Wacom Graphire USB Mini-HOWTO

Table of Contents

Wacom Graphire USB Mini-HOWTO

Arnaud Claden

01 June 2000

This is the Wacom Graphire USB Mini-HOWTO. This document describes how to setup a Wacom Graphire USB tablet for use with Linux (console and X), starting with the kernel configuration to the application level. But information should be accurate for the Intuos USB tablet.

1. Copyright

2. Introduction

This document is a guide to getting your Wacom Graphire USB tablet to working with Linux. Beware, this is a quite difficult task, but this guide will guide you, step-by-step, through the process.

This work is mainly derived from the Wacom Tablet HOWTO by Stephan Runkel, but brings some precisions about all the USB configuration that is absent from the original document. If you want more precision about general Wacom Tablets, please read Stephan Runkel's HOWTO.

I have written this document because I've had many difficulties in setting up my brand new Graphire tablet with Linux, but now it works quite well on my Toshiba 4030CDT Laptop.

2.1 New versions of this document

New version of this document will be made available at the following location :

http://perso.club-internet.fr/tiamat/computer/howto/Wacom_USB-Mini_HOWTO.html

2.2 Feedback and Corrections

Comments, corrections, and additions will greatly be appreciated. I can be contacted at:

<tiamat@club-internet.fr>

2.3 Acknowledgements

I wish to thank everyone that work on the USB part of the kernel, and special thanks go to Vojtech Pavlik for his work on the wacom module.

Of course, I thank Stephan Runkel for his great Wacom Tablet HOWTO that helped me much in the configuration process.

Special thanks also go to Frederic Lepied for his work on the xf86Wacom module.

2.4 Disclaimer

The information in this document is correct to the best of my knowledge, but there's a always a chance I've made some mistakes, so don't follow everything too blindly, especially if it seems wrong. Nothing here should have a detrimental effect on your computer, but just in case I take no responsibility for any damages incurred from the use of the information contained herein.This document is not a kernel howto, so I'll take the assumption that you know how to download/install a new kernel, and how to configure/compile it.

3. Requirements

3.1 Hardware

The main part is, of course, a computer with at least one USB port and a Wacom Graphire USB tablet.

3.2 Kernel

I've used the latest development kernel for my installation : linux-2.3.99-pre6. It's quite stable and has all the necessary features. You can also find some downports, which are patches for old 2.2.x kernels. You can find all the infos on the backport at http://www.suse.cz/development/usb-backport

3.3 Console Software

For the Linux console, the only program you need is gpm, available at ftp://ftp.prosa.it/pub/gpm

3.4 XWindow

For having all the abilities of your tablet, you'll need to use XInput. More specifically you'll need the xf86Wacom.so module (that is shipped with the most recent versions of XFRee86 starting from 3.3.5). Alternately, you may need the xf86WacomUSB.so, which is a work-in-progress of adapting the original xf86Wacom.so to use the USB. I'll speak of it later in this document.

3.5 Others

The applications supporting XInput are :

1. The Gimp - a powerful image manipulation program in the style of Photoshop. The 1.1.x instable

development version includes XInput support as a standard feature. You can obtain Gimp from http://www.gimp.org

2. gsumi - a simple B/W drawing program that supports drawing / erasing with pressure and tilt sensitivity. Get it from the gsumi web page: http://www.gtk.org/~otaylor/gsumi/
3. xink (By Ralph Levien) - another rudimentary drawing program for X. xink is available from: ftp://kiwi.cs.berkeley.edu/pub/raph/xink.tar.gz
4. RasMol - a molecular visualization program that supports a hardware dial box using XInput. look at http://www.umass.edu/microbio/rasmol/distrib/rasman.htm .
5. xinput (by Frederic Lepied) - a very useful utility for configuring and testing XInput devices. xinput is available from: ftp://ftp.x.org/contrib/utilities/xinput-1.2.tar.gz

4. Kernel and hard stuff

4.1 Kernel configuration

Like I said earlier, the options I'll give come from the 2.3.99pre6 kernel. If everything is not exactly the same, I think that taking a good guess should help ;)

**** Reminder : [] stands for Yes/No options, and <> stands for Yes/No/Module ****

First of all, you have to enable the experimental code :

Code Maturity level options --->

[*] Prompt for development and/or incomplete code/drivers

If you don't, you won't have access to some of the drivers.

Then, move on to the USB section.

USB support --->

<*> Support for USB

[*] USB verbose debug messages

--- Miscellaneous USB options

[*] Preliminary USB device filesystem

With this option, you'll activate the new features of the /dev/usb/* (which, of course, you'll have to create : to be seen later)

--- USB controllers

<*> UHCI (Intel PIIX4, VIA, ...) support

< > OHCI (Compaq, iMacs, OPTi, SiS, ALi, ...) support

You must choose one of these, and only one. So, if one doesn't work, then choose the other. Chances are that, if you use an x86 system, you'll take the UHCI driver.

--- USB Devices

< > USB Printer support

< > USB Scanner support

< > USB Audio support

< > USB Modem (CDC ACM) support

If you have one of these, feel free to activate the options

<M> USB Serial Converter support

This options must be activated for the Wacom tablet to work.

