EETI eGTouch Linux Programming Guide v2.5N

TABLE OF CONTENTS

Sec 1: Introduction

1.1 GuideLine

EETI provides all kinds of touch solution. EETI eGTouch is a touch daemon driver for EETI touch controller. Only is available for kernel **2.6.24** upward.

Support interfaces:

1. **USB**
2. **RS232**
3. **PS/2**

Having below features:

1. **Precise points.**
2. **Great calibration precision for Resistive controller.**
3. **Capable for 10+ points report.**
4. **Following Linux Standard Multitouch-protocol point report.**
5. **Rightclick, beep sound, constant touch filter, etc.**
6. **Support multi devices.**
7. **Available for detecting X-window rotation to do rotating coordinate.**
8. **Provide manually modify driver's behavior.**

This document would assist you to install eGTouch.

1.2 Support From Vendor

If you encounter any problem as running eGTouch driver, please refer to the Sec 8.
Sec 8 list few common question, it might be useful to you, if your problem still can NOT be solved, please refer to the **Sec 9-1**. **Need Support From EETI**.

如果你有任何 **driver** 使用上的問題，請參照此文件的第 8 節，該節列出常見的問題，對您可能很有用，如果仍然無法解決您的問題，請參閱第 9-1 節，取得 EETI 的支援。

如果您有任何驱动程序使用上的问题，请参照此文件的第 8 节，該节列出常见的问题，对您可能很有用，如果仍然无法解决您的问题，请参见第 9-1 节，取得 EETI 的支持。

Sec 2: Before install

2.1 Check kernel module

To install driver, please check module configuration as below:

Necessary:

1. **EVDEV**
2. **UINPUT**
3. **HIDRAW (USB Interface)**
4. **HID_MULTITOUCH (USB Interface & Kernel 3.0 upwards)**

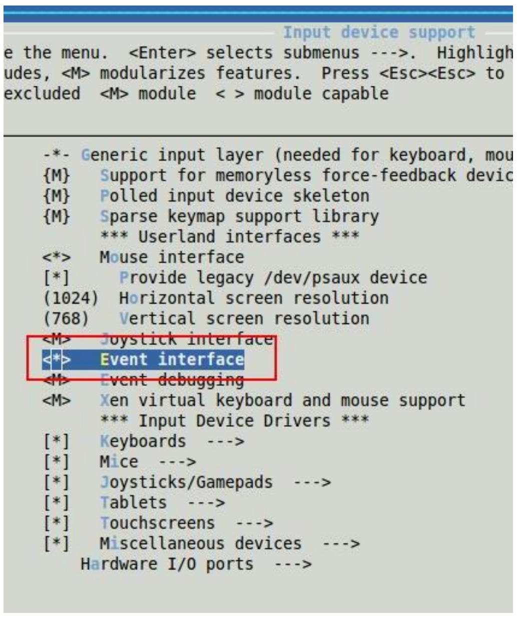
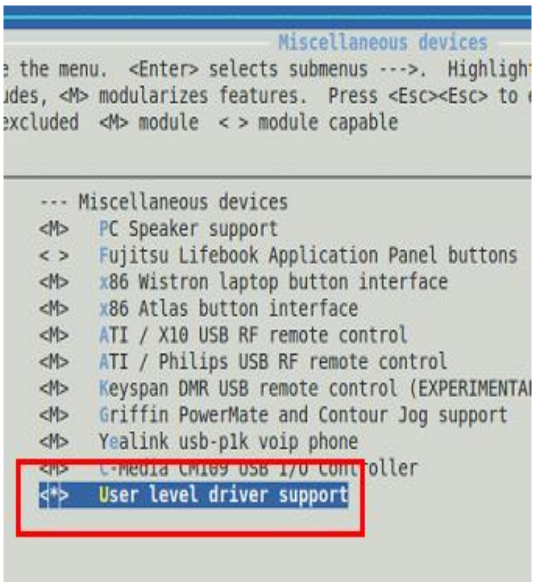
Remove:

CONFIG_TOUCHSCREEN_USB_COMPOSITE

(For USB Interface & PID 0001 controller)

You could check this by "make menuconfig" command or modify Kconfig file. Below is an example of "make menuconfig":

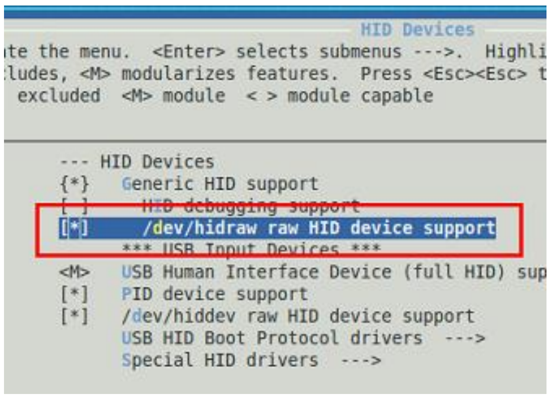
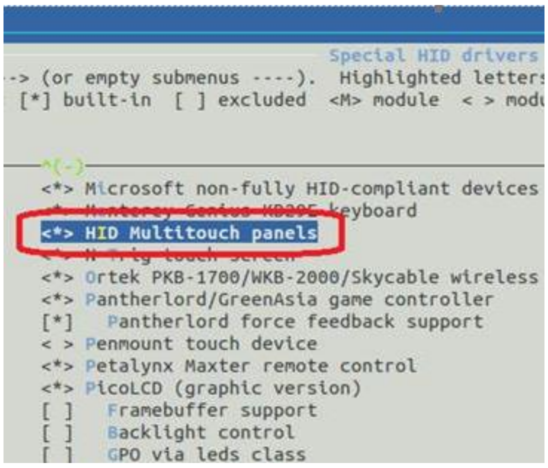
Necessary:

[Device Drivers] / [Input device support] / [Event interface]	[Device Drivers] / [Input device support] / [Miscellaneous devices] / [User level driver support]
	

```
                                  Input device support
e the menu.  <Enter> selects submenus --->.  Highligh
udes, <M> modularizes features.  Press <Esc><Esc> to
excluded  <M> module  < > module capable

    -*- Generic input layer (needed for keyboard, mou
    {M}   Support for memoryless force-feedback devic
    {M}   Polled input device skeleton
    {M}   Sparse keymap support library
          *** Userland interfaces ***
    <*>   Mouse interface
    [*]     Provide legacy /dev/psaux device
    (1024)  Horizontal screen resolution
    (768)   Vertical screen resolution
    <M>   Joystick interface
    <*>   Event interface
    <M>   Event debugging
    <M>   Xen virtual keyboard and mouse support
          *** Input Device Drivers ***
    [*]   Keyboards  --->
    [*]   Mice  --->
    [*]   Joysticks/Gamepads  --->
    [*]   Tablets  --->
    [*]   Touchscreens  --->
    [*]   Miscellaneous devices  --->
        Hardware I/O ports  --->
```

```
                                  Miscellaneous devices
e the menu.  <Enter> selects submenus --->.  Highligh
ides, <M> modularizes features.  Press <Esc><Esc> to
excluded  <M> module  < > module capable

    --- Miscellaneous devices
    <M>   PC Speaker support
    < >   Fujitsu Lifebook Application Panel buttons
    <M>   x86 Wistron laptop button interface
    <M>   x86 Atlas button interface
    <M>   ATI / X10 USB RF remote control
    <M>   ATI / Philips USB RF remote control
    <M>   Keyspan DMR USB remote control (EXPERIMENTAL
    <M>   Griffin PowerMate and Contour Jog support
    <M>   Yealink usb-p1k voip phone
    <M>   C-Media CM109 USB I/O Controller
    <*>   User level driver support
```

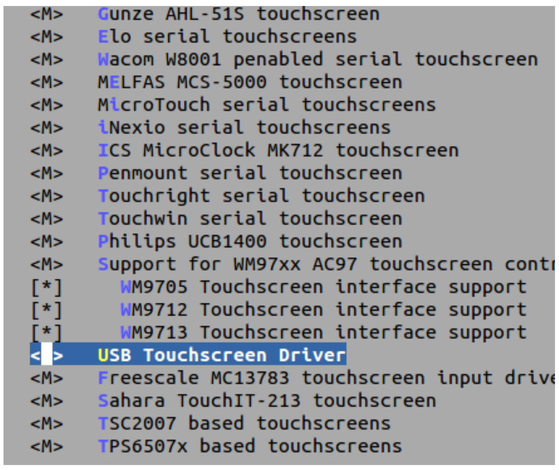
Necessary:

