`
- `tcc_sensor_work_func`
- `sensor_i2c_probe`
- `sensor_i2c_remove`
- `sensor_i2c_suspend`
- `sensor_i2c_resume`
- `tcc_sensor_write`
- `tcc_sensor_read`
- `tcc_sensor_ioctl`
- `tcc_sensor_release`
- `tcc_sensor_open`
- `tcc_sensor_exit`

## Telechips

To add the new configuration to Kconfig file.

File : kernel/drivers/char/Kconfig

```
if TCC_SENSOR_DRV

choice
    prompt "TCC Sensor Driver"
    default SENSOR_BMA150
    depends on I2C

config SENSOR_BMA150
    bool "BMA150"
    depends on I2C
...

config SENSOR_BMA250
    bool "BMA250"
    depends on I2C

config SENSOR_ABCDE
    bool "ABCDE"
    depends on I2C

endchoice
```

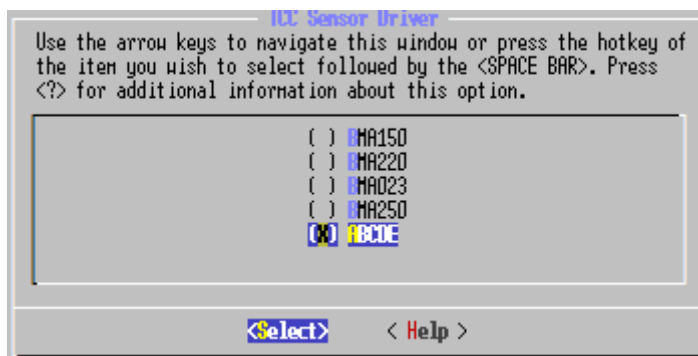
To modify makefile for new sensor driver.

File : kernel/drivers/char/Makefile

```
# S: telechips
obj-$(CONFIG_SENSOR_BMA150)      += sensor_bma150.o
...
obj-$(CONFIG_SENSOR_BMA250)      += sensor_bma250.o
obj-$(CONFIG_SENSOR_ABCDE)      += sensor_ABCDE.o
# E: telechips
```

To set the kernel configuration

Device Drivers -> Character devices -> TCC Sensor Driver -> TCC Sensor Driver ->



## Telechips

### 5. The Sensor Coordinate System & Orientation

#### 5.1. sensor coordinate system

The coordinate system is defined relative to the device's screen when the device is held in its default orientation. When a device is held in its default orientation, the X axis is horizontal and points to the right, the Y axis is vertical and points up, and the Z axis points toward the outside of the screen face. In this system, coordinates behind the screen have negative Z values.

...

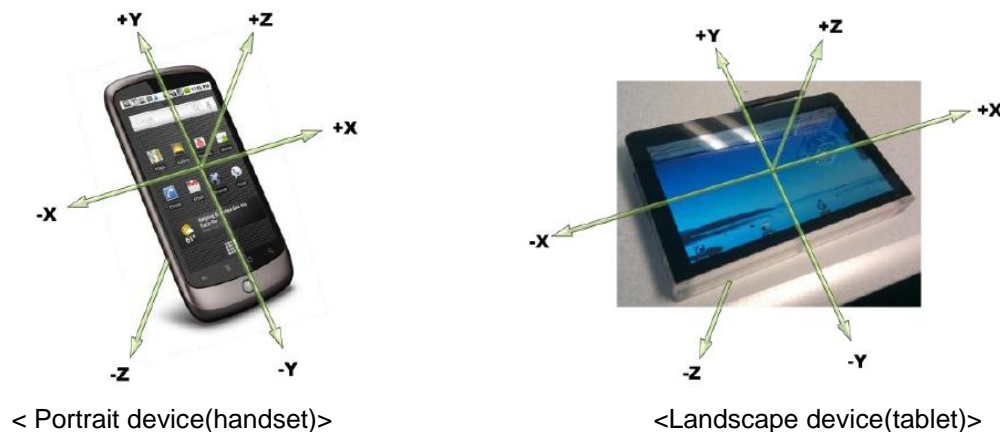
The most important point to understand about this coordinate system is that the axes are not swapped when the device's screen orientation changes.

