wangcq327 20160406

3轴Gsensor的输入系统代码

输入设备初始化:

idev = input\_allocate\_device();

idev->name = MIR3DA\_INPUT\_DEV\_NAME;

idev->evbit[0] = BIT\_MASK(EV\_ABS);

**input\_set\_abs\_params(idev, ABS\_X, -2048, 2048, 0, 0);**

**input\_set\_abs\_params(idev, ABS\_Y, -2048, 2048, 0, 0);**

**input\_set\_abs\_params(idev, ABS\_Z, -2048, 2048, 0, 0);**

input\_register\_device(idev);

上报3轴数据:

**input\_report\_abs(mir3da->input, ABS\_X, x);**

**input\_report\_abs(mir3da->input, ABS\_Y, y);**

**input\_report\_abs(mir3da->input, ABS\_Z, -z);**

input\_sync(mir3da->input);

光感+距离感 传感器

输入设备初始化:

input\_dev = input\_allocate\_device();

input\_dev->name = LTR558\_INPUT\_DEV;

input\_dev->phys = LTR558\_INPUT\_DEV;

input\_dev->id.bustype = BUS\_I2C;

input\_dev->dev.parent = &client->dev;

input\_dev->id.vendor = 0x0001;

input\_dev->id.product = 0x0001;

input\_dev->id.version = 0x0010;

ltr\_558als->input = input\_dev;

\_\_set\_bit(EV\_ABS, input\_dev->evbit);

**input\_set\_abs\_params(input\_dev, ABS\_DISTANCE, 0, 1, 0, 0);**

**input\_set\_abs\_params(input\_dev, ABS\_MISC, 0, 100001, 0, 0);**

input\_register\_device(input\_dev);

//PS数据上报

measured\_val = ltr558\_i2c\_read\_2\_bytes(LTR558\_PS\_DATA\_0);

**input\_report\_abs(pls->input, ABS\_DISTANCE, measured\_val);**

input\_sync(pls->input);

//ALS数据上报

measured\_val = ltr558\_als\_read(l\_gainrange);

**input\_report\_abs(pls->input, ABS\_MISC, measured\_val);**

input\_sync(pls->input);

触摸屏的驱动分析

**项目中的实例:**

**初始化:**

tpd->dev = input\_allocate\_device();

tpd->dev->name = TPD\_DEVICE;

**set\_bit(EV\_KEY, tpd->dev->evbit);**

set\_bit(BTN\_TOUCH, tpd->dev->keybit);

**set\_bit(EV\_ABS, tpd->dev->evbit);**

set\_bit(**ABS\_X**, tpd->dev->absbit);

set\_bit(**ABS\_Y**, tpd->dev->absbit);

input\_abs\_set\_res(tpd->dev, **ABS\_X**, TPD\_RES\_X);

input\_abs\_set\_res(tpd->dev, **ABS\_Y**, TPD\_RES\_Y);

input\_set\_abs\_params(tpd->dev, **ABS\_X**, 0, TPD\_RES\_X, 0, 0);

input\_set\_abs\_params(tpd->dev, **ABS\_Y**, 0, TPD\_RES\_Y, 0, 0);

set\_bit(**ABS\_PRESSURE**, tpd->dev->absbit);

input\_set\_abs\_params(tpd->dev, **ABS\_PRESSURE**, 0, 255, 0, 0);

set\_bit(**ABS\_MT\_TOUCH\_MAJOR**, tpd->dev->absbit);

set\_bit(**ABS\_MT\_TOUCH\_MINOR**, tpd->dev->absbit);

input\_set\_abs\_params(tpd->dev, **ABS\_MT\_TOUCH\_MAJOR**, 0, 100, 0, 0);

input\_set\_abs\_params(tpd->dev, **ABS\_MT\_TOUCH\_MINOR**, 0, 100, 0, 0);

set\_bit(**ABS\_MT\_TRACKING\_ID**, tpd->dev->absbit);

