TouchKit driver user guide for Linux with xorg

This driver/utility package supports TouchKit controllers only. All of versions support RS232 and USB interfaces of controllers. **The module version later than 1.06 supports PS/2 interface.** To support PS/2 interface, it needs to rebuild kernel and specific kernel module. For details about PS/2 support, please see another eGalax document **“How to rebuild kernel”**.

Note: *This driver does not support kernel 2.4.x. It is used for kernel 2.6.x only.*

1. Xorg driver module installation:

To install Xorg driver, all users have to do is copy Xorg driver module to specific directory and configure Xorg configuration file.

a.) copy Xorg driver module

User can use **“X –version”** instruction command in a terminal window to check the version of current X window.

The target file folders are different between the version before 7.0 and later.

For the version before 7.0, the target file folder is **“/usr/X11R6/lib/modules/input”**.

If the Linux system is **64 bit**, the target folder is **“/usr/X11R6/lib64/modules/input”**.

For the version 7.0 or later, the target file folder is **“/usr/lib/xorg/modules/input”**.

If the Linux system is **64 bit**, the target folder is **“/usr/lib64/xorg/modules/input”**.

In addition, there are also two different kinds of Xorg driver module format for different versions of X window.

For the version before 6.9, the X module file for TouchKit is **“egalax_drv.o”**.

For the version 6.9 or later, the X module file for TouchKit is **“egalax_drv.so”**.

After checking the version of X window with the instruction **“X –version”**, copy the correct X module files to the correct target path.

For example:

Before version 6.9:

“cp egalax_drv.o /usr/X11R6/lib/modules/input”

for version 6.9:

“cp egalax_drv.so /usr/X11R6/lib/modules/input”

for version 7.0 or later:

“cp egalax_drv.so /usr/lib/xorg/modules/input”

b.) edit /etc/X11/xorg.conf

(1) Add an input device declaration in “ServerLayout” section. For example:

```
Section “ServerLayout”
   …
        InputDevice    “EETI”    “SendCoreEvents”
EndSection
```

Note: *If many TouchKit controllers are used for the system, please add multiple InputDevice declaration in above “ServerLayout” section with different name.*

(2) Add “InputDevice” Section for TouchKit device. For example:

```
Section “InputDevice”
        Identifier  “EETI”
        Driver      “egalax”
        Option       “Device”       “/dev/ttyS0”
        Option      “Parameters”  “/etc/egalax.cal”
        Option      “ScreenNo”     “0”
 EndSection
```

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

Option “Device”

The “Device” option must be assigned so that the driver can read the data from this device node. This “Device” option must be a char device. If this “Device” is set to a pipe, the driver will not work correctly. User must be sure where the controller was connected to set the “Device” option.

This driver supports serial RS232, PS/2 and USB interfaces.

1.) For serial RS232 interface:

This “Device“ should be set to correct serial port device name, such as /dev/ttyS0 or /dev/ttyS1. Make sure the I/O address and IRQ are correct.

2.) For PS/2 interface:

This “Device” should be set to correct PS/2 auxiliary port device name, too, such as /dev/psaux. By default, the PS/2 interface devices will be directed to mouse devices automatically under kernel 2.6 or later. So, please make sure the kernel is rebuilt to support PS/2 auxiliary port and the using driver module version is later than 1.6. See another eGalax document **“How to rebuild kernel”** for details.

3.) For USB interface:

For details about USB device driver information, user can use **"cat"** instruction command in a terminal window to get more information from **/proc/bus/usb/devices** or **/proc/bus/input/devices** file.

3-1) Use TouchKit kernel module:

If vendor provided **tkusb.ko** kernel module was loaded for the USB device. And, the device node **"/dev/tkpanel0"** and **"/dev/tkpanel1"** were created for TouchKit USB device, this "Device" option should be set to **"/dev/tkpanel0"** or **"/dev/tkpanel1"**.

User can use **"insmod"** instruction command to insert kernel module into kernel and then use **"mknod"** instruction command to create device node, such as **"/dev/tkpanel0"** or **"/dev/tkpanel1"**.

