

GNU/Linux support for the Toshiba Satellite Pro 6000 series

Wouter Verhelst

First version: 17 February 2002.

This version: 27 June 2002

Updated link to 'most recent' version at 30 May 2003

Contents

1	Introduction	3
2	Framebuffer	3
3	The X Window System	4
4	Sound	6
5	Network	6
6	Power Management	7
7	USB	8
8	PCMCIA	9
9	Wireless LAN	9
10	Floppy disk drive	9
11	IDE Chipset	10
12	SlimSelect bay	10
13	IrDA	11

14 Other hardware	12
14.1 Modem	12
14.2 SD Card	12
14.3 FireWire?	12
15 lazy files	13
15.1 /proc/pci	13
15.2 /proc/cpuinfo	14
15.3 /proc/interrupts	15
15.4 outpu of 'toshset -q'	15
16 Other sites of interest	16
17 Legal stuff	16

1 Introduction

Employers rule. Especially if they give you a laptop and allow you to install GNU/Linux on it. Since not everything instantly worked the way I hoped under GNU/Linux, I wrote this document so that other people interested enough in running GNU/Linux on a Satellite Pro 6000 can profit from my findings.

Luckily, I found that most hardware is supported, and even some of the special features of the laptop are, too. Unfortunately, That doesn't mean it all works perfectly well, and even some (very nice) features are not (yet) supported.

If you get something to work in a different and/or better way than me, please be so kind to let me know so that I can include it here (and use it too, of course ;-). Thanks in advance.

Note: Since this laptop is not my personal property, and seen the fact that my employer and I both decided that it would be better that I quit, I no longer have one of these. Therefore, I can no longer check whether something someone sends me is correct before I put it here. I'm not going to remove these pages; however, note that Hereward Cooper took over maintaining this document; you can find his version at <http://zadok.org.uk/laptop/>.

Which seems to have disappeared (my proxy can't even find the domain anymore). However, while googling for myself (I do that every once in a while), I found a revised version of this document at <http://www.spierrel.freessurf.fr/>.

Oh, BTW: if you send me some mail that requires my reply, then **PLEASE** make sure my reply is not bounced by a stupid, dumb, non-functional spam-filter that requires me to waste *my* time doing stuff *I* don't need, just so *your* time isn't wasted by *other* morons. That would make both our lives easier.

Thanks. You know who you are.

And now... let's get into it.

2 Framebuffer

```
CONFIG_EXPERIMENTAL
CONFIG_FB
CONFIG_FB_VESA
```

There is an accelerated framebuffer available for the Trident CyberBlade family of graphical cards. Well, almost. With a literal “#error Floating

point maths. This needs fixing before the driver is safe” in the source, one can’t assume this’ll work. I did get some graphics using vesafb and kernel 2.4.16¹, though.

I’m using

```
vga=792
```

to specify a 1024x768x24 framebuffer. If you prefer 16 bit colors (which is a tiny bit faster), use

```
vga=791
```

but beware that it is impossible to switch modes once you chose one; so if you set up your framebuffer to use 1024x768x16, then that is what you’ll have to set your X server to use later on. See Documentation/fb/vesafb.txt in the kernel source for details on what those numbers exactly mean and how to specify other modes if you prefer.

3 The X Window System

The graphical card in this laptop is called a “Trident CyberBlade XP Ai1” by the preinstalled Microsoft Windows. However, this probably is a very new revision of that board, because “cat /proc/pci”, “lspci -vv”, “scanpci -v”, and “X -configure” all say the same thing: “unknown VGA-capable card found”(be it in their words instead of mine), although the X server mentions support for a CyberBlade XP and a CyberBlade Ai1 (but admittedly not a CyberBlade XP Ai1).

According to the Toshiba technical specifications², the name of this card is ‘CyberALLADIN-T’.

It might be possible to get the trident module to support this card; however, I did not get that to work. A workaround is to use the FBdev driver, but that one is dog slow; you can forget about full-screen multimedia thingies³.

According to Jeff Chua <mailto:jchua@fedex.com>, a working driver can be found at http://www.xfree86.org/~alanh/trident_drv.o. Which requires X 4.2.0, while the latest Debian/unstable packages are still on 4.1.0. Shit. And they’re so good. UPDATE: I can now confirm this. I installed X

¹vesafb in 2.4.18 (which is the latest version as of this writing) and 2.4.17 doesn’t like this card anymore.

