TSUSB Touchscreen – Linux Drivers Version 1.0.2

The TSUSB for Linux SetupPack included all source files for you to build and install drivers and programs for the TSUSB Touchscreen to your Linux system.

File List of TSUSB for Linux SetupPack

The following files are included in the TSUSB for Linux SetupPack:

tsusb-source-1.0.2-readme.pdf	*Installation document (this file)*
tsusb-source-1.0.2.tar.gz	the TSUSB Sources

Install the TSUSB Sources

To install the TSUSB Sources by running the followings:

(Note: replace the ***$sourcePath*** to the path of the tsusb-source-1.0.2.tar.gz file)

```
# cd /tmp
# tar xfvz $sourcePath/tsusb-source-1.0.2.tar.gz
```

The followings shows the directory structure of The TSUSB Sources:

tsusb-1.0.2	top directory of the TSUSB source		
	documents	serval type readme files (same as this file)	
	drivers	top directory of TSUSB driver source files	
		kernel-2.4	TSUSB kernel driver source files for Linux Kernel 2.4.x
		kernel-2.6	TSUSB kernel driver source files for Linux Kernel 2.6.x
		xfree86-4	TSUSB XFree86 input driver source files for XFree86 4.1.0 or later
		xorg-6	TSUSB XOrg input driver source files for XOrg 6.7.0 to 6.8.9
		xorg-7	TSUSB XOrg input driver source files for XOrg 7.0.0 or later
	programs	top directory of ths TSUSB programs	
		controlpanel	TSUSB control panel program source files

Get and Install the Kernel Sources

The Kernel Sources is required for build the TSUSB kernel driver. You need get and install the Kernel Sources on your Linux system. If you are newbie in Linux kernel, please read the Kernel-HOWTO document to start, you can get that document from http://www.linux.org/docs/howto/Kernel-HOWTO.

If you are using a packaged distribution than most likely the distributor will boundle a Kernel Sources package. You may get the Kernel Sources via the package installation method, whether RPM, apt, YAST, portage, etc. please consult your distribution's documentation for specifics.

The other option is to use the offical Kernel Sources via internat download from http://www.kernel.org. Follow the instructions in the Kernel-HOWTO document to install the Kernel Sources.

Prepare the Kernel Sources

As of this writing, if the HID kernel driver identify a pointer device as a HID mouse, it parse all Input Report that comes from the device to relative coordinate data because the driver always ignore the absolute coordinate flagbit in Input Report. To make the TSUSB Touchscreen work correctlly, some source codes of the HID kernel driver need to be modified.

The followings describe how to prepare the Kernel Sources for the TSUSB kernel driver:

*(NOTE: in the following steps, where **x.x.x-x** is the Kernel Sources Version that you using)*

1. **Modifing the HID Core Kernel Driver Source Code.**

   For kernel version 2.4.9 or earlly, edit the "/usr/src/linux-***x.x.x-x***/drivers/usb/hid.c" file.

   F*or kernel version 2.4.10 or later, edit the "/usr/src/linux-**x.x.x-x***/drivers/usb/hid-core.c" file.

   F*or kernel version 2.6.0 or later, edit the "/usr/src/linux-**x.x.x-x***/drivers/usb/input/hid-core.c" file.

   Find the line "struct hid_blacklist", above that line add two lines:

   ```
   #define USB_VENDOR_ID_TSUSB          0x16fd
   #define USB_DEVICE_ID_TSUSB          0x5453
   ```

   Scroll down to find the line "{ 0, 0 }" in the hid_blacklist structure, above that line add one line:

   ```
   { USB_VENDOR_ID_TSUSB, USB_DEVICE_ID_TSUSB, HID_QUIRK_IGNORE },
   ```

   The followings is a example shows how to add new codes to hid.c/hid-core.c:

   ```
   --- BEFORE  MODIFY ---
   #define USB_VENDOR_ID_WACOM         0x056a
   #define USB_DEVICE_ID_WACOM 0x0000
   struct hid_blacklist {
   __u16 idVendor;
   __u16 idProduct;
   unsigned quirks;
   } hid_blacklist[] = {
   { USB_VENDOR_ID_WACOM, USB_DEVICE_ID_WACOM, HID_QUIRK_IGNORE },
   { 0, 0 }
   };
   --- AFTER  MODIFY ---
   #define USB_VENDOR_ID_WACOM                 0x056a
   #define USB_DEVICE_ID_WACOM         0x0000
   #define USB_VENDOR_ID_TSUSB         0x16fd
   #define USB_DEVICE_ID_TSUSB         0x5453
   struct hid_blacklist {
   __u16 idVendor;
   __u16 idProduct;
   unsigned quirks;
   } hid_blacklist[] = {
   { USB_VENDOR_ID_WACOM, USB_DEVICE_ID_WACOM, HID_QUIRK_IGNORE },
   { USB_VENDOR_ID_TSUSB, USB_DEVICE_ID_TSUSB, HID_QUIRK_IGNORE },
   { 0, 0 }
   };
   ```

2. **Modifing the USB Mouse Kernel Driver Source Code.**

   For kernel version 2.4.x dit the "/usr/src/linux-***x.x.x-x***/drivers/usb/usbmouse.c" file.

   For kernel version 2.6.x dit the "/usr/src/linux-***x.x.x-x***/drivers/usb/input/usbmouse.c" file.

   Find the usb_mouse_probe function that the line looks like:

   ```
   static void *usb_mouse_probe(struct usb_device *dev, unsigned int ifnum,
   ```

   Scroll down to the blank line that below the last variable defintion, below that line add four lines:

For kernel 2.4.x:

```
if (dev->descriptor.idVendor == 0x16fd) {
        printk(KERN_INFO "usb_mouse: ignoring TSUSB Touchscreen devices\n");
        return NULL;
}
```

For kernel 2.6.x:

```
if ( dev->descriptor.idVendor == 0x16fd ) {
        printk(KERN_INFO "usb_mouse: ignoring TSUSB Touchscreen devices\n");
        return -ENODEV;
}
```

The followings shows how to add new codes to usmouse.c:

--- BEFORE MODIFY ---

```
static void *usb_mouse_probe(struct usb_device *dev, unsigned int ifnum,
                             const struct usb_device_id *id)
{
        struct usb_interface *iface;
        struct usb_interface_descriptor *interface;
        struct usb_endpoint_descriptor *endpoint;
        struct usb_mouse *mouse;
        int pipe, maxp;
        char *buf;

        ...
}
```

--- AFTER MODIFY ---

```
static void *usb_mouse_probe(struct usb_device *dev, unsigned int ifnum,
                             const struct usb_device_id *id)
{
        struct usb_interface *iface;
        struct usb_interface_descriptor *interface;
        struct usb_endpoint_descriptor *endpoint;
        struct usb_mouse *mouse;
        int pipe, maxp;
        char *buf;

        if (dev->descriptor.idVendor == 0x16fd) {
                printk(KERN_INFO "usb_mouse: ignoring TSUSB Touchscreen devices\n");
                return NULL; /* Note: for kernel 2.6.x,  return -ENODEV; */
        }
        ...
}
```

3. **Building and Installing the Kernel.**

   Follow the Kernel-HOWTO document to build and install the kernel.

   Make sure you enable the following options when you configure the kernel:

   CONFIG_USB "USB (Universal Serial Bus) support"

   CONFIG_USB_HID "USB Human Interface Device (full HID) support"

4. **Checking Your Linux System.**

   To make sure your Linux system is ready for the TSUSB Touchscreen by the following:

   For kernel version 2.4.x:

   # cat /proc/bus/usb/devices | grep TS2005F

   The output is “S: Product=TS2005F-USB”, If you plugged the TSUSB Touchscreen controller.

   The output is nothing, if you unpluged the TSUSB Touchscreen controller.