input\_set\_abs\_params(tpd->dev, **ABS\_MT\_TRACKING\_ID**, 0, 10, 0, 0);

set\_bit(**ABS\_MT\_POSITION\_X**, tpd->dev->absbit);

set\_bit(**ABS\_MT\_POSITION\_Y**, tpd->dev->absbit);

input\_set\_abs\_params(tpd->dev, **ABS\_MT\_POSITION\_X**, 0, TPD\_RES\_X, 0, 0);

input\_set\_abs\_params(tpd->dev, **ABS\_MT\_POSITION\_Y**, 0, TPD\_RES\_Y, 0, 0);

set\_bit(INPUT\_PROP\_DIRECT, tpd->dev->propbit);

input\_register\_device(tpd->dev)

**调用:**

if (cinfo.count > 0) {

for (i = 0; i < cinfo.count; i++)

tpd\_down(cinfo.x[i], cinfo.y[i], i + 1, cinfo.id[i]);

} else {

tpd\_up(cinfo.x[0], cinfo.y[0]);

}

input\_sync(tpd->dev);

static void tpd\_down(int x, int y, int p, int id)

{

input\_report\_abs(tpd->dev, **ABS\_MT\_TRACKING\_ID**, id);

input\_report\_key(tpd->dev, **BTN\_TOUCH**, 1);

input\_report\_abs(tpd->dev, **ABS\_MT\_TOUCH\_MAJOR**, 1);

input\_report\_abs(tpd->dev, **ABS\_MT\_POSITION\_X**, x);

input\_report\_abs(tpd->dev, **ABS\_MT\_POSITION\_Y**, y);

input\_mt\_sync(tpd->dev);

}

static void tpd\_up(int x, int y)

{

TPD\_DEBUG("%s x:%d y:%d\n", \_\_func\_\_, x, y);

input\_report\_key(tpd->dev, **BTN\_TOUCH**, 0);

input\_mt\_sync(tpd->dev);

}

**网络上的实例1**

<http://blog.csdn.net/wh_19910525/article/details/10828025>

我们知道每次触摸完成后都必须发送一个同步事件（EV\_SYN）来表明这次触摸的完成。 那么对于多点触控的屏幕事件发送分为两种方法，一是每次事件同步前包括多个点，一是每次事件同步前仅包含一个点。

**先来看包含多个点的**

**[cpp]** [view plaincopy](http://blog.csdn.net/dkleikesa/article/details/9415023)

1. **static** **void** ft5x0x\_report\_value(**struct** ft5x0x\_ts\_data \*data)
2. {
3. **struct** ts\_event \*event = &data->event;
4. **int** i;
5. **int** uppoint = 0;    //已经抬起的点数
7. **for** (i = 0; i < event->touch\_point; i++)  //循环处理 缓存中的所有点
8. {
9. input\_mt\_slot(data->input\_dev, event->au8\_finger\_id[i]);  //发送点的ID
11. **if** (event->au8\_touch\_event[i]== 0 || event->au8\_touch\_event[i] == 2)  //如果点按下
12. {
13. input\_mt\_report\_slot\_state(data->input\_dev, MT\_TOOL\_FINGER,  **true**);  //手指按下
14. input\_report\_abs(data->input\_dev,ABS\_MT\_POSITION\_X,event->au16\_x[i]); //x坐标
15. input\_report\_abs(data->input\_dev, ABS\_MT\_POSITION\_Y,event->au16\_y[i]);    //y坐标
16. input\_report\_abs(data->input\_dev,ABS\_MT\_TOUCH\_MAJOR,event->pressure); //触摸点长轴长度
17. }
18. **else**
19. {
20. uppoint++;                              //没有按下，则表明这个手指已经抬起
21. input\_mt\_report\_slot\_state(data->input\_dev, MT\_TOOL\_FINGER,**false**);   //报告手指抬起
22. }