For example,
```
insmod /[absolute path]/tkusb.ko
mknod /dev/tkpanel0 c 180 180
```

Note: *For details about building a kernel module **"tkusb.ko"**, please see another eGalax document **"How to build module"**.*

Note: *The driver module version 1.6 or later supports **"touchkitusb"** and **"usbtouchscreen"** internal kernel module for eGalax TouchKit USB devices. Please see 3-3) for details.*

3-2) Use HID compliant driver:

Linux kernel 2.6 supports HID compliant device with its inbox HID driver. If user is working with HID compliant TouchKit device and inbox HID driver. Then, there should be a device node for this HID compliant device in /dev. User has to identify which /dev/hiddev device node represents this HID compliant device. Then, set the proper Device option. For example, **"Device" "/dev/hiddev0"**

Note*: If the system has only one HID compliant TouchKit device, the Device option for driver module version 1.6 or later can be set to*

"Device" "hiddev*"

or

"Device" "hiddevs"

so that the driver module will scan HID compliant TouchKit device automatically.

3-3) Use internal kernel module:

Driver module version later than 1.6 supports event nodes. If the USB TouchKit device find the working driver as **"touchkitusb"** or **"usbtouchscreen"**, there should be one event device node in **/dev/input** for USB TouchKit device. If user can identify which event device node represents the USB TouchKit device, for example, **/dev/input/event2** represents the USB TouchKit device. Then, the "Device" option should be set as **"Device" "/dev/input/event2"**.

Note: *If the Device option is set to event device like "/dev/input/eventX", please modify the mouse setting in xorg.conf as well to prevent from the mouse driver read the data from the specified event device. Set the "Device" option for mouse to a real device like "/dev/input/mouseX" instead of device class "/dev/input/mice".*

User can check which real device node is used for mouse via below instruction in a terminal window.

cat /proc/bus/input/devices

Note: *If the system has only one USB TouchKit device with internal kernel module **"touchkitusb"** or **"usbtouchscreen"**, the Device option can be set to*

"Device" "event*"

or

"Device" "events"

so that the driver module will scan event device nodes for TouchKit device automatically.

Option "Parameters"

User should assign a writable file path for the driver module to save the driver parameters for the device. All of the driver parameters will be saved in this file.

Option "ScreenNo"

User can define which screen number the touchscreen will works with.

This value must be set to "0" if user has a touchscreen only.

c.) Restart X window

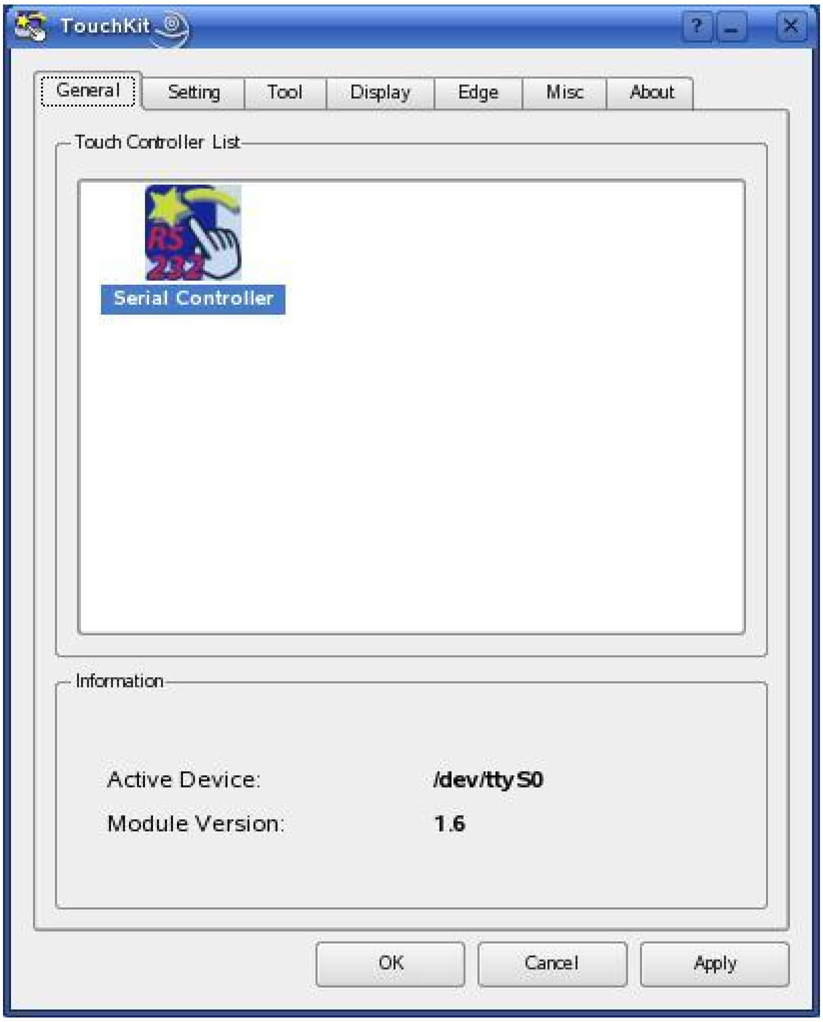
Restart X window to make sure the X module driver is loaded.

