Debian GNU/Linux on the IBM Thinkpad G40

David Härdeman <`david@2gen.com`>

Table of Contents

Introduction

This is an updated version of this page if you are for some reason looking for the old version, drop me a note.

I had an IBM™ ThinkPad 390E™ previously. After a few years of faithful service, being carried back and forth to different job sites and enduring abuse such as falling off a bike, coffee in the keyboard and other small incidents, it was a miracle it worked at all. However, it's age was beginning to show. The computer felt very, very slow on occasions (300MHz Pentium II™), and the hard drive (4GB) was really cramped with two OS's, applications and personal data. Add to that a keyboard where the **X** key wasn't always working, the battery was dead (meaning the computer always needed a power cable connection), a screen that varied wildly in intensity and a CD-ROM that refused to spit out the CD sometimes. It all added up to the decision that it was about time to get a new laptop.

Said and done, I decided that I wanted a ThinkPad since the previous one had been very faithful to me and they are virtually indestructible. I chose the G40 series because it offered a good price/performance ratio and included most of the hardware which I required. The G40's are equipped with a regular (non-mobile edition) Pentium 4 processor which means higher energy consumption and greater cooling requirements, but a lower price tag. The laptop has an air intake on top and one on the bottom and an exhaust on the side. It also has a track-point (essentially a small one-finger joy-stick), a feature which I have grown fond of while using my previous laptop (enough so to find touchpads quite awkward).

I am running Debian Unstable (and Windows XP) on this laptop right now. Since I had to spend some time getting everything up and running, I decided to document what I did on this page both for the benefit of others and as a support for my own memory. I generally expect the reader to be used to basic concepts such as installing modules, installing software etc. If not, I suggest you read up on these issues first.

The kernel I'm currently running is the latest 2.6.X kernel (2.6.11 at the time of writing) and the `.config` file can be found in the files section of this page. It's a pretty modular config that works well for me. Now, let's get on with the interesting stuff...

Hardware

This is a list of the hardware that can be found in my Thinkpad G40, there are some variations between different models, i.e. some G40's have builtin WiFi cards and the processor speed and memory size is customizable, but the differences shouldn't be huge.

- 2.8 GHz Intel Pentium 4 CPU
- 256MB RAM[1]
- Intel 82852/855GM graphics chipset
- 15" internal LCD screen and an external VGA connector
- Two cardbus slots driven by a TI PCI1410 controller
- Broadcom BCM5901 100Mbps NIC
- Integrated 1.44" FDD
- One 37GB ATA HD controlled by an Intel ICH4 chipset
- Integrated DVD/CD-RW combination
- Intel AC97 Modem
- Intel AC97 Audio
- Four USB 2.0 ports
- One LPT port and one PS/2 port for an external mouse *or* keyboard

This is the output from lspci:

```
00:00.0 Host bridge: Intel Corp. 82852/855GM Host Bridge (rev 01)
00:00.1 System peripheral: Intel Corp.: Unknown device 3584 (rev 01)
00:00.3 System peripheral: Intel Corp.: Unknown device 3585 (rev 01)
00:02.0 VGA compatible controller: Intel Corp. 82852/855GM Integrated Graphics
00:02.1 Display controller: Intel Corp. 82852/855GM Integrated Graphics Device
00:1d.0 USB Controller: Intel Corp. 82801DB USB (Hub #1) (rev 01)
00:1d.1 USB Controller: Intel Corp. 82801DB USB (Hub #2) (rev 01)
00:1d.2 USB Controller: Intel Corp. 82801DB USB (Hub #3) (rev 01)
00:1d.7 USB Controller: Intel Corp. 82801DB USB2 (rev 01)
00:1e.0 PCI bridge: Intel Corp. 82801BAM/CAM PCI Bridge (rev 81)
00:1f.0 ISA bridge: Intel Corp. 82801DBM LPC Interface Controller (rev 01)
```

