TouchKit driver user guide for Linux

The TouchKit driver archive 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. If the **serio_raw** kernel module does not exist, it may need to rebuild kernel or build the **serio_raw** kernel module for PS/2 touch device. Please take a reference to the document **"How to rebuild kernel"** for detailed procedure to rebuild kernel or build the **serio_raw** kernel module.

1. Installation with script file:

User can run the **setup.sh** script to install the driver archive 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 distribution is Ubuntu 8.04)

test@test-desktop:~$ **sh setup.sh**
(*) Linux driver installer for TouchKit controller

(I) Check 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)

(*) Driver archive installation:

root@test-desktop:~# **sh setup.sh**

(*) Linux driver installer for TouchKit controller

(I) Check user permission: root, you are the supervisor.

(I) Begin to setup TouchKit driver.

(I) Extract TouchKit driver archive to /usr/local/TouchKit32.

(I) Create TouchKit utility shortcut in /usr/bin.

(I) Create TKCal tool shortcut in /usr/bin.

(I) Check X window version: 1.4.x

(I) Copy X module: x14/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: /etc/X11/xorg.conf

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

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

(I) After booting, type “TouchKit” to do calibration.

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 has to 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 inbuilt 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.

(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 inbuilt kernel module for touch controller found.

(I) It is needed to 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 in the kernel.

(*) Driver archive un-installation:

root@test-desktop:~# **sh setup.sh uninstall**

(*) Linux driver installer for TouchKit controller

(I) Check user permission: root, you are the supervisor.

(I) Begin to remove TouchKit driver.

(I) Removed TouchKit driver archive from /usr/local/TouchKit32.

(I) Removed TouchKit utility shortcut.

(I) Removed TKCal tool shortcut.

(I) Removed X module.

(I) Removed blacklist usbtouchscreen from /etc/modprobe.d/blacklist.

(I) Removed touch configuration from /etc/X11/xorg.conf.

(I) The TouchKit driver has been removed successfully.

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

(*) The version of the installer:

root@test-desktop:~# **sh setup.sh version**

(*) Linux driver installer for TouchKit controller

(I) Version: 1.03.1301

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 Xorg module, user has to copy the Xorg module relating to your version of X window to the X input modules directory and then configure the Xorg configuration file.

a.) copy the Xorg module

There are two pre-compiled Xorg modules for X window as follows:

(1) For versions of X window prior to 6.9 use the **egalax_drv.o** module.

(2) For versions of X window from 6.9 to upwards use the **egalax_drv.so** module.

User can run **"X –version"** command to check your running version of X window.

Output:

X Window System Version 6.8.2

Release Date: 9 February 2005

X Protocol Version 11, Revision 0, Release 6.8.2

…

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

find /usr -name mouse_drv.*o

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 relating to your version of X window to the correct X input modules directory. For example:

(for 32 bit)

cp egalax_drv.o /usr/X11R6/lib/modules/input (for X version prior to 6.9)

or cp egalax_drv.so /usr/X11R6/lib/modules/input (for X version 6.9)

or cp egalax_drv.so /usr/lib/xorg/modules/input (for X version from 7.0 to upwards)

(for 64 bit)

cp egalax_drv.o /usr/X11R6/lib64/modules/input (for X version prior to 6.9)

cp egalax_drv.so /usr/X11R6/lib64/modules/input (for X version 6.9)

or cp egalax_drv.so /usr/lib64/xorg/modules/input (for X version from 7.0 to upwards)

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    "Touch"   "SendCoreEvents"
EndSection
```

Note: *If more than one TouchKit controllers are used for the system, the user must add multiple InputDevice declarations in the "ServerLayout" section with different names.*

For example:

```
InputDevice    "Touch1" "SendCoreEvents"
InputDevice    "Touch2" "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  "Touch"
    Driver      "egalax"
    Option      "Device"        "usbauto"
    Option      "Parameters"    "/var/lib/touch.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  “Touch1”
    Driver      “egalax”
    Option      “Device”       “/dev/ttyS0”
    Option      “Parameters”   “/var/lib/touch1.param”
    Option      “ScreenNo”     “0”
EndSection

Section “InputDevice”          ( For USB device )
    Identifier  “Touch2”
    Driver      “egalax”
    Option      “Device”       “usbauto”
    Option      “Parameters”   “/var/lib/touch2.param”
    Option      “ScreenNo”     “1”
EndSection
```

Note: *If more than one TouchKit controllers 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 input modules directory. For example:

