



¥ 99.00

## android 电容屏（四）：驱动调试之驱动程序分析篇 -- FocalTech

转载 2013年09月01日 15:56:42

13054



1

本人用的触摸屏IC是FocalTech公司的ft5306，是一款i2c的电容屏多点触控芯片。对于它的整体驱动官方已经给了，我们就触摸屏和按键部分的代码做相关说明。说明其中应该注意的地方。

对于所有的input设备，报告input事件时候都分这么几部分，首先在probe文件中设置设备发送的事件类型、按键类型、设置设备一些属性信息。然后在发送事件时候要根据probe的设置来发送事件，否则就会被判为无效忽略掉。

### 一、触摸屏部分

#### 1.设备配置

对于触摸屏，必须支持的事件类型有以下这么三个

`_set_bit(EV_SYN, input_dev->evbit);` //设备同步，每次触摸完成以后都要发送一个同步事件，来表明这次触摸已经完成

`_set_bit(EV_ABS, input_dev->evbit);` //绝对坐标事件，触摸屏每次发送的坐标都是绝对坐标，不同于鼠标的相对坐标

`_set_bit(EV_KEY, input_dev->evbit);` //按键事件，每次触摸都有一个BTN\_TOUCH的按键事件

触摸屏必须支持的按键类型

`_set_bit(BTN_TOUCH, input_dev->keybit);` //touch类型按键

触摸屏属性设置

`input_mt_init_slots(input_dev, CFG_MAX_TOUCH_POINTS);` //报告最大支持的点数

`input_set_abs_params(input_dev, ABS_MT_TOUCH_MAJOR, 0, PRESS_MAX, 0, 0);` //将触摸点看成一个椭圆，它的长轴长度。这个是可选项，并不影响正常使用。

`input_set_abs_params(input_dev, ABS_MT_POSITION_X, 0, ft5x0x_ts->x_max, 0, 0);` //x坐标取值范围

`input_set_abs_params(input_dev, ABS_MT_POSITION_Y, 0, ft5x0x_ts->y_max, 0, 0);` //y坐标取值范围

#### 2.事件发送

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]
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
6.
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.
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.         }
23.
24.     }
25.
26.     if(event->touch_point == uppoint)
27.     {
28.         // ... (rest of the function code) ...
```

### 联系我们



请扫描二维码联系

webmaster@

400-660-011

QQ客服

关于 招聘 广告服务

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

¥ 99.00

### 他的最新文章

git describe

查看linux系统重启之前的log -- g

Git中tag的用法

关于mtk GPIO口的定制

MTK GPIO操作

### 文章分类

android-源码开发

android之Preference

Wifi

C/C++技术

linux

嵌入式技术

展开

### 文章存档

2017年11月

2017年9月

2017年8月

2017年7月

2017年6月

2017年5月

展开

```
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
30.
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
35.     }
36.     input_sync(data->input_dev); //sync 设备同步
37.
38. }
39. }
```

1

1

然后是每次同步仅发送一个点

```
[cpp]
1. static ft5x0x_report_value(struct ft5x0x_ts_data *data)
2.
3. {
4.     struct ts_event *event = &data->event;
5.     int i;
6.
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.        }
21.
22.
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
25.
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。

## 二、触摸按键部分

对于触摸按键的发送可以分为两种方法，一是android提供的 virtualkey's 架构方法，一种是直接报告key event的方法。我们——来看

### 1.报告key event方法

在probe中添加所支持的按键类型，本人用的触摸屏上有三个按键因此

报告支持事件类型

```
_set_bit(EV_SYN, input_dev->evbit);
```

```
_set_bit(EV_KEY, input_dev->evbit);
```

报告支持的按键

```
_set_bit(KEY_HOME, input_dev->keybit);
```

```
_set_bit(KEY_BACK, input_dev->keybit);
```

```
_set_bit(KEY_MENU, input_dev->keybit);
```

触摸屏上的三个按键对应的坐标

(KEY\_BACK) 120:1400 (KEY\_HOME) 360:1400(KEY\_MENU) 500:1400

key event的报告方法很简单只要报告相应的key 和设备同步sync就可以了

```
[cpp]
1. static void ft5x0x_report_value(struct ft5x0x_ts_data *data)
2. {
```