...

Another point to understand is that your application must not assume that a device's natural (default) orientation is portrait. The natural orientation for many tablet devices is landscape. And the sensor coordinate system is always based on the natural orientation of a device.

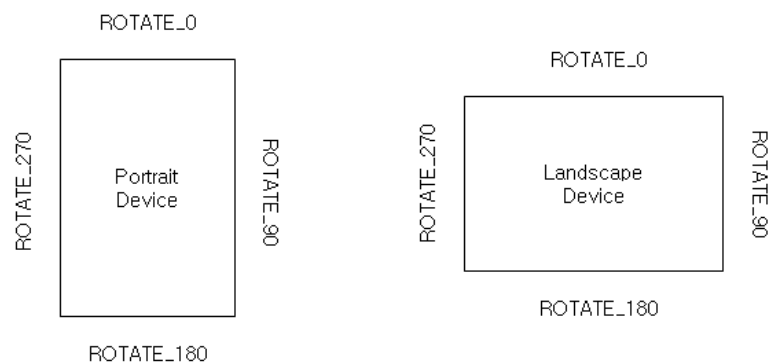
(refer to [http://developer.android.com/guide/topics/sensors/sensors\\_overview.html](http://developer.android.com/guide/topics/sensors/sensors_overview.html))

According to the LCD shape, the sensor coordinate-system is as below.



#### 5.2. Orientation of the device

According to the LCD shape in the application view , the orientation of the device is as below.





## Telechips

---

### 6. The General Issues About The Sensor

Some applications are made without considering about the landscape(tablet) device. In this case, the application uses the coordinate system like the portrait(handset) device regardless of the LCD shape( portrait / landscape). These applications work normally in the portrait(handset) device, but doesn't work normally in the landscape(tablet) device.

There is the blog of android developers which is mentioned about this issue.

(refer to <http://android-developers.blogspot.com/2010/09/one-screen-turn-deserves-another.html> )

Some applications are made without considering about the LCD shape. Any other applications are made with considering about the LCD shape.

In the landscape(table) device, two cases can not work normally at the same time.

***This phenomenon is caused by the wrong applications.***

There are many application which doesn't meet the policy of android about the sensor coordinate-system. Honeycomb is the new version of Android build especially for tablets. I think that applications, which is made after Honeycomb version is released, will meet the policy of android. When you can not judge whether the application is a wrong or not, it is helpful for your decision to compare with the reference tablet product.[ ex. Galaxy tab 10.1 (honeycomb or higher version) ]

The next <6.1> ~<6.3> are the issues that it can happen by the application which is made without the consideration about the tablet(landscape) device.



## Telechips

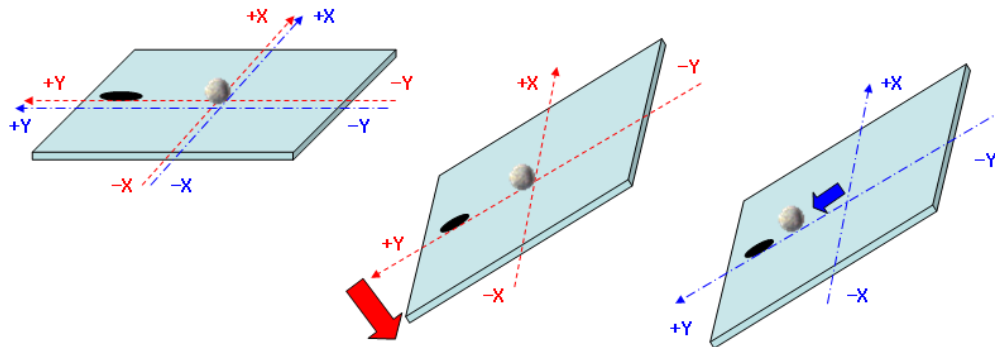
### 6.1. Issue about the g-sensor coordinate-system

It assumes that the game is made without the consideration for the landscape device

#### [ In the handset(portrait) device ]

The coordinate-system of sensor and application are same as below.



