Handspring−Visor mini−HOWTO

Table of Contents

Handspring−Visor mini−HOWTO.....1
Ryan VanderBijl.....1
1.Introduction.....1
2.Installing Stuff.....1
3.References.....1
1.Introduction.....1
1.1 About this Document.....1
1.2 Disclaimer.....2
1.3 History.....2
1.4 Copyright.....2
2.Installing Stuff.....2
2.1 Requirements.....2
2.2 Getting your kernel to support USB.....3
2.3 Installing pilot link.....4
2.4 Almost Done! (aka testing).....4
3.References.....5

Handspring−Visor mini−HOWTO

Ryan VanderBijl

v0.3, April 7, 2000

Using the Visor with Linux and your USB port

1.Introduction

- 1.1 About this Document
- 1.2 Disclaimer
- 1.3 History
- 1.4 Copyright

2.Installing Stuff

- 2.1 Requirements
- 2.2 Getting your kernel to support USB
- 2.3 Installing pilot link
- 2.4 Almost Done! (aka testing)

3.References

1.Introduction

1.1 About this Document

This document was first created by Ryan VanderBijl, April 6, 2:00am. It has been edited by Miles Lott. I recieved my Handvisor today, and have struggled getting it to work, and found no good HOWTO yet, so thought i'd make one. By some random mutation, you might be able to find a new version on my web page, but that is up to chance. Test your luck at:

http://www.calvin.edu/~rvbijl39/

or

http://usbvisor.sourceforge.net

I used the vi editor. Anything else would have been evil. No spell check has yet been run on this document. So, if you have a problem with my spellling, tuff luck.

1.2 Disclaimer

Hmm... what did I claim? Oh well. I am really a newbie at this usb/visor stuff, so i really couldn't answer to many of your questions. There are people who could help you much better, who maintain the sites I refer to at the end of the document. However, feel free to send me questions, or updates to this document. My email address is: rvbijl39@calvin.edu

1.3 History

- v0.1 − initial release
- v0.2 − HTML−ized version
- v0.3 − SGML−ized, Added modifications by Miles Lott

1.4 Copyright

2.Installing Stuff

2.1 Requirements

Some obvious things are required, such as a computer, USB ports, hand visor (and USB cradle), Linux, etc. You also should know how to compile stuff, like the kernel.

Most critical is determining which type of USB host controller you have. Intel chipset motherboards, i.e. the BX chipset, are typically UHCI controllers. Most addon cards are OHCI controllers. You could try

```
cat /proc/pci
```

or use

```
lspci −vv
```

and look for your controller in there. It should say which type you have. You should refer to http://www.linux−usb.org for further details. The kernel documentation in the directory (Documentation/usb/usb.txt) may also be helpful.

Now lets see... working USB support was first introduced in the Linux Development Kernels, around v2.3.40. The latest version, of this writing was v2.3.99pre3. I tried briefly, but that broke a number of things on my system, and I was too lazy to fix them. SO. What I did instead was to get 'The v2.2 Linux USB drivers backport'. This is available from http://www.linux−usb.org

http://www.suse.cz/development/usb−backport/usb−2.3.50−1−for−2.2.14.diff.gz

So, the bear minimum kernel version is 2.2.14, which is what I'm using. And it works. Check for the latest backport patch if still available or relevant when you are reading this. If kernel version 2.4 is out, use that kernel, otherwise use the latest backport patch.

2.2 Getting your kernel to support USB

To install the USB backport, you can do something like this: (cut and paste from:http://www.suse.cz/development/usb−backport/)

```
cd /usr/src
tar xzvf linux-2.2.14.tar.gz
cd linux
gzip -dc ../usb-2.3.50-1-for-2.2.14.diff.gz | patch -p1
make distclean
```

Configure, and then make your kernel. Make sure to enable:

- USB support (CONFIG_USB),
- the appropriate controller − UHCI, alternate UNCI, or OHCI (CONFIG_USB_UHCI, CONFIG_USB_UHCI_ALT, or CONFIG_USB_OHCI)
- USB Serial Converter support (CONFIG_USB_SERIAL)
- USB Handspring Visor Driver (CONFIG_USB_SERIAL_VISOR) (serial converter's sub−option)

I compiled these into the kernel, but as modules should work. The developer(s) prefer the module method, but that is up to you.

After you did that, finish installing your kernel, and reboot.

Not done yet! To make the Visor drivers work, we need to finish making the USB Serial Converter working. For that, you can refer to the Linux Kernel Documentation: Documentation/usb/usb−serial.txt. Basically, what it tells you to do is to make the devices. You can do this by excuting the commands:

```
mknod /dev/ttyUSB0 c 188 0
mknod /dev/ttyUSB1 c 188 1
etc...
```

Right now its limited to 16 connections/ports/devices, so just from 0 to 15 need to be done.

For some reason, the visor gets connected to ttyUSB1, instead of ttyUSB0, even when its the only USB device. A "neat" thing to do might be to create link to the device, so you can refer to /dev/visor (or palm, or pilot, depending on what floats your boat):

```
cd /dev
ln −s /dev/ttyUSB1 visor
```

2.3 Installing pilot link

pilot−link is the software package which contains many programs to talk with you're Visor (or your Palm, if you have one of those). They run the same OS, and that's what is important.

So you can get the version I got (0.9.3) from here:

ftp://ryeham.ee.ryerson.ca/pub/PalmOS/pilot−link.0.9.3.tar.gz

At the time of writing, http://pilot−link.sourceforge.net was the new home for this package. Download it, and compile. Version 0.9.0 did NOT work for me, but this one (0.9.3) worked fine. There are some packages (rpm,deb) available, but of older versions. (see the PalmOS HOWTO). You may also try newwer versions, when they appear.

```
tar xvzf pilot−link.0.9.3.tar.gz
cd pilot−link.0.9.3
./configure −−prefix=/usr/visor
make
make install
```

Also, add /usr/visor/bin to your path. Of course, you can also select whichever path you prefer. Redhat and other distributions should have this package available as well, and they will install in their respective system paths, i.e. /usr/bin.

2.4 Almost Done! (aka testing)

We are pretty much done. Now we need to test it. Of course, the visor needs to be in the cradle, and the cradle needs to be plugged into your USB port.

To test, we can use the pilot−xfer, in list mode. For the computer to realize that the visor really is there, you need to hit the hotsync button on the cradle. (When a USB device connects, I've noticed that a bunch of info is dumped to the screen). Then you can run pilot−xfer:

```
pilot−xfer /dev/visor −l
```

(Remember, /dev/visor is a symbolic link to /dev/ttyUSB1). You should get a list of programs/things on your visor. From there, use other documentation.

3.References

For this document, and, well, learning what i did, i used:

- http://www.orbits.com/Palm/ (for pilot−link info, PalmOS HOWTO)
- ftp://ryeham.ee.ryerson.ca/pub/PalmOS (for pilot−link program)
- http://www.linux−usb.org (for backport/kernel stuff)
- http://usbvisor.sourceforge.net