```
Driver      ”egalax”
```

Option “Device”

The “Device” option must be assigned so that the driver can read the data from the device node. The device node 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 name of serial device node, 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, take a reference to 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 name of PS/2 auxiliary device node, e. g. **/dev/serio_raw0**. By default, the PS/2 touch 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 device node 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 usbhid module for **HID** touch device.

(2) Inbuilt touchkitusb / usbtouchscreen module for **non-HID** touch device

(3) TouchKit tkusb module for **both HID and non-HID** touch devices.

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

Option “Device” “usbauto”

It is highly recommended to use inbuilt kernel module instead of tkusb kernel module for USB touch device, by doing this all users do NOT need to compile any source code during driver installation. The user can identify the USB class of touch device and check which kernel module is loaded for USB touch device by the following.

1. Run the command **"lsusb –v –d 0eef:0001"** to check the USB class of touch device in a terminal window.

Part of output:

Interface Descriptor:	**(HID compliant device)**
bLength	9
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	**3 Human Interface Device**
bInterfaceSubClass	0 No SubClass
bInterfaceProtocol	0 None
iInterface	0

Interface Descriptor:	**(non-HID compliant device)**
bLength	9
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	**255 Vendor Specific Class**
bInterfaceSubClass	255 Vendor Specific Subclass
bInterfaceProtocol	255 Vendor Specific Protocol
iInterface	0

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

Part of output:

P: Vendor=0eef ProdID=0001 Rev=1.00 **(HID compliant 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 **(non-HID compliant 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, there is always a default mouse driver which will run simultaneously with TouchKit driver if the user does NOT prevent the mouse driver from reading. It is extremely important that set the "Device" option for mouse to a real mouse device node like "/dev/input/mouseX" instead of default device node "/dev/input/mice". Otherwise, the user will get double click events and all kind of strange things. The user can run the following command to check which real mouse 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*

*Part of X configuration file: e.g. **/etc/X11/xorg.conf***

