

# ***TouchKit driver user guide for Linux***

The TouchKit driver package contains a pre-compiled Xorg module and utility for X window. This driver supports both RS232 and USB TouchKit controllers. **The Xorg module version later than 1.06 supports PS/2 TouchKit controller as well.** It may need to rebuild kernel or build the kernel module “serio\_raw” if the kernel module “serio\_raw” does not exist for PS/2 controller. Please take a reference to the document **“How to rebuild kernel”** for detail procedure to rebuild the kernel or build the kernel module “serio\_raw”.

## **1. Installation with script file:**

User can use the script file **“setup.sh”** to install the driver package easily in a terminal window.

Syntax:

```
      sh setup.sh          # To install the TouchKit driver.
or    sh setup.sh uninstall # To remove the TouchKit driver.
or    sh setup.sh version  # To get the version of the installer.
```

Note that **the root permission is required to run this installation script.** Otherwise, below error messages would be popped up. ( Assume, the working system is Ubuntu 8.04 )

```
test@test-desktop:~$ sh setup.sh
```

(\*) Linux driver installer for TouchKit controller

(I) Begin to setup TouchKit driver.

(I) Checking user permission: **test**, you are NOT the supervisor.

**(E) The root permission is required to run this installer.**

The user can get the root permission by the following command.

For example:

```
      su          ( For general Linux )
or    sudo -s    ( For Ubuntu series only )
```

root@test-desktop:~# **sh setup.sh**      ( Driver package installation )

(\*) Linux driver installer for TouchKit controller

- (I) Begin to setup TouchKit driver.
- (I) Checking user permission: **root**, you are the supervisor.
- (I) Extract TouchKit driver package to **/usr/local/TouchKit\_x14**.
- (I) Create TouchKit utility **shortcut** in **/usr/bin**.
- (I) Copy X module **egalax\_drv.so** to **/usr/lib/xorg/modules/input**.

(Q) **Which interface of controller do you use?**

(I) [1] RS232 [2] PS/2 [3] USB: **1**

(Q) **Which COM port will be connected? e.g. /dev/ttyS0 (COM1)**

(A) Please input: **/dev/ttyS0**

(I) Found X configuration file **xorg.conf** in **/etc/X11**

(I) Add **touch configuration** into **/etc/X11/xorg.conf**

(I) **Please reboot the system for some changes to take effect.**

#### **1.) For serial RS232 interface:**

(Q) **Which interface controller do you use?**

(I) [1] RS232 [2] PS/2 [3] USB: **1**

(Q) **Which COM port will be connected? e.g. /dev/ttyS0 (COM1)**

(A) Please input: **/dev/ttyS4**

**Note that the user must input correct serial device node where the controller connected.**

For example:

**/dev/ttyS4**      ( Connected to COM5 )

#### **2.) For PS/2 interface:**

(Q) **Which interface controller do you use?**

(I) [1] RS232 [2] PS/2 [3] USB: **2**

(I) Using interface: **PS/2**

(I) **Please make sure the kernel module for PS/2 controller is available.**

(I) For details, see the document **"How to rebuild kernel.pdf"**.

### 3.) For USB interface:

- (Q) Which interface controller do you use?
- (I) [1] RS232 [2] PS/2 [3] USB: 3
- (I) Using interface: USB
- (I) Found a HID compliant touch controller.
- (I) Found kernel module `usbtouchscreen`.
- (I) It is highly recommended that add it into blacklist.
- (Q) Do you want to add it into blacklist? (y/n) y
- (I) Add kernel module `usbtouchscreen` into `/etc/modprobe.d/blacklist`.

Note that it is highly recommended that add inbuilt kernel module “`usbtouchscreen`” or “`touchkitusb`” into blacklist to avoid conflict if the touch controller is HID compliant device and the inbuilt kernel module “`usbtouchscreen`” or “`touchkitusb`” is loaded in the kernel.

- (Q) Which interface controller do you use?
- (I) [1] RS232 [2] PS/2 [3] USB: 3
- (I) Using interface: USB
- (I) Found a non-HID compliant touch controller.
- (W) No suitable kernel module found.
- (I) The user must build `tkusb` kernel module for touch controller.
- (I) For details, see the document “`How to build module.pdf`”.

Note that the user needs to build the TouchKit kernel module “`tkusb`” for touch controller if the inbuilt kernel module “`usbtouchscreen`” or “`touchkitusb`” does NOT exist.

root@test-desktop:~# `sh setup.sh uninstall` ( Driver package un-installation )

(\*) Linux driver installer for TouchKit controller

- (I) Begin to remove TouchKit driver.
- (I) Checking user permission: `root`, you are the supervisor.
- (I) Removed TouchKit driver from `/usr/local/TouchKit_x14`.
- (I) Removed TouchKit utility `shortcut`.
- (I) Removed `X` module.
- (I) Found X configuration file `xorg.conf` in `/etc/X11`.
- (I) Removed `touch configuration` in `/etc/X11/xorg.conf`.
- (I) The TouchKit driver has been removed successfully.
- (I) Please reboot the system for some changes to take effect.

root@test-desktop:~# **sh setup.sh version** ( The version of the installer )

(\*) Linux driver installer for TouchKit controller

(l) **Version: 1.00.0324**

For more information about X configuration setting, please see the following section  
"Installation of the Xorg module manually".

