

# Globalmems Shell Script Manual

## General Introduction

This script is helpful for GMA driver to calibrate the accelerometer.

It has to provide two kinds of paths to gss.sh choice.

Path 1: use sysfs “/sys/class/input/inputX/calibration”

Path 2: use gmad . See detailed supplementary documents.

Reminder: This shell script should be used with gma30x/gme60x\_accel kernel driver. Must implement the following five interfaces.

1. /sys/class/input/inputX/offset
2. /sys/class/input/inputX/reg\_rx
3. /sys/class/input/inputX/reg\_tx
4. /sys/class/input/inputX/chipinfo
5. /sys/class/input/inputX/calibration

There are five Features, refer to the following

## Index

Globalmems Shell Script Manual.....	1
General Introduction.....	1
1. Features .....	2
2. Usage.....	2
3. File Usage .....	3
4. Supplement.....	5

## 1. Features

- **calib** – calibration , the offset archive to “/data/misc/gsensor\_offset.txt”
- **clear\_offset** - clear offset
- **read\_reg** - Read register value
- **write\_reg** - The value written to the register
- **read\_reg\_map** - Read all register value and archive to "/data/misc/reg\_map.txt"

## 2. Usage

The gss.sh copied to the device, and execute permissions to open.

Example:

```
# adb push gss.sh /system/bin
# adb remount
# adb shell
# chmod a+x /system/bin/gss.sh
```

```
lenovo@lenovo-THINK /cygdrive/q
$ adb remount
remount succeeded
lenovo@lenovo-THINK /cygdrive/q
$ adb shell
root@tiny4412:/ #
```

2.1 Prototype : `sh /system/bin/gss.sh read_reg REG`

```
# cd /system/bin
```

```
# sh gss.sh read_reg 0x04
```

chip	REG(Chip id)	Value	REG	Value
Gma302	0x00	0x02	0x04	0x55
Gma303	0x00	0x03	0x04	0x55
Gma305	0x00	0x05	0x04	0x55
Gme601	0x00	0x05	0x04	0x55
Gme603	0x00	0x05	0x04	0x55
Gme605	0x00	0xe5	0x04	0x55

Example: Reads the value of register 0x04, and its value is 0x55.

```
root@tiny4412:/ #
root@tiny4412:/ # cd /system/bin
cd /system/bin
root@tiny4412:/system/bin # sh gss.sh read_reg 0x00
sh gss.sh read_reg 0x00
0xE5 0x00 0x00 0x1F 0x55 0x40 ( Wed Jan 1 12:13:52 GMT 2014)
root@tiny4412:/system/bin #
root@tiny4412:/system/bin # sh gss.sh read_reg 0x04
sh gss.sh read_reg 0x04
0x55 0x00 0x37 0x5E 0x00 0x45 ( Wed Jan 1 12:14:06 GMT 2014)
```

2.2 Prototype : `sh /system/bin/gss.sh write_reg REG VALUE`

```
# sh /system/bin/gss.sh write_reg 0x0f 0x00
```

Example: Register 0x0f set its value 0x00.

```
root@panda:/ # sh /system/bin/gss.sh write_reg 0x0f 0x00
0x00 0x00 0x07 0x55 0xc0 0x06 ( Fri Jan 2 01:17:11 GMT 1970)
```

2.3 Prototype : `sh /system/bin/gss.sh calib asix`

`# sh /system/bin/gss.sh calib 9`

Examples: Do zero-G and save offset.

Note : path fixed !!!save offset to file `"/data/misc/gsensor_offset.txt"`

```
root@tiny4412:/system/bin # sh gss.sh calib 9
sh gss.sh calib 9
step 2: Get layout= 1 Set position= 1 ( Wed Jan 1 12:43:30 GMT 2014)
step 3: Start calibration. ( Wed Jan 1 12:43:30 GMT 2014)
step 4: GRAVITY_ON_Z_NEGATIVE
step 5: File gsensor_offset.txt exists.
Read old offset 0 0 0.
Save new offset 938 145 -410.
```