2. Utility

TouchKit driver package for Xorg provides user with a configuration tool utility for TouchKit controller. The utility contains property pages **General, Setting, Tool, Display, Edge, Misc** and **About**.

Note: *Make sure the X module is installed correctly and the root permission is required to run this utility. Otherwise, it does not work correctly.*

2.1) General Property:

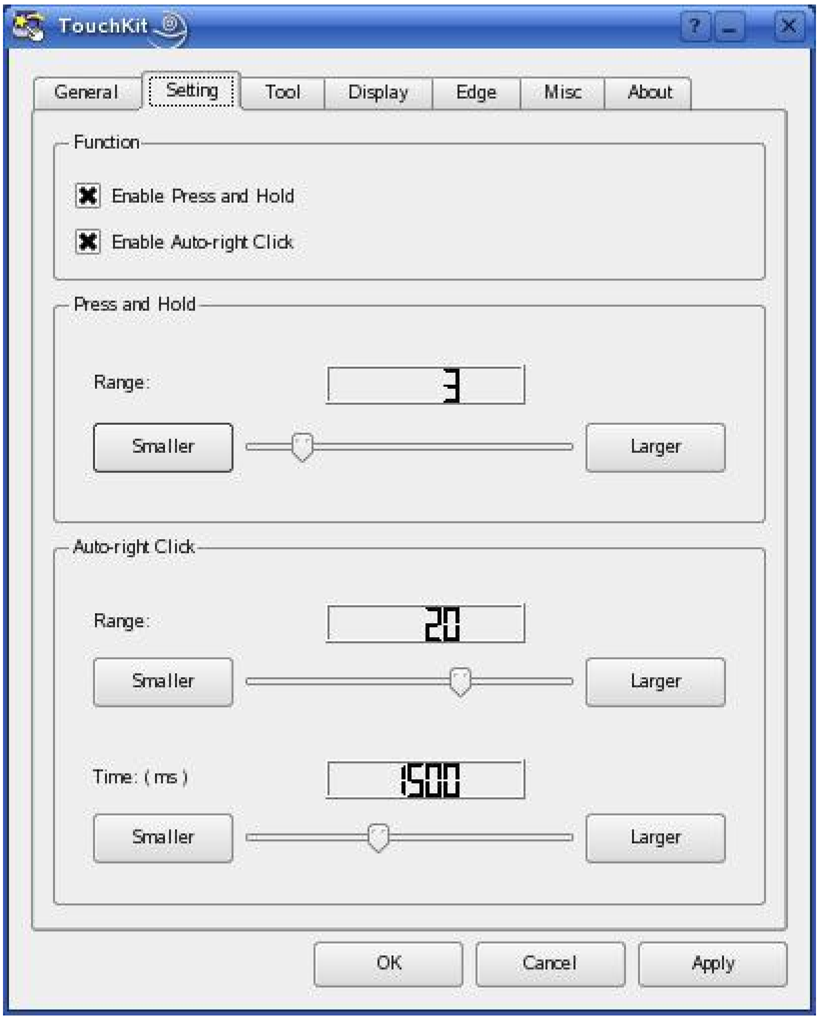


Active Device: /dev/ttyS0
Module Version: 1.6

The utility enumerates TouchKit touchscreen controller installed in this system. All of the enumerated TouchKit controllers will be list in the **"Touch Controller List"** Window. It also shows device node name which communication device node the device is connected.

In addition, the X driver module version will be shown in the Information window, too.

2.2) Setting Property:



Function

☒ Enable Press and Hold

☒ Enable Auto-right Click

Press and Hold

Range: 3

Auto-right Click

Range: 20

Time: (ms) 1500

Some options can be configured for mouse emulation.