25. }
27. **if**(event->touch\_point == uppoint)
28. {
29. input\_report\_key(data->input\_dev, BTN\_TOUCH, 0); //所有手指都抬起了 发送BTN\_TOUCH 抬起事件
31. }
32. **else**
33. {
34. input\_report\_key(data->input\_dev, BTN\_TOUCH, event->touch\_point > 0);//还有手指没抬起，发送BTN\_TOUCH 按下的事件
36. }
37. input\_sync(data->input\_dev); //sync 设备同步
39. }

**然后是每次同步仅发送一个点**

**[cpp]** [view plaincopy](http://blog.csdn.net/dkleikesa/article/details/9415023)

1. **static** ft5x0x\_report\_value(**struct** ft5x0x\_ts\_data \*data)
3. {
4. **struct** ts\_event \*event = &data->event;
5. **int** i;
7. **for** (i = 0; i < event->touch\_point; i++)  //循环处理 缓存中的所有点
8. {
9. input\_mt\_slot(data->input\_dev, event->au8\_finger\_id[i]);  //发送点的ID
10. **if** (event->au8\_touch\_event[i]== 0 || event->au8\_touch\_event[i] == 2)  //如果点按下
11. {
12. input\_mt\_report\_slot\_state(data->input\_dev, MT\_TOOL\_FINGER,  **true**);  //手指按下
13. input\_report\_abs(data->input\_dev,ABS\_MT\_POSITION\_X,event->au16\_x[i]); //x坐标
14. input\_report\_abs(data->input\_dev, ABS\_MT\_POSITION\_Y,event->au16\_y[i]);    //y坐标
15. input\_report\_abs(data->input\_dev,ABS\_MT\_TOUCH\_MAJOR,event->pressure); //触摸点长轴长度
16. }
17. **else**
18. {
19. input\_mt\_report\_slot\_state(data->input\_dev, MT\_TOOL\_FINGER,**false**);   //手指抬起
20. }

23. input\_mt\_report\_pointer\_emulation(input\_dev, **true**);//用模拟点的方法，来告知此次触摸已经完成。
24. input\_sync(data->input\_dev); //sync 设备同步
26. }
27. }

这两种方法都可以，但是建议选择上面那种，效率比较高。

**网络上的实例2**

typeA兼容typeB协议初始化:

input\_dev = input\_allocate\_device();

input\_dev->name = FOCALTECH\_TS\_NAME;

**input\_mt\_init\_slots(input\_dev, TS\_MAX\_FINGER,0);**

//支持KEY

set\_bit(EV\_KEY, input\_dev->evbit);

\_\_set\_bit(BTN\_TOUCH, input\_dev->keybit);

\_\_set\_bit(KEY\_MENU, input\_dev->keybit);

\_\_set\_bit(KEY\_BACK, input\_dev->keybit);

\_\_set\_bit(KEY\_HOMEPAGE, input\_dev->keybit);

//支持ABS

set\_bit(EV\_ABS, input\_dev->evbit);

\_\_set\_bit(ABS\_MT\_TOUCH\_MAJOR, input\_dev->absbit);

\_\_set\_bit(ABS\_MT\_POSITION\_X, input\_dev->absbit);

\_\_set\_bit(ABS\_MT\_POSITION\_Y, input\_dev->absbit);

\_\_set\_bit(ABS\_MT\_WIDTH\_MAJOR, input\_dev->absbit);

input\_set\_abs\_params(input\_dev,ABS\_MT\_POSITION\_X, 0, pdata->TP\_MAX\_X, 0, 0);

input\_set\_abs\_params(input\_dev,ABS\_MT\_POSITION\_Y, 0, pdata->TP\_MAX\_Y, 0, 0);

input\_set\_abs\_params(input\_dev,ABS\_MT\_TOUCH\_MAJOR, 0, 255, 0, 0);

input\_set\_abs\_params(input\_dev,ABS\_MT\_WIDTH\_MAJOR, 0, 255, 0, 0);

input\_register\_device(input\_dev);

typeA方式上报数据:

if(finger\_down){

for(i = 0; i < TS\_MAX\_FINGER; i++) {

input\_report\_abs(ft5x0x\_ts->input\_dev, ABS\_MT\_POSITION\_X, x);

input\_report\_abs(ft5x0x\_ts->input\_dev, ABS\_MT\_POSITION\_Y, y);

input\_report\_key(ft5x0x\_ts->input\_dev, BTN\_TOUCH, 1);

input\_mt\_sync(ft5x0x\_ts->input\_dev);