Sensor coordinate-system	Portrait shape	
Application coordinate-system	Portrait shape	

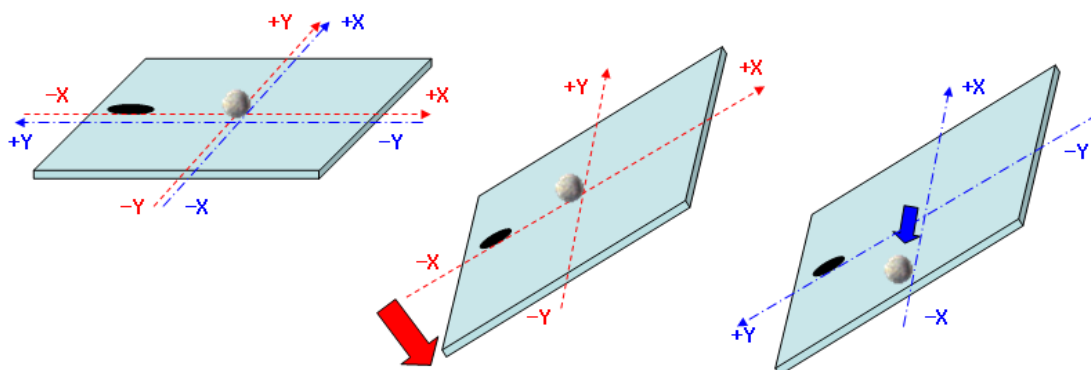


1. when the device is tilted to the left
2. the y acceleration value becomes negative(-Y)
3. the application interprets as the device is tilted toward (+Y)
4. the application draw the new ball image at the left position (+Y).

#### [ In the tablet(landscape) device ]

The coordinate-system of sensor and application are different as below.

Sensor coordinate-system	Landscape shape	
Application coordinate-system	Portrait shape	

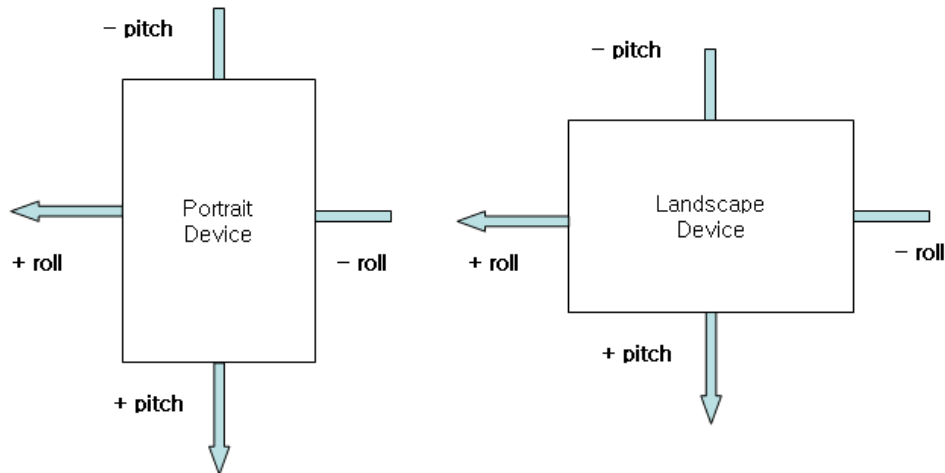


1. when the device is tilted to the left
2. the x acceleration value becomes positive(+X)
3. the application interprets as the device is tilted toward (-X)
4. the application draw the new ball image at the down position (-X).

## Telechips

### 6.2. Issue about the pitch/roll coordinate-system

When some application run on the landscape(tablet) device, there is the pitch/roll direction issue which is similar to the above g-sensor coordinate-system issue.



### 6.3. Issue about the azimuth

In normal application that the default orientation is a portrait, 0 degree is represented as below.



< Handset(portrait) device >



< Tablet(landscape) device >

In some application which is made without the consideration for the tablet device, 0 degree is misrepresented as below.



< Tablet(landscape) device >