```
Section "InputDevice"
    Identifier    "Mouse0"
    Driver        "mouse"
    Option        "Device" "/dev/input/mouse1"
    …
EndSection
```

If the above section does not exist in the X configuration file, the user has to append it manually.

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, there should be a device node for the HID compliant TouchKit device in /dev or /dev/usb. The user needs to identify which hiddevX device node represents the HID compliant TouchKit device and set the proper "Device" option. For example:

Option "Device" "/dev/hiddev0"

Since the kernel 2.6.27 or later does NOT support hiddevX device node, there should be a device node called hidrawX instead of hiddevX for HID compliant TouchKit device in /dev. The user needs to identify which hidrawX device node represents the HID compliant TouchKit device and set the proper "Device" option.

For example:

Option "Device" "/dev/hidraw0"

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"***
or ***Option "Device" "hidraw*"***
or ***Option "Device" "hidraws"***

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

Note: *The HID compliant touch device should NOT work with inbuilt 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* ***blacklist*** *file 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 the directory /dev/input, e.g. /dev/input/event4. The user needs to identify which event device node represents the USB TouchKit device and set the proper "Device" option. For example:

Option "Device" "/dev/input/event4"

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***

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.

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.

Option "Device" "/dev/tkpanel0"

For details about building the TouchKit USB kernel module **tkusb.ko**, see another document **"How to build module"**.

Note: *The Xorg module version 1.06 or later supports inbuilt kernel module **touchkitusb**, **usbtouchscreen** and **usbhid** for USB TouchKit device. 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 **tkusb** kernel module, it is suggested to add **touchkitusb** and **usbtouchscreen** into the **blacklist** file 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 an TouchKit device. It is recommended the files are saved in the variable library directory. For example:

Option "Parameters" "/var/lib/touch.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"

Option "AutoMapping"

When working with X server 1.4 or later, two different absolute coordination conversion algorithms are implemented in different Linux distributions. After calibration, if user can do draw test and get correct touch position with TouchKit utility, but the mouse cursor does NOT follow the touch position with big difference. The user can set this option with value "1" as example below to enable the mapping feature in Xorg module to work around this problem. For example:

Option "AutoMapping" "1"

Option “SkipClick”

When working touch controller is **HID compliant device** and the using option “Device” for mouse is **“/dev/input/mice”**, this option could be appended as example below in the Xorg configuration file (e.g. /etc/X11/xorg.conf) to avoid conflict of mouse click. But, this will cause the **Click On Release** function abnormally. A way to solve this issue, please see page 10 for details.