JD.COM 京东



¥99.00

的处理

联系我们



请扫描二维码联系  
webmaster@csdn.com  
400-660-0111  
QQ客服

关于 招聘 广告服务

©1999-2018 CSDN版权所有  
京ICP证09002463号

经营性网站备案信息  
网络110报警服务  
中国互联网举报中心  
北京互联网违法和不良信息举报中心



1



```

3. struct ts_event *event = &data->event;
4. int i;
5. for (i = 0; i < event->touch_point; i++)
6. {
7.     if (event->au16_y[i]==1400)
8.     {
9.         if(event->au8_touch_event[i]== 0 || event->au8_touch_event[i] == 2)
10.        {
11.
12.            switch(event->au16_x[i])
13.            {
14.                case 120:
15.                    input_report_key(data->input_dev, KEY_BACK, 1);
16.                    break;
17.                case 360:
18.                    input_report_key(data->input_dev, KEY_HOME, 1);
19.                    break;
20.                case 500:
21.                    input_report_key(data->input_dev, KEY_MENU, 1);
22.                    break;
23.                default: break;
24.            }
25.
26.        }
27.        else
28.        {
29.
30.            switch(event->au16_x[i])
31.            {
32.                case 120:
33.                    input_report_key(data->input_dev, KEY_BACK, 0);
34.                    break;
35.                case 360:
36.                    input_report_key(data->input_dev, KEY_HOME, 0);
37.                    break;
38.                case 500:
39.                    input_report_key(data->input_dev, KEY_MENU, 0);
40.                    break;
41.                default: break;
42.            }
43.
44.        }
45.        input_sync(data->input_dev);
46.        return;
47.    }
48. }
49.

```

对于这种方法有一个bug，就是事件发送上去，系统并不认为是触摸屏发送的按键，系统的 触屏震动反馈 并不起作用。这并不符合标准的android触摸设备标准。具体怎么破本人比较菜没有找到方法，大神们谁知道 求破。

## 2.virtualkeys方法

virtualkeys是android提供的架构使用起来简单方便，推荐大家使用。直接上代码

```

[cpp]
1. static ssize_t ft5x06_virtual_keys_show(struct kobject *kobj,          //按键的配置
2.                                         struct kobj_attribute *attr, char *buf)
3. {
4.     return sprintf(buf,
5.         __stringify(EV_KEY) ":" __stringify(KEY_BACK) ":120:1400:8:8" //键类型:键值:按键区域中心x坐标:按键区
域中心y坐标:按键区域宽:按键区域高
6.         ":" __stringify(EV_KEY) ":"
7.         __stringify(KEY_HOME) ":360:1400:8:8"
8.         ":" __stringify(EV_KEY) ":"
9.         __stringify(KEY_MENU) ":500:1400:8:8"
10.        "\n");
11. }
12.
13. static struct kobj_attribute ft5x06_virtual_keys_attr = {
14.     .attr = {
15.         .name = "virtualkeys.Ft5x0x_Touch_Screen", //这里的名字必须为virtualkeys.设备名字 否则系统不会识别
16.         .mode = S_IRUGO,
17.     },
18.     .show = &ft5x06_virtual_keys_show,
19. };
20.
21. static struct attribute *ft5x06_properties_attrs[] = {
22.     &ft5x06_virtual_keys_attr.attr,
23.     NULL,
24. };
25.
26. static struct attribute_group ft5x06_properties_attr_group = {
27.     .attrs = ft5x06_properties_attrs,

```



## 联系我们



请扫描二维码联系

webmaster@

400-660-011

QQ客服

关于 招聘 广告服务

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

```

28.     };
29.
30.     static void ft5x06_virtual_keys_init(void)
31.     {
32.         struct kobject *properties_kobj;
33.         int ret;
34.
35.         properties_kobj = kobject_create_and_add("board_properties", NULL); //添加目录board_properties
36.
37.         if (properties_kobj)
38.             ret = sysfs_create_group(properties_kobj, //生成/sys/board_properties/virtualkeys.Ft5x0x_Touch_Screen
虚拟按键配置文件
&ft5x06_properties_attr_group); //可以使用 cat /sys/board_properties/virtualkeys.Ft5x0x_Touch_Sc
reen命令来查看配置是否正确
39.
40.         if (!properties_kobj || ret)
41.             pr_err("failed to create board_properties\n");
42.     }