2.4 Prototype : `sh /system/bin/gss.sh clear_offset`

`# sh /system/bin/gss.sh clear_offset`

Examples: clear offset and save (0 0 0) to `gsensor_offset.txt`.

```
root@tiny4412:/system/bin # sh gss.sh clear_offset
sh gss.sh clear_offset
clear offset 0 0 0
step 5: File gsensor_offset.txt exists.
Read old offset .
Save new offset 0 0 0.
```

2.5 Prototype : `sh /system/bin/gss.sh read_reg_map`

`# sh /system/bin/gss.sh read_reg_map`

```
root@tiny4412:/system/bin # sh gss.sh read_reg_map
sh gss.sh read_reg_map
gme605a Register map save to /data/misc/reg_map.txt
root@tiny4412:/system/bin # cat /data/misc/reg_map.txt
cat /data/misc/reg_map.txt
Read Gme605_accel Register MAP
REG(0x00) 0xE5 0x00 0x00 0x1F 0x55 0x00
REG(0x01) 0x00 0x00 0x00 0x00 0x00 0x00
REG(0x02) 0x00 0x1F 0x55 0x00 0x34 0xD8
REG(0x03) 0x1F 0x55 0x00 0x35 0xD1 0x01
REG(0x04) 0x55 0x00 0x34 0xD1 0x01 0x47
REG(0x05) 0x34 0xD1 0x01 0x47 0x00 0x30
REG(0x06) 0xD1 0x01 0x47 0x00 0x30 0xFD
REG(0x07) 0x01 0x49 0x00 0x30 0xFD 0x00
REG(0x08) 0x49 0x00 0x30 0xFD 0x00 0x00
REG(0x09) 0x00 0x30 0xFD 0x00 0x00 0x55
REG(0x0a) 0x2D 0xFD 0x00 0x00 0x55 0x00
REG(0x0b) 0xFD 0x00 0x00 0x55 0x00 0x34
REG(0x0c) 0x00 0x00 0x55 0x00 0x35 0xD6
REG(0x0d) 0x00 0x55 0x00 0x34 0xD6 0x01
REG(0x15) 0x00 0x00 0xF0 0x40 0x00 0x00
REG(0x16) 0x00 0xF0 0x40 0x00 0x00 0x00
REG(0x17) 0xF0 0x40 0x00 0x00 0x00 0x00
REG(0x18) 0x40 0x00 0x00 0x00 0x00 0x00
REG(0x38) 0x5F 0x00 0x00 0x00 0x00 0x00
```

### 3. File Usage

- `/data/misc/gsensor_offset.txt` : Record lastest offset each reboot thus read the file offset.

`cat /data/misc/gsensor_offset.txt`

```
root@tiny4412:/ # cat /data/misc/gsensor_offset.txt
cat /data/misc/gsensor_offset.txt
940 142 -412
```

- `/data/misc/gsensor_offset.log` : Record each calibration offset and calibration time.

`cat /data/misc/gsensor_offset.log`

```
root@tiny4412:/ # cat /data/misc/gsensor_offset.log
cat /data/misc/gsensor_offset.log
940 142 -412 wed Jan 1 12:56:28 GMT 2014
0 0 0 wed Jan 1 12:45:47 GMT 2014
938 145 -410 wed Jan 1 12:43:30 GMT 2014
```

- `/data/misc/reg_map.txt`: The current value of the records of all registers.

