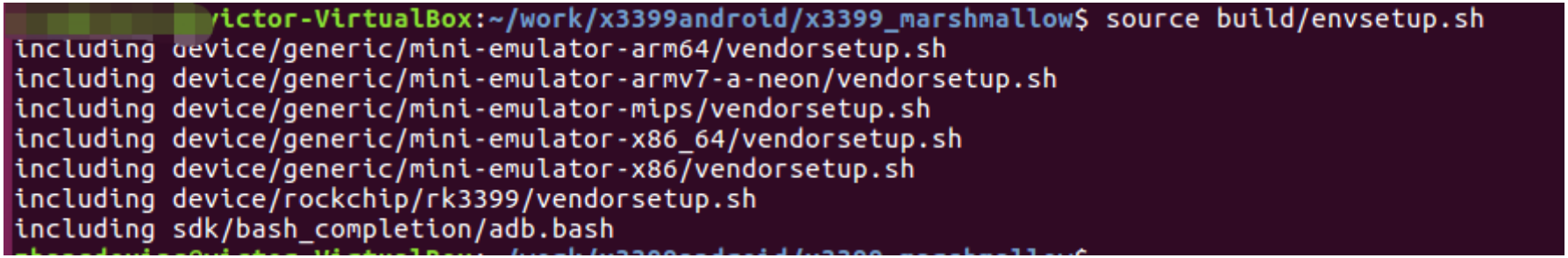
```
最近刚接触了电阻屏，跟电容屏不一样，电阻屏需要自行去校准，否则无法使用触摸功能。在网上查了一下关于Android下移植tslib库，发现资料很少，

我的环境：
平台：RK3399
Android版本6.0
触摸IC：TI-ADS7846
显示屏：ICN6202，分辨率：1024x768；
```

1：先拿到tslib库并适用于Android平台的，网上有一些版本，亲自使用了发现并不适用于Android。大家自己有可以用，本人用的库在下方地址：
下载Android-tslib:
https://download.csdn.net/download/qq_42233843/11131137

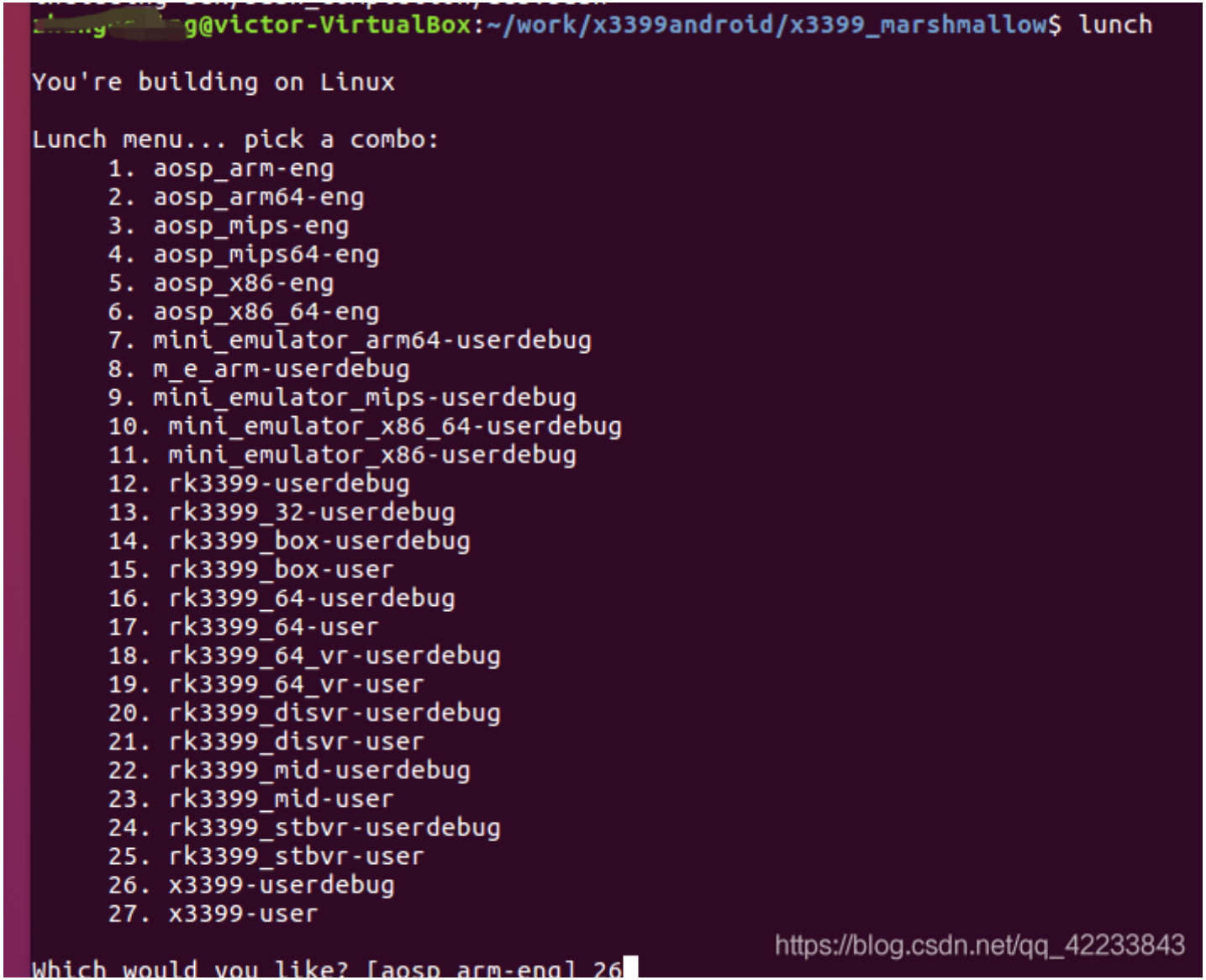
2，将tslib解压并放到平台下编译。
gunzip tslib.tar.gz
tar xvf tslib.tar
直接放在Android源码下即可。
当然首先先配一下Android环境为后面单独编译tslib做准备。
解压后cd tslib
./autogen.sh