[*] USB Generic Serial Driver

[ ] USB Handspring Visor Driver

[ ] USB ConnectTech WhiteHEAT Serial Driver (EXPERIMENTAL)

[ ] USB FTDI Single Port Serial Driver (EXPERIMENTAL)

[ ] USB Keyspan PDA Single Port Serial Driver (EXPERIMENTAL)

[ ] USB ZyXEL omni.net LCD Plus Driver (EXPERIMENTAL)

[*] USB Serial Converter verbose debug

If you want to have all the messages (which can be very helpful in the beginning). When satisfied with the functionning, you can deactivate it.

< > USB Kodak DC-2xx Camera support

< > USB Mustek MDC800 Digital Camera support (EXPERIMENTAL)

< > USB Mass Storage support (EXPERIMENTAL)

< > USS720 parport driver

< > DABUSB driver

< > PLUSB Prolific USB-Network driver (EXPERIMENTAL)

< > USB ADMtek Pegasus-based device support (EXPERIMENTAL)

< > USB Diamond Rio500 support (EXPERIMENTAL)

--- USB HID

The HID is kinda the interface between you and your devices, so you MUST activate it.

<M> USB Human Interface Device (HID) support

< > USB HIDBP Keyboard support

< > USB HIDBP Mouse support

<M> Wacom Intuos/Graphire tablet support

There we go, the Wacom support should be activated (unless you read this just for fun ;))

< > Logitech WingMan Force joystick support

< > Keyboard support

<M> Mouse support

To be activated to use the Wacom Intuos/Graphire mouse

(1024) Horizontal screen resolution

(768) Vertical screen resolution

<M > Joystick support

<M> Event interface support

4.2 Kernel compilation

Now the easy stuff :

When you exit the configuration do as they say

make dep bzImage modules modules_install

Type it just like this and the computer will run all the steps one after the other.

Then copy the file bzImage to /boot, copy the System.map to /boot, edit your lilo.conf to point to your new kernel and type **/sbin/lilo** at the prompt.

**** Of course all of this (kernel config, kernel compilation and lilo tweaking must be done as root) ****

4.3 Before rebooting

FStab

If you want to avoid error messages at the next reboot, you'll have to do the following :

Edit the file /etc/fstab and add the following lines

```
none /proc/bus/usb usbdevfs defaults 0 0
```

This one is to activate the usb device filesystem

```
none /var/shm shm defaults 0 0
```

This one is because of a change in the shared memory stuff in the kernel.

USB related devices

Next, if you want to be able to use the USB :

```
mkdir /dev/input
```

```
mknod /dev/input/mouse0 c 13 32
```

This will activate the mouse support

```
mknod /dev/input/event0 c 13 64

mknod /dev/input/event1 c 13 65

mknod /dev/input/event2 c 13 66

mknod /dev/input/event3 c 13 67
```

These will activate the stylus : moves, pressure, button and eraser.

Modules configuration

Edit the file /etc/modules.conf and add the following line :

```
alias char-major-13 usbcore
```

What I did next

Next I added in the /etc/rc.d/boot.local file (remember, I use a SuSE distribution, but you should fine an equivalent for RedHat in /etc/rc.d/init.d/boot.local or /etc/rc.d/init.d/rc.local)

```
/sbin/modprobe wacom

/sbin/modprobe mousedev

/sbin/modprobe evdev
```

Just to be sure the modules will load at boot time.

Last but not least

Reboot ;)

Check your boot messages (**dmesg | less**), and see if there are no errors related to USB. It may be because you've chosen UHCI and your system is an OHCI one.

On my system, the following messages are shown :

```
.....

.....

usb.c: registered new driver usbdevfs

usb.c: registered new driver hub

usb-uhci.c: $Revision: 1.9 $ time 17:54:28 May 8 2000

usb-uhci.c: High bandwidth mode enabled

usb-uhci.c: Intel USB controller: setting latency timer to 0

usb-uhci.c: USB UHCI at I/O 0xffe0, IRQ 11

usb-uhci.c: Detected 2 ports

usb.c: new USB bus registered, assigned bus number 1

usb.c: USB new device connect, assigned device number 1

usb.c: kmalloc IF c3fcd2e0, numif 1

usb.c: new device strings: Mfr=0, Product=2, SerialNumber=1

usb.c: USB device number 1 default language ID 0x0

Product: USB UHCI Root Hub

SerialNumber: ffe0

hub.c: USB hub found

hub.c: 2 ports detected

hub.c: ganged power switching

hub.c: standalone hub
hub.c: global over-current protection
hub.c: power on to power good time: 2ms
hub.c: hub controller current requirement: 0mA
hub.c: port 1 is removable
hub.c: port 2 is removable
hub.c: local power source is good
hub.c: no over-current condition exists
hub.c: enabling power on all ports
usb.c: hub driver claimed interface c3fcd2e0
.....
.....
hub.c: port 1 connection change
usb.c: USB new device connect, assigned device number 2
usb.c: kmalloc IF c3fcd620, numif 1
usb.c: skipped 1 class/vendor specific interface descriptors
usb.c: new device strings: Mfr=1, Product=2, SerialNumber=0
usb.c: USB device number 2 default language ID 0x409
Manufacturer: WACOM
Product: ET-0405-UV1.1-1
usb.c: unhandled interfaces on device
usb.c: This device is not recognized by any installed USB driver.
Length = 18
DescriptorType = 01
USB version = 1.00
Vendor:Product = 056a:0010
MaxPacketSize0 = 0
```

```
NumConfigurations =1

Device Version = 1.11

Device Class:SubClass:Protocol = 00:00:00

Per-interface classes

Configuration:

.....

.....
```