[Device Drivers] / [HID Devices] / [/dev/hidraw raw HID device support] **(for USB Interface)**	[Device Drivers] / [HID Devices] / Special HID drivers / HID Multitouch panels **(If Kernel Version 3.0 upwards & for USB Interface)**
	

```
                                         HID Devices
ite the menu.  <Enter> selects submenus --->.  Highli
:ludes, <M> modularizes features.  Press <Esc><Esc> t
 excluded  <M> module  < > module capable

    --- HID Devices
    {*}   Generic HID support
    [ ]     HID debugging support
    [*]     /dev/hidraw raw HID device support
          *** USB Input Devices ***
    <M>   USB Human Interface Device (full HID) sup
    [*]   PID device support
    [*]   /dev/hiddev raw HID device support
          USB HID Boot Protocol drivers  --->
          Special HID drivers  --->
```

```
                                        Special HID drivers
-> (or empty submenus ----).  Highlighted letters
 [*] built-in  [ ] excluded  <M> module  < > modu

  ^(-)
  <*> Microsoft non-fully HID-compliant devices
  <*> Monterey Genius KB29E keyboard
  <*> HID Multitouch panels
  < > N-Trig touch screen
  <*> Ortek PKB-1700/WKB-2000/Skycable wireless
  <*> Pantherlord/GreenAsia game controller
  [*]   Pantherlord force feedback support
  < > Penmount touch device
  <*> Petalynx Maxter remote control
  <*> PicoLCD (graphic version)
  [ ]   Framebuffer support
  [ ]   Backlight control
  [ ]   GPO via leds class
```

Remove:

[Device Drivers] / [Input device support] / [Touchscreens] / [USB Touchscreen Driver] **(PID 0001 USB controller)**


```
  <M>    Gunze AHL-51S touchscreen
  <M>    Elo serial touchscreens
  <M>    Wacom W8001 penabled serial touchscreen
  <M>    MELFAS MCS-5000 touchscreen
  <M>    MicroTouch serial touchscreens
  <M>    iNexio serial touchscreens
  <M>    ICS MicroClock MK712 touchscreen
  <M>    Penmount serial touchscreen
  <M>    Touchright serial touchscreen
  <M>    Touchwin serial touchscreen
  <M>    Philips UCB1400 touchscreen
  <M>    Support for WM97xx AC97 touchscreen contr
  [*]      WM9705 Touchscreen interface support
  [*]      WM9712 Touchscreen interface support
  [*]      WM9713 Touchscreen interface support
  < >    USB Touchscreen Driver
  <M>    Freescale MC13783 touchscreen input drive
  <M>    Sahara TouchIT-213 touchscreen
  <M>    TSC2007 based touchscreens
  <M>    TPS6507x based touchscreens
```

2.2 Conditions to patch kernel source code

If your system does not fulfill the conditions described below, please ignore this section.

If your system meets all below two conditions, please refer to Appendix 10-1 to do kernel blacklist patch first, or driver would NOT be functional.

1.	**Interface**	**USB**
2.	**X.org version**	**< 1.8.7 or no X-window**

If your system meets all below three conditions, please refer to Appendix 10-2 to do kernel HIDCORE patch first, or driver would NOT be functional.

1.	**Interface**	**USB**
2.	**Kernel version**	**3.8.x to 3.12.x**
3.	**ControllerType**	**Resistive or SCAP**

2.3 Check device

1.) If you did above modification, please rebuild your kernel to make it effect.

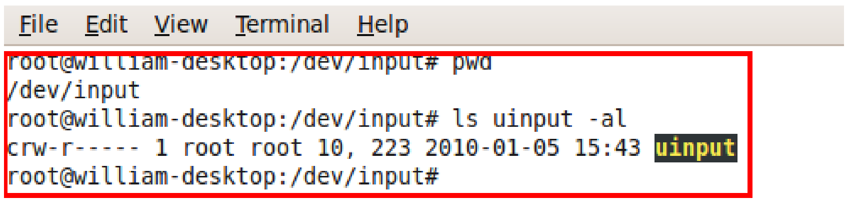
2.) After that, you could check those kernel functions enable or not through below steps.

All interface.

a. UINPUT device node

You should see uinput under **/dev/input/uinput** or **/dev/uinput.**

For example:

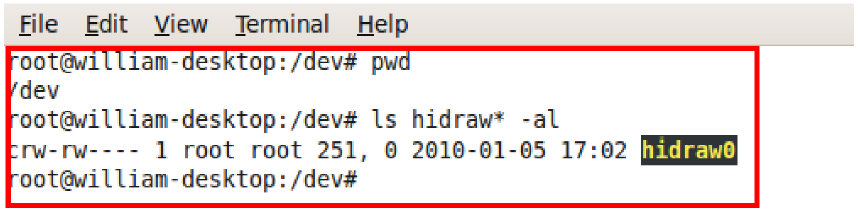


```
root@william-desktop:/dev/input# pwd
/dev/input
root@william-desktop:/dev/input# ls uinput -al
crw-r----- 1 root root 10, 223 2010-01-05 15:43 uinput
root@william-desktop:/dev/input#
```

USB interface only.

b. hidraw device node

As the usb device is plug-in, there would be a **hidraw** node generated under **/dev**

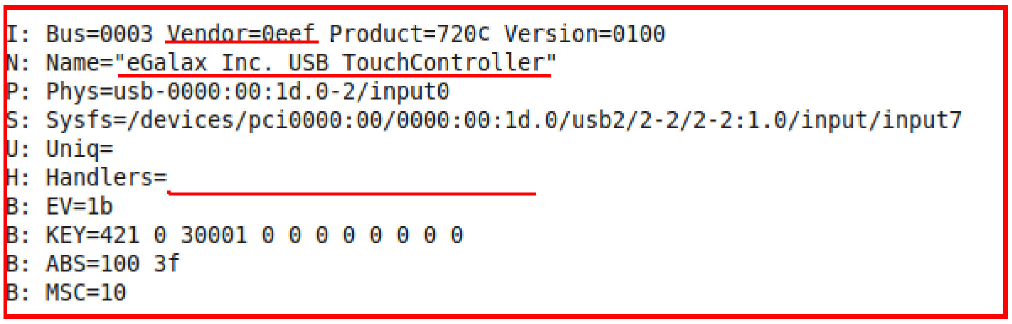


```
root@william-desktop:/dev# pwd
/dev
root@william-desktop:/dev# ls hidraw* -al
crw-rw---- 1 root root 251, 0 2010-01-05 17:02 hidraw0
root@william-desktop:/dev#
```

c. USB touch device handlers

Type command “**cat /proc/bus/input/devices**” and see the result.

If you need and have done the source code patch, you would see a **blank content** behind the **Handlers** item.



```
I: Bus=0003 Vendor=0eef Product=720c Version=0100
N: Name="eGalax Inc. USB TouchController"
P: Phys=usb-0000:00:1d.0-2/input0
S: Sysfs=/devices/pci0000:00/0000:00:1d.0/usb2/2-2/2-2:1.0/input/input7
U: Uniq=
H: Handlers=
B: EV=1b
B: KEY=421 0 30001 0 0 0 0 0 0 0 0
B: ABS=100 3f
B: MSC=10
```

Sec 3: Install Driver Package

3-1 Install Process

Before running install setup script, please plug-in the controller first. Then you could execute script file **setup.sh** to automatically install driver.

Syntex:

```
sh setup.sh              # To install the eGTouch driver.
sh setup.sh uninstall    # To remove the eGTouch driver.
```

You could also complete these steps manually.

1. Decompress eGTouch package which contains:

a)	eGTouchD:	a daemon service driver for EETI touch controller.
b)	eGTouchL.ini:	a parameter list loaded by driver
c)	GetEvent.c:	a sample code describes how to read EETI input event.

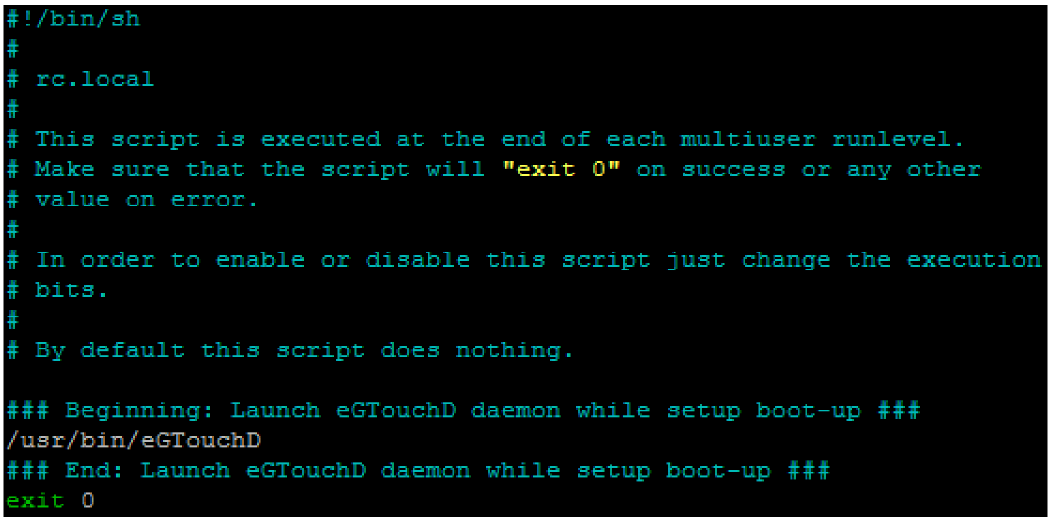
If you have X-window, you may also be available for these:

d)	eGTouchU:	a X-window utility tool for eGTouchD (x86 only)
e)	eCalib:	a command line X-window calibration tool.
f)	52-egalax-virtual.conf	X-window configure file for recognizing EETI touch

2. Place “eGTouchL.ini” into Linux system directory “/etc/eGTouchL.ini” where driver would load it. We can change driver behavior by modifying this file. **The detail descriptions of parameters are described in Section 5.** (You can see brief definitions in eGTouchL.ini)

3. Place **eGTouchD** , **eGTouchU** (x86 only) and **eCalib** (need X-window) under **/usr/bin**.

4. In general Linux distribution, please edit /etc/rc.local (/etc/rc.d/rc.local in RedHat or /etc/init.d/boot.local in Suse), to place /usr/bin/eGTouchD execution in /etc/rc.local to make eGTouchD execute at system boot.