[1]I've since upgraded to 1GB of memory

```
00:1f.1 IDE interface: Intel Corp. 82801DBM Ultra ATA Storage Controller (rev 0
00:1f.3 SMBus: Intel Corp. 82801DB/DBM SMBus Controller (rev 01)
00:1f.5 Multimedia audio controller: Intel Corp. 82801DB AC'97 Audio Controller
00:1f.6 Modem: Intel Corp. 82801DB AC'97 Modem Controller (rev 01)
02:00.0 Ethernet controller: Broadcom Corporation: Unknown device 170d (rev 01)
02:01.0 CardBus bridge: Texas Instruments PCI1410 PC card Cardbus Controller (r
```

ACPI

ACPI is working out of the box on kernels 2.4.23 and 2.6.0-test10 or later which means quieter operation, working suspend-to-ram (see the suspend section for details), battery status information and much more.

DRI

DRI, or Direct Rendering Infrastructure, is a technique to speed up access to graphics hardware in order to benefit bandwidth-hungry 3D applications under XFree86. In order to get this working on the 855GM chipset a few things are needed. The kernel needs DRM (Direct Rendering Manager) support for the chipset and the X server needs to be set up to take advantage of this.

First of all, the kernel. Both recent 2.4 and 2.6 kernels have DRM support for the 855GM chipset using the `i830` driver (note that AGP support needs to be enabled in the kernel to be able to use this driver). Additionally, the 2.6 kernel has the `i915` driver, which is used by more recent X servers, such as the latest X.Org server (make sure you use the right one as DRI might silently fail otherwise). In my case, I still use XFree86 4.3.0, which expects the `i830` driver.

Second of all, XFree86. You need 4.3.0 or later (which of course includes the X.Org server). Then configure the X server to use the `i810` module, add glx and dri modules as well as a DRI section to the config file. A link to my XF86Config-4 file can be found in the files section (you probably want to change the keyboard to be non-swedish), it also shows how to support external mice and the internal track-point simultaneously.

Warning

There were some occasional errors when using the `i830` kernel driver with earlier 2.4 kernels (verified with kernel 2.4.21). When using DRI with this module, a switch from X to console mode would occasionally kill X and display an error message similar to this one: `[drm:i830_wait_ring] *ERROR* space: 65520 wanted 65528, [drm:i830_wait_ring] *ERROR lockup`. The errors have been fixed in later kernels (I do not know exactly which), so make sure to run a recent version.

Once the kernel and XFree86 are set up, you should be good to go. Start the X server, open an xterm and type **glxinfo | grep rendering**. Hopefully you will get a positive message stating that direct rendering is enabled. If you are having troubles getting DRI to work, I can only refer you to the documentation on the DRI web site [http://dri.sourceforge.net/].

These are the (not very impressive) results of running glxgears with DRI enabled:

```
(david@hansolo:~)$ glxinfo | grep rendering
direct rendering: Yes
(david@hansolo:~)$ glxgears /* Windowed mode */
2657 frames in 5.0 seconds = 531.400 FPS
2791 frames in 5.0 seconds = 558.200 FPS
2792 frames in 5.0 seconds = 558.400 FPS
(david@hansolo:~)$ glxgears /* Full screen mode */
378 frames in 5.0 seconds = 75.600 FPS
378 frames in 5.0 seconds = 75.600 FPS
378 frames in 5.0 seconds = 75.600 FPS
```

Caution

I've been told that in order to actually get any hardware acceleration using DRI, you need to be running X with a display depth of 16 bits, not 24 as you'll get with the X config file provided below in the files section.

Under Windows XP it is possible to use the internal LCD in combination with an external CRT in dual-head mode. With dual-head I mean "desktop-of-twice-the-size" and not having duplicate views of the same thing. This feature is unfortunately not supported in XFree86 though, the log files mention "pipe A" and "pipe B" but they're not actually used for anything. Using the keyboard, one can switch between internal, external or "copied-to-both", but not dual-head. It's been indicated to me on the XFree86 mailing list that it's not supported right now and it's not known if it ever will. The discussion on the X.Org mailing list has been a bit more positive so far, but it still remains to be seen whether any progress will be made (details can be found in the X.Org bug tracking system [https://bugs.freedesktop.org/show_bug.cgi?id=1064]).