## 2. Installation of the Xorg module manually:

To install X module, it needs to copy the Xorg module "egalax\_drv.so" to the X window input module directory and configure the Xorg configuration file.

### a.) copy the Xorg module

The X window input module directory varies by distribution. The user can use the following command to give you a clue as to where the Xorg modules are located on your system:

```
find / -name mouse_drv.so
```

Output:

```
/usr/lib/xorg/modules/input/mouse_drv.so      ( for 32 bit )  
or /usr/lib64/xorg/modules/input/mouse_drv.so  ( for 64 bit )
```

Copy the Xorg module "egalax\_drv.so" to the correct X window input module directory.

For example:

```
cp egalax_drv.so /usr/lib/xorg/modules/input  ( for 32 bit )  
or cp egalax_drv.so /usr/lib64/xorg/modules/input ( for 64 bit )
```

## b.) configure the Xorg configuration file

Edit the Xorg configuration file (e.g. `/etc/X11/xorg.conf`) and add the configuration used by the driver to connect to the device installed on your system.

- (1) Add an Input device declaration in “ServerLayout” section.

For example:

```
Section "ServerLayout"
    ...
    ...
    InputDevice      "EETI"      "SendCoreEvents"
EndSection
```

**Note:** *If more than one TouchKit controllers ( 2 or more TouchKit touchscreens ) are used for the system, the user must add multiple InputDevice declarations in the “ServerLayout” section with different names.*

For example:

```
InputDevice      "EETI1"      "SendCoreEvents"
InputDevice      "EETI2"      "SendCoreEvents"
```

- (2) Configure the Xorg module configuration for TouchKit device.

For each InputDevice section declared in the “ServerLayout” section, the user will need to create additional separate configuration in the `xorg.conf` file.

For only one USB device in the system:

```
Section "InputDevice"
    Identifier      "EETI"
    Driver          "egalax"
    Option          "Device"      "usbauto"
    Option          "Parameters"  "/var/lib/eeti.param"
    Option          "ScreenNo"    "0"
EndSection
```

**Note:** *The Identifier line must be the same as the name declared it in the section “ServerLayout”.*

For multiple devices ( RS232 and USB ) in the multiple monitors system:

**Section "InputDevice" ( For RS232 device )**

Identifier **"EETI1"**  
Driver "egalax"  
Option "Device" **"/dev/ttyS0"**  
Option "Parameters" **"/var/lib/eeti1.param"**  
Option "ScreenNo" **"0"**

**EndSection**

**Section "InputDevice" ( For USB device )**

Identifier **"EETI2"**  
Driver "egalax"  
Option "Device" **"usbauto"**  
Option "Parameters" **"/var/lib/eeti2.param"**  
Option "ScreenNo" **"1"**

**EndSection**

**Note:** *If more than one TocuhKit controllers ( 2 or more TouchKit touchscreens ) are used for the system, the user must edit each option "Parameters" with different file names.*

**Driver**

The Driver line must be set to correct Xorg module name which is copied to the X window input module directory. For example:

**Driver "egalax"**

**Option "Device"**

The "Device" option must be assigned so that the driver can read the data from the device port. The "Device" is a char device usually found in /dev.

If the "Device" is set to a pipe, the driver will not work correctly. The user must determine where the controller was connected to set the "Device" option.

**Note:** *The driver supports three interfaces, serial RS232, PS/2 and USB so that all users need to indentify which interface controller was connected in the system before set the option "Device".*

### 1.) For **serial RS232** interface:

The “Device” should be set to correct serial device port name, e. g. **/dev/ttyS0** or **/dev/ttyS1**. Besides, the user must ensure the I/O address and the IRQ number are the same as the BIOS setting. For details about checking these settings, see the following command.

```
setserial /dev/ttyS0 -a
```

Some users connected the serial RS232 touch device on **/dev/ttyS4** or **/dev/ttyS5** and the IRQ setting for serial touch device is **NOT** available. In this case **the user can use IRQ 0 for serial touch device** by the following command so that the system will use polling instead of interrupt to access the serial touch device.

```
setserial /dev/ttyS4 irq 0
```

### 2.) For **PS/2** interface:

The “Device” should be set to correct PS/2 auxiliary device port name, e. g. **/dev/serio\_raw0**. By default, the PS/2 interface TouchKit device will be directed to mouse device automatically under Linux kernel 2.6 or later. **The user must make sure the kernel supports PS/2 auxiliary port as a char device like kernel 2.4 does and the using Xorg module version is later than 1.09. See another document “How to rebuild kernel” for details. Note that the PS/2 device should be connected correctly before power on.**

### 3.) For **USB** interface:

There are three kernel modules support USB TouchKit device.

- (1) Inbuilt HID kernel module: **“usbhid”**
- (2) Inbuilt USB kernel module: **“touchkitusb”** or **“usbtouchscreen”**
- (3) TouchKit USB kernel module: **“tkusb”**

**If the version of Xorg module is 1.08 or later and the system has only one USB TouchKit device, the “Device” option can be set to “usbauto” so that the Xorg module will attempt to determine the communication device port automatically. It might be better to manually configure the option “Device” declaration by user. For example:**

**Option “Device” “usbauto”**

It is highly recommended that use inbuilt kernel module instead of tkusb kernel module for USB touch device if the user does NOT want to compile any source code manually. The user can identify the USB device type and then load suitable kernel module for USB touch device by the following.