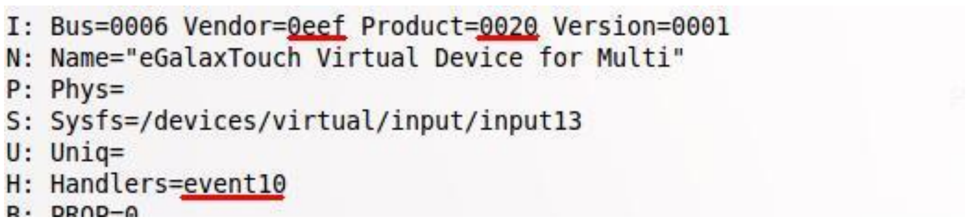
```
#!/bin/sh
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

### Beginning: Launch eGTouchD daemon while setup boot-up ###
/usr/bin/eGTouchD
### End: Launch eGTouchD daemon while setup boot-up ###
exit 0
```

5. To blacklist usbtouchscreen module run from the beginning of system operation. You could also manually modify **/etc/modprobe.d/blacklist.conf** to add usbtouchscreen into blacklist.

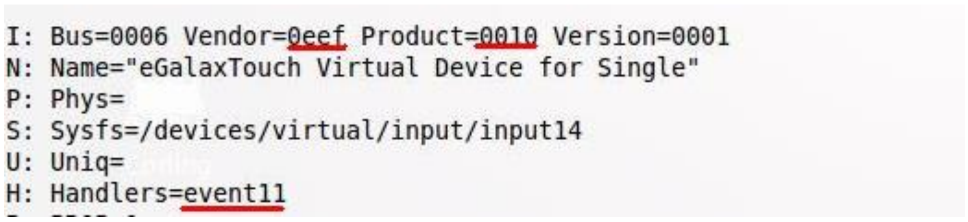
   ### Beginning: blacklist usbtouchscreen ###
   blacklist usbtouchscreen
   ### End: blacklist usbtouchscreen ###

6. If Xorg Version is 1.8.7 upwards, put **52-egalax-virtual.conf** xorg rule file into **/usr/share/X11/xorg.conf.d** folder

7. After launching eGTouchD with device plugged, check **/proc/bus/input/devices** file and you will find two virtual devices. Like below figures:



```
I: Bus=0006 Vendor=0eef Product=0020 Version=0001
N: Name="eGalaxTouch Virtual Device for Multi"
P: Phys=
S: Sysfs=/devices/virtual/input/input13
U: Uniq=
H: Handlers=event10
B: PROP=0
```



```
I: Bus=0006 Vendor=0eef Product=0010 Version=0001
N: Name="eGalaxTouch Virtual Device for Single"
P: Phys=
S: Sysfs=/devices/virtual/input/input14
U: Uniq=
H: Handlers=event11
```

   We could check event node which was assigned to the virtual device and read/get input event through this device node, e.g. /dev/input/eventX.

3-2 Tools

As you have **X-window**, these tools are available for use.

Please execute these tools under “root” permission!

eGTouchU x86 system only	The tool eGTouchU is a utility tool which could help you modify driver’s parameter through UI. The detail descriptions please refer to the document “EETI eGTouch Utility Guide” in driver package.
eCalib	The tool eCalib is a calibration tool with command line. Please type “eCalib -h” to see the usage content.

Sec 4: Touch Input Event Sequence

The eGTouchD daemon sends input event through kernel feature UINPUT so that the client program can get these events from /dev/input/eventX.

4-1 Two different event sequences

The eGTouchD daemon would report event based on different kernel version.

1. kernel version is 2.6.36 upwards:

Multi-touch Protocol Type B

```
ABS_MT_SLOT 0
ABS_MT_TRACKING_ID 0
ABS_MT_POSITION_X x[0]
ABS_MT_POSITION_Y y[0]
ABS_MT_SLOT 1
ABS_MT_TRACKING_ID 1
ABS_MT_POSITION_X x[1]
ABS_MT_POSITION_Y y[1]
```

you can see the detailed rule described in /Documentation/input/**multi-touch-protocol.txt** under Linux kernel source code.

2. kernel version is 2.6.35 downwards:

EETI protocol: Standard mouse event and custom extra event

Type = EV_KEY
Code = BTN_LEFT
Value = left mouse button state of **first point**,
1: pen down / 0: life off.
	Type = EV_KEY
	Code = BTN_EXTRA
	Value = the touch state of **second point**,
	1: pen down / 0: lift off.

Type = EV_ABS
Code = ABS_X
Value = the X axis position of **first point**.
The range is from 0 to 4095.
	Type = EV_ABS
	Code = ABS_RX
	Value = the X axis position of **second point**.
	The range is from 0 to 4095

Type = EV_ABS
Code = ABS_Y
Value = the Y axis position of **first point**.
The range is from 0 to 4095.
	Type = EV_ABS
	Code = ABS_RY
	Value = the Y axis position of **second point**.
	The range is from 0 to 4095.

Type = EV_SYNC

Code = SYN_REPORT

Value = 0

A Sync report event, all data will be valid after this event is received.

4-2 How to read touch event

EETI provide a sample code **GetEvent.c** to show how the event sequence behaves. Please compile the sample code and execute it corresponding to the event node (/dev/input/eventX). You would see the event sequence as panel is touched and design your own application based on this input sequence as well

Sec 5: eGTouchL.ini Parameter Explanations

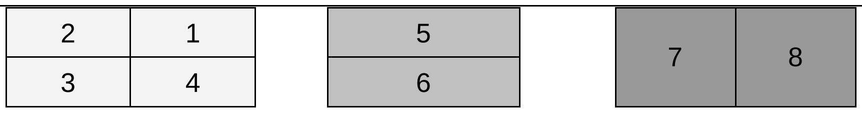
The file **eGTouchL.ini** has a parameter list which would be loaded by driver. Driver's behavior could be changed by these parameters. Please **DON'T** modify the front title as setting up eGTouchL.ini.

5-1 Parameter Table

This table describe the detailed usage of all parameters. There is also a simple description in eGTouchL.ini.

◆ DebugEnableBits	Debug message you want to show.
0	Close all Debug
1	Print initialization debug message **[Default]**
FFFFF	Open all Debug
◆ ShowDebugPosition	Position you want to show/store Debug message
0	Print in file located at /tmp **[Default]**
1	Print in terminal
2	Print in above both
◆ DeviceNums	How many devices you want to plug-in to the system. If you want more than one device, please modify this value.
1	Only one device **[Default]**
2-10	More than one device. **[Max = 10]**
◆ Baudrate	Choose the BaudRate
0	Auto detect Baudrate **[Default]**
X	Set Baudrate to X bps. (PCAP72: 57600 , Resis: 9600)
◆ ScanInterface	Choose scan interface
0	Scan all interface **[Default]** (USB / RS232 / PS/2)
1	Scan USB interface only.
2	Scan UART interface only.
3	Scan PS/2 interface only.
◆ ScanDevStartDelayTime	Driver booting delay time
0	No delay [Default]
X	Delay X millisecond to start driver.
◆ SerialPath	RS232 Serial Path

default /dev/serial/ttyS0	Default path /dev/ttySX (X could be equals to 0-10) **[Default]** Customized path. Please type in your specific serial path accordinating to the form.
◆ SupportPoints	The amount of points you want to report (This is also confined by Controller)
0 1 >=2	No point Single-touch Multi-touch **[Default = 10]**
◆ Direction	Change the X and Y direction
0 1 2 3 4	Don't make any invert **[Default]** Invert X Invert Y Invert both X and Y Swap X and Y
◆ Orientation	Change the orientation
0 1 2 3	0 degree **[Default]** 90 degree 180 degree 270 degree
◆ EdgeCompensate	Do edge compensate
0 1	Disable **[Default]** Enable
EdgeLeft, EdgeRight EdgeTop, EdgeBottom	Edge compensate value
X	If equals to 100, it means no change. If you set Left=50, you'll see the left-edge points are shrinks inward. And vice versa. **[Min 50 - 150 Max]** **[Default = 100]**
◆ HoldFilterEnable	Filter out constant touch or not
0 1	Disable **[Default]** Enable
HoldRange	Constant touch valid area
X	±X range of the point which would lead to constant touch **[Min 0 - 50 Max]** **[Default = 10]**
◆ SplitRectMode	Split the display into Specific Rect. Touch would just show on the specific Rect.
0 1-8	No change (Full Display) **[Default]** Driver in-built split Rect