²you can find these through <http://www.toshiba.com>. I don’t have a direct URL as they use Java Servlets whith a URL-encoded SessionID (yuck), so any URL I would give you would probably be invalid

³unless you don’t care about a frame rate of about 1 frame per 1/2 second, that is.

4.2.0 in a chroot environment, which indeed gives me accelerated graphics. I can play games now! Woohoo! ;-)

I also had the problem that my X server sometimes interpreted a keyboard event more than once. This was solved by adding a line like

Option ‘‘XkbDisable’’ ‘‘true’’

to /etc/X11/XF86Config-4. Since that fucks up your keyboard mapping, you’ll have to use xmodmap to fix it again. The easiest way to achieve that follows:

- Dump your keyboard map to .xmodmaprc:

```
xmodmap -pke > .xmodmaprc
```

- Use your \$EDITOR to edit the file so that the keyboard mapping is sane again. You’ll see that no key has a third value. You need those third values for the extended keys; so you’ll have to add them. The symbolic names of keys like @, and — are the same as those for XKB, so reading your XKB keymap may help. If your system is sane⁴, you should be able to find those at /usr/X11R6/lib/X11/xkb/symbols/.
- At the end of the file, add the following two lines:

```
remove mod1 = Mode_switch
add mod3 = Mode_switch
```

- Map Mode_switch to some key. You probably want Alt Gr, which has keycode 113. If you insist, you can map it to Return, the Windows key, heck, or even your spacebar if you really want to. Use “xev” to find out what the keycode of your chosen key is.
- Finally load your new keymap:

```
xmodmap .xmodmaprc
```

The keyboard should work now.

Many thanks go to Olaf Schnapauff<mailto:olaf@schnapauff.com> for showing me how great xmodmap actually is⁵.

⁴where “sane” should be read as “is FHS-compliant”

⁵Can you imagine how stupid I can be? I actually referenced xmodmap here, but didn’t take the time to look at it and fix my keyboard mapping. Oh well.

4 Sound

```
CONFIG_SOUND
CONFIG_SOUND_TRIDENT
```

Now for the good news: sound support is quite good. The sound card is not the best ever (it uses the AC97 chipset, which is of the “cheap, but acceptable quality” type), but at least it works. You’ll want the above options; you don’t need to supply module loading options, and you don’t have to mess with ALSA or the like (which is what I was most afraid of). Just make sure you only load the module when you need it (using the kernel module loader, for example), because some part of the code doesn’t like being idle; besides the problem with APM I mentioned above, I’m also seeing applications using the sound device that block until I kill them, in which case the solution is to reload the module.

If you’re masochistic and actually *want* to mess with ALSA⁶, it’s possible. According to a mail I got from J. Waalboer <mailto:jwaalboer@convolution.nl>, all you need to do is to compile ALSA as follows:

```
./configure --with-oss=yes --with-cards=ali5451
make install
```

Disclaimer: I didn’t test this, both because I don’t like ALSA, and because I received this hint after having a Toshiba Sattelite Pro⁷. If you need help, ask J. Waalboer.

According to `/proc/pci`, the sound card is called “Acer Laboratories Inc. M5451 PCI South Bridge Audio (rev. 1)”.

Unfortunately, this sound card has no internal MIDI sequencer. If you want to play MIDI files, you’ll have to go for “timidity” or something likewise.

5 Network

```
CONFIG_NET
CONFIG_NETDEVICES
CONFIG_NET_ETHERNET
CONFIG_NET_PCI
CONFIG_EEPRO100
```

⁶Just kidding, of course ;-)

⁷You read the introduction, right?

The networkcard is a standard EtherExpress Pro 100, which is pretty popular these days. Therefore, the support is quite good.

I think Linus has one too, BTW: if you don't modify anything about your kernel configuration when building from source, the EtherExpress Pro will be the only NIC supported by your kernel...

No real problems here.