1. Use the command "**lsusb**" to check the USB touch device type.

For example:

```
lsusb -v -d 0eef:0001
```

Part of output:

Interface Descriptor:           **( For HID device )**

bLength	9
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
<b>bInterfaceClass</b>	<b>3 Human Interface Device</b>
bInterfaceSubClass	0 No SubClass
bInterfaceProtocol	0 None
iInterface	0

Interface Descriptor:           **( For non-HID device )**

bLength	9
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
<b>bInterfaceClass</b>	<b>255 Vendor Specific Class</b>
bInterfaceSubClass	255 Vendor Specific Subclass
bInterfaceProtocol	255 Vendor Specific Protocol
iInterface	0

2. Use the command "**cat**" to get more USB information from the system file "devices" in /proc/bus/usb.

```
cat /proc/bus/usb/devices
```



Part of output:

P: Vendor=0eef ProdID=0001 Rev=1.00 ( For HID device )

S: Product=USB TouchController

I: If#= 0 Alt=0 #EPs=1 Cls=03(HID ) Sub=00 Prot=00 Driver=usbhid

P: Vendor=0eef ProdID=0001 Rev=1.00 ( For non-HID device )

S: Product=USB TouchController

I: If#= 0 Alt=0 #EPs=1 Cls=ff(vend. ) Sub=ff Prot=ff Driver=usbtouchscreen

**Note:** *If the inbuilt kernel module “usbhid”, “touchkitusb” or “usbtouchscreen” is loaded for USB touch device, the user needs to modify the mouse setting in the xorg.conf file as well to prevent from the mouse driver read the data from the specified device node. Set the “Device” option for mouse to a real device node like “/dev/input/mouseX” instead of default device class “/dev/input/mice”. The user can use the following command to check which real device node is used for mouse.*

*cat /proc/bus/input/devices*

Part of output:

N: Name="ImPS/2 Generic Wheel Mouse"

P: Phys=isa0060/serio1/input0

S: Sysfs=/class/input/input2

H: Handlers=mouse1 event2

For example:

Section "InputDevice"

Identifier "Mouse0"

Driver "mouse"

Option "Device" "/dev/input/mouse1"

...

...

EndSection

### 3-1) Use inbuilt HID kernel module:

The Linux kernel 2.6 supports HID compliant TouchKit device with inbuilt HID kernel module. If the user is working with HID compliant TouchKit device and HID kernel module. Then, there should be a device node for this HID compliant TouchKit device in `/dev` or `/dev/usb`. The user needs to identify which `/dev/hiddevX` device node represents the HID compliant TouchKit device and set the proper "Device" option. For example:

**Option "Device" `/dev/hiddev0`**

**Note:** *If the system has only one HID compliant TouchKit device, the "Device" option for Xorg module version 1.06 or later can be set to*

**Option "Device" `"hiddev"`**

or **Option "Device" `"hiddevs"`**

*so that the Xorg module will determine HID device node for HID compliant TouchKit device automatically.*

**Note:** *The HID compliant TouchKit device should NOT work with inbuilt USB kernel module `"touchkitusb"` or `"usbtouchscreen"`. Instead, it should work with `"usbhid"` or TouchKit USB kernel module `"tkusb"`. It is suggested to add `"touchkitusb"` and `"usbtouchscreen"` into the file `"blacklist"` in `/etc/hotplug` or `/etc/modprobe.d` to avoid conflicts.*

### 3-2) Use inbuilt USB kernel module:

The Xorg module version later than 1.06 supports event device node. If the USB TouchKit device finds the working kernel module as inbuilt `"touchkitusb"` or `"usbtouchscreen"`, there should be an event device node for USB TouchKit device in `/dev/input`, e.g. `/dev/input/event4`. The user needs to identify which event device node represents the USB TouchKit device. Then, set the proper "Device" option.

For example:

**Option "Device" `/dev/input/event4`**

**Note:** If the system has only one USB TouchKit device with inbuilt kernel module “touchkitusb” or “usbtouchscreen”, the “Device” option for Xorg module version 1.06 or later can be set to

Option “Device” “**event\***”  
or Option “Device” “**events**”

so that the Xorg module will determine event device node for USB TouchKit device automatically.

**Note:** The user can check which event device node is used for USB touch device manually by the following command.

```
cat /proc/bus/input/devices
```

Part of output:

I: Bus=003 Vendor=0eef Product=0001 Version=0210

N: Name=“USB TouchController”

H: Handlers=mouse2 **event4**

### 3-3) Use TouchKit USB kernel module:

If vendor provided the USB kernel module “tkusb.ko” was loaded and the device file “/dev/tkpanel0” were created for USB TouchKit device, this “Device” option should be set to “/dev/tkpanel0” as follow. For details about building the TouchKit USB kernel module “tkusb.ko”, see another document “**How to build module**”.

Option “Device” “**/dev/tkpanel0**”

**Note:** The Xorg module version 1.06 or later supports inbuilt kernel module “touchkitusb”, “usbtouchscreen” and “usbhid” for USB TouchKit devices. It is highly recommended to use inbuilt kernel module instead of tkusb, by doing this all users **do NOT need to compile any source code during driver installation**.

**Note:** If the user prefers to use the kernel module “tkusb”, it is suggested to add “touchkitusb” and “usbtouchscreen” into the file “blacklist” in /etc/hotplug or /etc/modprobe.d to avoid conflicts.