9	 Customized Rect.
CustomRectLeft CustomRectRight CustomRectTop CustomRectBottom	Theses parameters are valid as SplitRectMode=9. You can customize the Rect by these parameters.
0-4095	Four sides of the customized Rect
◆ MonitorName	Monitor Name
default null	Use for mapping touch data output to specific monitor. Check monitor name by command "xrandr", example: "eDP1". If there’s no roation and multi monitor requirement, just ignore it. Note: if you find your monitor name will change after suspend resume. You can set "eDP*", and then driver will search correct monitor number.
◆ DetectRotation (Only for x86 system)	**Enable:** Driver would map its coordinate corresponding to X window rotation or monitor status change. ***Please see Sec 6.** **Disable:** If there’s no roation and multi monitor requirement, just disable it.
0 1	Disable **[Default]** Enable
◆ ReportMode	Set different report type
1 2 3	Normal Mode. Report point normally. **[Default]** Click on Touch. Only report point as touch down. Click on Release. Only report point as touch up.
◆ EventType	Set events report type
0 1 2	Auto detect mode Single touch mode (if mouse cursor is disapeared, please try set EventType to 1) Multi touch even type mode
◆ BtnType	Set EETI protocol BtnType
0 1 2	Report single event as BTN_LEFT. **[Default]** Report single event as ABS_PRESSURE. (Generally for Tslib) Report single event as BTN_TOUCH.
◆ RightClickEnable	Report mouse Right Click after constant touch for a while
0 1	Disable Right Click Enable Right Click **[Default]**
RightClickDuration	Constant touch duration to trigger Right Click
X	X milliseconds **[Default = 1500]**

RightClickRange	Valid area of trigger-RightClick constant touch
X	±X range of the point would lead to constant touch for RightClick **[Min 0 - 50 Max]** **[Default = 10]**
◆ BeepState	Make a beep sound as touch *__Please see Sec 6-3.__
0 1 2 3	Disable Beep Make a beep sound as ”Touch Down” Make a beep sound as ”Touch Up” Make a beep sound as both two above conditions.
BeepDevice	Choose the beep sound device
0 1 2 3	No device Send beep sound by from system buzzer Send beep sound by from sound card (Only for x86 system) Send beep sound from both devices.
BeepFreq	You can modify buzzer beep frequency here.
X	(Only for buzzer) The buzzer beep frequency. **[Default = 1000]**
BeepLen	You can modify buzzer beep time length here.
X	(Only for buzzer) The buzzer beep time length (ms). **[Default = 200]**
◆ VKEYEnable	Enable this option if there’s virtual key on your touch sensor.
0 1	Disable **[Default]** Enable
VKEYReportMod	Virtual Key Sensitivity.
X	Smaller value means more sensitive. On the other hand, larger is less sensitive
VKEY_**X** **Y**	1. The value of **X** refer to the vkey package **reported by controller** as touching the specified virtual key on your sensor. 2. The value of **Y** refer to the **keyevent code in input.h** which you want to report to system. You can choose the keyevent you want and fill in the code.
Example A: **VKEY_0** **139**	1. As controller report vkey package [0], we’ll send keyevent code [139] to system. 2. The event code [139] in input.h refers to MENU_KEY. Note: 139 is a decimal number.

Example B: **VKEY_1 0x160**	1. You can fill in Hex number in **Y** by adding a symbol [0x] before the number. 2. As controller report vkey package [1], we'll send keyevent code [0x160] to system The event code [0x160] in input.h refer to KEY_OK.
Note: If you're not sure controller's vkey package value, please contact EETI vendor.	

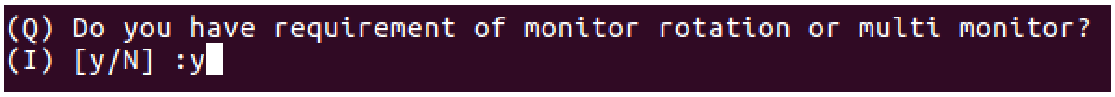
Sec 6: Annotation

6-1 DetectRotation Note

eGTouch driver support detect monitor rotation and multi monitor mapping. For enable these features, eGTouch driver have to be executed after X-server is ready(We use Xlib to do detection), and system need support these commands: ”xrandr” and ”xinput”.

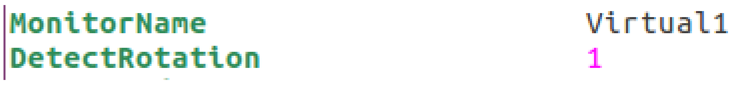
We recommand use **lightdm** to startup eGTouch driver until now, if your system not using lightdm, please install lightdm first. You can install by this command: ”apt-get install lightdm”. Since the ready time sequence of Xlib is different among diverse startup. We’re sorry that we couldn’t provide solution correspond to all startup. If there’s any further problem as setting up please contact us for technical support.

Notice:
When you execute ”setup.sh ”to install eGTouch driver, please set ”y” to enable functions.



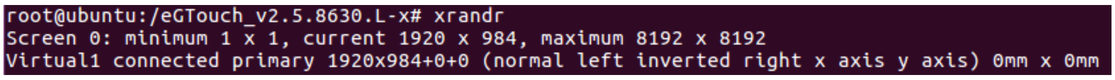
```
(Q) Do you have requirement of monitor rotation or multi monitor?
(I) [y/N] :y
```

And remember set ”MonitorName” in /etc/eGTouchL.ini



```
MonitorName                            Virtual1
DetectRotation                         1
```

You can check monitor name by command "xrandr".



```
root@ubuntu:/eGTouch_v2.5.8630.L-x# xrandr
Screen 0: minimum 1 x 1, current 1920 x 984, maximum 8192 x 8192
Virtual1 connected primary 1920x984+0+0 (normal left inverted right x axis y axis) 0mm x 0mm
```

6-2 Rotation and Beep for Embedded System

If you are using an embedded system (ex: ARM CPU), and you need support for rotation detection. There’s a necessary condition: **Xrandr** lib support since eGTouch detect rotation event by Xrandr lib.

And so on. If you are using an embedded system (ex: ARM CPU), you need support for sound card beep. There’s a necessary condition: **ALSA** lib support since eGTouch send beep sound by ALSA lib.

If you need this support and your system got target library, please contact us for a customized driver. Thanks.

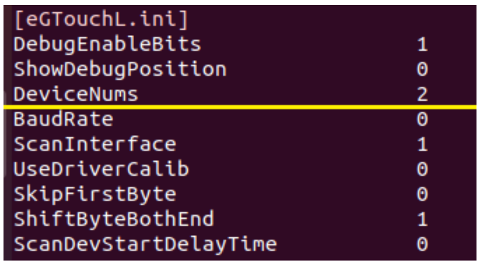
Sec 7: Multi-Monitor Setting

7-1 Numerous Devices

If you're going to use numerous devices, please do remember to modify the parameter "DeviceNums" in the ini file.

For example: If you've plug **two** EETI devices on your system, please modify the parameter as below:

DeviceNums 2



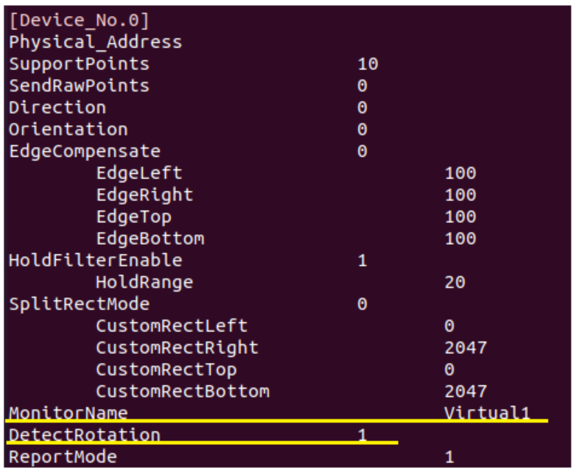
```
[eGTouchL.ini]
DebugEnableBits                 1
ShowDebugPosition               0
DeviceNums                      2
BaudRate                        0
ScanInterface                   1
UseDriverCalib                  0
SkipFirstByte                   0
ShiftByteBothEnd                1
ScanDevStartDelayTime           0
```

After modifying the parameter, please reboot your system or restart driver to make it valid.