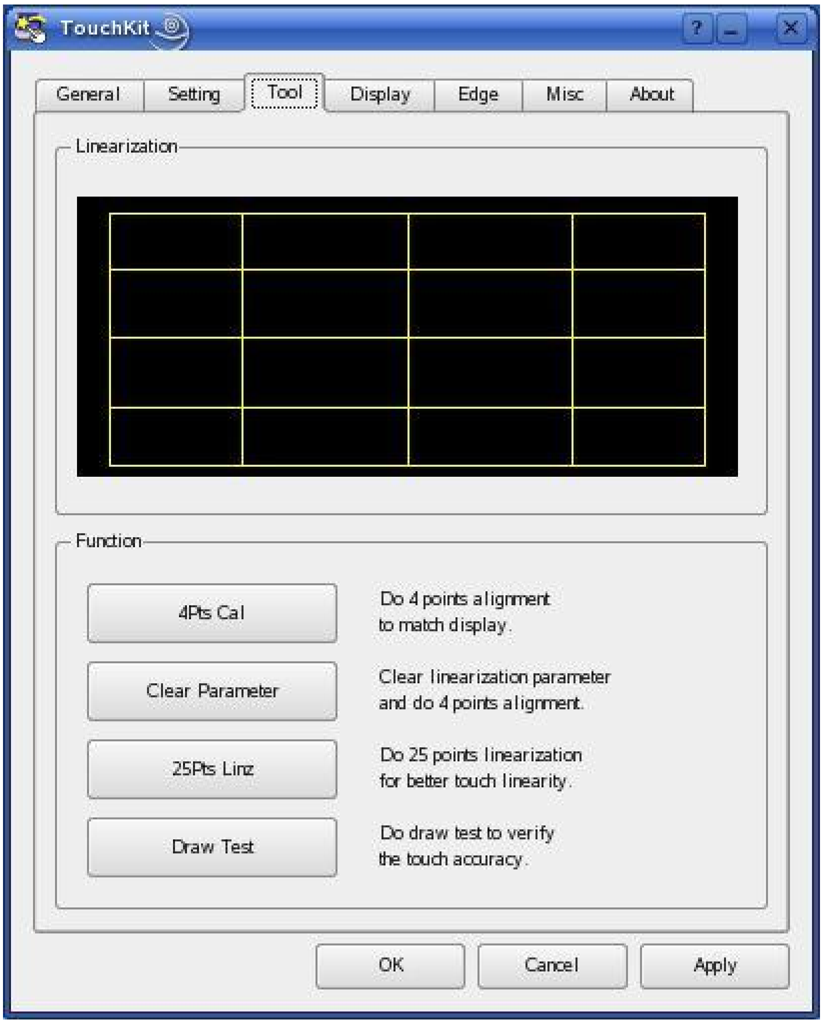
Press and Hold :

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 this checkbox to enable constant touch function so that the driver would not report second points unless the position is different or lift up. The range of the point difference can be configured with the Range slider control. If constant touch function enabled, the driver would report touch points only when the point difference between current point and last point is greater than this range value or when it detects a lift up.

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 this checkbox checked. The duration and the range for auto-right click emulation can be configured with the Range slider and the Time slider.

2.3) Tool Property:



4Pts Cal: Do 4 points alignment to match display.

Clear Parameter: Clear linearization parameter and do 4 points alignment.

25Pts Linz: Do 25 points linearization for better touch linearity.

Draw Test: Do draw test to verify the touch accuracy.