### Option “Parameters”

The user can assign a writeable file path for the driver to save the parameters. All of the control parameters will be saved in this file. A separate file will be needed for a TouchKit device. It is recommended the files are saved in the variable library directory. For example:

### Option “Parameters” “/var/lib/eeti.param”

### Option “ScreenNo”

The user can define which screen number the TouchKit touchscreen will work with. If the system has only one TouchKit touchscreen, this value should be set to “0”. For example:

### Option “ScreenNo” “0”

### c.) Restart X window

Restart X window to make sure the Xorg module is loaded. It is enough to logout of X window and log back in.

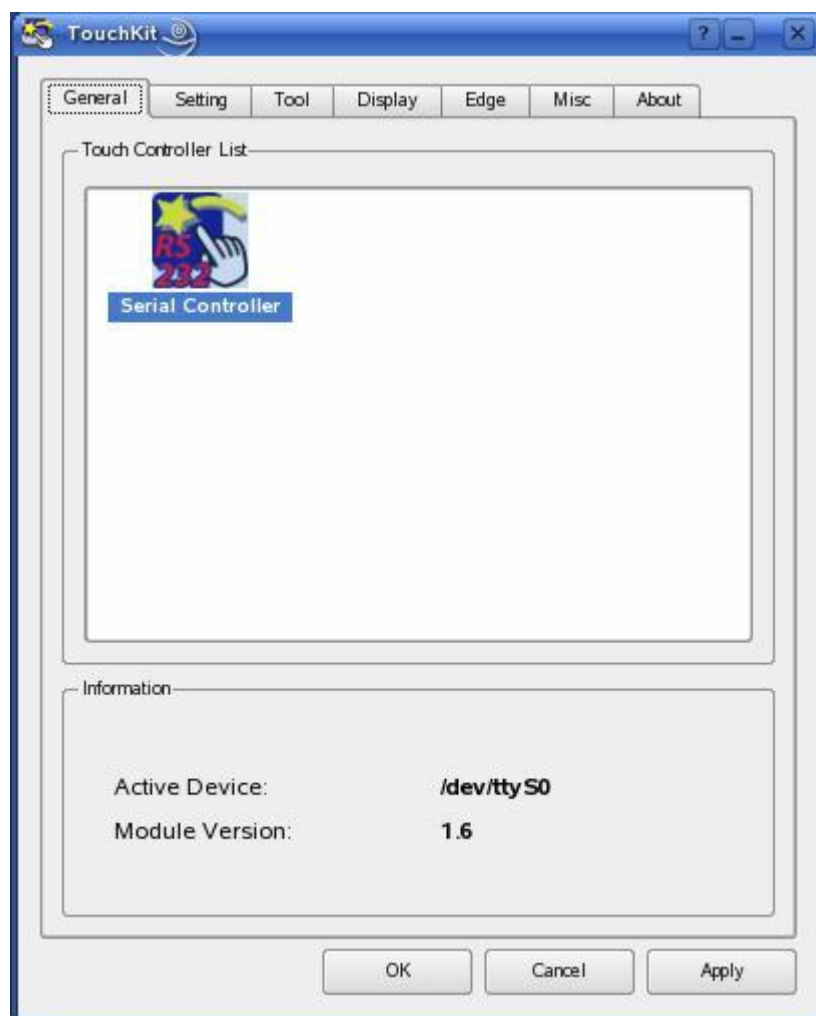
**Note:** *If the system is running SELinux. SELinux might block access from the driver to the TouchKit device. This can cause many different problems. Before installing the driver issue the “setenforce 0” command to disable SELinux enforcement. Once the driver is installed check the system audit logs for denials between Xorg and TouchKit devices. If see these, the user will need to create a policy or upgrade the latest policy via internet to allow those accesses before re-enabling SELinux with “setenforce 1”.*

### 3. Utility

TouchKit driver package for X window provides all users with a configuration tool utility for TouchKit touchscreen. The utility contains property pages **General**, **Setting**, **Tool**, **Display**, **Edge**, **Misc** and **About**.