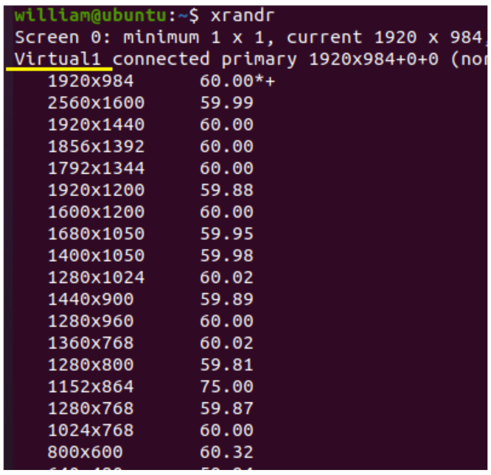
7-2 Monitor Name

After setting 7-1 Numerous Devices and reboot system or restart driver, /etc/eGTouchL.ini will have two devices configuration, please make sure MonitorName and DetectRotation are setting correctly by each device.

For example, you can get monitor name via **$ xrandr** and set it to the corresponding device, and check DetectRotation is set to 1, and set the second device as the same way.



```
[Device_No.0]
Physical_Address
SupportPoints                   10
SendRawPoints                   0
Direction                       0
Orientation                     0
EdgeCompensate                  0
        EdgeLeft                        100
        EdgeRight                       100
        EdgeTop                         100
        EdgeBottom                      100
HoldFilterEnable                1
        HoldRange                       20
SplitRectMode                   0
        CustomRectLeft                  0
        CustomRectRight                 2047
        CustomRectTop                   0
        CustomRectBottom                2047
MonitorName                             Virtual1
DetectRotation                  1
ReportMode                              1
```



```
william@ubuntu:~$ xrandr
Screen 0: minimum 1 x 1, current 1920 x 984
Virtual1 connected primary 1920x984+0+0 (nor
   1920x984      60.00*+
   2560x1600     59.99
   1920x1440     60.00
   1856x1392     60.00
   1792x1344     60.00
   1920x1200     59.88
   1600x1200     60.00
   1680x1050     59.95
   1400x1050     59.98
   1280x1024     60.02
   1440x900      59.89
   1280x960      60.00
   1360x768      60.02
   1280x800      59.81
   1152x864      75.00
   1280x768      59.87
   1024x768      60.00
   800x600       60.32
```

After modifying the parameter, please reboot your system or restart driver to make it valid.

7-3 Calibration Method.

If you are using PCAP devices, it is no need to do calibration, if not, please refer to the below statement.

The calibration tool eCalib currently does not support multi-monitor, you may have to calibrate your touch controller one by one.

For example, if you are using two touch controller and monitor, please disconnect your second monitor and touch controller, than do calibration and disconnect first monitor and controller after calibrating done, than reconnect second monitor and touch controller to your system, do calibration again, reconnect all of your monitors and controllers after calibrating done.

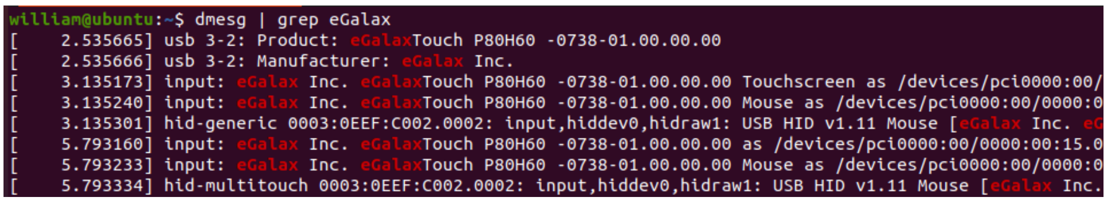
Sec 8: FAQ

8-1 Touch not working

1. **Check connection of controller.**

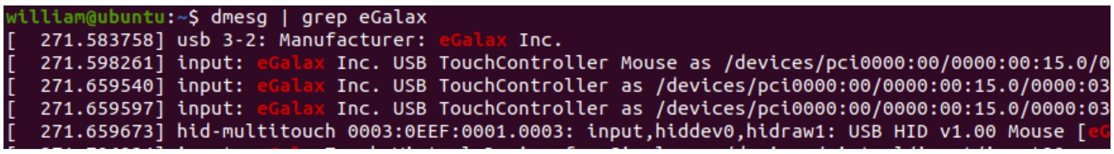
For USB interface: **$ dmesg | grep eGalax**

[PCAP example]



```
william@ubuntu:~$ dmesg | grep eGalax
[    2.535665] usb 3-2: Product: eGalaxTouch P80H60 -0738-01.00.00.00
[    2.535666] usb 3-2: Manufacturer: eGalax Inc.
[    3.135173] input: eGalax Inc. eGalaxTouch P80H60 -0738-01.00.00.00 Touchscreen as /devices/pci0000:00/
[    3.135240] input: eGalax Inc. eGalaxTouch P80H60 -0738-01.00.00.00 Mouse as /devices/pci0000:00/0000:0
[    3.135301] hid-generic 0003:0EEF:C002.0002: input,hiddev0,hidraw1: USB HID v1.11 Mouse [eGalax Inc. eG
[    5.793160] input: eGalax Inc. eGalaxTouch P80H60 -0738-01.00.00.00 as /devices/pci0000:00/0000:00:15.0
[    5.793233] input: eGalax Inc. eGalaxTouch P80H60 -0738-01.00.00.00 Mouse as /devices/pci0000:00/0000:0
[    5.793334] hid-multitouch 0003:0EEF:C002.0002: input,hiddev0,hidraw1: USB HID v1.11 Mouse [eGalax Inc.
```

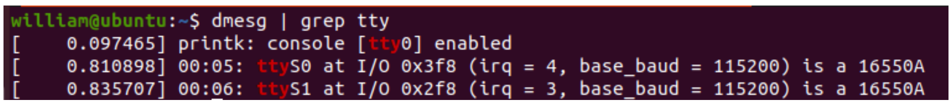
[Resistant touch example]



```
william@ubuntu:~$ dmesg | grep eGalax
[  271.583758] usb 3-2: Manufacturer: eGalax Inc.
[  271.598261] input: eGalax Inc. USB TouchController Mouse as /devices/pci0000:00/0000:00:15.0/0
[  271.659540] input: eGalax Inc. USB TouchController as /devices/pci0000:00/0000:00:15.0/0000:03
[  271.659597] input: eGalax Inc. USB TouchController as /devices/pci0000:00/0000:00:15.0/0000:03
[  271.659673] hid-multitouch 0003:0EEF:0001.0003: input,hiddev0,hidraw1: USB HID v1.00 Mouse [eG
```

For UART interface: **$ dmesg | grep tty**

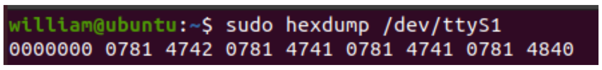
Find system current ttyS inputs.



```
william@ubuntu:~$ dmesg | grep tty
[    0.097465] printk: console [tty0] enabled
[    0.810898] 00:05: ttyS0 at I/O 0x3f8 (irq = 4, base_baud = 115200) is a 16550A
[    0.835707] 00:06: ttyS1 at I/O 0x2f8 (irq = 3, base_baud = 115200) is a 16550A
```

Use hexdump and touch screen, if connection is OK, it will print some information.

$ sudo hexdump /dev/ttySX , in this example, X should be 0 or 1.

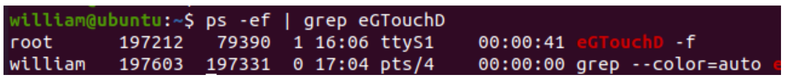


```
william@ubuntu:~$ sudo hexdump /dev/ttyS1
0000000 0781 4742 0781 4741 0781 4741 0781 4840
```

If you have trouble in this step, please check the hardware connection of your device and controller.

2. **Check eGTouch driver status.**

$ ps –ef | grep eGTouchD



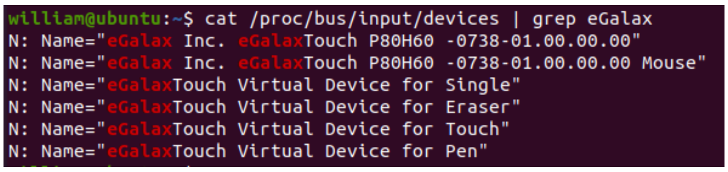
```
william@ubuntu:~$ ps -ef | grep eGTouchD
root       197212   79390  1 16:06 ttyS1    00:00:41 eGTouchD -f
william    197603  197331  0 17:04 pts/4    00:00:00 grep --color=auto e
```

If eGTouchD is not running, please running eGTouchD manually. **$ sudo eGTouchD**

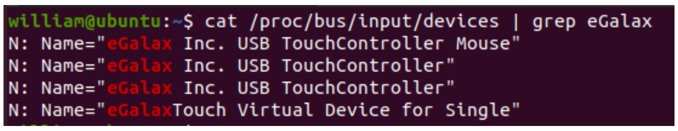
3. **Check input device has been created.**

$ cat /proc/bus/input/devices | grep eGalax To see is there any eGalaxTouch Virtual Device in list.