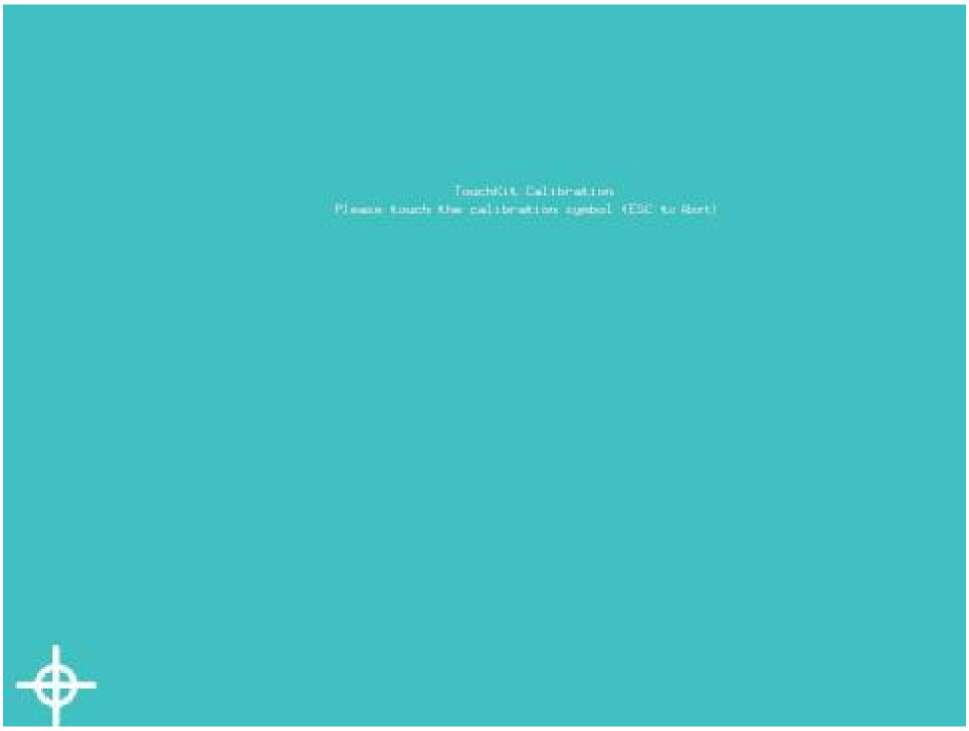
TouchKit utility provides users with tools for calibration and testing.

Linearization map:

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

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.



TouchKit Calibration
Please touch the calibration symbol (ESC to Abort)

The user just **press the calibration symbol until it goes to next point or disappears**. User can abort this calibration by pressing **<ESC>** key.

Clear Parameter:

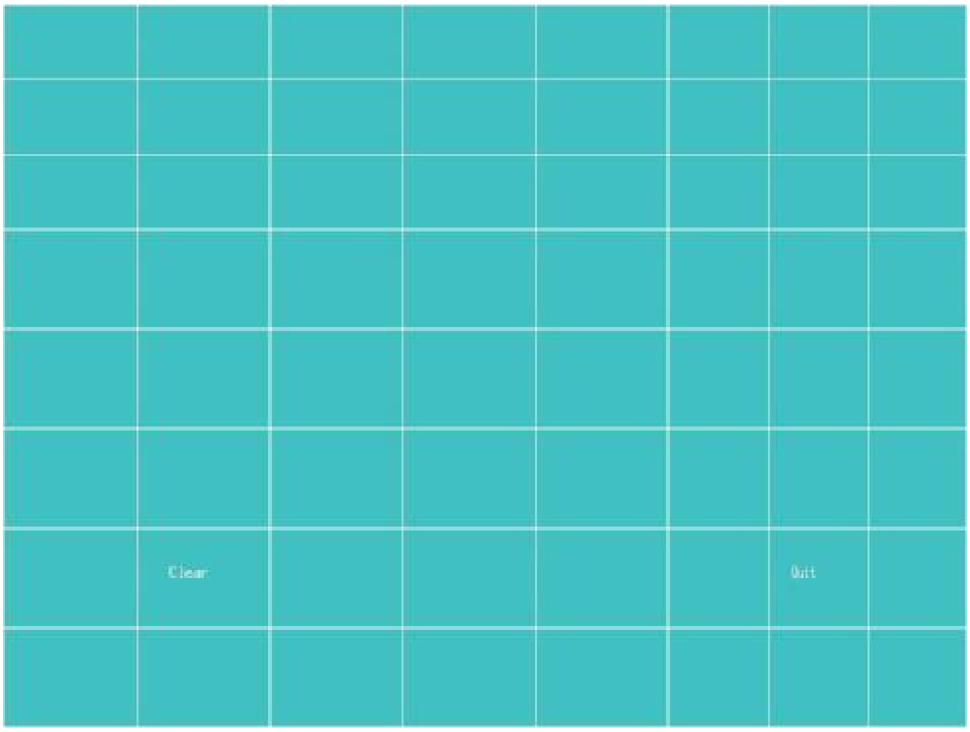
Press this button to clear the 25 points linearization parameters and do 4 points calibration again. All of the 25 points linearization parameters will be cleared if the button pressed.