Note that this option is used for HID compliant touch controller only.

For example:

Option “SkipClick” “1”

Option “HidOnEPC”

When the **HID compliant touch controller** is used on **EeePC** series, this option should be appended as example below in the Xorg configuration file (e.g. /etc/X11/xorg.conf) to make it function normally. For example:

Option “HidOnEPC” “1”

Option “Rotation”

Since touch driver supports auto-rotation function through RandR tool, the X system may crash once users touch the touchscreen under some Linux distributions. In order to solve this problem, users could append this option as example to disable auto-rotation function support. For example:

Option “Rotation” “0”

Option “Xinerama”

Since X window system supports multi-monitor with extend mode, totally there are two behaviors on different Linux distributions. One is the touch driver and the application window are both working across two monitors and other one is the touch driver is only working on one monitor, but the application window is working across two monitors. If users meet first situation, users could use Split-Display function to set which working area users need. Otherwise, users have to enable the function as example to workaround this problem.

For example:

Option “Xinerama” “1”

Option "SoundDevice"

The default sound device node is set to "/dev/audio". If users want to use other sound devices the system has, users could assign an absolute sound device path for the driver to generate sound through this sound device.

For example:

Option "Parameters" "/dev/audio"

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.

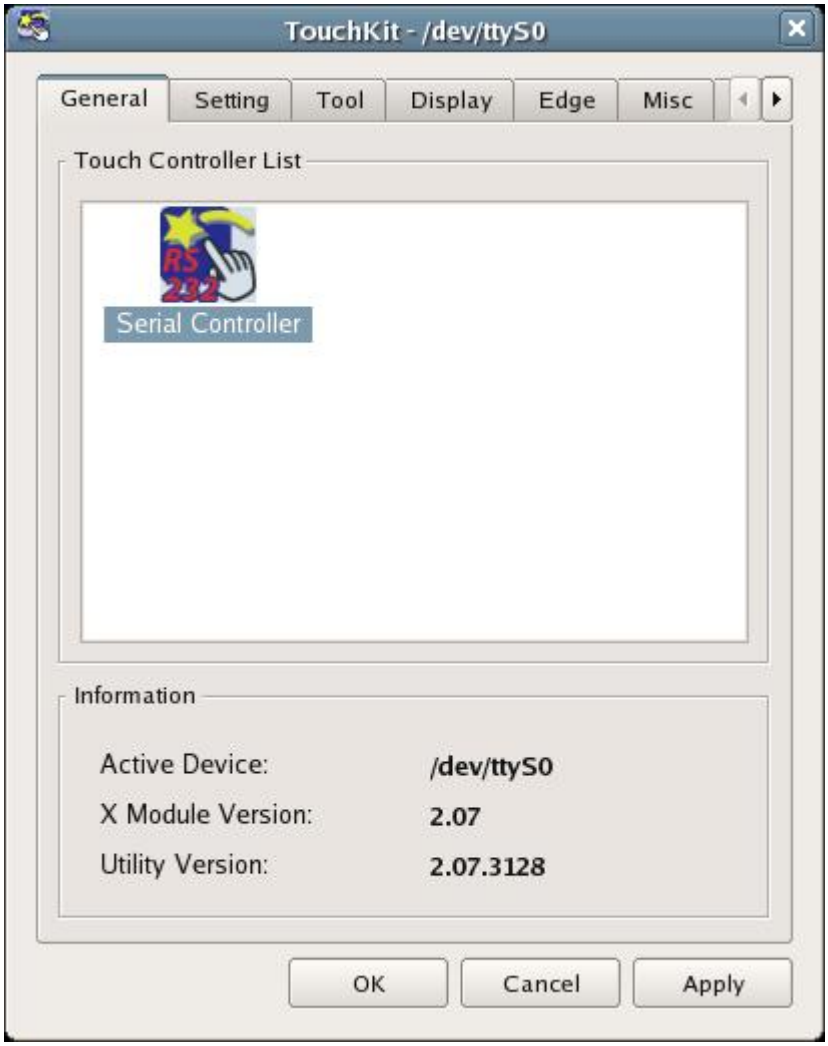
Note: *If the system is running SELinux. SELinux might block access from the driver to the TouchKit device. This can cause many different problems. Before installing the driver issue the "setenforce 0" command to disable SELinux enforcement. Once the driver is installed, check the system audit logs for 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 archive 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**.

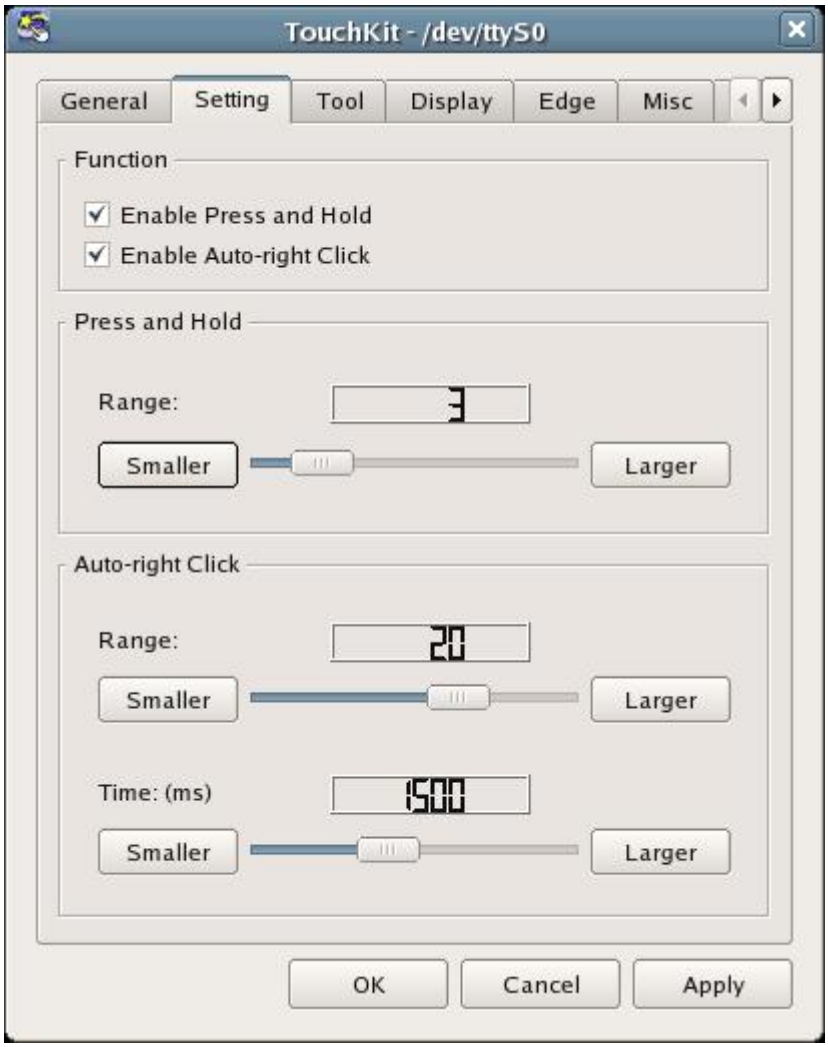
Note: *Please make sure the Xorg module is installed successfully. Otherwise, the utility does not work properly.*

3.1) General Property:



The utility enumerates TouchKit 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 active device node the device is connected. In addition, the Xorg module and utility versions will be shown in the Information window as well.

3.2) Setting Property:



Some options can be configured for mouse emulation.

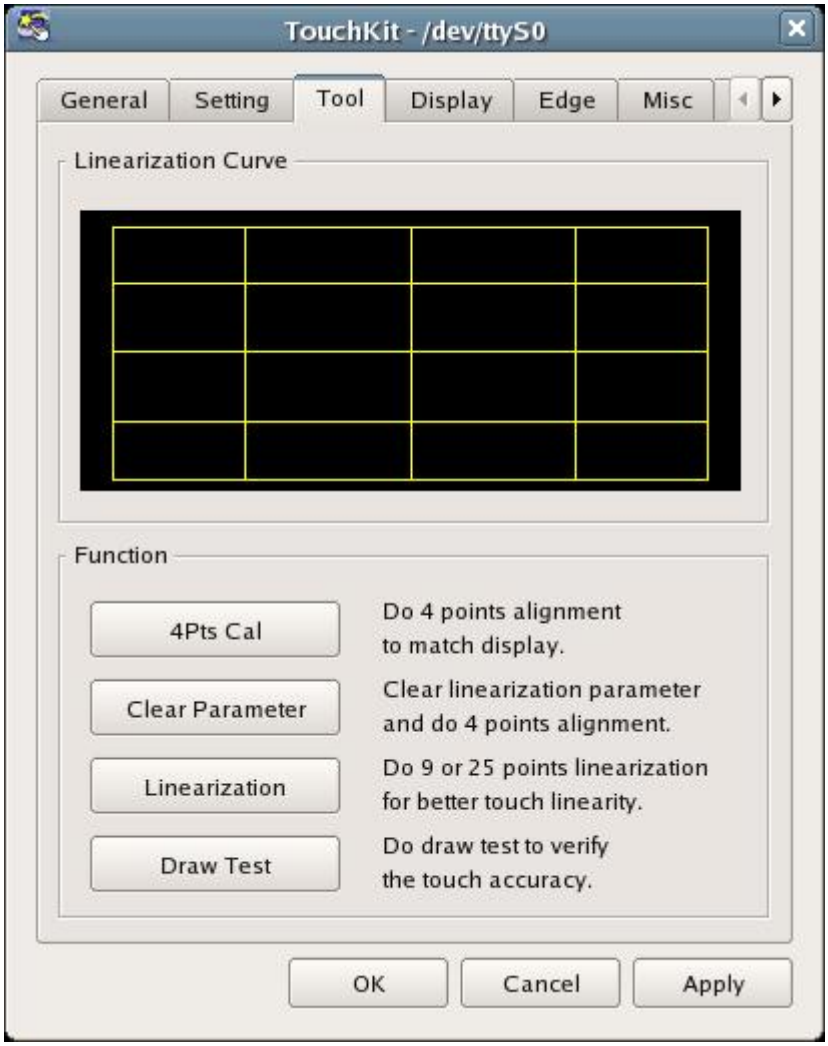
Press and Hold:

Press and hold at the same point. In some applications, 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 the touch device of event type.

Auto Right Click:

The driver generates a mouse right click event automatically whenever the driver detects the touch 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 Curve:

After linearization finished, the linearity of the touchscreen will be shown in this linearization curve.

4 Pts Calibration:

The 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.



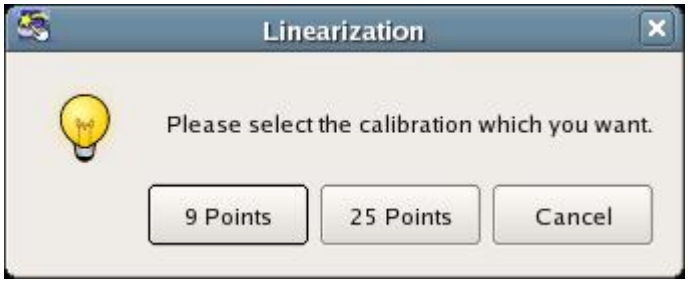
The user should **press the calibration symbol until it goes to next point or disappears**. The user can abort this calibration by pressing **<ESC>** key.