}

input\_sync(ft5x0x\_ts->input\_dev);

}else if(finger\_up){

input\_report\_key(ft5x0x\_ts->input\_dev, BTN\_TOUCH, 0);

input\_mt\_sync(ft5x0x\_ts->input\_dev);

input\_sync(ft5x0x\_ts->input\_dev);

}

adb shell getevent -lt /dev/input/event0

ABS\_MT\_POSITION\_X x[0]   
ABS\_MT\_POSITION\_Y y[0]

BTN\_TOUCH DOWN  
SYN\_MT\_REPORT   
ABS\_MT\_POSITION\_X x[1]   
ABS\_MT\_POSITION\_Y y[1]   
SYN\_MT\_REPORT   
…   
SYN\_REPORT

手抬起的时候是如下样子：

BTN\_TOUCH UP   
SYN\_MT\_REPORT   
SYN\_REPORT

typeB方式上报数据:

for(i = 0; i < TS\_MAX\_FINGER; i++) {

…//计算xy坐标及id,触摸的状态

input\_mt\_slot(data->input\_dev, buf[6\*i+5]>>4);

if(finger\_down){

input\_mt\_report\_slot\_state(input\_dev, MT\_TOOL\_FINGER, true);

input\_report\_abs(input\_dev, ABS\_MT\_POSITION\_X, x);

input\_report\_abs(input\_dev, ABS\_MT\_POSITION\_Y, y);

input\_report\_key(input\_dev, BTN\_TOUCH, 1);

}else if(finger\_up){

input\_report\_key(input\_dev, BTN\_TOUCH, 0);

input\_mt\_report\_slot\_state(input\_dev, MT\_TOOL\_FINGER, false);

}

}

input\_mt\_report\_pointer\_emulation(input\_dev,false);

input\_sync(ft5x0x\_ts->input\_dev);

adb shell getevent -lt /dev/input/event0

两个点同时按下抬起

[     208.775409] EV\_ABS       ABS\_MT\_SLOT          00000000 多点时标示属于哪个slot   标示slot 0   
[     208.775462] EV\_ABS       ABS\_MT\_TRACKING\_ID   0000000e 分配ID   
[     208.775484] EV\_ABS       ABS\_MT\_POSITION\_X 00000215   
[     208.775498] EV\_ABS       ABS\_MT\_POSITION\_Y 0000037c   
[     208.775533] EV\_SYN       SYN\_REPORT 00000000

[     208.859516] EV\_ABS       ABS\_MT\_SLOT          00000001 标示slot 1   
[     208.859534] EV\_ABS       ABS\_MT\_TRACKING\_ID   0000000f 分配ID 上一 次ID +1   
[     208.859538] EV\_ABS       ABS\_MT\_POSITION\_X 000000e6   
[     208.859541] EV\_ABS       ABS\_MT\_POSITION\_Y 0000031f   
[     208.859550] EV\_SYN       SYN\_REPORT 00000000

[     208.873597] EV\_ABS       ABS\_MT\_SLOT 00000000   
[     208.873637] EV\_ABS       ABS\_MT\_TRACKING\_ID   ffffffff 释放slot0 的ID   
[     208.873659] EV\_ABS       ABS\_MT\_SLOT 00000001   
[     208.873667] EV\_ABS       ABS\_MT\_TRACKING\_ID   ffffffff 释放slot1 的ID   
[     208.873688] EV\_SYN       SYN\_REPORT           00000000