In 2.4 kernels there is a specific frame-buffer driver for the 855GM chipset of the G40 laptop. This frame-buffer driver is not present in the 2.6 kernel and after asking David Dawes (the author of the driver) I got the reply that he does not have the access to the hardware anymore and that he has heard no interest from someone else to port the 2.4 driver to 2.6. Thus it seems like 855GM owners will be stuck with the VESA frame-buffer driver for now.

It should also be noted that Intel has made a driver of their own which might improve performance and enable more features of the card. I haven't tried it and their website creates horrible URLs so if you're interested, you'll have to find it yourself by searching their website [http://downloadfinder.intel.com/] (thanks to Daemon Bernstein for the tip).

USB

This is too easy. **modprobe uhci-hcd; modprobe ehci-hcd**. Works just fine. I've tried around ten different USB devices and all work flawlessly. Note that the `uhci-hcd` module is named `hcd-uhci` in kernel 2.4.

Storage

The computer came pre-configured with a large FAT32 partition. This is where Windows XP is installed and the first time Windows XP is booted it will spend some time converting this to NTFS. The second partition is used for "system restore", meaning that the first partition is wiped and Windows XP reinstalled. A cute feature that IBM has devised in order to avoid having to ship CD-ROM's with the OS.

I promptly wiped both partitions and created new partitions for Windows XP and Linux. I could have resized the partitions using Partition Magic, but the install of Windows XP that the computer came with was so littered with utility programs and other stuff that I would never use that I opted for a clean reinstall instead. I've also heard from others that you can receive "Rescue CD's" by calling IBM customer support and telling them that the system restore partition is damaged in some way.

Warning

If you do wipe the "system restore" partition (or break/replace the HD), you *cannot* use the Windows XP serial printed on the bottom of the machine to reinstall Windows XP from normal installation CD's. Only the Windows XP install CD's provided by IBM works with the serial number. If your machine is still under warranty, you can get the CD's for free, but if it is no longer under warranty, you will have the pleasure of paying cirka 55 EUR in order to get the install media for an OS you've already paid for. You've been warned.

The CD-RW/DVD combination works without a problem. I've tried watching DVD's using mplayer or xine and burning some CD-R:s using xcdroast. No problems at all.

Audio

Very easy, I used the ALSA driver in the 2.6 kernels. The following command **modprobe snd_intel8x0; modprobe snd_mixer_oss; modprobe snd_pcm_oss** should be all you need. You might also want to install the Debian ALSA packages such as alsa-base which does some nice things for you like make sure that hotplug doesn't load oss drivers for the card. If you want to use the volume up/down/mute buttons, see the keyboard section.

Suspend

Both suspend-to-ram (S3) and suspend-to-disk (S4) works with a recent kernel (tested with 2.6.10).

For suspend-to-ram, the only problem to overcome was that the monitor remains powered off after the computer resumes from a suspension. This is easilly solved by installing the vbetool and acpid packages, and some scripting. I've personally set things up so that closing the lid of the computer triggers a suspend-to-ram.

The script which accomplishes this is split into two parts, the first is placed at `/etc/acpi/events/lidbtn` with the following contents:

Example 1. Specifying to acpid which script to run when the lid is closed

```
# /etc/acpi/events/lidbtn
# This is called when the user closes the lid and calls
# /etc/acpi/lidbtn.sh for further processing.

# Optionally you can specify the placeholder %e. It will pass
# through the whole kernel event message to the program you've
# specified.

# We need to react on "button lid.*" and "button/lid.*" because
# of kernel changes.

event=button[ /]lid
action=/etc/acpi/lidbtn.sh "%e"
```