```

然后将ft5x06\_virtual\_keys\_init()加入到 触摸屏的init 或者probe 函数中，这样触摸键就可以使用了。

### 三、触摸屏驱动流程

### i2c中加入平台初始化代码

```
[cpp]
1. static struct ft5x0x_platform_data ft5x0x_platform_i2c_data = {
2.     .x_max=540,
3.     .y_max=960,
4.     .irq= SABRESD_CHARGE_FLT_1_B,<span style="white-space:pre"> </span> //中断引脚
5.     .reset=SABRESD_DISP0_RST_B,<span style="white-space:pre"> </span> //复位引脚
6. };
```

## 触摸屏驱动初始化

```
[cpp]
1. static int __init ft5x0x_ts_init(void)
2. {
3.     int ret;
4.     ret = i2c_add_driver(&ft5x0x_ts_driver);
5.     if (ret) {
6.         printk(KERN_WARNING "Adding ft5x0x driver failed "
7.             "(errno = %d)\n", ret);
8.     } else {
9.         pr_info("Successfully added driver %s\n",
10.             ft5x0x_ts_driver.driver.name);
11.     }
12.     return ret;
13. }
```

## probe函数

```

1. #define VIRTUAL_LI      0
2. #define EVENT_LI        1
3. #define TOUCH_KEY        VIRTUAL_LI
4. static int ft5x0x_ts_probe(struct i2c_client *client,
5.                             const struct i2c_device_id *id)
6. {
7.     . . . . .
8.     ft5x0x_ts = kzalloc(sizeof(struct ft5x0x_ts_data), GFP_KERNEL); //分配参数内存
9.
10.    . . . . .
11.    i2c_set_clientdata(client, ft5x0x_ts); //参数地址传给i2c 内核
12.
13.    初始化一些参数
14.    ft5x0x_ts->irq = client->irq;
15.    ft5x0x_ts->client = client;
16.    ft5x0x_ts->pdata = pdata;
17.    ft5x0x_ts->x_max = pdata->x_max - 1;
18.    ft5x0x_ts->y_max = pdata->y_max - 1;
19.    ft5x0x_ts->pdata->reset = FT5X0X_RESET_PIN;
20.    ft5x0x_ts->pdata->irq = ft5x0x_ts->irq;
21.    . . . . .
22.    err = request_threaded_irq(client->irq, NULL, ft5x0x_ts_interrupt, //注册读取数据中断
23.                                IRQF_TRIGGER_FALLING, client->dev.driver->name,
24.                                ft5x0x_ts);
25.    . . . . .
26.    input dev = input_allocate_device(); //分配设备

```



联系我们



请扫描二维码联系

✉ webmaster@

 400-660-0111

QQ客服

关于 招聘 广告服务 

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

1

1

1

27. ....

28. \_\_set\_bit(EV\_SYN, input\_dev->evbit); //注册设备支持event类型

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

30. \_\_set\_bit(EV\_KEY, input\_dev->evbit);

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

32. #if TOUCH\_KEY == EVENT\_LI //如果使用event key的方法

33. \_\_set\_bit(KEY\_HOME, input\_dev->keybit);

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

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

36. #endif

37. input\_mt\_init\_slots(input\_dev, CFG\_MAX\_TOUCH\_POINTS); //设备属性

38. input\_set\_abs\_params(input\_dev, ABS\_MT\_TOUCH\_MAJOR,

39. 0, PRESS\_MAX, 0, 0);

40. input\_set\_abs\_params(input\_dev, ABS\_MT\_POSITION\_X,

41. 0, ft5x0x\_ts->x\_max, 0, 0);

42. input\_set\_abs\_params(input\_dev, ABS\_MT\_POSITION\_Y,

43. 0, ft5x0x\_ts->y\_max, 0, 0);

44.