./configure --host=arm-none-linux-gnueabi --prefix=+存放文件的绝对路径。
其实这两步的作用就是生成一个config.h的文件，否则后面mm编译步过。也不用太计较这两步的写法。对应Android下移植没什么多大用也不用make 和make install。Android移植tslib直接往下看

在源码下
source build/envsetup.sh



lunch + 你平台的选项 一般都是平台+userdebug



cd tslib目录

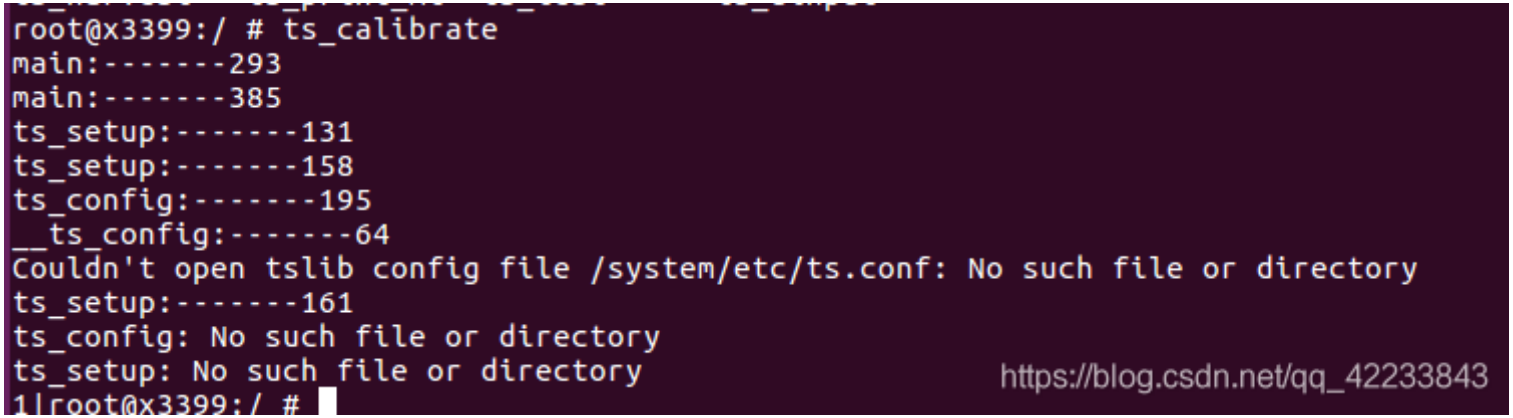
直接mm单独编译，不配上面的环境是不能直接mm的会报错。

```
Install: out/target/product/x3399/system/bin/ts_print_mt
target  C: ts_print_raw <= tslib/tests/ts_print_raw.c
target Executable: ts_print_raw (out/target/product/x3399/obj/EXECUTABLES/ts_print_ra
s/LINKED/ts_print_raw)
target Unpacked: ts_print_raw (out/target/product/x3399/obj/EXECUTABLES/ts_print_raw_
PACKED/ts_print_raw)
target Symbolic: ts_print_raw (out/target/product/x3399/symbols/system/bin/ts_print_r
target Strip: ts_print_raw (out/target/product/x3399/obj/EXECUTABLES/ts_print_raw_int
print_raw)
Install: out/target/product/x3399/system/bin/ts_print_raw
target  C: ts_uinput <= tslib/tools/ts_uinput.c
target Executable: ts_uinput (out/target/product/x3399/obj/EXECUTABLES/ts_uinput_inte
ED/ts_uinput)
target Unpacked: ts_uinput (out/target/product/x3399/obj/EXECUTABLES/ts_uinput_interm
/ts_uinput)
target Symbolic: ts_uinput (out/target/product/x3399/symbols/system/bin/ts_uinput)
target Strip: ts_uinput (out/target/product/x3399/obj/EXECUTABLES/ts_uinput_intermedi
)
Install: out/target/product/x3399/system/bin/ts_uinput
make: Leaving directory '/home/zhangdexing/work/x3399android/x3399_marshmallow'

#### make completed successfully (12 seconds) ####
```