I received a mail from Jeff Chua <mailto:jchua@fedex.com> who told me that he experienced slowdowns on network traffic with the `eeepro100` driver, something he didn't notice using the "e100" one. Maybe I'm lucky, maybe I don't use my network as much as he does, maybe I'm just not used to 100MBit, but I don't have that problem; if you do, you might want to try this suggestion.

If you're compiling your own kernel, you'll want to compile this as a module, so that you can unload the module when you're not using your network card. Your battery will like that, and live a bit longer in return ;-)

6 Power Management

```
CONFIG_PM
CONFIG_APM
CONFIG_APM_DO_ENABLE
CONFIG_APM_DISPLAY_BLANK
CONFIG_APM_CPU_IDLE
CONFIG_TOSHIBA
```

The above options are what I use, and basic APM support works (in conjunction with `apmd`, that is). However, the Toshiba Satellite Pro 6000 has some extra powermanagement features, which are an extension to the basic APM spec. To support these features, you'll want to activate `CONFIG_TOSHIBA` and download the tools at <http://www.buzzard.org.uk/toshiba/>; they are also available in the Debian package "toshtools". They work pretty well, and make some features supported by Windows available under Linux (and even some that are, AFAICS, *not* supported by Windows, like the alarm power on. Correct me if I'm wrong).

Alternatively, there's the `toshset` stuff. I have no more information about this than the `toshset -q` output near the end of this document (thanks go to J. Waalboer <mailto:jwaalboer@convolution.nl> for that one) and the name of the software. Ask Mr. Waalboer for details.

ACPI support doesn't work; don't try. The kernel mumbles something about "buggy BIOS" and disables ACPI. Besides, current Linux ACPI support is incomplete; APM works much better. You're not losing anything.

That doesn't mean it all works flawlessly. Toshiba engineers seem to like putting all possible and impossible hardware so that it uses IRQ 11. My system tends to lock up when suspending, unless I give the extra bootparameter "pci=biosirq" (as the kernel suggests if you boot without it). Has some impact on sound, though; it sometimes gives lags, and sometimes doesn't. If it does, you just need to reload the module.

One note: if you suspend your system, you need to reload the usb-ohci module when resuming, else the bus will no longer report when something connects or disconnects. You can do this automatically by creating two shell-scripts. This one comes from my /etc/apm/suspend.d:

```
#!/bin/sh
/etc/init.d/usbmgr stop
rmmod usb-ohci
```

and this one from /etc/apm/resume.d:

```
#!/bin/sh
modprobe usb-ohci
/etc/init.d/usbmgr start
```

the stopping and starting of usbmgr is because if I don't do that, usb-ohci is still in use, and thus can't be removed. It's not strictly necessary, but then again...

Clever people might notice that I'm not reloading the trident module to fix the sound problem. This is because the sound device may be in use, which means that the module won't be unloaded, which probably means that the script errors out, and thus that the machine doesn't suspend. I don't like that "feature" ;-)

7 USB

```
CONFIG_PCI
CONFIG_USB
CONFIG_USB_OHCI
```

You'll probably also want CONFIG_SCSI and CONFIG_USB_STORAGE for the USB floppy you get from Toshiba. If you load the right modules, it works pretty good. See the section on the floppy drive, below.

If you have any other USB hardware, make sure you have drivers for that hardware, too.

8 PCMCIA

CONFIG_PCMCIA
CONFIG_CARDBUS

Some Toshiba Pro 6000's have a built-in wireless LAN card. Mine does not, but I bought a PCMCIA version. The PCMCIA-chipset in this laptop is a yenta-compatible thing, which is pretty standard. It works flawlessly.

9 Wireless LAN

As said in the section about PCMCIA, above, I don't have an integrated LAN-adapter. However, Ivan Brunello <mailto:ibrunello@dgtworld.it> mailed me, telling that his integrated Wireless LAN-adapter uses the orinoco-chipset, and the PCMCIA-interface. Thus, you need the orinoco_cs driver.

I later got a note from J. Waalboer, saying that the builtin ORiNOCO-stuff doesn't work with the default settings of the pcmcia packages. To make it work, you'll want to:

- add the following line to `/etc/pcmcia/network`, on line 38 (right before `". ./wireless"`):

```
if [ "$ACTION" == "start" ]; then ifconfig $DEVICE up; fi
```

- make sure you enter the ESSID in `/etc/pcmcia/wireless` as the *first*, and *not* the last item.