25Pts Linz:

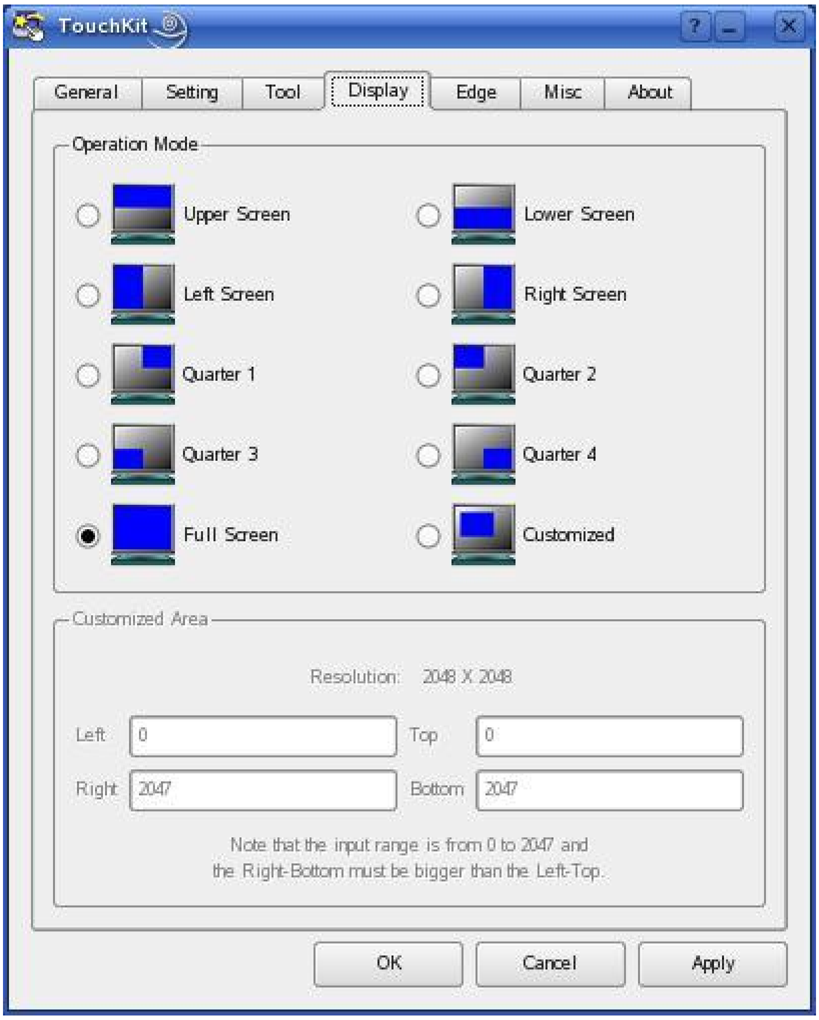
Press this button to do 25 points calibration. After calibration, the previous 25 points linearization parameters will be overwritten by the new parameters.

Draw Test:

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



2.4) Display Property:



Operation Mode

○ Upper Screen ○ Lower Screen
○ Left Screen ○ Right Screen
○ Quarter 1 ○ Quarter 2
○ Quarter 3 ○ Quarter 4
● Full Screen ○ Customized