编译完后，生成的可执行文件都放在out/target/product/平台/system/bin 库在lib64/ts/plugins下， 配置文件ts.conf在tslib下目录下。

现在可以编译你们的system及安卓的img，后烧录进你们的板子。

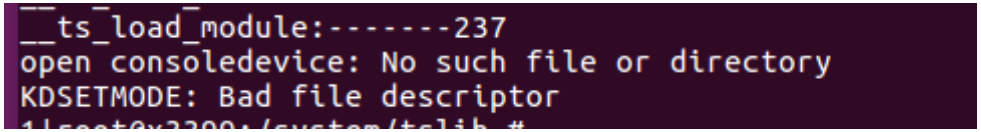
一开始以为这样就成功了，烧录完之后很开心，直接执行ts_calibrate
结果



有些打印是我自己加的，直接在源码加即可，这样方便调试，不然你后面都不知道自己卡在哪里。在tslib/tests/目录下打开你们的ts_calibrate.c文件，自己加一下打印跟踪问题。
上面就是一个典型的报错 找不到你的配置文件ts.conf。直接在out目录生成的ts.conf adb push进对应目录。

修改完后继续执行，都是要慢慢调试，不加打印失败都是会报
couldn’t load module input
no raw modules loaded的错，一开始鬼知道错在哪。最好在每个源码文件中#define DEBUG 因为tslib源码有一些打印虽然不够但是可以自己在加，跟踪后你会发现会联系到很多的.c文件。但是都不难理解。

现在又是另一个错，看报错就知道怎么改了吧。要么改一下环境变量的
TSLIB_PLUGINDIR找到你编译进去的库，都放在 /system/下，自己根据平台上找。
修改完后。
继续执行ts_calibrate。很多人觉得奇怪，为什么我没有配环境变量先，我的做法就是根据错误来配置。抱一个错就配哪个环境，不然很多教程都是先配置环境变量，我根本就不知道有什么用，甚至有些根本就不用配 配了反而不对，因为tslib库有默认帮我们配置一些。



简单易懂.

