



Linux Installation

Revision 1.33, 27th May 2010
www.touch-base.com/documentation/installation

[History](#) [Deliverables](#) [General](#) [Limitations](#) [Requirements](#) [Important notes](#) [Installation](#)
[Calibration](#) [Settings](#) [Serial port](#) [Uninstall](#) [Mouse](#) [Utilities](#) [Contact](#)

There is no single standard for interfacing touch devices in the Linux environment and there are a number of different 'touch interfaces' available. The most standard interface method is via the X (the Linux graphical layer) interface as this offers the low level, mouse emulation type touch support. However, there are other interfaces, such as direct file access to touch hardware (special device files), X multi touch, (since kernel 2.6.30) Tslib, Evdev, Evtouch – there may be others. Linux application development kits, such as [SDL Multimedia Library](#) , [Clutter Library](#) , [Pigment Toolkit](#), [GTK](#) etc, may also dictate the touch interface in use. For example, applications written with SDL or Clutter may utilise the Tslib interface if direct access to the touch device is required by the application.

Under Linux we offer UPDD in two forms, UPDD Linux B(inary) (pre compiled programs) and UPDD Linux S(ource).

The notes below refer to the UPDD Linux driver supplied in binary form *for UPDD Linux B version 4.1.1, initial release April 08 and 4.1.8 available since May 2010*. The source driver is available to touch manufacturers and integrators that require a source based Linux driver. Notes on the source driver are available [here](#).

The UPDD Linux B driver utilizes the X interface and should satisfy 99% of touch users. However, where applications have been written to utilize other interface methods then please contact us to discuss further as we can modify the driver accordingly.

Very important general note: Linux is an open source environment utilised and supported by highly knowledgeable and capable developers. Many aspects of the Linux system are maintained by Linux groups and communities. Nearly all software created by these Linux groups is made available under open source license agreements, including any touch screen drivers that have been written, mainly by individuals. Linux distributions are many and varied and all have slight subtle differences. Many users, developers and integrators in the Linux community do not expect to pay for software. All of this makes it commercially very difficult to operate in the Linux environment with a niche product such as our touch screen driver products.

These notes refer to our Linux driver, supplied in binary form, and is mainly aimed at the non technical Linux users or larger commercial organisations that need a tried and tested touch driver solution that comes with comprehensive support. To this end this driver offers a generalised installation package with basic system requirements and in many cases should work 'out the box', especially with the main stream Linux distributions. If you are a Linux technician, with access to open source touch driver code that you can modify and make the necessary system configuration changes then UPDD is unlikely to match your specific requirements particularly as there are license costs involved.

Further, given the number of Linux distributions which undergo constant updates and modifications, if UPDD has any issues on distributions not listed and tested we may not be readily able to offer free support as the amount of support needed does not match the commercial viability of Linux sales.

License Notice

The software is licensed software and as such requires a license per system when the production version of the software is installed. Production software is either supplied by pointer device manufacturers or system integrators (who are entitled to distribute the driver) or is available directly from [Touch-Base sales](#).

Evaluation software, which is available from our [Download Centre](#) has a 100 'mouse click' restriction at which point the driver needs to be reloaded or a calibration performed to gain a further 100 clicks.

Copyright Notice

This software, Universal Pointer Device Driver – TBUPDD, is copyright © 1998 – 2009 by Touch-Base Ltd. All rights reserved.

The Linux version utilizes a software library, libusb, which is used under the terms of the GNU Lesser General Public License as published by the Free Software Foundation and under the terms of this license the following applies:

Copyright © 2000-2003 Johannes Erdfelt johannes@erdfelt.com All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Under the terms of this license we will also make the Universal Pointer Device Driver binaries available to allow the libusb module to be replaced with an alternative version. Please contact technical@touch-base.com for further information.

Full details of the LGPL are available [here](#).

Linux driver build history

Release	Date	Change
4.0.2	10 th Oct 06	Initial Version 4 release for X86 systems
	22 nd Nov 06	Support new X interface on Fedora Core 5
4.0.4	5 th Jan 07	Added support for Power PC based systems
	16 th Jan 07	Changes required for SUSE 10.1
	17 th Jan 07	Interactive touch fix. Mouse settings called from Click Mode dialog for KDE and Gnome.
	9 th March 07	Multi-monitor support added
4.0.6	15 th Nov 07	Linux 64 bit support
4.1.1	25 th Apr 08	Much improved PnP support based on common code across all platforms
	30 th July 08	Ubuntu 8.04 support with updated X interface
	18 th Aug 08	Slackware 12.1, Fedora Core 8 and 9 support added
	26 th Sept 08	Uninstall utility added
	Mar 09	Debian Lenny support added
	Apr 09	Ubuntu 9.04 (Jaunty Jackalope) support added
		Supports rotation from Ubuntu display control panel
		X interface via 'built in' method – remove need for own X interface module
		Updated KDE UPDD desktop icon mechanism
		KDE X interface installation fix
	May 09	Fedora Core 10 supported added
	Aug 09	Fedora Core 11 supported added
	Oct 09	Suse 11.1 64 bit support added (Gnome WM only!)
Jan 10	Unbuntu 9.10 support added	
Mar 10	Suse 11.2 32 bit support added	

Deliverables

The main installation package is held within the compressed file called linuxupdd.tgz. Software requested from Touch-Base to be delivered electronically will be delivered in one of three ways:

- As an email attachment.
- Delivered to an FTP folder for manual download. The FTP link, user name and password details will be sent in an email.
- Automatically downloaded from a HTTP download link as sent in an email.

General Notes

UPDD for Linux has two driver components. A core processing engine which exists outside the kernel and an [X extension](#) to provide the mouse interface for X based GUI applications. This architecture allows for maximum portability. As the core driver is not embedded in the kernel a single compiled module can be executed on any kernel version. The vast majority of users use X to host GUI applications; however, UPDD has the flexibility to support other GUI's by reimplementing of this minimal module.

Hardware access is performed through operating system defined device interfaces allowing UPDD to operate with all compliant hardware, rather than just those tested by Touch-Base. UPDD's flexible modular architecture does however allow for direct hardware access to be added for specific device categories if required.

The initial UPDD for Linux was developed under Linux 2.4 + X (or X-Windows). The driver has been [tested on various Linux distributions](#) using Xfree86, mainly version 4.x and more recently X.org. It has also been extensively tested under 2.6.x kernel. In theory the driver should install and run on most of the standard Linux distributions. Please contact us if you should experience any difficulties installing or running the driver as it is likely to be a minor issue that is preventing the software from working.

You should be aware that it is possible to load multiple X sessions on a machine and switch between them as required. At any one time an X session will be active on the desktop and the other X sessions will be in the background. The driver can be installed from any X session. Each X session running will receive pointer movements and click requests which might not be desirable as you will not know what is being selected in the hidden, background, X sessions. We believe that in a touch environment most users will only run with one X session and therefore we do not think this is a significant issue. Please contact us if this causes any problems, as it may be possible for us to limit mouse movement and clicks to the active X session if you must run in a multiple X session environment.