Clear Parameter:

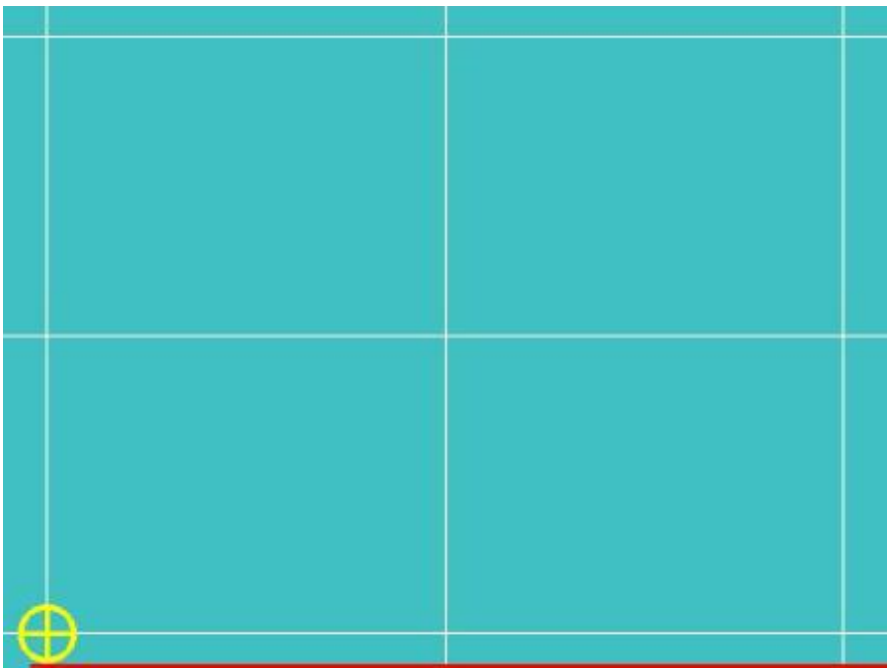
Press this button to clear the linearization parameters and do 4 points calibration. All of the linearization parameters will be cleared if the button pressed.

Linearization:

Press this button to do linearization, a message box as below will pop up to hint user to select 9 points or 25 points calibration.



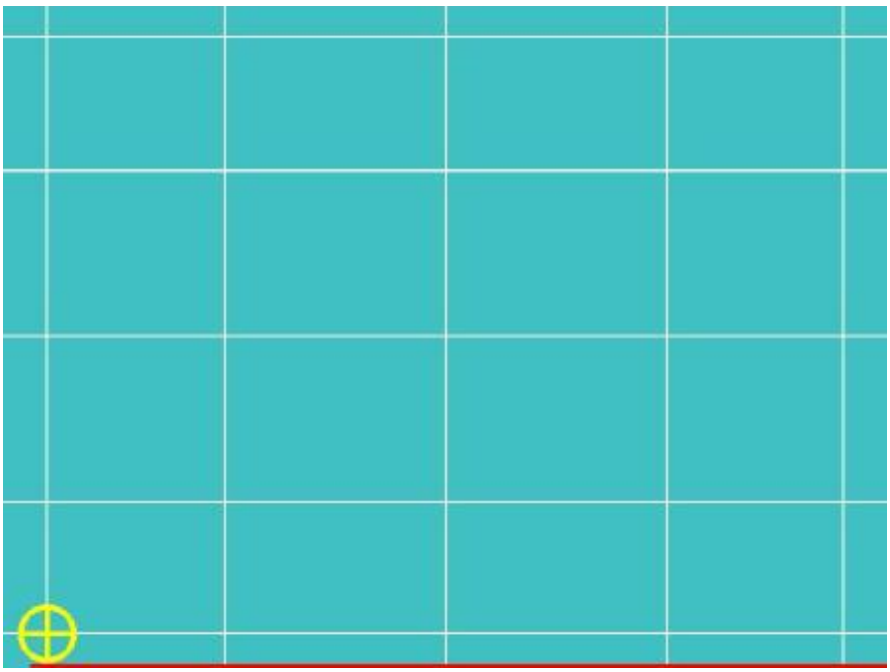
9 Pts Linearization:



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.

25 Pts Linearization:



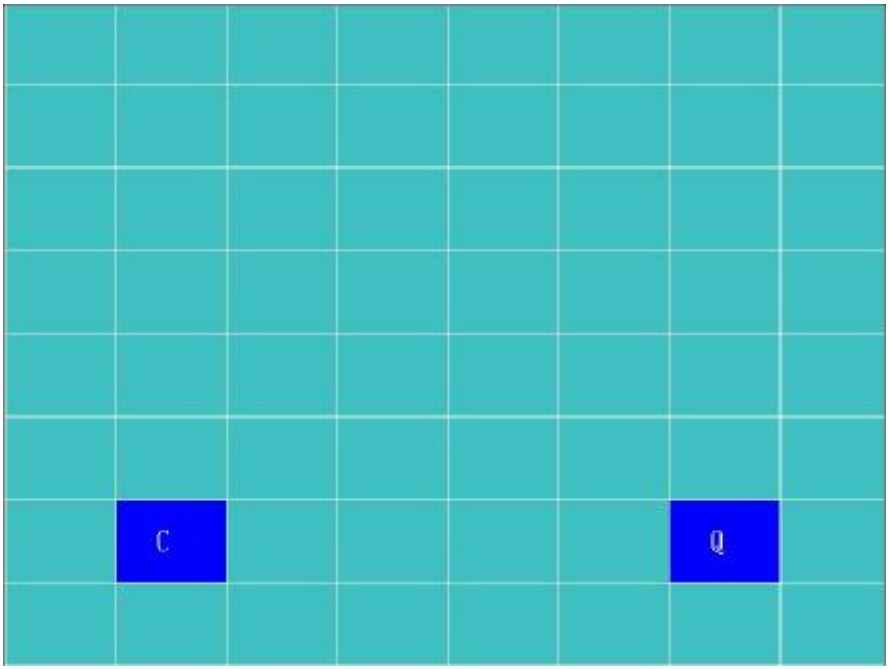
The user should **press the calibration symbol until it goes to next point or disappears**.

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

After linearization, the previous 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…



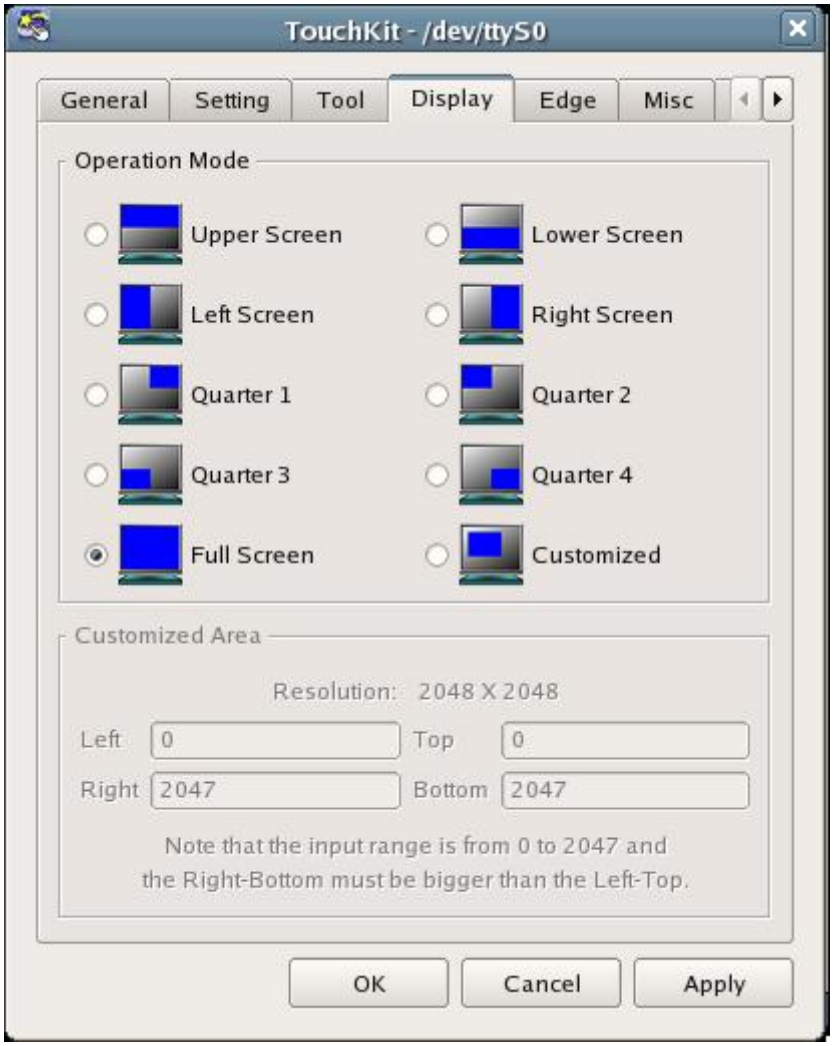
[C]: Clear

The user can touch within the rectangle of [C] to clear the window of draw test.

[Q]: Quit

The user can quit this window by touching within the rectangle of [Q], or pressing **[ESC]** key.

3.4) Display Property:

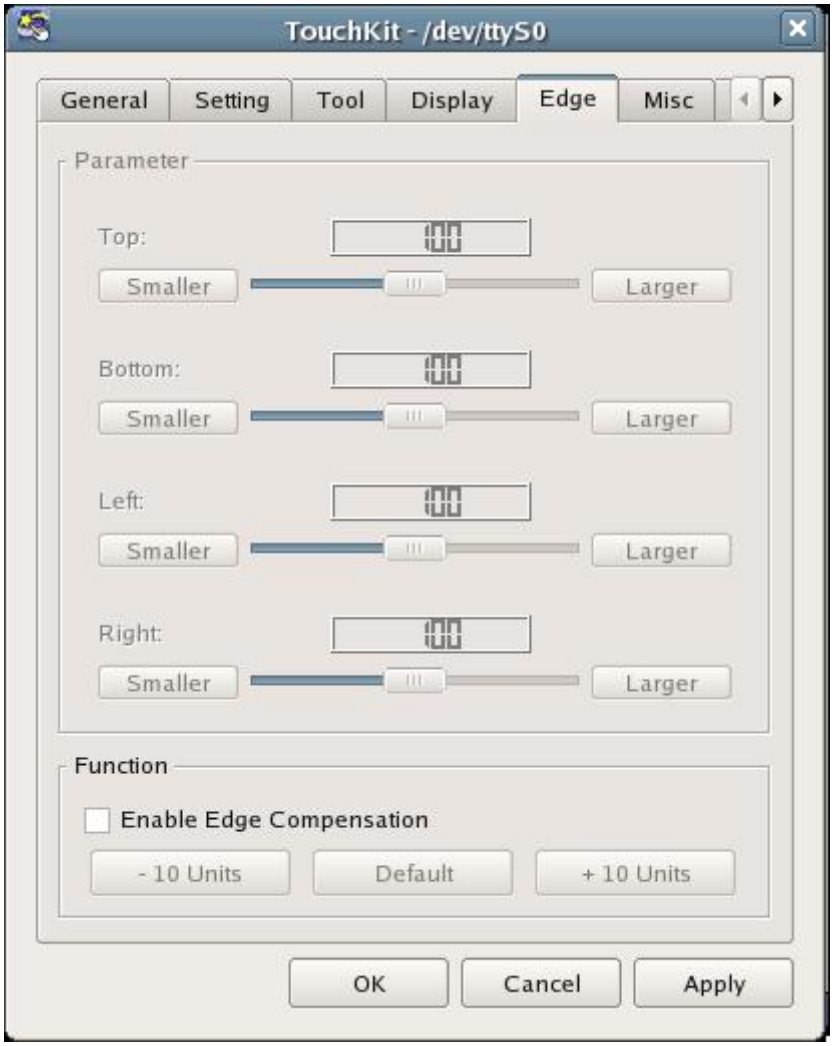


TouchKit utility supports split display feature.

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

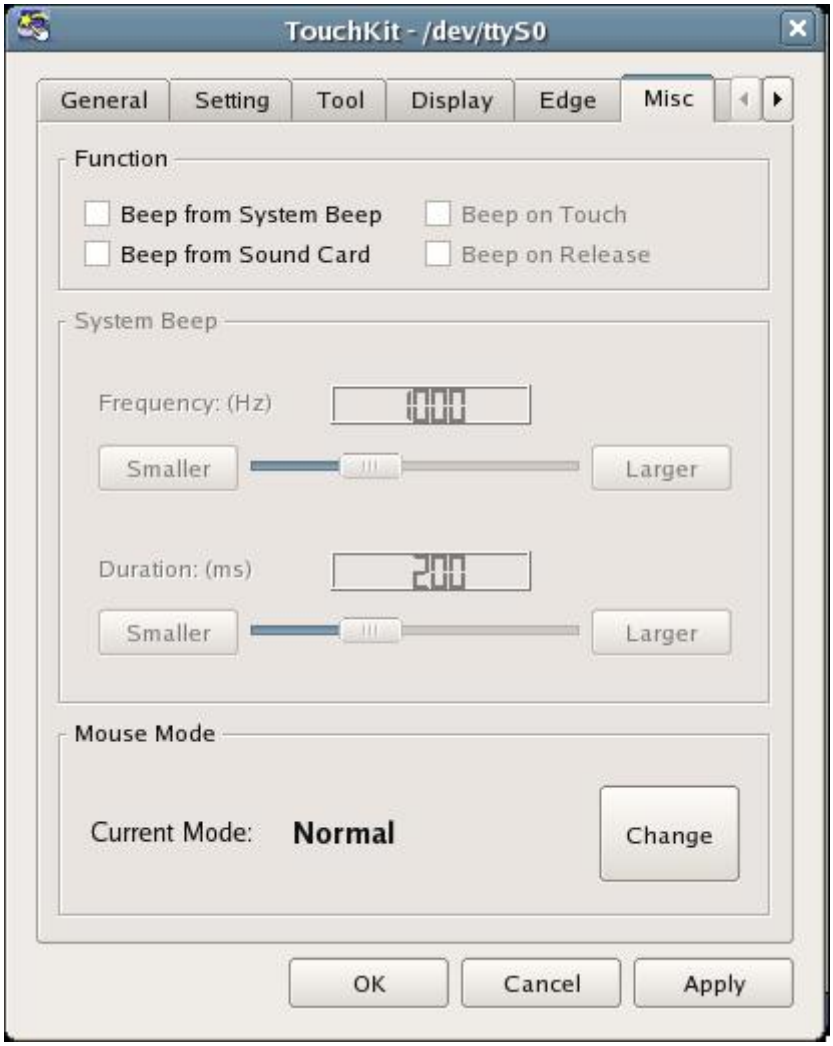
The user can choose any options to define where the touchscreen will be mapped. However, if the **Customized** is selected, it needs to enter the area to map to. The utility always **assumes the resolution is 2048 X 2048. The user has to calculate the area manually if the video resolution is not 2048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