   For kernel version 2.6.x:

   # cat /proc/bus/input/devices | grep TS2005F

   The output is always nothing, if you prepared the Kernel Sources .

Build the TSUSB Kernel Driver

To build the TSUSB kernel driver, you need installed and prepared the Linux kernel sources on your Linux system (see the “Prepare the Kernel Sources” that shows on the above).

The followings describe how to build the TSUSB kernel driver:

*(NOTE: in the following steps, where **x.x.x-x** is the Kernel Sources Version that you using)*

1. **Changing the TSUSB Device Major Number.**

   This step is required for kernel 2.4.x only.

   The TSUSB kernel driver used device major number 234 to create a character device named “/dev/touch”. If your Linux system had another device used the same device major number, you need change the TSUSB device major number by edit the “/tmp/tsusb-1.0.2/drivers/kernel-2.4/tsusbhid.c” file:

   Find the line looks like “#define TSUSB_MAJOR 234” that near the top of the tsusbhid.c, where 234 is the device major number that used by the TSUSB kernel driver, change it to a free device major number on your Linux system.

2. **Adding the TSUSB Source Files to the Kernel Sources.**

   To add the TSUSB source files to the Linux kernel sources tree by running the followings.

   For kernel version 2.4.x:

   # cd /tmp/tsusb-1.0.2/drivers/kernel-2.4

   # cp tsusbhid.c /usr/src/linux-***x.x.x-x***/drivers/usb

   # cp tsusbhid.h /usr/src/linux-***x.x.x-x***/drivers/usb

   For kernel version 2.6.x:

   # cd /tmp/tsusb-1.0.2/drivers/kernel-2.6

   # cp tsusbhid.c /usr/src/linux-***x.x.x-x***/drivers/usb/input

   # cp tsusbhid.h /usr/src/linux-***x.x.x-x***/drivers/usb/input

   # cp tsusbhid.rules /etc/udev/rules.d

3. **Adding TSUSB Build Option to the Kernel Sources.**

   For kernel 2.4.x, edit the "/usr/src/*linux-**x.x.x-x***/drivers/usb/Makefile" file.

   For kernel 2.6.x, edit the "/usr/src/*linux-**x.x.x-x***/drivers/usb/input/Makefile" file.

   Find the line looks like "obj-$(CONFIG_USB_MOUSE)+= usbmouse.o", below that line add one line:

   obj-$(CONFIG_USB_TSUSB) += tsusbhid.o

4. **Adding TSUSB Configure Option to the Kernel Sources.**

   This step is required for kernel 2.4.x only, edit the "/usr/src/linux-x.x.x-x/drivers/usb/Config.in" file:

   Find the line "comment 'USB Human Interface Devices (HID)'", below that line is look like:

```
if [ "$CONFIG_INPUT" = "n" ]; then
        comment '  Input core support is needed for USB HID'
else
        dep_tristate ...
        if [ "$CONFIG_USB_HID" != "y" ]; then
                dep_tristate ...
        fi
        dep_tristate ...
fi
```

   Scroll down to the last line "fi", above that line add one line:

```
dep_tristate '  TSUSB Touchscreen support' CONFIG_USB_TSUSB$CONFIG_USB $CONFIG_INPUT
```