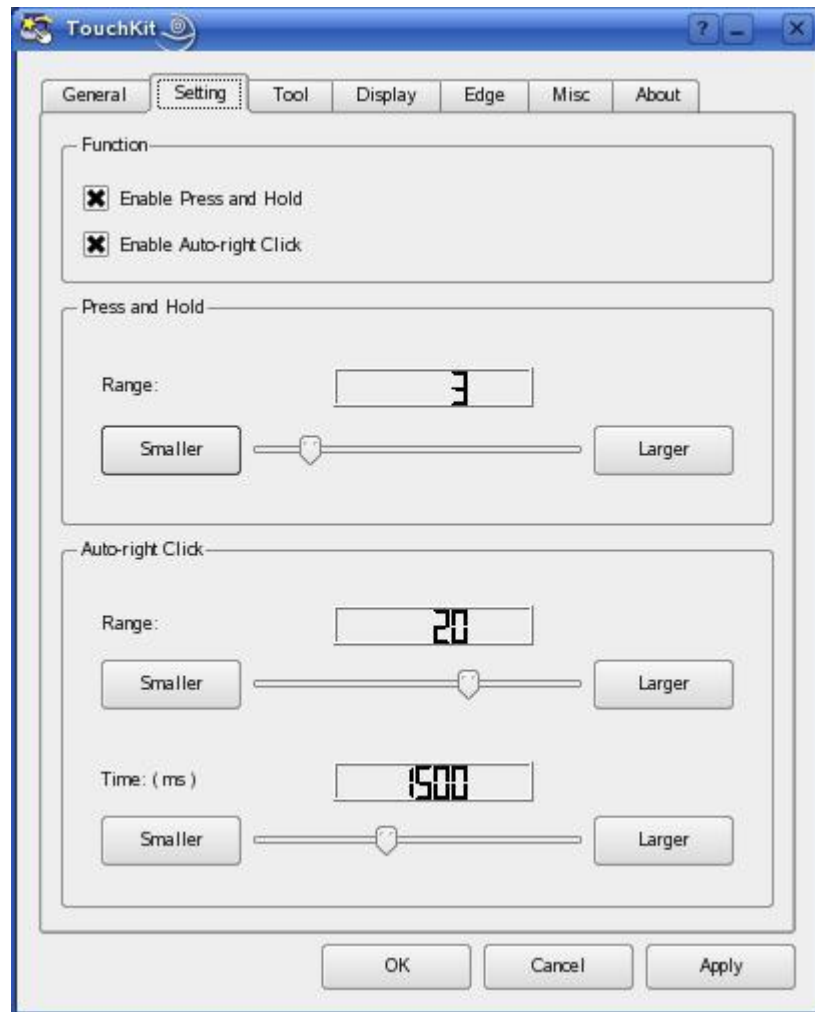
**Note:** *Please make sure the Xorg module is installed successfully. Otherwise, the utility does not work properly.*

#### 3.1) General Property:



The utility enumerates TouchKit touchscreen controller installed in this system. *All of the enumerated TouchKit controllers will be listed in the “Touch Controller List” Window.* It also shows device name which communication device node the device is connected. In addition, the Xorg module version will be shown in the Information window, too.

### 3.2) Setting Property:



Some options can be configured for mouse emulation.

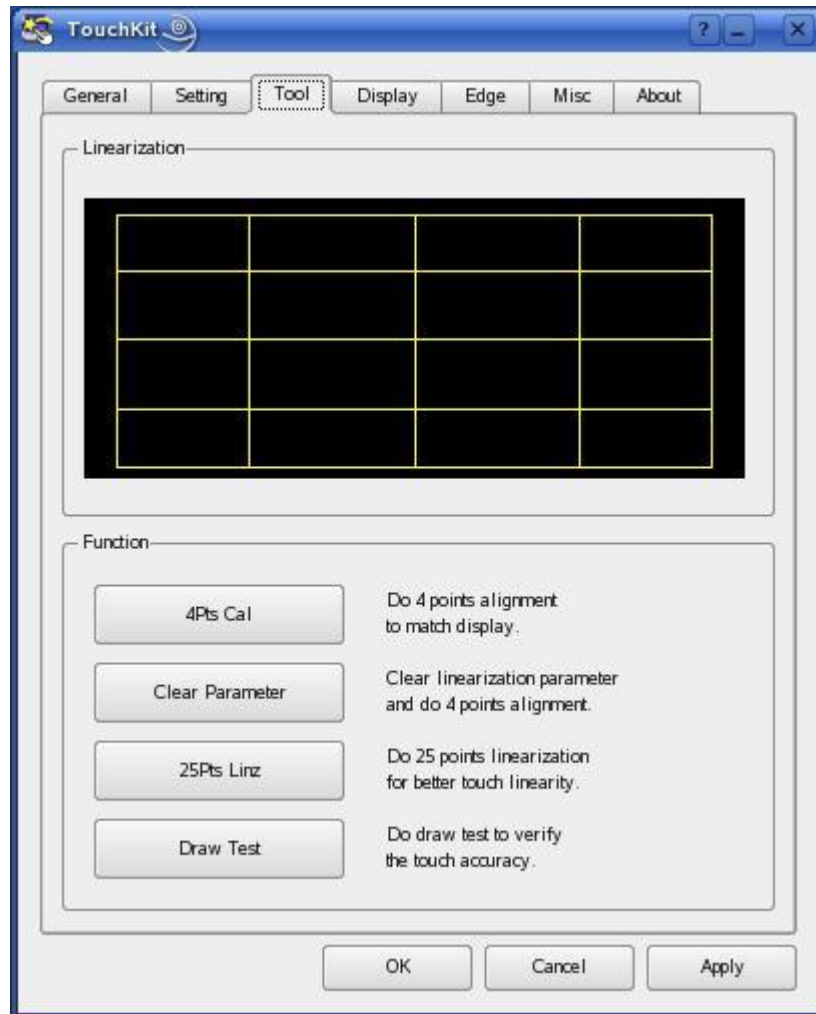
#### Press and Hold:

Press and hold at the same point. In some application, the application program does not want to receive too many touch points for the touch held at same position, **the user can check the checkbox to enable constant touch function so that the driver would not report other points unless the position difference between current position and last position is greater than the “Range” value or lift up.** The range of the point difference can be configured with the “Range” slider. **This feature does NOT support event type device.**

#### Auto Right Click:

**The driver generates a mouse right click event automatically whenever the driver detects the touchscreen was press and hold for a while** if the checkbox checked. The duration and the range for auto-right click emulation can be configured with the “Range” slider and the “Time” slider.

### 3.3) Tool Property:



TouchKit utility provides all users with tools for calibration and testing.