As standard the driver is dynamically linked and relies on [certain system libraries being available](#). This is normally not an issue for standard Linux desktop systems but can be an issue in older distributions or specialised, cut down or embedded Linux environments. Please contact technical@touch-base.com if you require a statically linked version of the driver. The driver utilises QT cross platform graphics and therefore the installation does install this required QT graphics library.

There are a number of hardware considerations:

USB

1. If you are installing a USB controller please be advised that in our experience USB port initialisation can take up to 20 seconds to complete, during which time USB connected touch screens will not be active. The driver initialises the USB port when it loads, and currently, after calibration.
2. **Important USB controller observation:** *Not all USB controllers are supported by Linux. We have found a number of USB controllers that do not conform to USB standards and hence cause issues with the underlying OS interface. Most of*

these controllers have successfully worked in Windows by fluke or driver changes to overcome shortcomings in the firmware. It is only when they are used on a different OS that the problems appear.

- Important USB technical interface note:** Prior to UPDD version 3.7.10, the driver's USB interface method resulted in a timeout warning in the system log every time a USB pending read timed out. This could result in 1000's of entries in the log given sufficient time. Although we considered this to be an unnecessary warning message issued by the OS we investigated alternative approaches to overcome this issue.

Unfortunately, finding a generic solution has not been as straight-forward as envisaged, given that USB support in some Linux distributions appears to be flakey in places. We found that terminating pending read requests does not work as expected (we found that although pending reads can be cancelled the USB stack goes mad issuing log messages until the next interrupt occurs). It should be possible to circumvent this by use of the USB reset command, but this forces a bus re-enumeration and several touch screen controllers we tested with stopped working after this. We also found that in some cases, if we do successfully cancel outstanding read requests the thread that requested the read will not terminate until the timeout period on the read elapses – all very frustrating! In addition to this we also found that older versions of the kernel do not work consistently after the pending i/o cancellation. Given that the USB stack support appears to be so flakey in some distributions, we have had to go for a non optimal solution that appears to work in all cases. The new USB interface method works perfectly well. However, to receive USB data there is always a pending read outstanding and drivers cannot be unloaded in this state, which in some cases we need to do (trouble shooting etc) so we deliberately placed a timeout on this read of 10 minutes at which time we reissue the read. This does, however, result in a timeout message being written to the timeout log every 10 minutes, which we trust is acceptable.

PS/2

As of version 24/09/09 we have removed support for ps/2 devices due to the difficulty of maintaining support of a hardware touch product rarely used in a Linux environment.

System Requirements

Apart from the Linux kernel software our software requires the following components:

Processor	X86	Fully supported and tested in-house			
	X86 – 64bit	Fully supported and tested in-house			
	ARM	Have generated an ARM build with a cross compiler but have not tested in-house due to lack of ARM based hardware.			
	PowerPC	Fully supported and tested in-house			
X Interface	This interface is required to access the mouse port to emulate mouse movement and clicks.				
	Standard	The driver has been tested with these standard interfaces. The "86" in the config file names refers to the fact that it was originally developed on x86 and doesn't actually reflect what processor/platform it's running on.			
	XF86config-4	The driver utilises two different types of user mode X interfaces either via the UPDD kernel component and more recently (April 2009) via the Linux built in mouse interface. Often the UPDD kernel interface needs to be updated to cater for changes made to the X interface with new Linux distributions. It is our hope that the direct interface to the built in module will overcome this issue. Built in interface is utilised by the following distributions:			
	XF86config				
	Xorg.conf				
		Ubuntu 8.04	Aug 2009	Ubuntu 9.04	April 2009
		Fedora Core 10	May 2009	Redhat	May 2009
		Cent OS 4.6	May 2009	Fedora Core 9	June 2009
		Debian Lenny 32bit	April 2009	Ubuntu 91.0	Jan 2010
	Suse 11.2	Mar 2010			
	X Multi-touch (Kernel 2.6.30)	Multi-touch extensions were first implemented in kernel 2.6.30 and continue to be enhanced. Since May 2010 UPDD Linux B interfaces with the X MT extensions as described here .			
	Non-standard	This implementation of X is usually found on minimized /embedded Linux distributions and does not use xorg.conf configuration files and does not work with standard X modules. UPDD has been tested with this X and as of June 09 we have a working solution via the built in interface. In these environments it is likely that a manual setup is required as described here .			
	X11 Kdrive/TinyX				
	Xorg nanoX	Graphic system for smaller devices and platform – see http://www.microwindows.org/ . Untested and listed here for completeness. We would need to build a system using this X interface and test/address any issues that arise. Unlikely to work until this has been done.			
	None	Driver will load and work without X allowing for application UPDD API interface to the driver.			
	X is normally enhanced with a Window Manager that sits on top of X. See the Windows Manager installation note below.				
File utilities	mkdir, cd etc, Sound utility sox if using calibration beeps, see Hardware requirement below.				
Libraries	C	UPDD V4 uses dynamic linked library calls so requires the C library to be available. UPDD V4 utilises release 6 (libstdc++.so.6). C library version 6 has been available for a number of years and should be shipped as standard in most Linux distributions. For legacy distributions with earlier C libraries either install the V6 lib or use UPDD V3 (uses V5 C library).A C++ version 6 library that may be suitable in some distributions can be found here . If this library is not available in your distribution then we can supply a driver with statically linked components.			
Hardware	Sound card	If calibration beeps are enabled in the UPDD Console a sound card is required as we have not been able to access the internal PC speaker under Linux.			
	USB	For USB devices the USB file system must be a component of the distribution and enabled.			

If you need a driver for other environments, such as a different graphics manager or different processor we can update our driver to match your requirements but there will be a cost involved on a time and materials basis.

UPDD Linux has been around many years and used on many distribution and the old Linux documentation lists previous distributions tested. The following table shows the distributions tested since 4.1.1 release:

Distribution	OS id	OS rel	Proc	UPDD	Date	Test status	Install script
Redhat	Enterprise	4 & 5	x86 X64	4.1.1 4.1.1	Apr 08 Sept 09	Tested in-house	Redhat
Kubuntu	KDE Desktop Manager is the default desktop for Kubuntu. Jaunty Jackalope		9.04	X86	4.1.1	Apr 09	Tested in-house Kubuntu-Jaunty-Jackalope
Xubuntu	Utilises the XFCE desktop environment. Designed for low-specification computers.						
Linutop	- mini PC	8.04	X86	4.1.1	May 09	Reported OK	Ubuntu-Hardy-Herron
Ubuntu	GNOME Desktop Manager is the default desktop for Ubuntu.						
EEEbuntu	- EeePC	2.0	X86	4.1.1	Feb 09	Reported OK	
	Feisty Fawn	7.04	X86	4.1.1	Apr 07	Tested in-house	Ubuntu-Feisty-Fawn
	Gutsy Gibbon	7.10	x86	4.1.1	Apr 08	Tested in-house	Ubuntu-Gusty-Gibbon
LTS	Hardy Heron	8.04	x86	4.1.1	July 08	Tested in-house	Ubuntu-Hardy-Herron
	Intrepid Ibex	8.10	X86	4.1.1	Jan 09	Tested in-house	Ubuntu-Intrepid-Ibex
	Jaunty Jackalope	9.04	X86	4.1.1	Apr 09	Tested in-house	Ubuntu-Jaunty-Jackalope
Fedora Core	Karmic Koala	9.10	X86	4.1.1	Jan 10	Tested in-house	Ubuntu-Karmic-Koala
	Moonshine	7	x86	4.1.1	Apr 08	Tested in-house	Fedora-Core
	Werewolf	8	x86	4.1.1	Aug 08	Tested in-house	Fedora-Core-8
	Sulphur	9	x86	4.1.1	Aug 08	Tested in-house	Fedora-Core-9
	Cambridge	10	X86	4.1.1	May 09	Tested in-house	Fedora-Core-10
	Leonidas	11	X86	4.1.1	Aug 09	Tested in-house	Fedora-Core-11
	Unite	12	X86	4.1.8	May 10	Tested in-house for multi-touch extensions	
Suse		10.3	x86	4.1.1	Apr 08	Tested in-house	SuSE
	(KDE WM only!)	11.1	X86	4.1.1	Mar 09	Reported OK	SuSE-11
	(Gnome only!)	11.1	X64	4.1.1	Oct 09		
		11.2	X86	4.1.8	Mar 10	Tested in-house	SuSE-11.2
CentOS	Based on RedHat Enterprise Linux	4.6	x86	4.1.1	Apr 09	Reported OK	CentOS-4.6
		5.1	x86 x64	4.1.1 4.1.1	Jun 08 Sept 09	Reported OK	
Slackware		12.1	x86	4.1.1	Aug 08	Tested in-house	Slackware-12.1
Debian	Etch	4.x	x86	4.1.1	Aug 08	Reported OK.	Debian-Etch
	Lenny	5.x	X86	4.1.1	Mar 09	Tested in-house	Debian-Lenny
Guardlinux			X86	4.1.8	Feb 10	Tested in-house	Guada-Linux
Lliurex	Based on ubuntu		X86	4.1.8	Mar 10	Reported OK	Ubuntu-Jaunty-Jackalope
NimbleX	2008		X86	4.1.1	Jan 09	Reported OK	
Mandriva	2009		X86	4.1.1	Mar 09	Reported OK	
Gentoo	(Kernel 2.6.24)	10.1	X86	4.1.1	Jan 10	Reported OK	Manual install

Important – The table above shows the list of distributions that have been tested or reported as ok. However, given the number of Linux distributions it is impossible to guarantee that our install and/or driver will work in all environments or in all distributions, even if the same underlying kernel, X interface and desktop manager is in use or if a distribution is based on one of the supported distributions. To this end we have selected a number of main stream distributions to regularly test our latest driver and to try and keep the driver current, these being ubuntu, Fedora-Core, Redhat and Suse. Issues with our driver in any other distribution may incur a cost to investigate.

Linux Notes

We have observed some general and distribution specific issues that are documented below.

If you are using the graphical install on 'tested distributions' then, where possible, we have catered for the known issues in the individual Linux distribution setup scripts.

If you are manually installing, or installing on untested distributions or there are issues after install on tested distributions then please read the appropriate section below for your distribution:

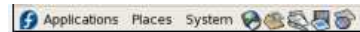
General

General

Issues

Window management issue – first noted on Fedora Core 5 and 6 and also reported on Suse 10.1 but could effect other distributions:

Generating a click via touch is sometimes ignored. In some areas of the Linux menu system we have found that unless the mouse is moved to the position of the click the click is ignored, such as the 2nd level of menus from the Application start menu option:



Some items within the applications menu will expand to a 2nd level menu. Lifting the stylus off the first menu and clicking into the second menu item *will not* activate the selected item. You need to slide your finger from the first menu to the second menu and lift the stylus off when over the desired item. This problem can also be seen with a mouse by having the mouse at the position of the second level item and clicking when it is shown. With no mouse movement prior to click, the click is ignored.

Calibration issues

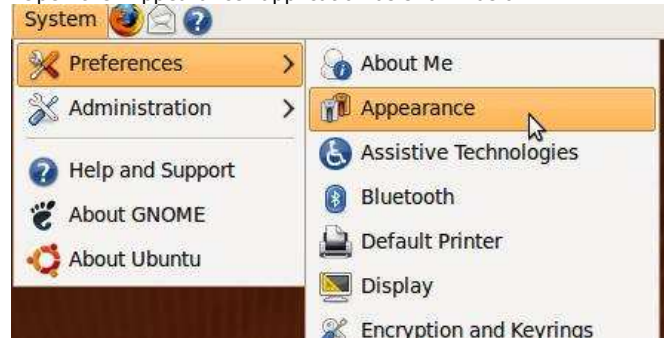
1) Full screen calibration

Correct operation of the calibration requires the calibration screen to be displayed in full screen mode. Some Linux distributions or certain visual effect settings cannot handle the method used by our graphical calibration program to force full screen with unpredictable results. To date we have seen this with Linux distribution Maemo and main stream distributions with certain visual effects enabled.

On systems where the full screen issue occurs, one of the following actions will be necessary:

1. Execute the calibration program in an environment that supports full screen display.
 - For example if a window manager is preventing full screen mode try running without the window manager active. (This has been necessary on systems using the TWM window manager).
 - Where visual effects are preventing full screen temporarily disable the effect. We have had reports that on some Ubuntu systems the calibration screen is windowed (having a titlebar) instead of taking up the entire desktop. This is due to the desktop effects interfering with the window manager. If this effect is experienced then follow these instructions to temporarily disable desktop effects for the duration of the calibration.

-Open the "Appearance" application as shown below:.



-Navigate to the "Visual Effects" tab as shown in the screenshot.



- If "Normal" or "Extra" is selected then select "None".
- Calibrate the touchscreen

- When calibration is complete select the visual effect that was in use before and click "Close"
- In cases where full screen is not possible we have utilised a UPDD calibration style, named Custom2Point, whereby calibration does not attempt full screen mode (and therefore cannot draw accurate calibration points) but instead requests that the top left and bottom right corners of the calibration area be used as the calibration reference points. No other calibration patterns are catered for. Either request a build from Touch-Base with this calibration style pre-defined or use the UPDD Console, calibration dialog to add the style (deleting the existing style) and then calibrate to invoke corner calibration.
- 2) Changing desktop resolution
- The driver does not automatically monitor desktop resolution and the resolution is captured at the point calibration is performed and used as part of the X interface. If resolution is changed a new calibration will be required.
- TWM window manager In order to calibrate you must run the calibration program (tbcilib) without the TWM window manager active so that the entire screen area is used. If this is not possible see calibration notes above.
- Driver background process The driver uses a background process to monitor video rotation and regularly calls xrandr for this purpose. One customer reported that this process caused their animated graphics to temporarily freeze every time xrandr was called. Should the background task cause issues on your system you can safely kill the process "tlinuxdaemon" to stop xrandr getting called. To disable it permanently remove the line "/bin/bash /opt/tbupddl/startedaemon #UPDD" from the file /etc/profile.