[PCAP example]


```
william@ubuntu:~$ cat /proc/bus/input/devices | grep eGalax
N: Name="eGalax Inc. eGalaxTouch P80H60 -0738-01.00.00.00"
N: Name="eGalax Inc. eGalaxTouch P80H60 -0738-01.00.00.00 Mouse"
N: Name="eGalaxTouch Virtual Device for Single"
N: Name="eGalaxTouch Virtual Device for Eraser"
N: Name="eGalaxTouch Virtual Device for Touch"
N: Name="eGalaxTouch Virtual Device for Pen"
```

[Resistant touch example]


```
william@ubuntu:~$ cat /proc/bus/input/devices | grep eGalax
N: Name="eGalax Inc. USB TouchController Mouse"
N: Name="eGalax Inc. USB TouchController"
N: Name="eGalax Inc. USB TouchController"
N: Name="eGalaxTouch Virtual Device for Single"
```

If you have trouble in this step, please check the required modules have been installed correctly. (Refer to **2.1 Check kernel module**)

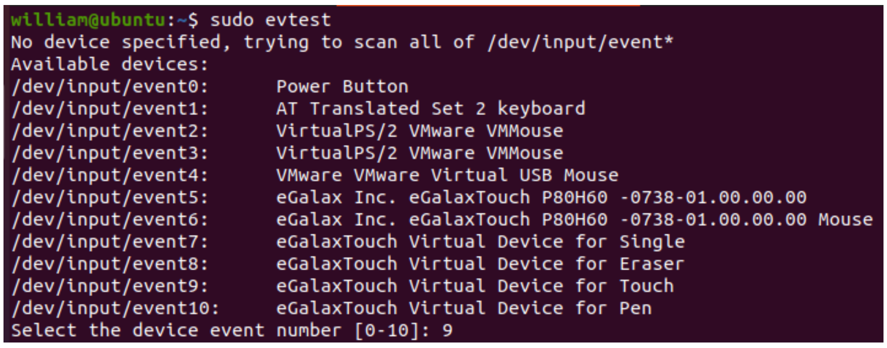
If you are using UART interface, please also have a check to **8.5** section.

4. Check input data flow.

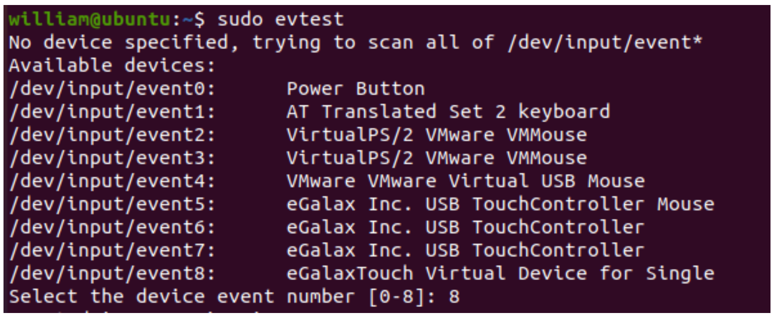
Use evtest to get the data from input device, (Installation: **$ sudo apt-get install evtest**)

Select the respond device as below example. **$ sudo evtest**

[PCAP example] Select eGalax Virtual Device for Touch and touch screen.


```
william@ubuntu:~$ sudo evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      Power Button
/dev/input/event1:      AT Translated Set 2 keyboard
/dev/input/event2:      VirtualPS/2 VMware VMMouse
/dev/input/event3:      VirtualPS/2 VMware VMMouse
/dev/input/event4:      VMware VMware Virtual USB Mouse
/dev/input/event5:      eGalax Inc. eGalaxTouch P80H60 -0738-01.00.00.00
/dev/input/event6:      eGalax Inc. eGalaxTouch P80H60 -0738-01.00.00.00 Mouse
/dev/input/event7:      eGalaxTouch Virtual Device for Single
/dev/input/event8:      eGalaxTouch Virtual Device for Eraser
/dev/input/event9:      eGalaxTouch Virtual Device for Touch
/dev/input/event10:     eGalaxTouch Virtual Device for Pen
Select the device event number [0-10]: 9
```

[Resistant touch example] Select eGalax Virtual Device for Single and touch screen.

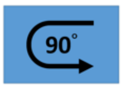

```
william@ubuntu:~$ sudo evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      Power Button
/dev/input/event1:      AT Translated Set 2 keyboard
/dev/input/event2:      VirtualPS/2 VMware VMMouse
/dev/input/event3:      VirtualPS/2 VMware VMMouse
/dev/input/event4:      VMware VMware Virtual USB Mouse
/dev/input/event5:      eGalax Inc. USB TouchController Mouse
/dev/input/event6:      eGalax Inc. USB TouchController
/dev/input/event7:      eGalax Inc. USB TouchController
/dev/input/event8:      eGalaxTouch Virtual Device for Single
Select the device event number [0-8]: 8
```

If you see data printing from screen, you may check has inbox touch driver has been blocked correctly. (Refer to **2.2** section)

If you do not see data printing from screen, please refer to **9.1** section, describe your issue and contact us.

8-2 Touch position and direction don’t match rotated display

1. Set Orientation value in **eGTouchL.ini** to match your using scenario.

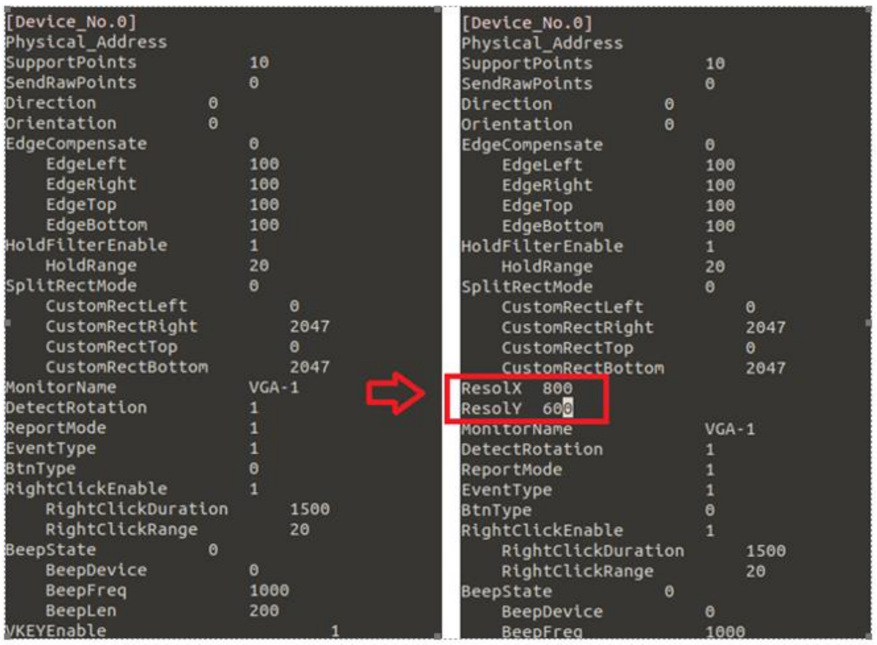
Screen Rotation	Setting Orientation
	Orientation = 0
	Orientation = 1
	Orientation = 2
	Orientation = 3

2. Reboot device.

8-3 OS can not map touch position to screen position

Some Linux embedded system do not have display server (X, Wayland..), so the resolution of touch input cannot be mapped to screen, in this case, you may need to specific your screen resolution manually, please follow the below steps.