终端下输入: export TSLIB_CONSOLEDEVICE=none;
解决后继续

```
__ts_load_module:-------237
open fbdevice: No such file or directory
```

简单易懂.
终端下输入: export TSLIB_FBDEVICE=/dev/graphics/fb0;
这个是LCD的设备文件 安卓的大多数都是这个，根据平台自己修改吧。简单的做法就是自己进dev目录找fb设备

继续

```
xres = 1024, yres = 768
tslib: Selected device is not a touchscreen (must support ABS event type)
ts_read_raw:-------29
ts_read_raw: No such file or directory
```

现在好像可以执行了，屏幕变黑 也有打印LCD的分辨率。
报错原因找不到触摸设备，很清楚自己从头到尾都没告诉他我的触摸设备。
终端下输入: export TSLIB_TSDEVICE=/dev/input/event0;

再次执行ts_calibrate，无报错，完成！！最后会生成pointercal文件，
可以配置一下保存的位置
export TSLIB_CALIBFILE=/data/pointercal.

点击完屏幕的5个点后可以cat一下你的pointercal。
对应的7个数据就是校准参数了，后面的是你的分辨率 0不用管

```
99$ cat ~/Desktop/7846-backup/pointercal/pointercal1
-417 17862 -3228339 -13694 231 53965176 65536 1024 768 0
```

后写进我的校准程序里，为后面自己写校准界面做准备

```
static unsigned char                    v_Calibrated = 0;
static CALIBRATION_PARAMETER            v_CalcParam ={
    .a1     =   -63,
    .b1     =    17232,
    .c1     =   -2373345,
    .a2     =   -13502,
    .b2     =    85,
    .c2     =    52533400,
    .delta  =    65536,
};
static CALIBRATION_PARAMETER v_CalcParam_bak = {
    .a1     =   -102,
    .b1     =    17230,
    .c1     =   -1970907,
    .a2     =   -13547,
    .b2     =    226,
    .c2     =    52204164,
    .delta  =    65536,

};
```


校准程序是通用的，
可以在我这里下载，内带ADS7856触摸驱动，XPT2046触摸驱动和一个通用校准程序calibration。
地址：
https://download.csdn.net/download/qq_42233843/11131270
下面我在总结一下遇到的问题：

1：如果执行没有报错但是界面弹不起来。
原因：是因为Android UI界面的原因，
方法1：把/system/bin/surfaceflinger ui界面文件改个名字 然后重启一下板子。会发现没有安卓界面了那么你就可以去试一下（重启后环境变量没有了，重新配置一下把，很快，或者把写进你的init.rc里面）
方法2：我是用了第一种方法发现不行，针对平台把。后来我直接写了个APP，设置成全屏黑，然后你在去执行校准程序ts_calibrate。发现可以！。

2：什么都正常后，校准也成功后，发现触摸还是不正常。
原因：1，检查一下自己的触摸驱动，是否可以正确获取到AD值，不然校准绝对是不准的。
2，Android6.0要是的触摸正常必须加入触摸IC的 idc文件，我的是7846，具体在源码下的/device/rockchip/rk3399/x3399/ 目录下有idc文件。编译idc文件的mk在上一级目录。自己可查。

```
d/x3399_marshmallow/device/rockchip
$ cat ADS7846_Touchscreen.idc
# Basic Parameters
touch.deviceType = touchScreen
touch.orientationAware = 1
```

我配置成这样就正常了，是把滑鼠模式改成单点模式，（现象：光标不会移动到我点击的位置，但是会跟随你的触摸点移动而移动）。idc文件名有讲究，大家可以自己去查，很简单的说就是要和你触摸驱动里面的ts->name一样。

```
snprintf(ts->phys, sizeof(ts->phys), "%s/input0", dev_name(&spi->dev));
snprintf(ts->name, sizeof(ts->name), "ADS%d Touchscreen", ts->model);
```

我的ts_name：ADS7846 Touchscreen 中间有个空格，idc的话要把空格当成_ 下划线，所以我的idc文件名字为ADS7846_Touchscreen.idc。
不知道自己的触摸文件的话可以
cat /proc/bus/input/devices

我的配置如下

```
export TSLIB_FBDEVICE=/dev/graphics/fb0;
export TSLIB_CONSOLEDEVICE=none;
export TSLIB_TSDEVICE=/dev/input/event0;
export TSLIB_PLUGINDIR=/system/lib64/ts/plugins;
export TSLIB_CALIBFILE=/data/pointercal;
export TSLIB_CONFFILE=/system/tslib/ts.conf
```

大概就这么多了，希望大家喜欢，本人也是做个记录，方便自己帮助他人。