45.

46.

47. input\_dev->name = "Ft5x0x\_Touch\_Screen";//lijianzhang

48. err = input\_register\_device(input\_dev); //注册这个input设备

49. ....

50. #if TOUCH\_KEY == VIRTUAL\_LI //如果使用虚拟键盘设定

51.

52. ft5x06\_virtual\_keys\_init();

53. #endif

54. ....

55. }

中断处理

1. [cpp]

2. static irqreturn\_t ft5x0x\_ts\_interrupt(int irq, void \*dev\_id)

3. {

4. struct ft5x0x\_ts\_data \*ft5x0x\_ts = dev\_id;

5. int ret = 0;

6. disable\_irq\_nosync(ft5x0x\_ts->irq);

7. ret = ft5x0x\_read\_Touchdata(ft5x0x\_ts); //读取数据

8. if (ret == 0)

9. ft5x0x\_report\_value(ft5x0x\_ts); //报告数据

10.

11. enable\_irq(ft5x0x\_ts->irq);

12. return IRQ\_HANDLED;

13. }

报告事件

1. [cpp]

2. static void ft5x0x\_report\_value(struct ft5x0x\_ts\_data \*data)

3. {

4. struct ts\_event \*event = &data->event;

5. int i;

6. int uppoint = 0;

7.

8. /\*protocol B\*/

9. for (i = 0; i < event->touch\_point; i++)

10. {

11. #if TOUCH\_KEY == EVENT\_LI //如果使用 key event方法

12. if (event->au16\_y[i]==1400)

13. {

14. if(event->au8\_touch\_event[i]== 0 || event->au8\_touch\_event[i] == 2)

15. {

16. switch(event->au16\_x[i])

17. {

18. case 120:

19. input\_report\_key(data->input\_dev, KEY\_BACK, 1);

20. break;

21. case 360:

22. input\_report\_key(data->input\_dev, KEY\_HOME, 1);

23. break;

24. case 500:

25. input\_report\_key(data->input\_dev, KEY\_MENU, 1);

26. break;

27. default: break;

28. }

29.



联系我们

请扫描二维码联系  
webmaster@  
400-660-011  
QQ客服

关于 招聘 广告服务

©1999-2018 CSDN版权所有  
京ICP证09002463号

经营性网站备案信息  
网络110报警服务  
中国互联网举报中心  
北京互联网违法和不良信息举报中心



```
30.     }
31.     else
32.     {
33.
34.         switch(event->au16_x[i])
35.         {
36.             case 120:
37.                 input_report_key(data->input_dev, KEY_BACK, 0);
38.                 break;
39.             case 360:
40.                 input_report_key(data->input_dev, KEY_HOME, 0);
41.                 break;
42.             case 500:
43.                 input_report_key(data->input_dev, KEY_MENU, 0);
44.                 break;
45.             default: break;
46.         }
47.         uppoint++;
48.
49.     }
50.     input_sync(data->input_dev);
51.     return;
52. }
53.
54. #endif
55.
56. input_mt_slot(data->input_dev, event->au8_finger_id[i]);
57.
58.
59. if (event->au8_touch_event[i]== 0 || event->au8_touch_event[i] == 2)
60. {
61.     input_mt_report_slot_state(data->input_dev, MT_TOOL_FINGER, true);
62.     input_report_abs(data->input_dev, ABS_MT_POSITION_X, event->au16_x[i]); //lijianzhang
63.     input_report_abs(data->input_dev, ABS_MT_POSITION_Y, event->au16_y[i]);
64.     input_report_abs(data->input_dev, ABS_MT_TOUCH_MAJOR, event->pressure);
65. }
66. else
67. {
68.     uppoint++;
69.     input_mt_report_slot_state(data->input_dev, MT_TOOL_FINGER, false);
70. }
71.
72. }
73.
74.
75. if(event->touch_point == uppoint)
76. {
77.     input_report_key(data->input_dev, BTN_TOUCH, 0);
78. }
79. else
80. {
81.     input_report_key(data->input_dev, BTN_TOUCH, event->touch_point > 0);
82. }
83. input_sync(data->input_dev);
84.
85. }
```



