

To use USB touch controller on Android, please switch your controller to digitizer mode first, and follow the steps below:

1. Please enable USB HID support in kernel, first find .config in Android kernel source folder, and enable HID related flags :

```
CONFIG_HID_SUPPORT=y
```

```
CONFIG_HID=y
```

```
CONFIG_USB_HID=y
```

```
CONFIG_HID_PID=y
```

```
CONFIG_USB_HIDDEV=y
```

```
CONFIG_HIDRAW=y
```

2. Run "make clean" then run "make" to build kernel image

3. Make your file system image (Detail steps please check user manual of your development board)

4. Extracts the touch driver installation package in the Android_USB_XXXXXXX/

5.

x86 :

Copy your .iso to Android_USB_XXXXXXX /

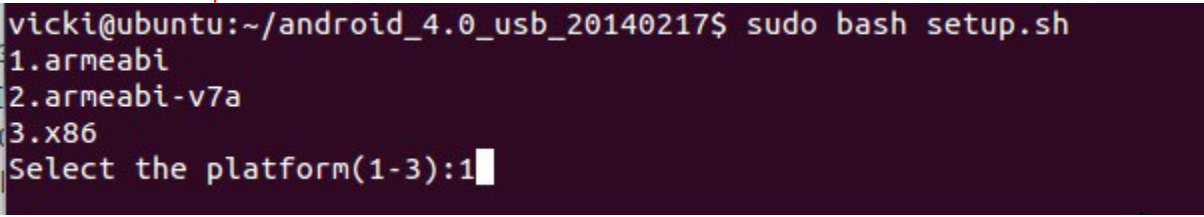
ARM:

Copy your system.img and ramdisk.img to Android_USB_XXXXXXX /

6. Install Calibrate.apk to system.img

```
# cd Android_USB_XXXXXXX/
```

```
# sudo bash setup.sh
```



```
vicki@ubuntu:~/android_4.0_usb_20140217$ sudo bash setup.sh
1.armeabi
2.armeabi-v7a
3.x86
Select the platform(1-3):1
```

7. The install script will automatically run "gedit ueventd.rc" , please add a line :

```
/dev/hiddev*      0777 radio radio
```

```
/dev/usb/hiddev*  0777 radio radio
```

```
/dev/hidraw*      0777 radio radio
```

,then save and close gedit .

```
*ueventd.rc (~/.android_installer/android_4.0_usb_20140117/android-image/ramdisk) - gedit
開啟 儲存 復原
*ueventd.rc X
/dev/ntc-acousltc 0000 system audio
/dev/vdec 0660 system audio
/dev/q6venc 0660 system audio
/dev/smd0 0640 radio radio
/dev/qmi 0640 radio radio
/dev/qmi0 0640 radio radio
/dev/qmi1 0640 radio radio
/dev/qmi2 0640 radio radio
/dev/ttyUSB* 0666 radio radio
/dev/bus/usb/* 0660 root usb
/dev/mtp_usb 0660 root mtp
/dev/usb_accessory 0660 root usb
/dev/tun 0660 system vpn

# CDMA radio interface MUX
/dev/ts0710mux* 0640 radio radio
/dev/ppp 0660 radio vpn

# sysfs properties
/sys/devices/virtual/input/input* enable 0660 root input
/sys/devices/virtual/input/input* poll_delay 0660 root input
/sys/devices/virtual/usb_composite/* enable 0664 root system

/dev/hiddev* 0777 radio radio
```

純文字 ▾ Tab 字元寬度: 8 ▾ 第 86 行第 30 字 插入

8.