This means that when the lid is closed, acpid executes the `/etc/acpi/lidbtn.sh` script which performs the following:

Example 2. /etc/acpi/lidbtn.sh

```
#!/bin/sh
# /etc/acpi/lidbtn.sh
# Initiates a suspend-to-ram when the lid has been closed

# Initial settings
exec > /dev/null 2>&1
umask 0077
PATH="/sbin:/bin:/usr/sbin:/usr/bin"

# Suspend
echo "Going to sleep at `date`"
echo "* chvt 1"
/usr/bin/chvt 1
echo "* save vbestate"
/usr/sbin/vbetool vbestate save > /etc/acpi/vbestate
echo "* writing to /sys"
echo -n "mem" > /sys/power/state
echo ""

# Resume
```

```
echo "Back from sleep at `date`"
echo "* restore vbestate"
/bin/cat /etc/acpi/vbestate | /usr/sbin/vbetool vbestate restore
echo "* restore clock"
/sbin/hwclock --hctosys
echo "* chvt 7"
/usr/bin/chvt 7
echo "* done"
echo ""

# Done
exit 0
```

This script changes to a text console, as required by vbetool, saves the current state of the video card to `/etc/acpi/vbestate` and then continues to suspend. On resume, which happends when the computer is opened again and the "Fn" key is pressed, the script continues executing where it left off, restores the vbe state, ensures that the system clock is correct (it seems to drift when suspended) and then switches back to the console where X is usually running.

Suspend-to-disk needs a similar setup to that of suspend-to-ram, first you need to make sure that the kernel is compiled with swsusp support (swsusp2 also seems to be working fine, but I haven't spent lots of time with it as swsusp does everything I need). Then install acpid and create the file `/etc/acpi/events/sleepbtn` as follows:

Example 3. Specifying to acpid which script to run when the sleep (Fn + F4) button is pressed

```
# /etc/acpi/events/powerbtn
# This is called when the user presses the power button and calls
# /etc/acpi/powerbtn.sh for further processing.

# Optionally you can specify the placeholder %e. It will pass
# through the whole kernel event message to the program you've
# specified.

# We need to react on "button power.*" and "button/power.*" because
# of kernel changes.

event=button[ /]sleep
action=/etc/acpi/sleepbtn.sh "%e"
```

This means that when the sleep button **Fn**+**F4** is pressed, acpid executes the `/etc/acpi/sleepbtn.sh` script which has the following contents:

Example 4. /etc/acpi/sleepbtn.sh

```
#!/bin/sh
# /etc/acpi/sleepbtn.sh
# Initiates a suspend when the sleep button has been pressed

#  Initial settings
exec >> /dev/null 2>&1
umask 0077
PATH="/sbin:/bin:/usr/sbin:/usr/bin"
# These are troublesome modules you wish to remove prior to suspending
MODULES=""

# Suspend
echo "Going to sleep at `date`"
echo -n "* remove modules:"
for i in $MODULES; do
        echo -n " $i"
        rmmod $i
done
echo ""
echo "* sending commands"
echo -n "shutdown" > /sys/power/disk
echo -n "disk" > /sys/power/state

# Resume
echo ""
echo "Back from sleep at `date`"
echo -n "* inserting modules:"
for i in $MODULES; do
        echo -n " $i"
        modprobe $i
done
echo ""
echo "* done"
echo ""

# Done
exit 0
```

This script unloads any troublesome modules (in my case, a binary-only WLAN driver), and then performs a suspend-to-disk. Before it all works, you must also make sure that you have a swap partition (not file), and that you pass the `resume=partition` command to your kernel (eg. `resume=/dev/hda2`) every time you boot (which you probably want to do by editing the config files for LILO or GRUB). Once this is all complete, and you've rebooted with the correct kernel parameters, pressing **Fn**+**F4** should suspend the computer to disk.

I've added links to the four files which are needed under `/etc/acpi` below so that you can download them.