#### Linearization map:

After 25 points calibration finished, the linearity of the touchscreen will be shown in this linearization map.

#### 4 Pts Cal

TouchKit utility provides 4 points calibration for touchscreen alignment. **The touchscreen can work correctly only after calibration.** When the user presses this “4Pts Cal” button to do 4 points calibration, a calibration window will pop up to guide user to complete the calibration.



The user should **press the calibration symbol until it goes to next point or disappears.**

The user can abort this calibration by pressing **<ESC>** key.

#### Clear Parameter:

Press this button to **clear the 25 points linearization parameters and do 4 points calibration.**

All of the 25 points linearization parameters will be cleared if the button pressed.

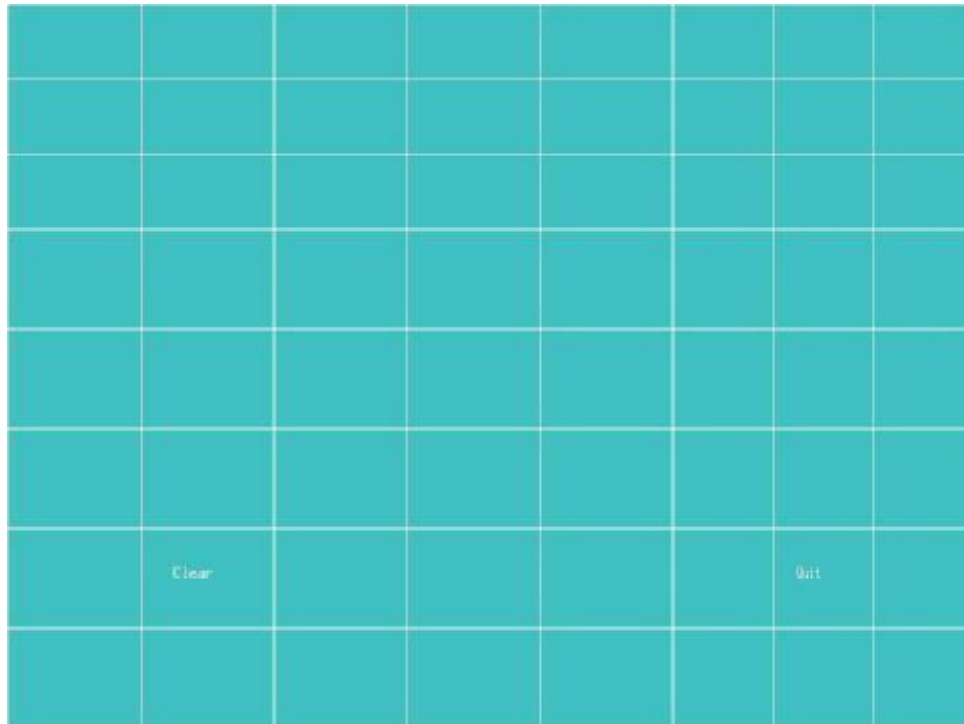
#### 25Pts Linz:

Press this button to do 25 points calibration. **After calibration, the previous 25 points linearization parameters will be overwritten by the new parameters.**

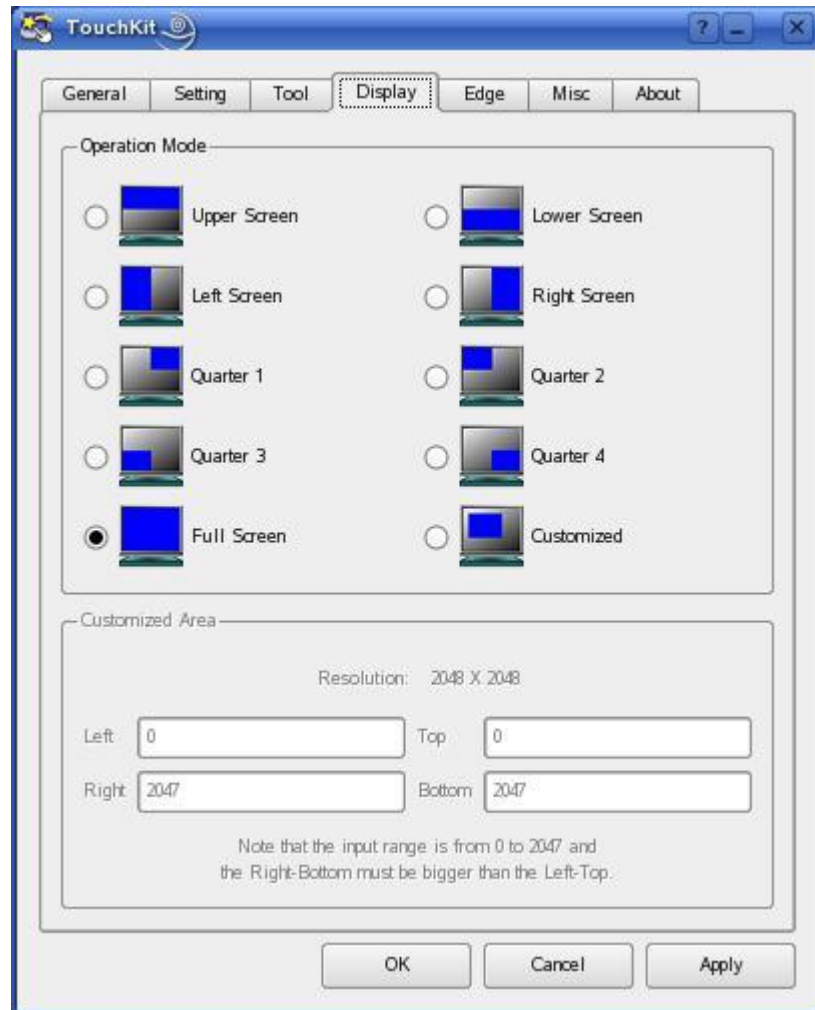


**Draw Test:**

After linearization or alignment, the user can press this button to **check the touch accuracy, linearization, response, etc...**