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 from System Beep:

When this function enabled, the driver will generate a beep sound from **system beep** if **Beep On Touch** or **Beep On Release** checkbox is checked.

Beep from Sound Card:

When this function enabled, the driver will generate the beep sound from **sound card** if **Beep On Touch** or **Beep On Release** checkbox is checked.

Beep On Touch:

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

Frequency:

Adjust this **Frequency** to control the frequency of the system beep.

Duration:

Adjust this **Duration** to control the duration of the system beep.

If check **Beep On Touch** and **Beep On Release** at the same time, the user might heard the beep sound for a long time when touched many times, to minimum the **Duration** time to avoid the long beep sound.

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

Mouse Emulation Mode:

TouchKit driver supports four mouse emulation modes.

1.) Normal Mode:

With this 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:

With this mode, 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:

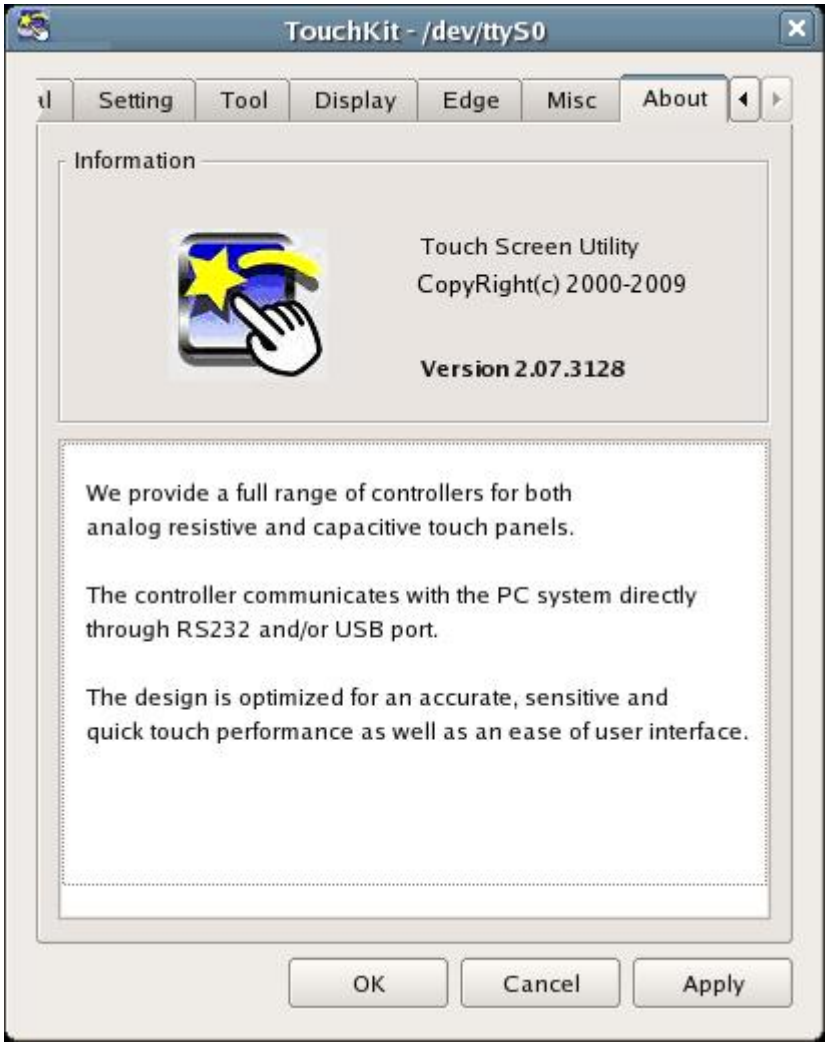
With this mode, the touch driver does not report any event until it detects a lift off. It reports **a left button click event** when it detects a lift off.

4.) Desktop Mode:

With this mode, the driver behaves similar as Normal Mode. But the driver will not report mouse button down immediately after user touches down. User needs to touch and stay at one point for a few milli-second, then the driver report mouse touch down.

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.