The rest would look as a "normal" ORiNOCO-configuration. Also, he's told me that for a real good wireless performance, you'll want to use a more recent kernel than 2.4.18 (like 2.4.19-pre10 or such); there would be problems with Tx timeouts on 2.4.18.

Note: I couldn't check any of this; if you have problems, don't ask me.

10 Floppy disk drive

Besides support for USB, you'll also need:

CONFIG_SCSI
CONFIG_BLK_DEV_SD
CONFIG_USB_STORAGE

The floppy registers as a USB “Mass Storage” (yeah, right) device. If you load the right modules, plug it in and put a floppy in it, you can mount it as `/dev/sda`. You can have your system to do this automatically, without running commands as root, by using `usbmgr`. It polls the USB bus for status changes and loads the required modules on demand. In Debian, the package you need has the same name...

I compared its speed against a normal floppy disk by using the following command:

```
time dd if=/dev/sda,fd0 of=/dev/null
```

and got about 41 secs on a normal floppy drive, and about 10 secs more on this one. Pretty good.

11 IDE Chipset

`CONFIG_BLK_DEV_ALI15X3`

I never tried this⁸, although I should've. Your IDE chipset will work without these options, but it'll probably be quite a lot faster if you do use them.

Thanks go to J. Waalboer <mailto:jwaalboer@convolution.nl> for finding this out and mentioning it. You'll most likely also want some `hdparm`-stuff to be done; according to Mr. Waalboer, you'll have optimal harddisk performance with

```
hdparm -X70 -k1 -c1 -d1 /dev/hda
```

however, if you do not trust me or him, and/or if you do not know what you are doing, don't play with `hdparm` too much. Please do remember that misuse of `hdparm` might fry your harddisk.

12 SlimSelect bay

These things can be found in an increasing number of systems today. One can put a floppy drive, a secondary battery, or an IDE device in it. Since there's a USB floppy with this toshiba, one CD or DVD module and an 'empty' module (to reduce weight without the risk of damaging the SlimSelect bay) are sold with it.

⁸you'll know why if you read the introduction...

Since it can be hotswapped, it's nice to know that Linux also supports hotswapping. You'll need some extra tool to register and deregister IDE-devices; you can find it at <http://users.ox.ac.uk/~univ1377/>. Although written for a Dell C600, it works on this system too⁹. I don't think there's extra kernel-support needed, but might be wrong there.

13 IrDA

CONFIG_IRDA
CONFIG_SMC_IRCC_FIR

Previously, IrDA was put with the "Doesn't work" chapter. I never got it working in Linux. However, I J. Waalboer <mailto:jwaalboer@convolution.nl> told me that it's not impossible. Which is a good thing, of course.

The problem is that the SMC chipset is, by default, *disabled* at boot. You'll need to enable it with a tool you can download at <http://www.csai.unipa.it/peri/toshsat1800-irdasetup/>. Then, assuming you compile this tool and install it as /usr/sbin/irdasetup, you add the following to /etc/modutils/irda (on a Debian system; add it to /etc/modules.conf or /etc/conf.modules on any other system):

```
options smc-ircc ircc_dma=1 ircc_irq=10 ircc_sir=0x2e8 ircc_fir=0x130
pre-install smc-ircc /usr/sbin/irdasetup --dma=1 --irq=10 --firbase=0x130 --sirb
```

There's one minor gotcha, though: if you have an ORiNOCO-card, it will by default overlap with the 0x130 I/O range, which obviously won't work. To make sure that doesn't happen, add the following to /etc/pcmcia/config.opts:

```
exclude port 0x2e8-0x2ef
exclude port 0x130-0x13f
```

Now, the only other thing you'll have to do is find some IrDA-hardware ;-)

Disclaimer: I never tested this, since I didn't have my TSP6000 anymore at the time I received this hint¹⁰. Ask J. Waalboer if you need help.

⁹as in "I tried a few times, and didn't fry my system

¹⁰You did read the introduction, right?

14 Other hardware

So far for the good news. There's still some more hardware on this laptop, but this is all I got to work. What follows is some other hardware that may work, but that I either didn't even try to get it to work, or that I did try, but gave up because I lost interest because I probably won't use that particular bit of hardware in the near future anyway.