path fixed !!! `/data/misc/reg_map.txt`

cat /data/misc/reg\_map.txt

Note : path fixed !!!save offset to file **"/data/misc/reg\_map.txt"**

```
root@panda:/ # cat /data/misc/sensor/reg_map.txt
Read Gma301 Register MAP
REG(0x00) 0x06 0x00 0x00 0x00 0x00 0x00
REG(0x0C) 0x8F 0x74 0x00 0x00 0x00 0x07
REG(0x0D) 0x74 0x00 0x00 0x00 0x07 0x55
REG(0x0E) 0x00 0x00 0x00 0x07 0x55 0xC0
REG(0x0F) 0x00 0x00 0x07 0x55 0xC0 0x08
REG(0x11) 0x07 0x55 0xC0 0x08 0x00 0x09
REG(0x12) 0x55 0xC0 0x06 0x00 0x09 0x00
REG(0x13) 0xC0 0x06 0x00 0x09 0x00 0xBC
REG(0x14) 0x06 0x00 0x09 0x00 0xBC 0xFF
REG(0x15) 0x00 0x09 0x00 0xBC 0xFF 0xA4
REG(0x16) 0x09 0x00 0xBC 0xFF 0xA4 0x00
REG(0x17) 0x00 0xBC 0xFF 0xA4 0x00 0x55
REG(0x18) 0xBC 0xFF 0xA4 0x00 0x55 0xC0
REG(0x19) 0xFF 0xA3 0x00 0x55 0xC0 0x08
REG(0x1A) 0xA4 0x00 0x55 0xC0 0x08 0x00
REG(0x1B) 0x00 0x55 0xC0 0x08 0x00 0x09
REG(0x1C) 0xAA 0x40 0x00 0x06 0x00 0x09
REG(0x1F) 0x48 0x48 0x48 0x48 0x48 0x48
REG(0x21) 0x50 0x50 0x50 0x50 0x50 0x50
```

## 4. Supplement

### 4.1 Zero-G introduction.

Check the file " /sys/class/input/inputX/calibration" exists.

If the file exists on the use of the above path to zero-G.

else :Use ioctl to calibrate gsensor.(/system/bin/gmad)

Code reference ....

```
=====
if [ -f "$inputpath/$ATTR_CALIB" ]; then
    #echo file calibration path was $inputpath/$ATTR_CALIB
    break;
else
    echo did not find $inputpath/$ATTR_CALIB
    inputpath=NULL
    echo "The second calibration method. Run EXE"
    $RUN_CALIB -c $2

```

### 4.2 自動校準方向的調整

如果第一次開機是反向(Z 軸與螢幕朝上方向差 180 度),可以調整如下 AutoZeroZ 部份的

If (position > 0 )

```
#if AutoZeroZ
//if(1)ABS(gma->accel_data.y.x) < LevelValueRange_2_0 && ABS(gma->accel_dat
//2015_0619 modified.
int position = atomic_read(&gma->position);
if (position > 0)
{gma_acc_calibration(gma, GRAVITY_ON_Z_POSITIVE);}
else
{gma_acc_calibration(gma, GRAVITY_ON_Z_NEGATIVE);}
GMA_WriteCalibration(gma , GMA_Offset_TXT);
#endif

```

### 4.3 Glevel.apk 手動校準 Z 軸方向調整

(只考慮螢幕朝上,校正後水平擺放 z=+9.81, 180 度反面 z 要趨近於-9.81)

修改/system/bin/gss.sh

```
"calib")
    if[-f"$inputpath/$ATTR_POSITION"];then
        layout=`cat$DIR_INPUT/$inputpath/$ATTR_POSITION`
        if["$layout"-lt0];then
            position=2
        else
            position=1
        fi
    fi

```

If apk z-axis 方向校準錯誤要調整就改為

```
if [ "$layout" -lt 0 ] ; then
    position=1
else
    position=2
fi
```

PS.為何需要調整,因為 position 沒有正確,所以底層驅動的 Layout information 沒有正確的調整

```
pdata = client->dev.platform_data;
```

```
if (pdata) {
    /* Platform data is available. copy its value to local. */
    position = atomic_set(&s_gma->position, pdata->layout);
    s_gma->rstn = pdata->gpio_RSTN;
}
else {
    /*      Platform data is not available.
           Layout information should be set by each application.
           Acceleration Sensor Mounting Position on Board */
    position = atomic_set(&s_gma->position, GMS_GMA30x_DEFAULT_POSITION);
    s_gma->rstn = 0;
}
```

如果 chip mounting position 要正確調整,請參閱驅動

/\* Transformation matrix for chip mounting position

first pin top:Positive bottom:Negative

1: top/upper-left	(-y,-x,-z)
2: top/upper-right	(-x, y,-z)
3: top/lower-right	( y,-x,-z)
4: top/lower-left	( x,-y,-z)
-2: bottom/upper-left	\$Equal position 2 ( x, y, z)
-1: bottom/upper-right	\$Equal position 1 ( y, -x, z)
-4: bottom/lower-right	\$Equal position 4 (-x, -y, z)
-3: bottom/lower-left	\$Equal position 3 (-y, -x, z)