X86:

Replace your old .iso.

```
Writing: End Volume Descriptor Start Block 19
Done with: End Volume Descriptor Block(s) 1
Writing: Version block Start Block 20
Done with: Version block Block(s) 1
Writing: Path table Start Block 21
Done with: Path table Block(s) 4
Writing: Joliet path table Start Block 25
Done with: Joliet path table Block(s) 4
Writing: Directory tree Start Block 29
Done with: Directory tree Block(s) 2
Writing: Joliet directory tree Start Block 31
Done with: Joliet directory tree Block(s) 2
Writing: Directory tree cleanup Start Block 33
Done with: Directory tree cleanup Block(s) 0
Writing: Extension record Start Block 33
Done with: Extension record Block(s) 1
Writing: The File(s) Start Block 34
8.87% done, estimate finish Tue Apr 23 16:17:05 2013
17.75% done, estimate finish Tue Apr 23 16:17:05 2013
26.61% done, estimate finish Tue Apr 23 16:17:05 2013
35.48% done, estimate finish Tue Apr 23 16:17:05 2013
44.34% done, estimate finish Tue Apr 23 16:17:05 2013
53.22% done, estimate finish Tue Apr 23 16:17:05 2013
62.07% done, estimate finish Tue Apr 23 16:17:05 2013
70.95% done, estimate finish Tue Apr 23 16:17:05 2013
79.80% done, estimate finish Tue Apr 23 16:17:06 2013
88.68% done, estimate finish Tue Apr 23 16:17:06 2013
97.53% done, estimate finish Tue Apr 23 16:17:06 2013
Total translation table size: 4495
Total rockridge attributes bytes: 1372
Total directory bytes: 2048
Path table size(bytes): 26
Done with: The File(s) Block(s) 56215
Writing: Ending Padblock Start Block 56249
Done with: Ending Padblock Block(s) 150
Max brk space used 0
56399 extents written (110 MB)
***tegav2.iso created successfully***
vicki@ubuntu:~/android_usb_20130422$ ls
android-image Calibrate mkyaffs2image setup.sh tegav2.iso tegav2.iso.old unyaffs
vicki@ubuntu:~/android_usb_20130422$
```

ARM:

Replace your old system.img and ramdisk.img.

```
vicki@ubuntu:~/android_usb_20130422$ cp ../UT6410-android/ut6410-android2.1-v2.0/out/target/product/ut6410/system.img ./
vicki@ubuntu:~/android_usb_20130422$ cp ../UT6410-android/ut6410-android2.1-v2.0/out/target/product/ut6410/ramdisk.img ./
vicki@ubuntu:~/android_usb_20130422$ ./setup.sh
1.armeabi
2.armeabi-v7a
3.x86
Select the platform(1-3):2
[sudo] password for vicki:
Modify ramdisk.img
496 blocks
496 blocks
496 blocks
Modify system.img
end of image
system.img created successfully
vicki@ubuntu:~/android_usb_20130422$ ls
Calibrate mkyaffs2Image ramdisk ramdisk.img ramdisk.img.old setup.sh system system.img system.img.old unyaffs
vicki@ubuntu:~/android_usb_20130422$
```

If calibrate not work , please find /dev/hiddev0 by terminal , If the device is not found,

To add the following code in /your_android_src/system/core/init/devices.c

```
if (!strcmp(name, "hiddev", 6))
```

```
{
```

```
    base = "/dev/";
```

```
    mkdir(base, 0777);
```

```
} else
```

should look like this :

```
if (!strcmp(uevent->subsystem, "usb", 3)) {
```

```
    if (!strcmp(name, "hiddev", 6))
```

```
    {
```

```
        base = "/dev/";
```

```
        mkdir(base, 0777);
```

```
    } else if (!strcmp(uevent->subsystem, "usb")) {
```

```
/* This imitates the file system that would be created
```

```
* if we were using devfs instead.
```

```
* Minors are broken up into groups of 128, starting at "001"
```

```
*/
```

```
int bus_id = uevent->minor / 128 + 1;
```

```
int device_id = uevent->minor % 128 + 1;
```

```
/* build directories */
```

```
mkdir("/dev/bus", 0755);
```

```
mkdir("/dev/bus/usb", 0755);
```

```
snprintf(devpath, sizeof(devpath), "/dev/bus/usb/%03d", bus_id);
```

```
mkdir(devpath, 0755);
```

```
snprintf(devpath, sizeof(devpath), "/dev/bus/usb/%03d/%03d", bus_id, device_id);
```

```
devpath_ready = 1;
```

```
    } else {  
        /* ignore other USB events */  
        return;  
    }  
}
```

Make your file system image, then re-run step 5