Further, you might want to install the laptop-mode-tools package to have the laptop-mode mode of more recent kernels automatically enabled when you are on battery power (depending on how you configure it in `/etc/laptop-mode/laptop-mode.conf`). laptop-mode can keep your hard-drive spun down for as much as 10 minutes at a time, saving writes in memory, and then writing it out to disc all in one go, which should conserve power (at the risk of losing more data upon a crash). See the documentation of the package, and the comments in the config file for more details.

Jean Sébastien reported that the vbetool trick doesn't work reliably with DRI and that all problems can be avoided by using `acpi_sleep=s3_bios` as a boot parameter instead.

Network

When I first got the computer, the `tg3` driver included in the 2.4/2.6 kernels did not work with the integrated Broadcom 5901 NIC, which meant that I had to use the driver available from Broadcom's website [http://www.broadcom.com/support/downloaddrivers.php]. I was never quite happy with this situation though, it's always a hassle to have yet another external dependency to remember when compiling a new kernel and the Broadcom driver module was almost 75% larger than the in-kernel one which made me a bit uneasy.

Fortunately, as kernel development continued, the `tg3` driver improved, and with recent kernels (verified with 2.6.10), the integrated NIC works beautifully.

Some versions of the G40/G41 come pre-equipped with an integrated WLAN card. It is based on an Atheros chipset so you will have to use the proprietary madwifi driver [http://madwifi.org/] if you have one of these models (ndiswrapper probably works as well). Thanks to "rv" for the tip.

Keyboard

Some functions can be accessed using the **Fn** key. Screen brightness can be controlled using **Fn**+**Home** and **Fn**+**End**. Internal/External/Both displays can be cycled trough using **Fn**+**F7**. Those combinations work without any extra effort (though the tpb package below adds on-screen-display support for the brightness level). **Fn**+**F4**, lid close and the power button are reported as acpi events (which can be acted upon using acpid as detailed in the suspend section).

Additionally there are two keys located by the keypad, tentatively called forward and backward. These keys generate regular keyboard scancodes, so they can be assigned to actions using the regular keyboard shortcuts application under GNOME (and I assume there is something similar for KDE).

Finally, the keyboard has four non-standard buttons: "Access IBM" (which is the same as the "Thinkpad" button on some other models), volume up, volume down and mute. The buttons can be handled by including nvram support in the kernel and installing the tpb (ThinkPad Buttons) package (can also be found here [http://savannah.nongnu.org/projects/tpb/] if your distribution doesn't carry it). Edit `/etc/tpdrc` to your liking and everything should be good to go. Audio and brightness should now be tunable using special keys and the changes should also be displayed on screen in X using the libosd package.

Modem

While I've never used the internal modem myself, and probably never will, I have received confirmation from "rv" <herve add-at-sign-here hawaii dot edu> that the modem does work with version 2.9.2 (and presumably later) of the smartlink [http://linmodems.technion.ac.il/packages/smartlink/] driver.

The following sites might also be interesting: First of all, the Linmodems homepage [http://linmodems.technion.ac.il/] which, among other things, hosts the Smartlink [http://linmodems.technion.ac.il/packages/smartlink/] driver. Secondly, the Unofficial PCTel Winmodem [http://pctelcompdb.sourceforge.net/viewdetails.php?id_no=144] website seems to give the modem two thumbs up. There are also some other end user reports here [http://seehuhn.de/comp/toshiba.html#modem] and here [http://www.vgcomputing.com.au/lrtecras1.html#Modem] that seems to report success on similar modems after some efforts.

PCMCIA

PCMCIA works fine most of the time, I'm using it daily with a WLAN card and I've also tried it using regular network card. The kernel module `yenta_socket` and the pcmcia-cs package (or in experimental kernels, the hotplug package) should do the trick.