That doesn't mean I wouldn't like getting it to work. If you know or find out how to get one of those to work, let me know.

14.1 Modem

This is a softmodem. Since I don't do modems anymore (I work for a registrar, and have broadband Internet access at home), you'll have to find someone else to get this working. If you didn't know already: softmodems are a pain in the ass to get them working under anything else than Windows, usually. I couldn't even find out what the chipset is, but if it's not Lucent, it'll probably not work. Check out <http://www.linmodems.org> if you want to give it a shot. Definite answers on whether it works are welcome.

14.2 SD Card

An SD Card (Secure Digital Card) seems to be a CompactFlash-like thing. This laptop has a slot to read it, but only supports one of the types of cards that can be put into this type of slots.

Since I don't have any of those cards, I don't use them. Or the slot. Didn't try to get it to work.

14.3 FireWire?

There's one pretty confusing message box when you use the "extreme long battery life" setting in the "Power Saver" control panel under the preinstalled Windows, and use the laptop on battery power. It's got a title saying "IEEE 1394", while the text of the message box is a question asking something about some devices being shut down, and whether or not you agree on that.

This may lead one to think that there's some FireWire-hardware in this laptop, since IEEE 1394 is the document that describes that standard. There's no FireWire in this laptop, though. I suppose this is some bug in Windows (big surprise, there): the message refers to the network card being shut down.

Oh well, why not.

15 lazy files

Some people like to know what files in /proc looks like. Lemme give you some in case you're one of them.

15.1 /proc/pci

PCI devices found:

Bus 0, device 0, function 0:

Host bridge: PCI device 10b9:1644 (Acer Laboratories Inc. [ALi]) (rev 1).
Prefetchable 32 bit memory at 0xf0000000 [0xf3ffffff].

Bus 0, device 1, function 0:

PCI bridge: Acer Laboratories Inc. [ALi] M5247 (rev 0).
Master Capable. No bursts. Min Gnt=8.

Bus 0, device 2, function 0:

USB Controller: Acer Laboratories Inc. [ALi] M5237 USB (rev 3).
IRQ 11.
Master Capable. Latency=64. Max Lat=80.
Non-prefetchable 32 bit memory at 0xf7eff000 [0xf7efffff].

Bus 0, device 4, function 0:

IDE interface: Acer Laboratories Inc. [ALi] M5229 IDE (rev 195).
Master Capable. Latency=64. Min Gnt=2.Max Lat=4.
I/O at 0xeff0 [0xefff].

Bus 0, device 6, function 0:

Multimedia audio controller: Acer Laboratories Inc. [ALi] M5451 PCI South Br
IRQ 11.
Master Capable. Latency=64. Min Gnt=2.Max Lat=24.
I/O at 0x1000 [0x10ff].
Non-prefetchable 32 bit memory at 0x10000000 [0x10000fff].

Bus 0, device 7, function 0:

ISA bridge: Acer Laboratories Inc. [ALi] M1533 PCI to ISA Bridge [Aladdin IV

Bus 0, device 8, function 0:

Bridge: Acer Laboratories Inc. [ALi] M7101 PMU (rev 0).

Bus 0, device 10, function 0:

Ethernet controller: Intel Corp. 82557 [Ethernet Pro 100] (rev 13).
IRQ 11.
Master Capable. Latency=64. Min Gnt=8.Max Lat=56.
Non-prefetchable 32 bit memory at 0xf7efe000 [0xf7efefff].
I/O at 0xeec0 [0xeeff].
Non-prefetchable 32 bit memory at 0xf7ec0000 [0xf7edffff].