Distribution

Ubuntu 8.04, 8.10
and others

Issues

Some users of these Linux distros have reported that after calibrating, the cursor tracks incorrectly on the X axis (increasingly out to the right of the screen – we do not see this issue on any of our systems!). When this is seen we believe the driver has not correctly determined the X element of the screen resolution. To correct the problem it is necessary to store the screen resolution values in the X configuration file. Follow the instructions below to configure this:-

Either login as the "root" user OR open a terminal and type "su" to become the root user (entering the root password when prompted), or type "sudo bash" (entering your password when prompted) to get access to a root shell

Open /etc/X11/xorg.conf in a text editor (eg. by typing "emacs /etc/X11/xorg.conf" or "vi /etc/X11/xorg.conf" or use any other text editor)

Navigate to the section that reads:

```
Section "InputDevice" #UPDD
    Identifier "Updd0"
    Driver "xf86_tbuiddlx"
Option "SwapXY" "0"
    Option "Device" "/opt/tbupddl/comReadPipe"
EndSection #UPDD
```

Modify it to read

```
Section "InputDevice" #UPDD
    Identifier "Updd0"
    Driver "xf86_tbuiddlx"
Option "SwapXY" "0"
    Option "Width" "1024"
    Option "Height" "768"
    Option "Device" "/opt/tbupddl/comReadPipe"
EndSection #UPDD
```

Save the file

Now either reboot the computer OR shutdown the X server and then restart it
Replace the values of 1024 and 768 in the modified section to reflect the resolution that you are using.

If this fix is used then it is important to remember to update these values if a different screen resolution is used.

Slackware 10

We found two issues with this distribution:

- The Linux files are installed in a different file structure so UPDD install does not locate the configuration file and therefore there is some extra work to do after installation, as follows:

Install UPDD as normal.

Then create a file called "rc.updd" in the "/etc/rc.d" directory with the following content:-

```
#!/bin/sh

if [ "$1" = "stop" ]; then
    echo "Stopping updd..."
    killall tbupddwu
elif [ "$1" = "restart" ]; then
    echo "Restarting updd..."
```

```

killall tbupddwu
sleep 1
/opt/tbupddlx/tbupddwu &
else # assume $1 = start:
echo "Starting updd: /opt/tbupddlx/tbupddwu"
/opt/tbupddlx/tbupddwu &
fi

```

It should be marked executable by typing "chmod +x /etc/rc.d/rc.updd"

The file "/etc/rc.d/rc.local" should be modified to include the following 3 lines (the location of these lines is not significant so long as they do not split an existing code block):-

```

if [ -x /etc/rc.d/rc.updd -a -x /usr/sbin/syslogd -a -d /var/log ]; then
. /etc/rc.d/rc.updd start
fi

```

The system should then be rebooted.

We will automate this installation in a forthcoming UPDD release.

2. The default Linux kernel provided with Slackware 10.1 has problems with USB that prevent UPDD communicating with a USB device. However there is a 2.6 kernel supplied on CD2 of the distribution which functions correctly. To install and use this kernel follow the instructions located in mountpath>/testing/packages/linux-2.6.10/README.initrtd.

Mandrake 9.2
Elinos Embedded
KNOPPIX

Installs X 3.n.n. X needs to be upgraded to 4 to work.

Ships with X3.3.6 to keep embedded components as small as possible.

KNOPPIX is a live variant of Linux that is run completely off a CD by creating a RAMDrive. It is used a great deal for kiosk stations. This may require a manual install because KNOPPIX normally runs out of a RAMDrive and some of the storage areas are read-only which prevents the automatic installer writing files. In this environment the driver needs to be embedded into the Knoppix disk image and the file "/tbupddlx/tbupdd.reg" needs to be a symlink to this file on the RAMDrive. It also important that the files retain their case. The files norm*.gif and logo.gif should all be in uppercase.

Yellowdog

The UPDD Linux version 3 driver supports both USB and serial devices. However, given the lack of serial ports on the Macs, hardware serial support depends on the availability of a suitable serial to USB adaptor such as the Keyspan interface. We do not believe that the Keyspan or any other adaptor is available for Yellowdog (due to lack of required drivers), so to all intent and purpose there is no serial support. Work is required to produce a UPDD Version 4 power pc version.

Fedora Code 3

Fedora Core 3 has a built in HID driver that will take control of HID compatible touch controllers. For our driver to work with this distribution when using an HID compatible touch controller you will need to rebuild the kernel to make HID a loadable module.

With UPDD Version 4 this should now be resolved!

SUSE 9.1

We have found that Suse 9.1 has a non-standard implementation of HID and this stops our driver working with HID controllers. Although we could cater for this in UPDD we have found that SUSE 9.3 reverts back to a standard HID implementation and all is well.

Suse 10.x

One customer reported that the driver would not work with their USB touch controller and we asked for a USB device listing (as documented in our [Support Document](#)). To produce the device list the USB file system needed to be enabled and in doing this the driver worked OK. In other Suse 10.1 implementations it has not been necessary to do this so the UPDD installation does not automatically enable the USB file system. If you find that the driver does not work in Suse 10.1, enable the file system and test again. If it still does not work produce the Linux info as specified in the Support Document and we will investigate further

To enable the file system edit the file "/etc/fstab" and change the line that says:

```
usbfs /proc/bus/usbfs noauto 00 to usbfs /proc/bus/usbfs auto 00
```

and then reboot the system.

Another customer reported "In Suse 10.2 there's no USBFS, meaning with default Suse 10.2 kernel your driver cannot access USB devices as there is no /proc/bus/usb. In this case the usbfs system will have to be enabled. This particular customer used a non-default (but still Open-SUSE) kernel that gave backward compatible libusb access on /proc/bus/usb!

10.3 was tested 21st April 2008 and all was found to be OK.

SUSE 11.1 – X64

GNOME Window manager support only until we investigate driver issues when KDE is in use.