There is one gotcha though: PCMCIA fails to function if you have 1GB of RAM and use an older Linux kernel. The problem has been fixed in version 2.6.11 (see this thread [http://marc.theaimsgroup.com/?t=110885956400001&r=1&w=2] for details), so an update is strongly recommended (I haven't checked the 2.4 series).

Should you have an older 2.6 kernel, and no wish to update, or a 2.4 kernel you can try either the following patch [pcmcia-1gb-fix1.patch] which was found by Vincenzo Belloli `<vicky@belloli.it>` in this thread [http://www.ussg.iu.edu/hypermail/linux/kernel/0306.3/0956.html], or this patch [pcmcia-1gb-fix2.patch] as suggested by Peng Li `<lipeng@cis.upenn.edu>`.

Miscellaneous

Another interesting feature is that the Intel chipset includes a hardware random number generator. To use this, load the `hw_random` (2.6 kernels) or `i810_rng` (2.4 kernels) module and (if you use udev or devfs) a character device should appear at either `/dev/misc/hw_random` or `/`

`dev/hwrng`. Then install rng-tools, uncomment the Intel settings in `/etc/default/rng-tools` and restart the rng-tools daemon by running **/etc/init.d/rng-tools restart**. The rng-tools daemon should now be running, reading the random data from the proper dev file, verifying that it is indeed random and feeding it to the random pool.

I've experimented a bit with cpu frequency scaling. It seems as if the only driver which works is `p4_clockmod`. I've tested it together with the `cpufreq_ondemand` governor. Unfortunately, it seems to make very little difference in energy consumption (and consequently, heat output and battery life). If you want to run it anyway, make sure that the two above modules are loaded (usually done by adding them to `/etc/modules`), install this script [cpufreq] to `/etc/init.d/cpufreq`, make it executable and run **update-rc.d cpufreq defaults 20**.

Files

- ACPI patch [../../files/thinkpad/g40-acpi.patch]
- XFree86 configuration file [../../files/thinkpad/XF86Config-4]
- Kernel .config file (used with 2.6.11) [../../files/thinkpad/kernel.config]
- Fix for PCMCIA on systems with 1GB RAM or more [../../files/thinkpad/pcmcia-1gb-fix1.patch]
- Alternative fix for PCMCIA on systems with 1GB RAM or more [../../files/thinkpad/pcmcia-1gb-fix2.patch]
- ACPI sleep button config file [../../files/thinkpad/sleepbtn]
- ACPI suspend-to-disk script [../../files/thinkpad/sleepbtn.sh]
- ACPI lid close config file [../../files/thinkpad/lidbtn]
- ACPI suspend-to-ram script [../../files/thinkpad/lidbtn.sh]
- cpufreq setup script [../../files/thinkpad/cpufreq]

Links

- Broadcom drivers [http://www.broadcom.com/support/downloaddrivers.php]
- Bugzilla entry for ACPI problems with the G40 [http://bugzilla.kernel.org/show_bug.cgi?id=1265]
- X.Org Bugzilla entry tracking the dualhead support [https://bugs.freedesktop.org/show_bug.cgi?id=1064]
- Linmodems homepage [http://linmodems.technion.ac.il/]
- A Toshiba users workaround for the smartlink driver [http://seehuhn.de/comp/toshiba.html#modem]
- Another page detailing the same workaround [http://www.vgcomputing.com.au/lrtecras1.html#Modem]
- The Smartlink driver [http://linmodems.technion.ac.il/packages/smartlink/]
- AC97 modem entry in PCTel Winmodem database [http://pctelcompdb.sourceforge.net/viewdetails.php?id_no=144]
- The DRI website [http://dri.sourceforge.net/]

- The TPB website [http://savannah.nongnu.org/projects/tpb/]
- A report to the LKML of problems with PCMCIA and 1GB of RAM [http://www.ussg.iu.edu/hypermail/linux/kernel/0306.3/0956.html]
- The discussion on the LKML which led to a proper fix which was added to the 2.6.11 kernel [http://marc.theaimsgroup.com/?t=110885956400001&r=1&w=2]