When the kernel says "**This device is not recognized by any installed USB driver**", it's because, at boot time, the wacom module is not yet loaded. But as you can see, the whole USB bus and interfaces are recognized. So if your system doesn't work and doesn't have such messages, try to check if you haven't forgotten an USB option (something I indicated to be put in the kernel and that you put as a module : if it's the case, in the **/etc/rc.d/init.d/boot.local**, put the necessary modprobe **BEFORE** the 3 lines I indicated, for example **/sbin/modprobe usb-uhci**).

When the wacom module loads, here are the messages :

```
.....

.....

usb.c: registered new driver wacom

input0: Wacom Graphire on usb2

usb.c: wacom driver claimed interface c3fcd620

mouse0: PS/2 mouse device for input0

mice: PS/2 mouse device common for all mice

event0: Event device for input0

.....

.....
```

Retry and check everything until you get all these messages (or corresponding ones).

5. The Linux console

Now that you have your USB activated, it's just about time to try it all with gpm :

If gpm is activated at boot time, you'll have to kill it : **gpm -k**

Then try your Wacom tablet : **gpm -tIMPS/2 -m /dev/input/mouse0**

Move your stylus, or your mouse, and TADA ... the cursor moves on the screen.

6. XWindow

6.1 Having 2 mouses on the same system

Have you ever dreamed of having 2 mouses on your system ? If the answer is yes, do the following.

Edit the file **/etc/XF86Config**

The modules section should look like this

Section "Module"

Load "xf86Wacom.so"

EndSection

Then check the XInput section :

Section "XInput"

SubSection "Mouse"

DeviceName "WacomTablet" --> or whatever you like

Protocol "IMPS/2" --> I didn't manage to have imwheel recognize the wheel, but if you do, drop me an email

Port "/dev/input/mouse0"

Buttons 5

ZAxisMapping 4 5

AlwaysCore --> means that the Wacom mouse is always considered as the core pointer, ie if you move your normal mouse or the Wacom mouse (or stylus), it'll be just the same.

EndSubSection

EndSection

And there you are, the Wacom mouse (or stylus) is now considered as a second mouse, but your normal one still works.

6.2 Now on to the real stuff

Back to the XInput section, after the parameters for the mouse, here is the real Wacom stuff :

```
SubSection "WacomStylus"

Port "/dev/input/event0" --> the USB event device

DeviceName "WacomStylus" --> or whatever you want to call it

Mode Absolute --> ie the tablet surface is equivalent to the screen surface

#Suppress 6

DebugLevel 10 --> once everything works, set it to 0

#TiltMode

#HistorySize 200

#AlwaysCore

#KeepShape

#TopX 20412 --> the 4 lines are used to fine-tune the coordinate to fit with the screen

#TopY 14730

#BottomX 0

#BottomY 0

EndSubSection

SubSection "WacomEraser"

DeviceName "/dev/input/event0"

DeviceName "WacomEraser"

Mode Absolute

EndSubSection

SubSection "WacomCursor"

Port "/dev/input/event0"

DeviceName "GraphireMouse"
```

Mode Absolute

EndSubSection

6.3 The new module

The xf86Wacom.so is not specifically written for the USB tablets, and it's even not working correctly with these tablets. A new driver has been written by Matsumura Namihiko <po-jp@geocities.co.jp> has written one named xf86WacomUSB (available as source code at the same location than this Mini-HOWTO). However, to be able to compile it, you'll need the kernel source code and the source tree for XFree86 (I compiled it with success against XFree86 3.3.5, but it should compile with 3.3.6).

This driver doesn't handle all the possibilities of the tablet, and crashes if you pass it the **AlwaysCore** parameter.

To correct this, I'll keep in touch with Frederic Lepied for a modification of the original module to have it handle all of the possibilities of the tablet.

****NEWS** At Frederic Lepied's site, you'll find a link to a new USB driver that works perfectly well, but exclusively with the possibility to have your tablet working as a mouse : ie either you use your tablet in The Gimp to draw, or you use it as an XFree mouse, but not both together. I'll investigate further, and update this document accordingly.**

7. Further documentation

For more information about Wacom tablets in general, try the great Wacom tablet HOWTO from Stephan Runkel <runkel@runkeledv.de> : http://www.linuxdoc.org/HOWTO/Wacom-Tablet-HOWTO.html

About the xf86Wacom.so, check Frederic Lepied's <lepied@xfree86.org> : http://www.lepied.com/xfree86

xf86WacomUSB.so, available at ftp://ftp.aros.org/pub