Gentoo Linux 2006.1

Perl needs to be installed (emerge -av perl)

CONFIG_USB_TOUCHSCREEN needs to be unset in the kernel

Install driver as per the installation instructions

The three lib files (libTBapi.so libACE.so.5.6.0 and libhbutton.so) need to be copied or symlinked to the /lib directory

Create file: /etc/init.d/updd with the following contents:

```

#!/sbin/runscript
# Copyright 1999-2006 Gentoo Foundation
# Distributed under the terms of the GNU General Public License v2
# $Header: $

```

```

depend() {
    use hotplug logger
}

start() {
    ebegin "Starting UPDD"
    /opt/tbupddlx/tbupddwu &
    eend ${?}
}

stop() {
    ebegin "Stopping UPDD"
    killall -quiet tbupddwu
    eend ${?}
}

```

Make the file above startable (chmod +x /etc/init.d/updd)

Start updd (/etc/init.d/updd start)

Include updd at startup (rc-update add updd default)

Running up the xsession will allow you to run the UPDD Console and calibrate the touchscreen.

Ubuntu 7.10

The USB file system needs to be enabled:

Edit the file: /etc/init.d/mountdevsubfs.sh

e.g gksudo gedit /etc/init.d/mountdevsubfs.sh

Change the lines:

```

#mkdir -p /dev/bus/usb/.usbfs
#domount usbfs "" /dev/bus/usb/.usbfs -obusmode=0700,devmode=0600,listmode=0644
#ln -s .usbfs/devices /dev/bus/usb/devices
#mount --rbind /dev/bus/usb /proc/bus/usb

```

to:

```

mkdir -p /dev/bus/usb/.usbfs
domount usbfs "" /dev/bus/usb/.usbfs -obusmode=0700,devmode=0600,listmode=0644
ln -s .usbfs/devices /dev/bus/usb/devices
mount --rbind /dev/bus/usb /proc/bus/usb

```

Restart and the usb devices will be recognised via the file system

Controllers

ELO

Issues

"If using an ELO controller and it doesn't work with UPDD then check the contents of your /etc/X11/xorg.conf file for the following lines:

```

Section "InputDevice"
    Identifier "elo"
    Driver      "elo"
    Option      "Device"      "/dev/input/elo_ser"
    Option      "SendCoreEvents" "true"
EndSection

```

If they exist then delete them and also delete the corresponding line "Inputdevice "elo"" in the "ServerLayout" section of your /etc/X11/xorg.conf file. The system should then be rebooted."

CD Distribution

We believe the best approach is to expand the contents of the linuxupdd.tgz file to the CD ensuring the permissions on the extracted files are preserved. The end user should then be able to run the installer directly from the CD. Please test before creating many copies!

Installation

You can install the driver using the supplied GUI installation program or manually ([standard Linux](#) / [embedded Linux](#)):

GUI installation Procedure

Ensure you have the correct setup file to match your processor. .e.g. x86 setup will install on X86-64bit but the driver will not work. In this instance you must have the 64 bit setup. Installation of the driver must be performed as the "root" user. It is usually possible to "become" the root user by typing command "su root" or preceding the setup command with "kdesu" or "sudo", but we have found on some systems this is not sufficient in which case you must log on at system start as root. Note that on SuSE 10.3 "sudo" does not allow the installer to work and you must use either "su" or "kdesu".

The main installation package is held within the compressed file called linuxupdd.tgz. Copy the file into **a directory other than "/opt" or "/"** such as a users home directory on the Linux system, change to that directory, then decompress it by using the command "tar zxvf linuxupdd.tgz".

Important - Do not decompress the software in the root directory or "/opt/" as the install will fail.

To install the software open a terminal window and either use "su" or "kdesu" to become the root user then type "./setup". Alternatively type "sudo ./setup" and enter your password (not on SuSE 10.3; see note above). This will launch the setup program:

- From the device list dialog select the touch device or, if it is a PnP device, optionally leave the PnP mechanism to pick up

the device(s).

- From the distribution list dialog select the distribution script if it is listed. Select the script that most closely matches your distribution. In no script matches select "Other". E.g. If an exact match is listed, say Ubuntu Gusty Gibbon (Ubuntu 7.1), select that.
- Install scripts can invoke question dialogs, such as 'OK to enable USB file system'. Answer these dialogs as appropriate.

Following installation **you must reboot the system** for the driver to load

Installation notes

1. The installation procedure is used to install the software for a single touch screen / UPDD supported pointer device. In a multi-touch screen or pointer device environment invoke the [UPDD Console - Hardware dialog](#) to add additional devices after installation. See the [Multi monitor and multi device documentation](#) for further information.
2. After installation it is a requirement to reboot the system as the graphical subsystem needs to be reloaded for any new drivers to work.
3. **Window Manager:** Most Linux distributions use a Window Manager on top of the X graphical sub-system. The window manager enhances window management such as window borders and minimise/maximise buttons, the taskbar and the functionality to manipulate windows. The underlying X system provides the graphics primitives to allow drawing to the screen. KDE and Gnome are the most popular window managers so if they are used the installation procedure will **attempt to**** create desktop icons for UPDD. ****** Given the nature of Linux as a moving development target the method used to create the icons may not always work in all distributions

When the system has restarted and user log in is completed there are two ways to configure the driver. If KDE or GNOME are the Windows managers then there **should be** two new icons on the desktop, Console and Calibrate, which can be used to change driver settings and calibrate the touchscreen respectively. On systems where the icons are not seen then a Terminal program should be executed and the user should type "/opt/tbupddlx/upddconsole" to run the Console, and "/opt/tbupddlx/upddcalib" to run the Calibration program. Alternatively manually create a Linux shortcut to the UPDD Console and Calibration program.

4. **Root User:** Calibration can not be activated when running as a root user. The problem is that when starting an X session, the user is authenticated by X and given permission to connect to the server. When switching users by using "su", the new user no longer has permission to connect to the X server and as a consequence cannot run graphical programs.
5. UPDD for Linux uses components of the kernel to provide access to the various hardware ports, such as USB, PS/2 and serial ports. In order to access controller hardware using a distribution that does not implement or mount these sub-systems by default the integrator will need to use the kernel documentation for the distribution in question to enable the appropriate interface.
6. For serial devices the driver handles the serial device via standard COM port names (/dev/ttySnn) or USB to serial adaptors (/dev/ttyUSBn), so to use a serial device with a different name it is required to create a symbolic link to one of these port types. In the UPDD serial dropdown you have a choice of COMn for hardware serial ports (maps to /dev/ttysnn-1 (e.g COM1 = TTYS0)) or Adaptorn for virtual serial ports (maps to /dev/ttyUSBn-1 (e.g Adaptor1 = TTYUSB0)) created via a Serial to USB adaptor. See [Serial Port notes](#) for more information.