   The followings is a example shows how to modify Config.in:

```
--- BEFORE MODIFY ---
comment 'USB Human Interface Devices (HID)'
if [ "$CONFIG_INPUT" = "n" ]; then
    comment '  Input core support is needed for USB HID'
else
    dep_tristate '  USB Human Interface Device (full HID) support' CONFIG_USB_HID $CONFIG_USB $CONFIG_INPUT
    if [ "$CONFIG_USB_HID" != "y" ]; then
        dep_tristate '  USB HIDBP Keyboard (basic) support' CONFIG_USB_KBD $CONFIG_USB $CONFIG_INPUT
        dep_tristate '  USB HIDBP Mouse (basic) support' CONFIG_USB_MOUSE $CONFIG_USB $CONFIG_INPUT
    fi
    dep_tristate '  Wacom Intuos/Graphire tablet support' CONFIG_USB_WACOM $CONFIG_USB $CONFIG_INPUT
fi
--- AFTER MODIFY ---
comment 'USB Human Interface Devices (HID)'
if [ "$CONFIG_INPUT" = "n" ]; then
    comment '  Input core support is needed for USB HID'
else
    dep_tristate '  USB Human Interface Device (full HID) support' CONFIG_USB_HID $CONFIG_USB $CONFIG_INPUT
    if [ "$CONFIG_USB_HID" != "y" ]; then
        dep_tristate '  USB HIDBP Keyboard (basic) support' CONFIG_USB_KBD $CONFIG_USB $CONFIG_INPUT
        dep_tristate '  USB HIDBP Mouse (basic) support' CONFIG_USB_MOUSE $CONFIG_USB $CONFIG_INPUT
    fi
```

```
    dep_tristate '  Wacom Intuos/Graphire tablet support' CONFIG_USB_WACOM $CONFIG_USB $CONFIG_INPUT
    dep_tristate '  TSUSB Touchscreen support' CONFIG_USB_TSUSB $CONFIG_USB $CONFIG_INPUT
fi
```

5. **Adding TSUSB Configure Help to the Kernel Sources.**

   For kernel 2.4.x, edit the “/usr/src/linux-x.x.x-x/Documentation/Configure.help” file:

   Add the following lines (Note: the first and second line must start without leading space and the other lines must start with leading space):

```
 TSUSB Touchscreen support
 CONFIG_USB_TSUSB
 Say Y here if you want to use TSUSB Touchscreen.
 Make sure to say Y to "USB (Universal Serial Bus) support"
 CONFIG_USB and  "USB Human Interface Device (full HID) support"
 CONFIG_USB_HID as well.

 This driver is also available as a module ( = code which can be
 inserted in and removed from the running kernel whenever you want).
 The module will be called tsusbhid.o.  If you want to compile it as a
 module, say M here and read <file:Documentation/modules.txt>.
```

   For kernel 2.6.x, edit the “/usr/src/linux-x.x.x-x/drivers/usb/input/Kconfig” file:

   Add the following lines (Note: the first line must start without leading space and the other lines must start with leading space):

```
config USB_TSUSB
  tristate "TSUSB Touchscreen support"
  depends on USB && USB_HID
  ---help---
        Say Y here if you want to use the TSUSB Touchscreen.
        Make sure to say Y to "USB (Universal Serial Bus) support"
        CONFIG_USB and  "USB Human Interface Device (full HID) support"
        CONFIG_USB_HID as well.

        To compile this driver as a module, choose M here: the
        module will be called tsusbhid.
```

6. **Enabling the TSUSB Touchscreen Hotplug.**

   Edit the “/lib/modules/***x.x.x-x***/modules.usbmap” file:

   Add the following lines:

```
tsusbhid 0x0003 0x16fd 0x5453 0x0000 0x0000 0x00 0x00 0x00 0x00 0x00 0x00 0x00000000
```

7. **Configuring the Kernel to Enable the TSUSB Build Option.**

   To enable the TSUSB Build Option by running the followings:

cd /usr/src/linux-x.x.x-x

make oldconfig

Say Y or M when asking “TSUSB Touchscreen support” depends on your project.

8. **Building and Installing the Kernel for TSUSB Support.**

   Read the Kernel-HOWTO document (http://www.linux.org/docs/howto/Kernel-HOWTO) and follow the instructions to rebuild and reinstall the kernel.

9. **Checking TSUSB Support on Your Linux System.**

   To make sure the TSUSB kernel driver is working on your Linux system by the following command:

   # ls /dev/touch

   The output is “tsusb”, If none TSUSB Touchscreen controller plugged.

   The output is “tsusb tsusb0”, if one TSUSB Touchscreen controller plugged.

   The output is “tsusb tsusb0 tsusb1”, if two TSUSB Touchscreen controller pluged.