#### 联系我们



请扫描二维码联系

webmaster@

400-660-011

QQ客服

关于 招聘 广告服务

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

这里驱动流程做了简略的说明，关键的代码都已经贴出来了。与设备相关代码都是厂商给的没有太实际参考价值。

从android input的流程分析我们知道，驱动编译完成以后，要使触摸屏工作，还需要三个文件：触摸屏配置文件（idc文件，用来配置触摸屏的一些属性）、keylayout文件（kl文件，安卓层面的按键映射文件）、characterMap文件（kcm文件，安卓层面的字符映射文件）

我们——来看这三个文件

#### 1.触摸屏配置文件

文件所在目录访问顺序：

首先ANDROID\_ROOT/usr/idc目录下找相应名字的文件并返回完整的路径名，如果找不到就从ANDROID\_DATA/system/devices/idc下面去找，这里ANDROID\_ROOT一般指的是/system目录，ANDROID\_DATA一般指/data目录。

文件名称的查找顺序首先是Vendor\_XXXX\_Product\_XXXX\_Version\_XXXX.idc，然后是Vendor\_XXXX\_Product\_XXXX.idc最后是DEVICE\_NAME.idc

总结来看安卓为输入设备打开配置文件依次会访问

/system/usr/idc/Vendor\_XXXX\_Product\_XXXX\_Version\_XXXX.idc

```

/system/usr/idc/Vendor_XXXX_Product_XXXX.idc
/system/usr/idc/DEVICE_NAME.idc
/data/system/devices/idc/Vendor_XXXX_Product_XXXX_Version_XXXX.idc
/data/system/devices/idc/Vendor_XXXX_Product_XXXX.idc
/data/system/devices/idc/DEVICE_NAME.idc

```

👍 我们驱动里并没有写版本号等信息，因此我们设备访问的idc文件会是/system/usr/idc/DEVICE\_NAME.idc。因此我

1 们在这个目录下增加文件Ft5x0x\_Touch\_Screen.idc。对于idc文件的内容，下面是我使用的idc文件的具体内容，仅供参考

```
5  
[cpp]  
1. touch.deviceType = touchScreen  
2. touch.orientationAware = 1  
3.  
4. touch.size.calibration = none  
5. touch.orientation.calibration = none
```

## 2.key layout文件

key layout文件是android层面的按键映射文件，通过这个文件，用户可以对kernel发送上来的按键功能进行重新定义。也就是说，kernel发送上来一个home键，你可以在这里把它映射成一个back键或者其他的。一般情况下不会修改这个文件，因此我么完全可以使用默认的配置文件

这个文件访问顺序

```
/system/usr/keylayout/Vendor_XXXX_Product_XXXX_Version_XXXX.kl

/system/usr/keylayout/Vendor_XXXX_Product_XXXX.kl

/system/usr/keylayout/DEVICE_NAME.kl

/data/system/devices/keylayout/Vendor_XXXX_Product_XXXX_Version_XXXX.kl

/data/system/devices/keylayout/Vendor_XXXX_Product_XXXX.kl

/data/system/devices/keylayout/DEVICE_NAME.kl

/system/usr/keylayout/Generic.kl

/data/system/devices/keylayout/Generic.kl
```

这里不用修改因此不用做改变

### 3.characterMap文件

characterMap文件是android层面的字符映射文件，比如：你摁下了一个'e'键，平时代表'e'，shift+'e'代表'E'，cas  
plk+'e'代表'E'，alt+'e'可能代表别的意思，这个配置文件就是，做这些映射的。一般情况下这个文件也不用修改。使用  
默认的就是可以。这个文件的访问顺序：

```
/system/usr/keychars/Vendor_XXXX_Product_XXXX_Version_XXXX.kcm

/system/usr/keychars/Vendor_XXXX_Product_XXXX.kcm

/system/usr/keychars/DEVICE_NAME.kcm

/data/system/devices/keychars/Vendor_XXXX_Product_XXXX_Version_XXXX.kcm

/data/system/devices/keychars/Vendor_XXXX_Product_XXXX.kcm

/data/system/devices/keychars/DEVICE_NAME.kcm

/system/usr/keychars/Generic.kcm
```