### 3.4) Display Property:

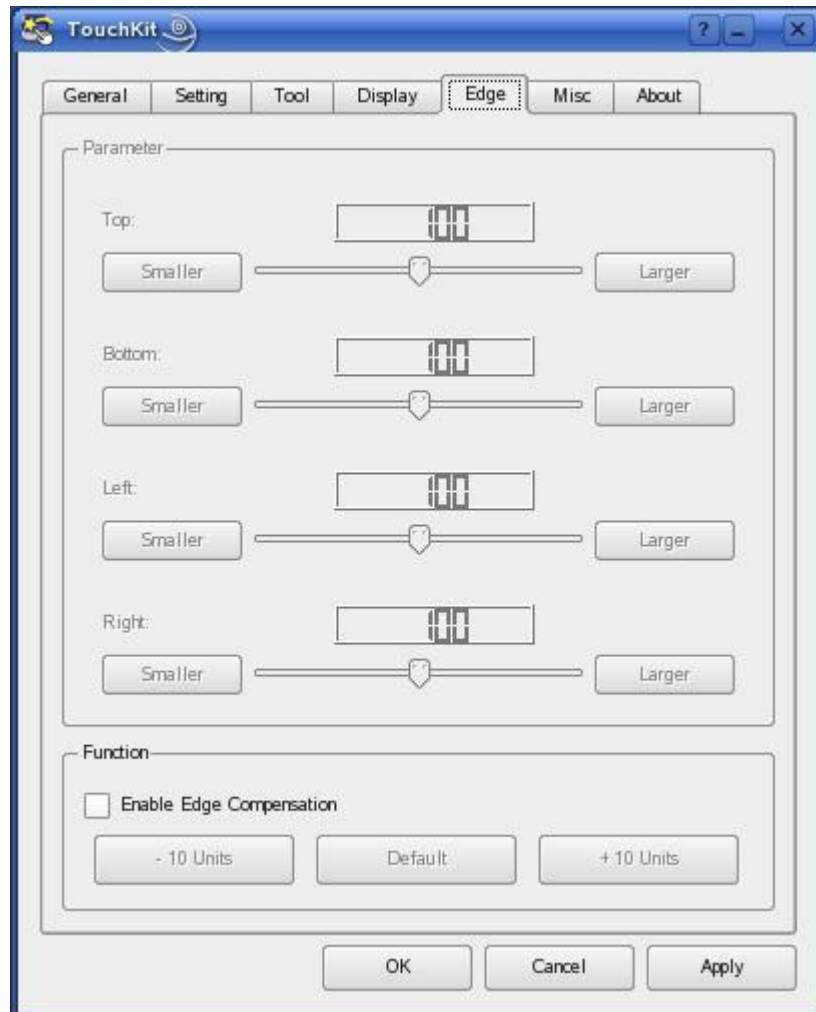


TouchKit utility supports split display feature.

The working area of the touchscreen can be mapped to anywhere on the video display.

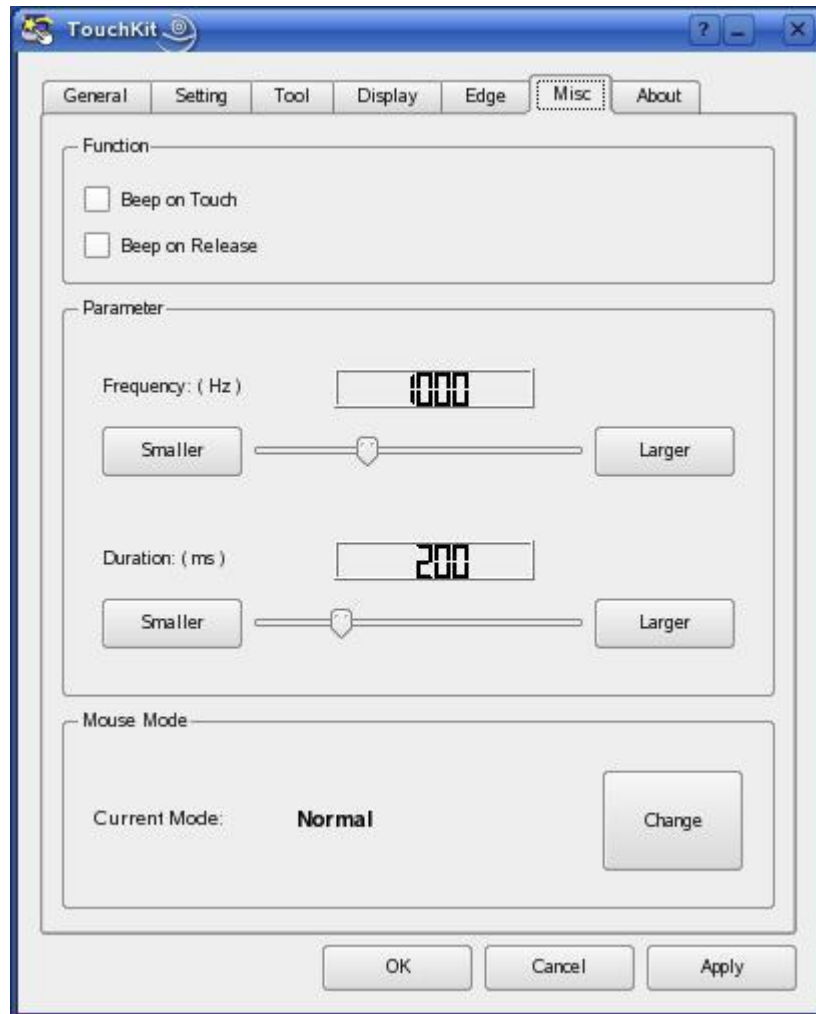
The user can choose any options to define where the touchscreen will be mapped. However, if the **“Customized”** is selected, it needs to enter the area to map to. The utility TouchKit always **assume the resolution is 2048 X 2048. If the video resolution is not 2048 X 2048, the user has to calculate the area manually.**