Folder structure

Following installation the following folder structure will have been created/updated on your Linux system:

/opt/tbupddlx/*

Contains the ini file, calib gif files, etc

/usr/X11R6/lib/modules/input/xf86_tbupddlx.o

This is the X module for systems using X11R6 based X Window Systems such as Xfree86 and early versions of x.org

/usr/lib/xorg/modules/input/xf86_tbupddlx_drv.so

This is the X module for systems using X11R7 based X Windows Systems such as the latest x.org

/etc/init.d/tbupdd

/etc/rc2.d/S90tbupdd

/etc/rc3.d/S90tbupdd

/etc/rc5.d/S90tbupdd

These automatically load the daemon on system boot

/etc/X11/XF86Config-4 or /etc/X11/XF86Config or /etc/X11/xorg.conf (depending on X version used)

This pre-existing file gets modified to add a section to load the UPDD XFree86 module

Manual Install

UPDD for Linux is shipped with a graphical installation program. In general this program should be used in conjunction with the supplied instructions to install the software as defined above. In some circumstances however it might be necessary to manually install the software, if for example to use Linux in an embedded environment. The instructions below show the steps required to take the contents of the file linuxupdd.tgz and install the contents manually on a Linux system. The instructions are for the use of personnel familiar with the Linux command line interface. The commands should be issued by a user logged on as root.

The manual installation instructions are as follows:

```
-cp <location of tgz>/linuxupdd.tgz /
-cd /
-tar zxvf linuxupdd.tgz
-rm /setup
```

```
-cd /opt/tbupddlx
```

[Standard Linux]

```
-cp S90tbupddlx /etc/rc.d/init.d/tbupdd
-ln -s /etc/rc.d/init.d/tbupdd /etc/rc.d/rc2.d/S90tbupdd
-ln -s /etc/rc.d/init.d/tbupdd /etc/rc.d/rc3.d/S90tbupdd
-ln -s /etc/rc.d/init.d/tbupdd /etc/rc.d/rc5.d/S90tbupdd
[Suse 10.1]
-cp S90tbupddlx /etc/rc.d/tbupdd
-ln -s /etc/rc.d/tbupdd /etc/rc.d/rc2.d/S90tbupdd
-ln -s /etc/rc.d/tbupdd /etc/rc.d/rc3.d/S90tbupdd
-ln -s /etc/rc.d/tbupdd /etc/rc.d/rc5.d/S90tbupdd
```

```
-chmod +x tbupddwu
-chmod +x tbcilib
-chmod +x dcu
-chmod +x xins
-chmod +x upddconsole
-chmod +x upddcalib
-chmod +x helpviewer
-chmod +x updddrawing
```

The X interface sections below, marked ===== are not required if the UPDD driver is being installed on a distribution using the built-in X interface, as [listed here](#):

=====

For X11R6 based systems

```
-chmod +x xf86_tbupddlx.o
-mv xf86_tbupddlx.o /usr/X11R6/lib/modules/input (32 bit systems)
-mv xf86_tbupddlx.o /usr/X11R6/lib64/modules/input (64 bit systems)
```

For X11R7 based systems

```
-chmod +x xf86_tbupddlx_drv.so
-mv xf86_tbupddlx_drv.so /usr/lib/xorg/modules/input (32 bit systems)
-mv xf86_tbupddlx_drv.so /usr/lib64/xorg/modules/input (64 bit systems)
```

=====

NOTE: For linux distributions where the driver does not automatically utilise the built in user mode mouseport X interface but it is desirable to do so these notes may help:

"Remove xmodule references from the xorg.conf file and run "touch /opt/tbupddlx/usermodemouseport" to enable the driver daemon to use the built in interface. You now have to configure the system to run "tlinuxmouse" when X starts (otherwise the cursor won't move). This is usually done by modifying the gdm startup script (/etc/gdm/Init/Default) to add "tlinuxmouse &" near the end. Also "/opt/tbupddlx/tlinuxmouse" sometimes needs to be copied to somewhere in the standard path otherwise gdm won't load it for security reasons. E.g. it could go into /usr/local/bin, or /usr/bin, etc."

=====

For systems using SELinux:-

```
"chcon -t texrel_shlib_t *.so" should appear after the line "chmod +x updddrawing"
```

For systems that have a "/usr/local/lib" directory:-

```
-cp libTBApi.so /usr/local/lib
-cp libhbutton.so /usr/local/lib
-cp libqt-mt.so.3 /usr/local/lib
-cp libACE.so.5.6.0 /usr/local/lib
-ldconfig /usr/local/lib
..otherwise:-
-cp libTBApi.so /usr/lib
-cp libhbutton.so /usr/lib
-cp libqt-mt.so.3 /usr/lib
-cp libACE.so.5.6.0 /usr/lib
-ldconfig /usr/lib
```

For USB and PS/2

```
./xins
```

For Serial

```
./xins COM<X>
```

```
-rm xins
```

You then need to run a script according to the specific distribution you are using. The scripts are [shown here](#).

Choose the appropriate script for your distribution and type the command "perl <script>"

e.g. For SuSE Linux - "perl SuSE.ins"

If your distribution isn't listed then use "Other.ins".

For SUSE 10.1 only

The file /etc/rc.d/.depend.start should be loaded in an editor and the line "tbupdd: " should be added to the bottom of this file.

The following line appears at the top of the file which reads similar to:-

```
"TARGETS = kbd nfs fbset earlykbd splash microcode network dbus "
```

This should be amended to add "tbupdd" to the end:-
i.e. "TARGETS = kbd nfs fbset earlykbd splash microcode network dbus tbupdd"

Then restart the system.

Once the system has restarted, start the UPDD Console as per the user guide and click "Add a new device" and follow the on screen directions to add your touchscreen configuration.

A list of dependencies for the UPDD Linux software can be found [here](#).

Manual install - Embedded Linux systems

Given the nature of embedded Linux systems, such as those utilising Tinyx/Kdrive, it is possible that our standard install scripts may not set up the driver requirements correctly, especially the launching of the daemon processes. To this end it is recommended that the individual driver modules be manually installed in these environments.

Basic instructions to manually install and run these programs follow and should be read in conjunction of the full manual install section above. These instructions are for the use of personnel familiar with the Linux command line interface. The commands should be issued by a user logged on as root. We cannot provide detailed instructions without details of the specific target system but we hope these basic instructions will suffice. We can of course answer any specific questions you might have.

```
-Copy the driver package (linuxupdd.tgz) to the root ("/") directory on the target system. ie "cp <location of
tgz>/linuxupdd.tgz /"
-Type the following commands:-
-"cd /"
-"tar zxvf linuxupdd.tgz"
-"rm /setup"
-"cd /opt/tbupddlx"
-"touch usermodemouseport"
```

This will copy the components to the correct locations. You will also need a copy of "libusb-0.1.so.4" which should be copied into the directory "/opt/tbupddlx". If you don't have a copy then we can provide it.

You then need to start two daemons: tbupddwu, and tlinuxdaemon. The standard place to start them from is from your "rc" files so that they will run automatically when the system starts. You will also need to set the LD_LIBRARY_PATH environment variable to include "/opt/tbupddlx" in your "rc" script. Note that tbupddwu must be launched before tlinuxdaemon.

To calibrate you need to run the script "/opt/tbupddlx/upddcalib".

To run the UPDD Console (settings) you need to run the script "/opt/tbupddlx/upddconsole". Please note - in some embedded systems we have seen that the UPDD Console and it's related test utility (updddraw) does not load due to missing font files. Only when the dependant modules are available will these programs load without error.