联系我们



请扫描二维码联系

 webmaster@

 400-660-0100

QQ客服

关于 招聘 广告服务 

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心



```
data/system/devices/keychars/Generic.kcm

/system/usr/keychars/Virtual.kcm

'data/system/devices/keychars/Virtual.kcm
```

1

到了这里 我们的触摸屏已经完成了，烧写以后应该可以正常使用了。

在这里分享一个小技巧，getevent 这个工具，在/dev/input/目录下使用这个命令，会首先得到系统中所有input设备的描述，然后会得到，kernel发送的所有input事件，当我们写完驱动以后，可以用这个命令将发送的事件打印出来，看驱动写的是否正确。

写下你的评论...

qq\_15337501

2017-08-09 10:56

#4楼

楼主的研究比较透彻，我从事这一行业也获益匪浅。另有一个刁钻问题请教：TP的吸边问题有没有好的解决方法？

回复

a785984

2015-05-31 14:23

#3楼

修改DEVICE\_NAME.idc 文件，但是还是一直在触摸屏滑动模式

回复

cai521cai

2014-08-05 15:52

#2楼

感谢博主分享

回复

查看 4 条热评

多点触控协议(MTP)

linux

Tommy\_wxie

2013年07月04日 11:57

3625

Multi-touch (MT) Protocol ----- Copyright (C)2009-2010 Henrik Rydberg In...

linux input子系统 — TP A/B(Slot)协议

linux

u012719256

2016年12月13日 15:09

613

将A/B协议这部分单独拿出来一方面是因为这部分内容是比较容易忽视的，周围大多数用到input子系统的开发人员也不甚理解；另一方面是由于这部分知识一旦扩展到TP(触摸屏Touch Panel)的多点触...

你和高级程序员的差距，不只是年龄

1200+门实战课程，89元就能拿下，结果可能就会不一样啦

你不要搞事情



Android驱动之 Linux Input子系统之TP——A/B(Slot)协议

点击打开链接将A/B协议这部分单独拿出来一方面是因为这部分内容是比较容易忽视的，周围大多数用到input子系统的开发人员也不甚理解；另一方面是由于这部分知识一旦扩展到TP(触摸屏Touch Pane...

LoongEmbedded

2016年04月13日 14:25

2726

JD.COM 京东



¥ 99.00

联系我们



请扫描二维码联系  
webmaster@csdn.net  
400-660-0111  
QQ客服

关于 招聘 广告服务

©1999-2018 CSDN版权所有  
京ICP证09002463号

经营性网站备案信息  
网络110报警服务  
中国互联网举报中心  
北京互联网违法和不良信息举报中心

https://blog.csdn.net/wh\_19910525/article/details/10828025

8/10



android底层驱动学习之focaltech触屏实例理解

android底层驱动学习之focaltech触屏实例理解 这个星期接触了具体的驱动实例-----focaltech公司的触控IC。下面总结下我个人对于触屏驱动的理解,分为两...

happyguys12345 2016年12月21日 16:00 1087

ui界面设计

ui界面设计流程讲解

百度广告



【黑苹果教程】ELAN , FocalTech 以及 Synaptics 触控板驱动

触控板上的手势操作应该是macOS的一大特色，但是黑苹果装完之后，很多人发现别说手势，很多基本的功能触控板都不能实现，此时你需要安装触控板驱动。本文参照osxlatitude论坛的ELAN, Foca...

wr132 2017年01月30日 22:12 12672

focaltech ( 敦泰 ) 触摸屏驱动Ft5306.c学习记录 liujiaoyage 2014年06月13日 15:53 2960  
1 概述 linux触摸屏驱动基于

Focaltech ( 敦泰 ) 触屏通用MTK驱动

2016年09月14日 14:36 103KB 下载



Multi-touch (MT) Protocol 小结 tianruxishui 2012年01月03日 18:26 10612  
Android 中多点触摸协议： 参考： http://www.kernel.org/doc/Documentation/input/multi-touch-pr...

input输入子系统分析 Diwat 2013年09月16日 23:27 2602  
参考：国嵌教育和http://blog.csdn.net/lidaqiang99/article/details/6605700 一、输入子系统：input 1、首先我们要知道什么时候我们可以用到in...