1. Get screen resolution, and set resolution manually.
   (You can get screen resolution via **$ xrandr**)
2. For example, if your resolution is 800x600, add **ResolX** and **ResolY** to **eGTouchL.ini**



```
[Device_No.0]
Physical_Address
SupportPoints           10
SendRawPoints           0
Direction           0
Orientation         0
EdgeCompensate          0
    EdgeLeft            100
    EdgeRight           100
    EdgeTop             100
    EdgeBottom          100
HoldFilterEnable        1
    HoldRange           20
SplitRectMode           0
    CustomRectLeft          0
    CustomRectRight         2047
    CustomRectTop           0
    CustomRectBottom        2047
MonitorName             VGA-1
DetectRotation          1
ReportMode              1
EventType               1
BtnType                 0
RightClickEnable        1
    RightClickDuration      1500
    RightClickRange         20
BeepState           0
    BeepDevice          0
    BeepFreq            1000
    BeepLen             200
VKEYEnable                      1
```

```
[Device_No.0]
Physical_Address
SupportPoints           10
SendRawPoints           0
Direction           0
Orientation         0
EdgeCompensate          0
    EdgeLeft            100
    EdgeRight           100
    EdgeTop             100
    EdgeBottom          100
HoldFilterEnable        1
    HoldRange           20
SplitRectMode           0
    CustomRectLeft          0
    CustomRectRight         2047
    CustomRectTop           0
    CustomRectBottom        2047
ResolX  800
ResolY  600
MonitorName             VGA-1
DetectRotation          1
ReportMode              1
EventType               1
BtnType                 0
RightClickEnable        1
    RightClickDuration      1500
    RightClickRange         20
BeepState           0
    BeepDevice          0
    BeepFreq            1000
```

3. Reboot device.

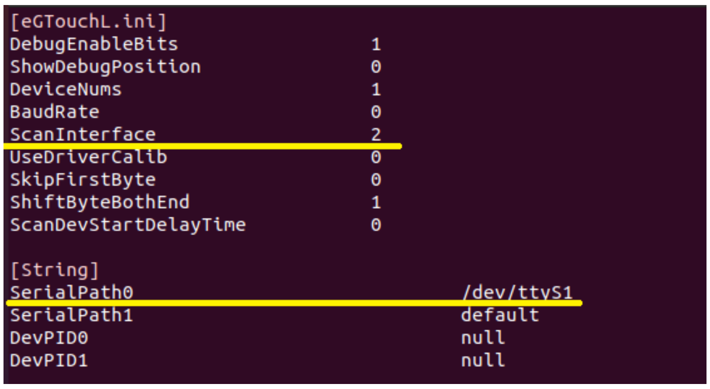
8-4 Touch is working, but click some items without reaction

Some desktops of Linux distribution have compatible issue to multi-touch event. In this case, connect touch device to other system, and observe is it can reproduce at the different system. If so, please upgrade your system or change to other system.

If you are using **PCAP** and you do not have multi-touch requirement, you also can change event type to single event, for details, please refer to 5.1 section.

8-5 eGTouchD can NOT find UART interface device

Check the setting of ScanInterface and SerialPath in **eGTouchL.ini**, please try to change ScanInterface setting to **2** and assign serial path with your input path, the definition of ScanInterface refer to 5.1 section, to figure out serial path can refer to 8.1 section.



```
[eGTouchL.ini]
DebugEnableBits              1
ShowDebugPosition            0
DeviceNums                   1
BaudRate                     0
ScanInterface                2
UseDriverCalib               0
SkipFirstByte                0
ShiftByteBothEnd             1
ScanDevStartDelayTime        0

[String]
SerialPath0                          /dev/ttyS1
SerialPath1                          default
DevPID0                              null
DevPID1                              null
```

After modifying the parameter, please reboot your system or restart driver to make it valid.

8-6 My UART device receive unexpected data from eGTouchD

As default, eGTouchD will scan USB and UART interface to find touch controller, eGTouchD use Vendor ID (VID) and Product ID (PID) to interpret USB interface touch device and send some data to UART to interpret UART interface touch device, this issue can be avoided by the below setting.

If you are using USB interface touch controller, please modify the ScanInterface to **1** in **eGTouchL.ini**, than eGTouchD will skip to scan UART interface.

If you are using UART interface touch controller, please refer to 8.5 section to specify ScanInterface and SerialPath in **eGTouchL.ini**, eGTouchD will only scan specified serial path. After modifying the parameter, please reboot your system or restart driver to make it valid.

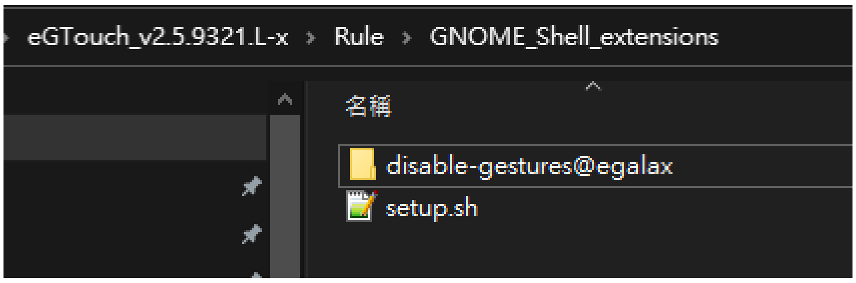
8-7 GNOME Gesture

GNOME 3.14 Support Touchscreen gestures for application and for system-wide actions. If you'd like to disable GNOME Gesture for your KIOSK system.

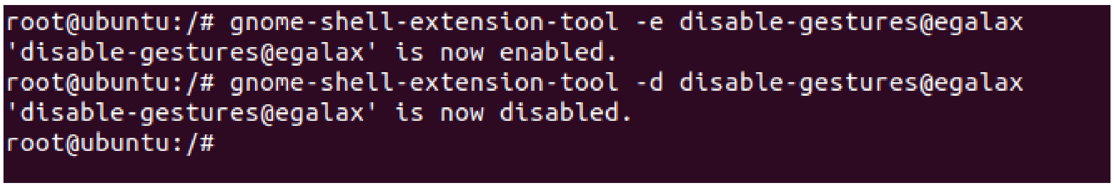
You can install gnome-shell-extensions-tool, and use EETI disable-gestures setup.sh script to install "disable-gestures" extensions.

./setup_ges.sh: #Install Gnome Extension: Disable-Gnome-Gesuture.

./setup_ges.sh uninstall: #Uninstall Gnome Extension: Disable-Gnome-Gesuture



After install "disable-gestures" extensions, you can use command "gnome-shell-extension-tool" to enable & disable this extensions.



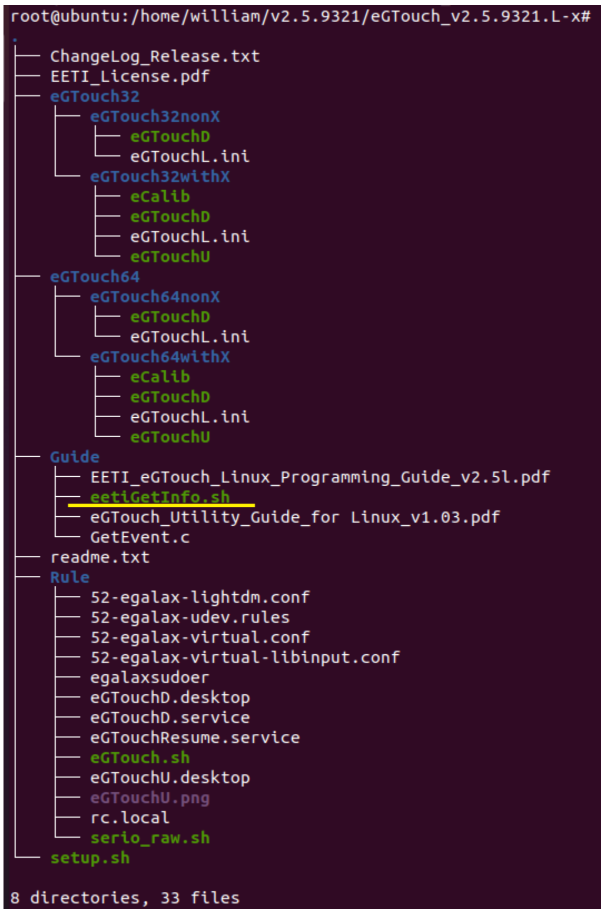
```
root@ubuntu:/# gnome-shell-extension-tool -e disable-gestures@egalax
'disable-gestures@egalax' is now enabled.
root@ubuntu:/# gnome-shell-extension-tool -d disable-gestures@egalax
'disable-gestures@egalax' is now disabled.
root@ubuntu:/#
```

Sec 9: Support

9-1 Need Support From EETI

If you have any problems when running the eGTouchD driver and the above FAQs still cannot solve your problem, please help to collect debugging information and provide a description of your issue. Collecting required information according to the following steps can help us understand your problem and provide assistance as soon as possible.