Calibration

After the restart and if using the KDE or Gnome window manager, there should be an icon on the desktop to calibrate:



If using a different window manager, open up a shell and type:-
/opt/tbupddlx/upddcalib

or

```
cd /opt/tbupddlx
./upddcalib
```

or alternatively create a link to it using the window manager.

To calibrate, double click the calibration icon on the desktop or run the calibration program and touch the calibration crosses, or arrows, as they appear. **Important note:** If the calibration screen does not cover the full desktop area see [calibration notes](#) above. Full calibration procedure information can be found in the [Calibration document](#).

Calibration can be performed in any screen resolution and the calibration data held is relative to the screen resolution. However, if the screen resolution is changed then a new calibration will be required as currently we do not automatically track screen resolution changes.

Driver/Controller settings

The UPDD Console defines the functionality of the pointer device(s) and the UPDD driver environment.

If using the KDE or Gnome window manager, there should be an icon on the desktop to invoke the DCU



If using a different window manager, open up a shell and type:-

```
/opt/tbupddlx/upddconsole
```

or

```
cd /opt/tbupddlx  
./upddconsole
```

To configure the UPDD settings invoke the UPDD Console program and change the setting as required.

See the [UPDD Console documentation](#) and [on-line help](#) for further information.

Serial port notes

Change serial port connection

The [UPDD Console - Hardware tab](#) allows the COM port name to be reassigned after installation.

Standard serial ports

If using a standard serial port, Select Com1, Com2 etc in the COM port selector. Serial ports should be registered in the system as ttySn which is mapped to our driver to COMn+1 (e.g. COM1 = ttyS0)

Serial to USB adaptors

If using a serial to USB adaptor, select Adaptor 1, Adaptor 2, etc in the COM port selector. This has only been tested with the Keyspan adaptors so far but the Linux documentation states that the interface is the same for all serial adaptors; hence UPDD should work for all serial/USB adaptors. Serial to USB adaptors should be registered within the system as ttyUSBn which is mapped by our driver to Adaptor n+1. (e.g Adaptor1 = ttyUSB0)

Serial port reassignment

The driver handles serial devices via standard COM port names (/dev/ttySnn) or USB to serial adaptors (/dev/ttyUSBn), so to use a serial device with a different name it is required to create a symbolic link to one of these port types.

Example: Assuming you have a serial port referenced as ttyC1P3 to be reassigned. You need to open a terminal with root privileges and type the following:

```
In -s /dev/ttyC1P3 /dev/ttyUSB0 (for Adaptor 1)  
or  
In -s /dev/ttyC1P3 /dev/ttyS0 (for com port 1)
```

You will then need to open up the UPDD Console and change the COM port for your device to "Adaptor 1" or "Com 1".

Serial port testing

Should the serial port connection not be working there are a number of procedures to follow to help identify the problem as described in the knowledge base article [here](#).

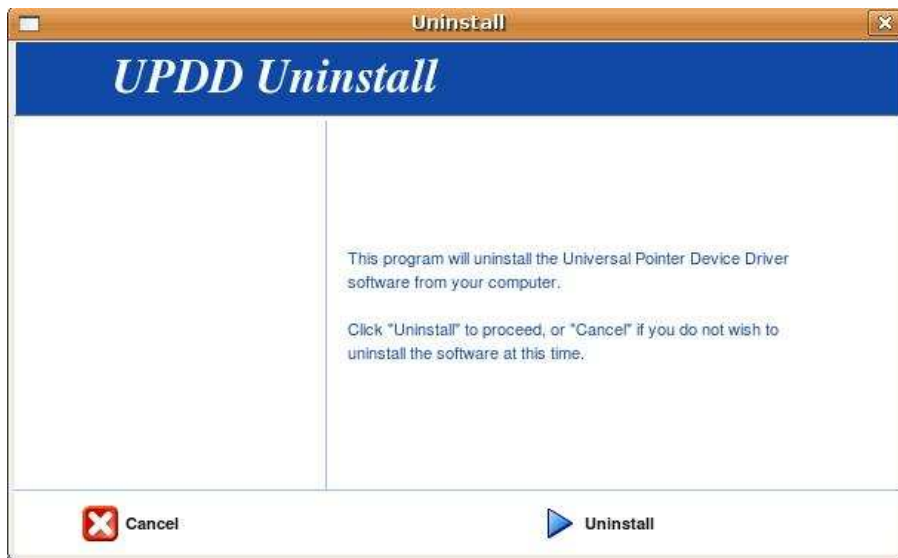
Uninstall

Automatic (4.1.1 - 26th Sept 2008 onwards)

Note: *The automated uninstall will only fully remove installations that have been installed from this build and later. If installing this build over an earlier build and then uninstalling using the "uninstall" program the changes to /etc/X11/xorg.conf will not be removed correctly. In the case of installing this build on a system which has had another build on it, the best scenario is to manually remove the old software and then install the new.*

Removal of the driver must be performed as the "root" user. It is usually possible to "become" the root user by typing command "su root" or preceding the setup command with "kdesu" or "sudo", but we have found on some systems this is not sufficient in which case you must log on at system start as root. Note that on SuSE 10.3 "sudo" does not allow the uninstaller to work and you must use either "su" or "kdesu".

To uninstall the software open a terminal window and either use "su" or "kdesu" to become the root user then type "/opt/tbupddlx/uninstall". Alternatively type "sudo /opt/tbupddlx/uninstall" and enter your password (not on SuSE 10.3; see note above). This will launch the uninstall program:



To continue and remove the software click on "Uninstall". To cancel, click "Cancel" and the system will remain unmodified.

Manual Uninstall

Type the following commands being careful to use the same case and spacing.

su

Enter the root password

rm -rf /opt/tbupddl *If this is mistyped, the whole system could be wiped.

For Systems using X11R6

rm /usr/X11R6/lib/modules/input/xf86_tbusddl.o

For Systems using X11R7

rm /usr/lib/xorg/modules/input/xf86_tbusddl_drv.so

rm /etc/rc.d/init.d/tbusdd

rm /etc/rc.d/rc2.d/S90tbusdd

rm /etc/rc.d/rc3.d/S90tbusdd

rm /etc/rc.d/rc5.d/S90tbusdd

rm /usr/local/lib/libTBApi.so

rm /usr/local/lib/libhbutton.so

rm /usr/local/lib/libqt-mt.so.3

rm /usr/local/lib/libACE.so.5.6.0

Edit the file "/etc/X11/XF86Config-4" for Xfree86, or "/etc/X11/xorg.conf" for x.org and remove the following section:-

Section "InputDevice"

Identifier "Updd0"

Driver "xf86_tbusddl"

Option "Device" "/tbusddl/comReadPipe"

EndSection

In the section that begins with:-

Section "ServerLayout"

Remove the line:-

InputDevice "Updd0" "SendCoreEvents"

Multi-monitor and multi-device support

Support for multiple monitors was added in UPDD Version 4.0.4 and is covered in full in the multi-monitor and device document, [Linux section](#).