Customized Area

Resolution: 2048 X 2048

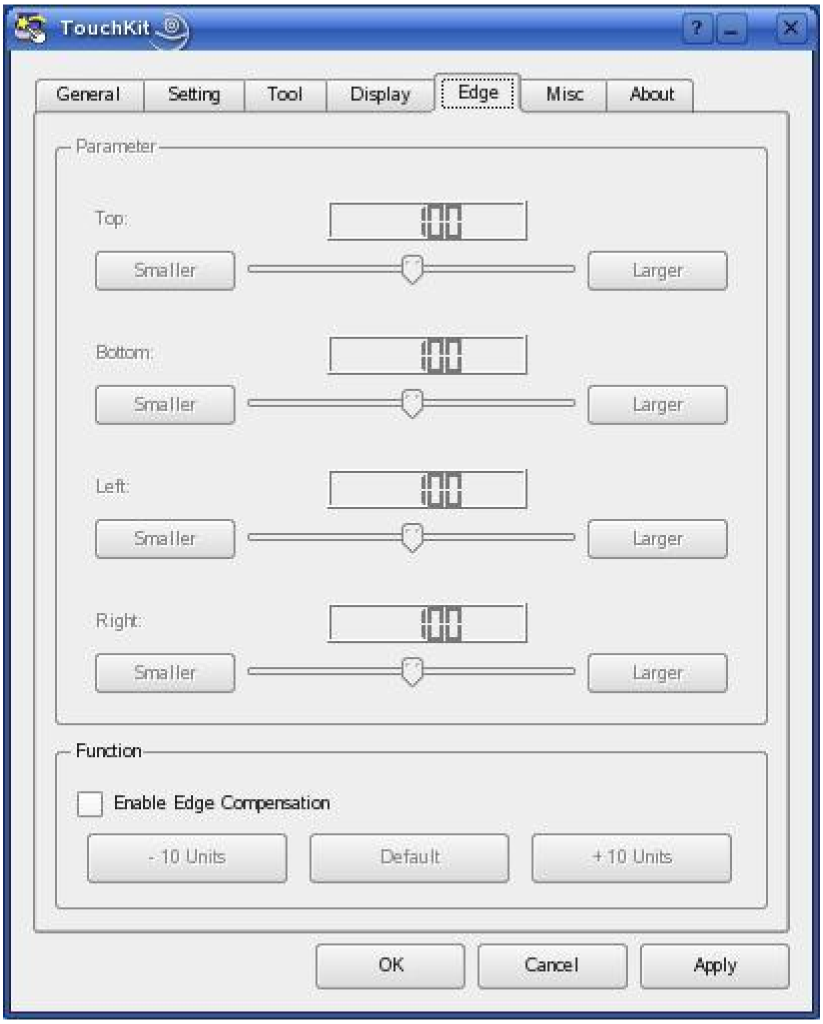
Left: 0 Top: 0
Right: 2047 Bottom: 2047

Note that the input range is from 0 to 2047 and
the Right-Bottom must be bigger than the Left-Top.

TouchKit utility supports split display feature.
The working area of the touchscreen can be mapped to anywhere on the video display.

User can choose any options to define where the touchscreen will be mapped. However, if the **"Customized"** is selected, it needs to key in the area to map to. TouchKit always **assume the resolution is 2048X2048**. **If the video resolution is not 2048X2048, the user has to calculate the area manually**.