Build the TSUSB X Input Driver (XFree86-4 and XOrg-6)

To build the TSUSB X Input driver for XFree86 (4.1.0 or later) or XOrg (6.7.0 to 6.9.0), you need install and prepare the XFree86/XOrg sources on your Linux system.

The followings describe how to build the TSUSB X Input driver for XFree86-4 and XOrg-6:

1. **Getting the XFree86/XOrg Sources.**

   You may get the offical sources via internat download:

   To get the XFree86 sources, visit http://www.xfree86.org.

   To get the XOrg sources, visit http://www.x.org.

   Follow the instructions in the XFree86/XOrg documents to install the XFree86/XOrg sources.

2. **Adding the TSUSB Source Files to the XFree86/XOrg Sources.**

   From the “xc/programs/Xserver/hw/xfree86/input” directory in the Xfree86/XOrg sources tree, run the followings:

   For XFree86 4.1.0 or later

   # cp -r /tmp/tsusb-1.0.2/drivers/xfree86-4 tsusb

   For XOrg 6.7.0 to 6.9.0

   # cp -r /tmp/tsusb-1.0.2/drivers/xorg-6 tsusb

3. **Adding TSUSB Option to the XFree86/XOrg Configure Files.**

   From the “xc/config/cf” directory in the Xfree86/XOrg sources tree, run the followings:

   For XFree86 4.1.0 or later, edit the the “xfree86.cf” and “xf86site.def” file.

   For XOrg 6.7.0 to 6.9.0, edit the “xorg.cf” and “xorg.def” file.

   Find the line that start with “#define XinputDrivers” (or looks like that), add a “tsusb” word to the end of XinputDrivers defintion, The followings is a example shows how to do that:

   --- BEFORE MODIFY ---

   #define XinputDrivers mouse dynapro elo2300 elographics magellan microtouch \

   mutouch spaceorb wacom void

--- AFTER MODIFY ---

```
#define XinputDrivers       mouse dynapro elo2300 elographics magellan microtouch \
                            mutouch spaceorb wacom void tsusb
```

4. **Building the TSUSB X Input Driver.**
   Follow the instructions in the XFree86/XOrg documents to build the Xfree86/XOrg, then the TSUSB X input driver will be builded when you complete that.
   If you want to build only the the TSUSB X input driver after “make World” or “make Everything”, run “make” from the “xc/programs/Xserver/hw/xfree86/input/tsusb” directory.

5. **Installing the TSUSB X Input Driver.**
   Follow the instructions in the XFree86/XOrg documents to install the Xfree86/XOrg, then the TSUSB X input driver will be installed when you complete that.
   If you want to install only the the TSUSB X input driver, run “make install” from the “xc/programs/Xserver/hw/xfree86/input/tsusb” directory.

Build the TSUSB X Input Driver (XOrg-7)

The followings describe how to build the TSUSB X Input driver for XOrg-7:

1. **Installing SDK for Xorg-X11 Package.**
   To build the TSUSB X Input driver for XOrg 7.0 or later, you need install the SDK for Xorg-X11 package that match your Xorg version on your Linux system.

2. **Building the TSUSB X Input Driver for XOrg-7.**
   To build the TSUSB X Input driver for XOrg 7.0 or later, run the followings:

```
# cd /tmp/tsusb-1.0.2/drivers/xorg-7
# ./configure --prefix=/usr
# make clean
# make
```

3. **Installing the TSUSB X Input Driver for XOrg-7.**
   To install the TSUSB X Input driver for XOrg 7.0 or later, run the followings:

```
 # cd /tmp/tsusb-1.0.2/drivers/xorg-7
# make install
```

Build the TSUSB Control Panel Program

For quick start, we recommanded that you using the TSUSB Control Panel program precompiler binary:

Get “tsusb-binary-1.0.2-program-controlpanel.tar.gz” and extract it to the “/etc” directory:

(Note: replace the ***$sourcePath*** to the path of the tsusb-binary-1.0.2-program-controlpanel.tar.gz file)

cd /etc

tar xfvz ***$sourcePath***/ tsusb-binary-1.0.2-program-controlpanel.tar.gz

To build the the TSUSB Control Panel program, you need install the GTK+ 2.4 libraries and include files on your Linux system. The followings describe how to build the TSUSB Control Program:

1. **Installing the GTK+ 2.4 Libraries and Include Files.**
   The TSUSB Control Panel program required the GTK+ libraries and include files, you need install the GTK+ 2.4 or later on your Linux system before build and/or execute that program.
   If you hard to find the GTK+ libraries and include files for your system, you may try our precompiler binary file named “tsbuild-gtk2-2.4.14-bin.tar.gz” and extract it to the “/tmp” directory:
   (Note: replace the ***$sourcePath*** to the path of the tsbuild-gtk2-2.4.14-bin.tar.gz file)
   # cd /tmp
   # tar xfvz ***$sourcePath***/ tsbuild-gtk2-2.4.14-bin.tar.gz
   Note: In the next step, modify the “mPATH_GTK = /usr” to “mPATH_GTK = /tmp/GTK/2.4/usr”

2. **Modify the Makefile for TSUSB Control Panel Program.**
   if you did not installed the GTK+ libraries and include files in the /usr directory, you need modify the Makefile for TSUSB Control Panel by the following:
   Edit the “/tmp/tsusb-1.0.2/programs/controlpanel/Makefile” file.
   Modify the first line that looks like “mPATH_GTK = /usr”, change the /usr to the path where you installed the GTK+ libraries and include files.

3. **Building the TSUSB Control Panel Program.**
   To build he TSUSB Control Panel program by running the followings:
   # cd /tmp/tsusb-1.0.2/program/controlpanel
   # make clean
   # make

4. **Installing the TSUSB Control Panel Program.**
   To install the TSUSB Control Panel program by running the followings:
   # cd /tmp/tsusb-1.0.2/program/controlpanel
   # make install

Using the TSUSB Touchscreen with the X Window

In order to use TSUSB Touchscreen with the X Window System, you need to configurate the X Window System by editting the X Config File, the followings describe how to do that:

1. **Login as root.**

2. **Configurating the X Window System.**

   To edit the X Config File by followings:

   - Opening the X Config File.

     Use any text editer to open the X Config File in edit mode. The path and filename of the X Config File is depended on the Linux Distribution that you using, locate to the “/var/log” directory, open the file named “XFree86.0.log” or “Xorg.0.log”, search in the file with “Using config file:” string, the path and filename of the X Config File shows following the string.

   - Add a InputDevice Section.

     ```
     Section    "InputDevice"
                Identifier    "TSUSB_Touchscreen"
                Driver        "tsusb"
                Option        "Device"          "/dev/touch/tsusb"
                Option        "ScreenNo"        "0"
                Option        "SendCoreEvents"  "yes"
     EndSection
     ```

   - Add a InputDevice lines to the ServerLayout Section.

     ```
     Section    "ServerLayout"
                InputDevice "TSUSB_Touchscreen" "SendCoreEvents"
     EndSection
     ```

   - Save changes to the X Config File.

3. **Restart the X Window System**.

   After you complete the install, you must reboot your computer or restart the X Server for the new settings to take effect, and make the TSUSB Touchscreen start to working.

Using the TSUSB Control Panel Program

To calibrate or configure the TSUSB Touchscreen, run the “tsusb_controlpanel” program in the “/etc/TSUSB” directory, following the instructions show on the screen to complete the calibrate and/or configure.

Note:

The user who runs the TSUSB Control Panel program must have permissions to access (read/write) the “tsusb-config” file in the “/etc/TSUSB” directory.

The TSUSB Control Panel program read the “CalibrationMode” setting from the tsusb-config” in the “/etc/TSUSB” directory at start, if the setting is set to 1, it running in 4 Points Calibration Mode 2 (center of each eddge of screen), if the setting is set to 0, it running in 4 Points Calibration Mode 1 (each conner of screen).