**参考linux多点触控协议修改板卡测试**

注：

程序修改文件路径为/external/pcba-test/minuitwrp/event.cpp

修改函数名称为 static int vk\_modify(struct ev \*e, struct input\_event \*ev)

1. **板卡测试产生漂移**

**修改原因：**

由于现有的MT-207S板卡测试程序的编写，未考虑到linux多点触控B协议的协议内容，导致在第一个触点点击屏幕获得的X,Y坐标，会被第二个触点点击所修改，导致有可能上报的点X,Y的值会有三种情况，如下：

假设第一个触点获得的坐标为X1，Y1；第二个触点获得的坐标为X2，Y2；则上报的可能为：

1. 上报 X1，Y1; (X2，Y2的值完全与X1，Y1相同；)
2. 上报 X1，Y2; (X2与X1的值相同，Y2与Y1的值不同；)
3. 上报 X2，Y1; (Y2与Y1的值相同，X2与X1的值不同；)

以上的三种上报情况，在第一个触点点击后，再点击第二个触点时，屏幕会出现触摸的位置不在两个触摸点的任意一个点上，而偏向了X1，Y2或X2,Y1的位置，产生漂移。

**修改目的：**

为了解决MT-207S板卡测试程序产生的漂移，与多点触摸获得坐标的二义性，屏蔽除第一个点外的输入。

**修改依据：**

根据MT-207S机型所使用的是linux多点触控B协议，可跟据ABS\_MT\_SLOT来标识触摸，屏蔽不感兴趣的触摸。上报的协议如下：（只说明X,Y坐标）

ABS\_MT\_SLOT 0 //第一个触摸的SLOT标识

ABS\_MT\_TRACKING\_ID 0 //第一个触摸的TRACKING标识

ABS\_MT\_POSITION\_X x[0] //第一个触摸的X坐标

ABS\_MT\_POSITION\_Y y[0] //第一个触摸的Y坐标

SYN\_REPORT //若两个点同时按下时，不产生此同步信号，两个点按下时有先后顺序才会产生此同步信号。

ABS\_MT\_SLOT 1 //第二个触摸的SLOT标识

ABS\_MT\_TRACKING\_ID 1 //第二个触摸的TRACKING标识

ABS\_MT\_POSITION\_X x[1] //第二个触摸的X坐标

ABS\_MT\_POSITION\_Y y[1] //第二个触摸的Y坐标

SYN\_REPORT //对按下动作上报

已有的MT-207S板卡测试程序处理如下：（对X,Y坐标处理的程序片段）

case ABS\_MT\_POSITION\_X:

e->mt\_p.x = ev->value;

break;

case ABS\_MT\_POSITION\_Y:

e->mt\_p.y = ev->value;

break;

case ABS\_MT\_TRACKING\_ID:

if ( ev->value < 0)

{

touchReleaseOnNextSynReport = 2;

use\_tracking\_id\_negative\_as\_touch\_release = 1;

}

从程序中看，MT-207S板卡测试，只要有获得X,Y坐标的动作，就将已有的X或Y的值覆盖。导致了漂移现象的产生。

**修改位置与修改内容：**

1. 在变量定义处，新建一个标识limit\_modify,作用是标识谁能改X,Y坐标。

static int limit\_modify = 1;

1. 添加case ABS\_MT\_SLOT动作的判断，设定只有SLOT的值为0时标识limit\_modify才有效。

case ABS\_MT\_SLOT:

if (ev-value == 0)

{

limit\_modify = 1;

}

else if (ev-value > 0)

{

limit\_modify = 0;

}

1. 在ABS\_MT\_TRACKING\_ID动作的判断中。设定松开触点，且只有limit\_modify的值为1时,才执行赋值语句。

case ABS\_MT\_TRACKING\_ID:

if ( ev->value < 0)

{

if (limit\_modify)

{

touchReleaseOnNextSynReport = 2;

use\_tracking\_id\_negative\_as\_touch\_release = 1;

}

}

1. 在case ABS\_MT\_POSITION\_X 处，判断只有limit\_modify标识有效时，才允许更新X坐标的值，并设置已获得X坐标的旗标。

case ABS\_MT\_POSITION\_X:

if (limit\_modify == 1)

{

e->mt\_p.synced |= 0x01;

e->mt\_p.x = ev->value;

}

break;

1. 在case ABS\_MT\_POSITION\_Y 处，判断只有limit\_modify标识有效时，才允许更新Y坐标的值，并设置已获得Y坐标的旗标。

case ABS\_MT\_POSITION\_Y:

if (limit\_modify == 1)

{

e->mt\_p.synced |= 0x02;

e->mt\_p.y = ev->value;

}

break;

1. 在判断上报动作，当获得了X坐标，或Y坐标中任意一个时，就会调用vk\_tp\_to\_screen函数。

else if (e->mt\_p.synced &=0x03)

{

vk\_tp\_to\_screen(&e->mt\_p, &x, &y);

}