2.5) Edge Property



Top: 100

Bottom: 100

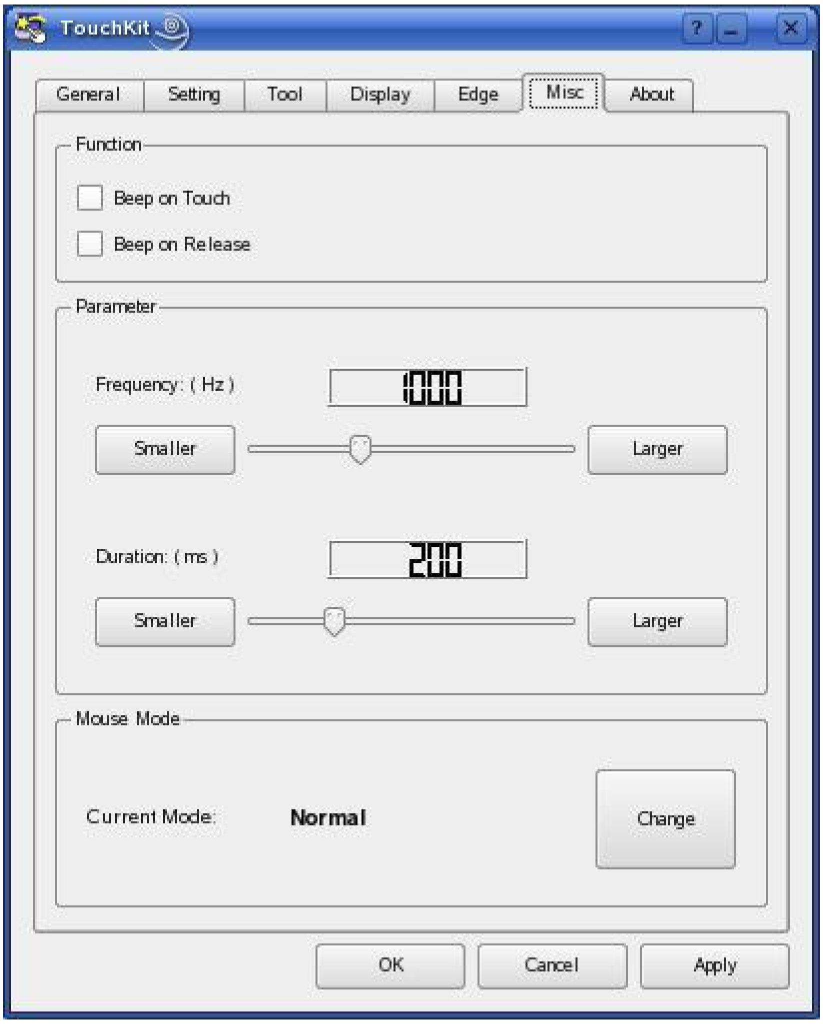
Left: 100

Right: 100

☐ Enable Edge Compensation

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

2.6) Misc Property



☐ Beep on Touch

☐ Beep on Release

Frequency: (Hz) 1000

Duration: (ms) 200

Current Mode: **Normal**

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:

Which 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.

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.

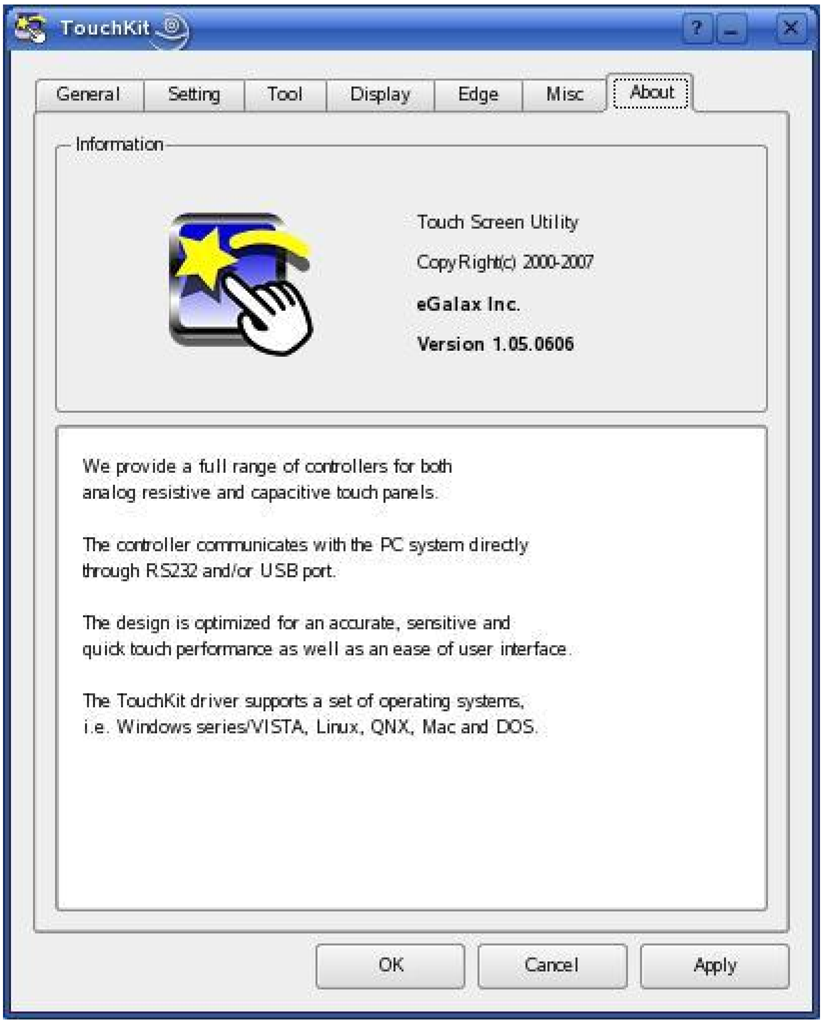
2.) **Click On Touch:**

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.

2.7) About Property



Touch Screen Utility

CopyRight(c) 2000-2007

eGalax Inc.

Version 1.05.0606

We provide a full range of controllers for both
analog resistive and capacitive touch panels.

The controller communicates with the PC system directly
through RS232 and/or USB port.

The design is optimized for an accurate, sensitive and
quick touch performance as well as an ease of user interface.

The TouchKit driver supports a set of operating systems,
i.e. Windows series/VISTA, Linux, QNX, Mac and DOS.

Information about TouchKit.