Display rotation considerations

Linux supports rotated video modes for supported video cards under both Xfree86 and X.org. UPDD will work with rotated video and this is explained in detail in the separate [rotate documentation](#).

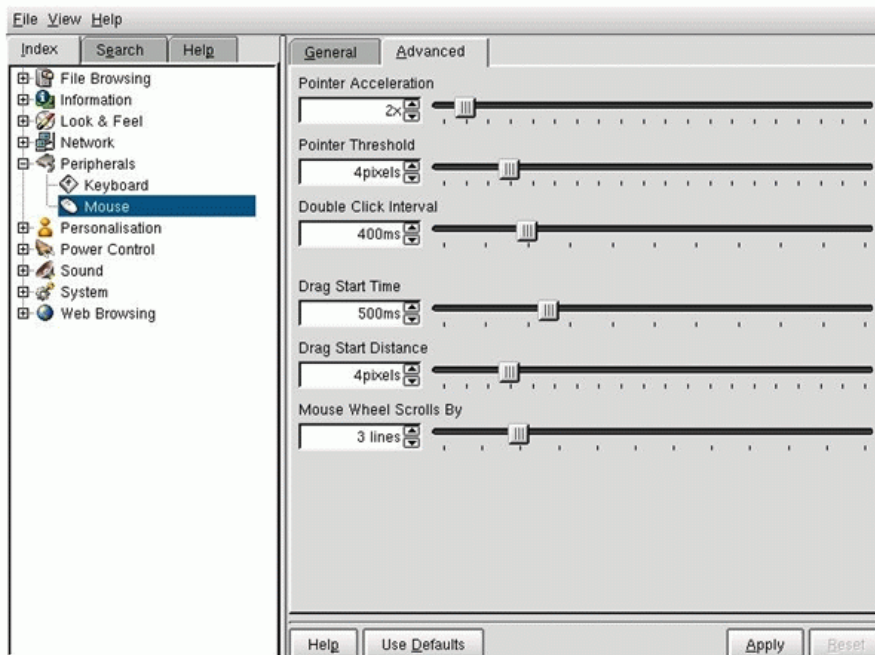
Display resolution / calibration considerations

The calibration mapping is based on the screen resolution setting at the time of calibration so if the resolution is changed the calibration will be inaccurate. To cater for this you will need to manually recalibrate after changing video resolution.

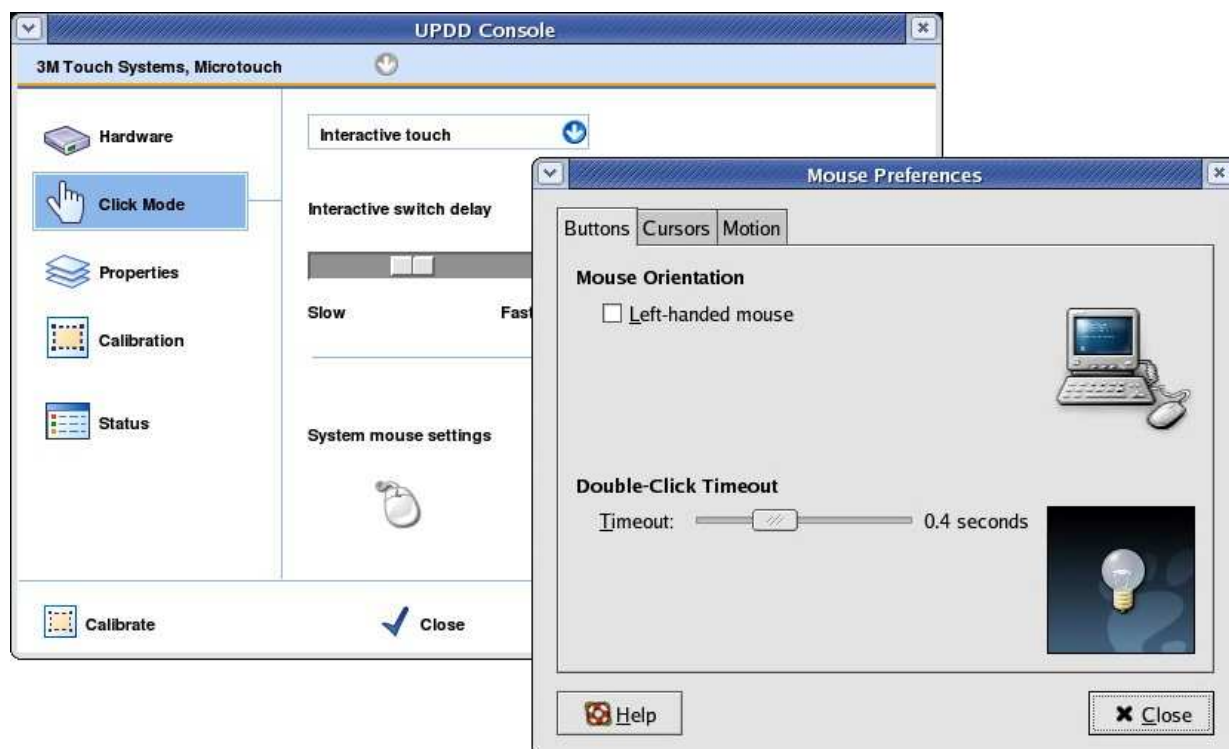
Future releases of the driver may well introduce a daemon process to automatically monitor video resolution and adjust automatically but until such times as this is available manual intervention is required.

Mouse settings

Double click capabilities are affected by the system's Mouse settings. To achieve a double click using the pointer device these settings need to cater for the type of device in use. A touch screen may well require different settings to that required by a mouse. The main setting that affects the ability to double click is the double click speed. If this is set too fast it may be impossible to produce a double click. Ensure this is set to an appropriate value in the mouse settings to allow for double clicks via a stylus. In this example the mouse settings screen is from the KDE Control Centre. Other window managers will have different ways of configuring the settings.



The UPDD Console, Click Mode dialog, System Mouse settings will invoke the Mouse settings for KDE and Gnome desktops, as shown in the following example:



Touch Utilities

Virtual Keyboards

A number of Virtual keyboards are available on the Web for Linux as detailed in the [UPDD Virtual keyboard documentation](#).

Mouse Cursor

At the time of writing we are not aware of any specific end user utility to change the mouse cursor or turn it on/off. Please contact us if you find such a suitable utility that we can document for other users.

Current Limitations

UPDD was originally developed for Windows and has since been ported to other OS. Not all features have been ported to Linux, they include:

- Dynamic detection of system language.
- Serial port auto-detection
- Interactive touch – visual notification of right click count down
- Anchor mouse function
- No sound on touch or during calibration
- Light pen calibration (the white lines on black mode)
- Toolbar actions
- Extensions

Contact

For further information or technical assistance please email the technical support team at technical@touch-base.com