Bus 0, device 17, function 0:

```

CardBus bridge: Toshiba America Info Systems ToPIC95 PCI to Cardbus Bridge w
IRQ 11.
Master Capable. Latency=168. Min Gnt=128.Max Lat=5.
Non-prefetchable 32 bit memory at 0x10001000 [0x10001fff].
Bus 0, device 17, function 1:
CardBus bridge: Toshiba America Info Systems ToPIC95 PCI to Cardbus Bridge w
IRQ 11.
Master Capable. Latency=168. Min Gnt=128.Max Lat=5.
Non-prefetchable 32 bit memory at 0x10002000 [0x10002fff].
Bus 0, device 18, function 0:
System peripheral: PCI device 1179:0805 (Toshiba America Info Systems) (rev
IRQ 11.
Non-prefetchable 32 bit memory at 0x10003000 [0x100031ff].
Bus 1, device 0, function 0:
VGA compatible controller: PCI device 1023:8820 (Trident Microsystems) (rev
IRQ 11.
Master Capable. Latency=8.
Non-prefetchable 32 bit memory at 0xfc000000 [0xfdffffff].
Non-prefetchable 32 bit memory at 0xfbc00000 [0xfbffffff].
Non-prefetchable 32 bit memory at 0xf8000000 [0xf9ffffff].
Non-prefetchable 32 bit memory at 0xf7ff8000 [0xf7ffffff].

```

15.2 /proc/cpuinfo

```

processor : 0
vendor_id : GenuineIntel
cpu family : 6
model : 11
model name : Intel(R) Celeron(TM) CPU          1066MHz
stepping : 1
cpu MHz : 1062.351
cache size : 256 KB
fdiv_bug : no
hlt_bug : no
f00f_bug : no
coma_bug : no
fpu : yes
fpu_exception : yes
cpuid level : 2
wp : yes
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36

```

bogomips : 2116.81

15.3 /proc/interrupts

```

          CPU0
 0:        604694      XT-PIC  timer
 1:        35746      XT-PIC  keyboard
 2:           0      XT-PIC  cascade
 3:       13546      XT-PIC  orinoco_cs
11:          15      XT-PIC  ALi Audio Accelerator, Toshiba America Info Sys
14:       38759      XT-PIC  ide0
15:           1      XT-PIC  ide1
NMI:           0
LOC:       604683
ERR:           0
MIS:           0
```

No, it doesn't come with a builtin orinoco. As I said in the PCMCIA section above, mine is a model without builtin wavelan (although models with wavelan builtin do exist). Since I like wavelan, I bought an ORiNOCO-card myself (and they're pretty damn expensive).

15.4 output of 'toshset -q'

I didn't run `toshset -q` on my system, since I didn't know about it until Mr. J. Waalboer told me about it. His system showed the following output:

```

machine id: 0xfc71      BIOS version: 1.40      SCI version: 2.81
toshset version: 1.53      battery save mode: user settings
power source: external      LCD backlight: on
fan: off      select bay: CDROM
HciFeature::query: HciFunction for feature select bay lock returned
NOT_SUPPORTED
Video out: internal      Hibernation LBA: 0x0 (0)

flat panel: 1024x768, 18 bit TFT      lcd brightness: super-bright
CPU speed: fast      CPU sleep mode: off
Display stretch: off      CPU cache: on
cache policy: write back      panel power: off
hard disk auto-off time: 30 minutes      display auto-off time: 30 minutes
HciFeature::query: HciFunction for feature HCI power-up mode returned
```

NOT_SUPPORTED

power-up mode: boot	battery percent: 547%
cooling method: performance	power-up alarm: disabled
auto-off time: disabled	parallel port mode: ecp
Hibernation: configured	Pointer: 0
boot method: hard disk->floppy->CDROM	user password: registered
supervisor password: registered	owner string: [max length: 513]

16 Other sites of interest

You may be looking for how to run GNU/Linux on an other type of laptop. I'm not that experienced on running Linux on a laptop, but at <http://www.linux-on-laptops.com> you'll find a whole bunch of links to other documents like this one. If you're lucky, yours is in there when you buy it. If you're not (like me), then you'll have to do it yourself.

Also, <http://www.linuxdoc.org> may be of interest; this is the central site for Linux HOWTO's, FAQs and guides. But you probably already knew this one, right? ;-)

If you would like to get these pages in some other format, go to <http://people.debian.org/~wouter/>, where you'll find links to HTML, PDF, DVI and L^AT_EX-files of this thing.

Corrections and suggestions are accepted at <mailto:wouter@debian.org>.

17 Legal stuff

Copyright © 2002, Wouter Verhelst

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover texts and no Back-Cover texts. A copy of the Free Documentation License can be downloaded from the GNU website at <http://www.gnu.org/licenses/fdl.txt>