8个按键的电容式触摸屏驱动程序

2009年12月04日 21:37 326KB 下载



5分钟实现软件限时限次授权管理

许可管理平台-深思云授权平台



android4.0触摸屏(touchscreen)以及屏幕按键驱动解析

本人用的触摸屏IC是FocalTech公司的ft5306，是一款i2c的电容屏多点触控芯片。对于它的整体驱动官方已经给了，我们就触摸屏和按键部分的代码做相关说明。说明其中应该注意的地方。...

dkleikesa 2013年07月23日 01:13 6854

Linux uinput驱动分析 mcgrady\_tracy 2014年04月04日 22:49 3119  
Linux版本：linux-3.13.3 uinput是Linux提供的一个可以在用户空间创建input设备的驱动程序，init部分代码如下： 877 static in t \_init uinp...

Goodix系列-----input device的init u013308744 2016年05月16日 15:25 689  
在probe函数里会有 static s8 gtp\_request\_input\_dev(struct goodix\_ts\_data \*ts){ s8 ret = -1; u8...

JD.COM 京东



¥ 99.00

联系我们



请扫描二维码联系  
webmaster@csdn.net  
400-660-0111  
QQ客服 4006600111

关于 招聘 广告服务 1999-2018 CSDN版权所有 京ICP证09002463号

经营性网站备案信息  
网络110报警服务  
中国互联网举报中心  
北京互联网违法和不良信息举报中心

### Input子系统 KeyPad-Touch上报数据格式与机制

----- 本文系本站原创,欢迎转载! 转载请注明出处:http...

android\_huber    2012年04月17日 14:48    9803

### nux输入子系统：多点触控协议 -- multi-touch-protocol.txt

1

Multi-touch (MT) Protocol ----- Copyright(C) 2009-2010 Henrik Rydberg...

DroidPhone    2012年12月25日 19:30    22528

### 一点点加盟费

一点点奶茶店加盟费要多少钱

百度广告

### Android 中多点触摸协议

http://blog.csdn.net/zuosifengli/article/details/7398661 Android 中多点触摸协议：参考...

wh\_19910525    2013年09月04日 12:29    4975

### ABS\_MT\_TOUCH\_MAJOR和ABS\_MT\_WIDTH\_MAJOR

通常的多点触摸周期可以描述为：ABS\_MT\_\*|SYN\_MT\_REPORT|ABS\_MT\_\*|SYN\_MT\_REPORT|SYN\_REPORT 有些ABS...

SUKHOI27SMK    2013年10月10日 00:39    1915

### ABS\_MT\_TOUCH\_MAJOR和ABS\_MT\_WIDTH\_MAJOR

通常的多点触摸周期可以描述为：ABS\_MT\_\*|SYN\_MT\_REPORT|ABS\_MT\_\*|SYN\_MT\_REPORT|SYN\_REPORT 有些ABS...

panzhun    2014年05月16日 11:20    252

### 设备树学习之（十三）电容触摸屏驱动

转载自：http://blog.csdn.net/lizuobin2/article/details/55264211 开发板：tiny4412SDK + S702 + 4GB F...

u013779722    2017年06月10日 07:31    660

### android 电容屏（三）：驱动调试之驱动程序分析篇

关键词：android 电容屏 tp 工作队列 中断 坐点计算 电容屏主要参数 平台信息: 内核：linux2.6/linux3.0 系统：android/andro  
id4.0 平台...

xubin341719    2012年08月05日 23:20    39506

JD.COM 京东

¥ 99.00

### 联系我们

请扫描二维码联系

webmaster@csdn.net

400-660-0111

QQ客服 1121780891

关于 招聘 广告服务

©1999-2018 CSDN版权所有  
京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

https://blog.csdn.net/wh\_19910525/article/details/10828025

10/10