### 3.5) Edge Property



TouchKit utility supports edge compensation to make sure that the touchscreen can **achieve the display edge area**.

### 3.6 ) Misc Property



#### Beep On Touch:

When this function enabled, the driver will generate a beep sound whenever it detects the touch state changed from untouched state to touched state.

#### Beep On Release:

When this function enabled, the driver will generate a beep sound whenever it detects the touch state changed from touched state to untouched state.

#### Frequency:

Change this **Frequency** value to change the beep sound frequency.

#### Duration:

Change this **Duration** value to change the duration of the beep sound.

***Note that this feature is NOT supported in some Linux distributions.***

### Mouse Emulation Mode:

TouchKit driver supports three mouse emulation modes.

#### 1.) Normal Mode:

The touch driver reports a left button down event when it detects a pen down and a left button up event when it receives a lift off.

#### 2.) Click On Touch:

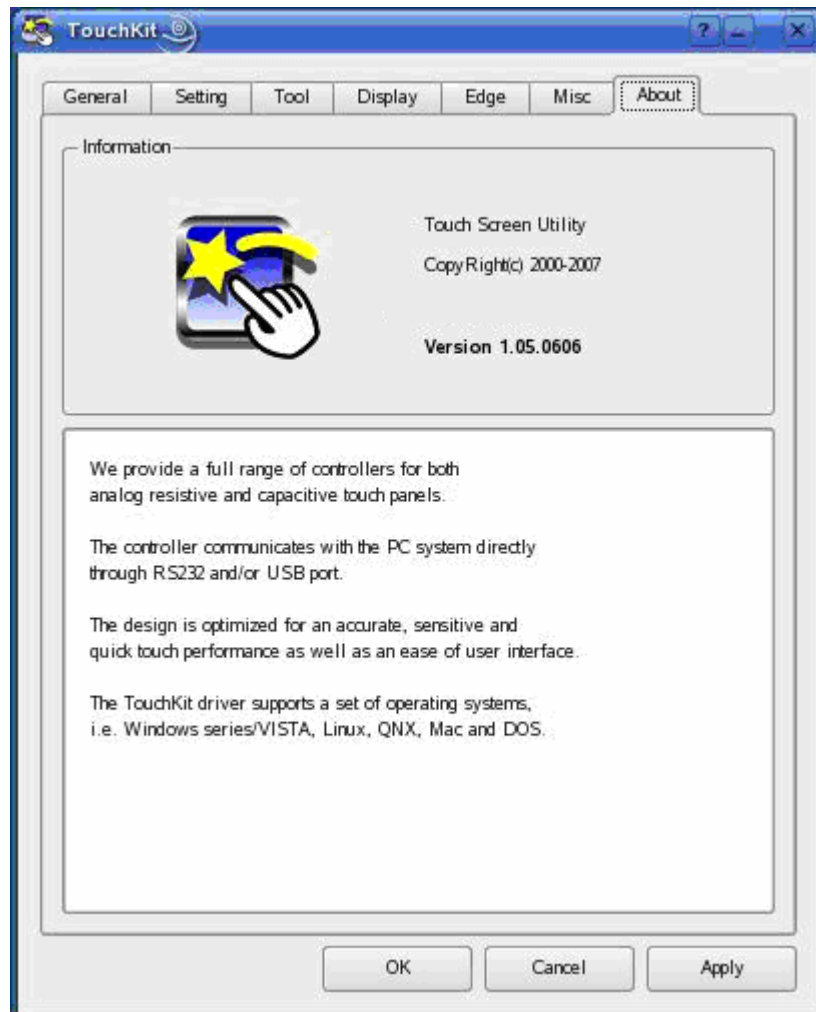
The touch driver reports a left button click event when it detects a pen down. Then, it does not report other events until it detects next a pen down.

#### 3.) Click On Release:

The touch driver does not report any event until it detects a lift off. It reports a left button click when it detects a lift off.

**Note:** *If the mouse emulation mode is changed to **Click On Touch** or **Click On Release**, the features **Press and Hold** and **Auto-right Click** will be **disabled** automatically.*

### 3.7 ) About Property



Information about TouchKit.