Please run **eetiGetInfo.sh** script as root to collect debugging information, the information may include your system information, kernel log, driver setting and driver information, we only use these information to assist your question.



```
root@ubuntu:/home/william/v2.5.9321/eGTouch_v2.5.9321.L-x#
.
├── ChangeLog_Release.txt
├── EETI_License.pdf
├── eGTouch32
│   ├── eGTouch32nonX
│   │   ├── eGTouchD
│   │   └── eGTouchL.ini
│   └── eGTouch32withX
│       ├── eCalib
│       ├── eGTouchD
│       ├── eGTouchL.ini
│       └── eGTouchU
├── eGTouch64
│   ├── eGTouch64nonX
│   │   ├── eGTouchD
│   │   └── eGTouchL.ini
│   └── eGTouch64withX
│       ├── eCalib
│       ├── eGTouchD
│       ├── eGTouchL.ini
│       └── eGTouchU
├── Guide
│   ├── EETI_eGTouch_Linux_Programming_Guide_v2.5l.pdf
│   ├── eetiGetInfo.sh
│   ├── eGTouch_Utility_Guide_for Linux_v1.03.pdf
│   └── GetEvent.c
├── readme.txt
├── Rule
│   ├── 52-egalax-lightdm.conf
│   ├── 52-egalax-udev.rules
│   ├── 52-egalax-virtual.conf
│   ├── 52-egalax-virtual-libinput.conf
│   ├── egalaxsudoer
│   ├── eGTouchD.desktop
│   ├── eGTouchD.service
│   ├── eGTouchResume.service
│   ├── eGTouch.sh
│   ├── eGTouchU.desktop
│   ├── eGTouchU.png
│   ├── rc.local
│   └── serio_raw.sh
└── setup.sh

8 directories, 33 files
```

This may takes few seconds, and you will find an **eeti.tar.gz** file in your current path.

Please attach **eeti.tar.gz**, describe your issue and contact us by mail: touch_fae@eeti.com

Sec 10: Appendix

If your system's <u>X-window version is 1.8.7 upwards</u>, <u>kernel version is not 3.8 to 3.12</u>, and <u>EETI touch controller type is not USB Resistive/SCAP touch</u>, please IGNORE this section.

10-1 Kernel patch: (for X-window version < 1.8.7)

If your system meets all below two conditions:

1.	**Interface**	**USB**
2.	**X.org version**	**< 1.8.7 or no X-window**

Please refer below instructions to do kernel blacklist patch, or driver would NOT be functional.

Please append following **RED** section into your source code.

If your kernel is **2.6.33 downwards**, please follow section **8-1.1**
If your kernel is **2.6.34 upwards**, please follow section **8-1.2**

10-1.1 kernel 2.6.33 downwards

1. /SourceCode/drivers/input/**evdev.c**

```
static struct input_device_id evdev_blacklist[] =
{ /* Added by EETI */
        {
        .flags  = INPUT_DEVICE_ID_MATCH_BUS | INPUT_DEVICE_ID_MATCH_VENDOR,
        .bustype = BUS_USB,
        .vendor = 0x0EEF,
        },
        {},     /* Terminating entry */
};

static struct input_handler evdev_handler = {
        .event = evdev_event,
        .connect = evdev_connect,
        .disconnect = evdev_disconnect,
        .fops = &evdev_fops,
        .minor = EVDEV_MINOR_BASE,
        .name = "evdev",
        .id_table = evdev_ids,
        .blacklist = evdev_blacklist, /* Added by EETI */
};
```

2. /SourceCode/drivers/input/**mousedev.c**

```
static struct input_device_id mousedev_blacklist[] =
{       /* Added by EETI */
        {
        .flags = INPUT_DEVICE_ID_MATCH_BUS | INPUT_DEVICE_ID_MATCH_VENDOR,
        .bustype = BUS_USB,
        .vendor = 0x0EEF,
        },
        {
        .flags  = INPUT_DEVICE_ID_MATCH_BUS | INPUT_DEVICE_ID_MATCH_VENDOR,
        .bustype = BUS_VIRTUAL,
        .vendor = 0x0EEF,
        },
        {},     /* Terminating entry */
};


static struct input_handler mousedev_handler = {
        .event = mousedev_event,
        .connect = mousedev_connect,
        .disconnect = mousedev_disconnect,
        .fops = &mousedev_fops,
        .minor = MOUSEDEV_MINOR_BASE,
        .name = "mousedev",
        .id_table = mousedev_ids,
        .blacklist = mousedev_blacklist, /* Added by EETI */
};
```

3. /SourceCode/drivers/input/**joydev.c**

```
static const struct input_device_id joydev_blacklist[] =
{
        {
        .flags = INPUT_DEVICE_ID_MATCH_EVBIT | INPUT_DEVICE_ID_MATCH_KEYBIT,
        .evbit = { BIT_MASK(EV_KEY) },
        .keybit = { [BIT_WORD(BTN_TOUCH)] = BIT_MASK(BTN_TOUCH) },
        },      /* Avoid itouchpads and touchscreens */
        {
        .flags = INPUT_DEVICE_ID_MATCH_EVBIT | INPUT_DEVICE_ID_MATCH_KEYBIT,
        .evbit = { BIT_MASK(EV_KEY) },
        .keybit = { [BIT_WORD(BTN_DIGI)] = BIT_MASK(BTN_DIGI) },
        },      /* Avoid tablets, digitisers and similar devices */
        {
        .flags = INPUT_DEVICE_ID_MATCH_BUS | INPUT_DEVICE_ID_MATCH_VENDOR,
        .bustype = BUS_VIRTUAL,
        .vendor = 0x0EEF,
        },      /* Added by EETI */
        { }     /* Terminating entry */
};

static struct input_handler joydev_handler = {
        .event = joydev_event,
        .connect = joydev_connect,
        .disconnect = joydev_disconnect,
        .fops = &joydev_fops,
        .minor = JOYDEV_MINOR_BASE,
        .name = "joydev",
        .id_table = joydev_ids,
        .blacklist = joydev_blacklist,
};
```

10-1.2 kernel 2.6.34 upwards

1. /SourceCode/drivers/input/**evdev.c**

```
static bool evdev_match(struct input_handler *handler, struct input_dev *dev)
{
        /* Avoid EETI USB touchscreens */
        #define VID_EETI 0x0EEF
        if ((BUS_USB == dev->id.bustype) && (VID_EETI == dev->id.vendor))
            return false;
        return true;
}

static struct input_handler evdev_handler = {
        .event = evdev_event,
        .match = evdev_match,  /* Added by EETI*/
        .connect = evdev_connect,
        .disconnect = evdev_disconnect,
        .fops = &evdev_fops,
        .minor = EVDEV_MINOR_BASE,
        .name = "evdev",
        .id_table = evdev_ids,
};
```

2. /SourceCode/drivers/input/**mousedev.c**

```
static bool mousedev_match(struct input_handler *handler, struct input_dev *dev)
{
        /* Avoid EETI USB touchscreens */
        #define VID_EETI 0x0EEF
        if ((BUS_USB == dev->id.bustype) && (VID_EETI == dev->id.vendor))
            return false;
        /* Avoid EETI virtual devices */
        if ((BUS_VIRTUAL == dev->id.bustype) && (VID_EETI == dev->id.vendor))
            return false;
        return true;
}
static struct input_handler mousedev_handler = {
        .event = mousedev_event,
        .match = mousedev_match,   /* Added by EETI */
        .connect = mousedev_connect,
        .disconnect = ousedev_disconnect,
        .fops = &mousedev_fops,
        .minor = MOUSEDEV_MINOR_BASE,
        .name = "mousedev",
        .id_table = mousedev_ids,
};
```

3. /SourceCode/drivers/input/**joydev.c**

```
static bool joydev_match(struct input_handler *handler, struct input_dev *dev)
{
        /* Avoid touchpads and touchscreens */
        if (test_bit(EV_KEY, dev->evbit) && test_bit(BTN_TOUCH, dev->keybit))
            return false;
        /* Avoid tablets, digitisers and similar devices */
        if (test_bit(EV_KEY, dev->evbit) && test_bit(BTN_DIGI, dev->keybit))
            return false;
        /* Avoid EETI virtual devices */
        #define VID_EETI 0x0EEF
        if (( BUS_VIRTUAL == dev->id.bustype) && (VID_EETI == dev->id.vendor))
            return false;
        return true;
}

static struct input_handler joydev_handler = {
        .event = joydev_event,
        .match = joydev_match,
        .connect = joydev_connect,
        .disconnect = joydev_disconnect,
        .fops = &joydev_fops,
        .minor = JOYDEV_MINOR_BASE,
        .name = "joydev",
        .id_table = joydev_ids,
};
```

10-2 Kernel patch (kernel 3.8~3.12 with USB resistive)

If your system meets all below three conditions:

1.	**Interface**	**USB**
2.	**Kernel version**	**3.8.x to 3.12.x**
3.	**ControllerType**	**Resistive or SCAP**

Please comment the following **RED** section in your source code.

/SourceCode/drivers/hid/**hid-core.c**

```
bool hid_ignore(struct hid_device *hdev)
{
   …
   switch (hdev->vendor) {
        …
        /*case USB_VENDOR_ID_DWAV:*/
            /* These are handled by usbtouchscreen. hdev->type is probably
             * HID_TYPE_USBNONE, but we say !HID_TYPE_USBMOUSE to match
             * usbtouchscreen. */
             /*if ((hdev->product == USB_DEVICE_ID_EGALAX_TOUCHCONTROLLER ||
                  hdev->product == USB_DEVICE_ID_DWAV_TOUCHCONTROLLER) &&
                  hdev->type != HID_TYPE_USBMOUSE)
                      return true;
              break;*/
        …
   }
   …
}
```