

A Brief Review of the Acer Aspire One A110

This document describes the Acer Aspire One (AAO) A110. This is the least expensive of the various AAO models, with 512MB RAM, a 3 cell 2200mAH battery, and 8 gig SSD drive. It also has wireless and a webcam fitted. The A110 comes with an Acer-modified version of Linpus Linux Lite installed. Typical battery duration for the standard three-cell battery is 2 to 2.5 hours. There are six and nine cell battery packs also available, with correspondingly greater duration. The unit was bought from Amazon in the UK for £179.00 sterling, around US\$250 at current exchange rates. AAO versions with hard drives, 1GB RAM, and Windows XP installed are also available.



The A110 RAM can be upgraded, but although not difficult, it is not a task for the faint-hearted – the unit has to be dismantled and the motherboard removed in order to do this. A detailed HOW-TO will appear later.

The wireless card is the Atheros AR5007EG which requires the MadWifi drivers. However, if you are using kernel 2.6.26 or later a suitable driver (ath5k) is built in to the kernel, though the LEDs won't work.

Sound card is Intel 820 and works well, providing also $\geq 1.0.17$ is installed. The speakers are on the bottom, so sound quality isn't terribly good. With headphones, it improves quite a bit.



The trackpad has a button on either side, running vertically. This can be a bit annoying, but one quickly gets used to it. I don't like trackpads anyway, and always carry my Trust wireless laser mini-mouse with the AAO.

The keyboard is a bit small, but bigger than Asus EEE's keyboard. The machine itself is also slightly bigger than the Eee. It has an 8.9 inch LCD which is very good and clear. It weighs 995g, about 2.19lbs.

It boots up extremely quickly, going right to Acer's custom desktop in around 15 seconds. Upon first boot, the user gives a password. This password is for a user called, oddly enough, user.

This user is automatically in the wheel group. The `/etc/sudoers` file has the line allowing any member of the wheel group to run any command without a password uncommented. I imagine this was to imitate XP home, which creates a user (although at least you're allowed to choose the user name) who by default, has admin privilege with no password. This is easily fixed by commenting out the NOPASSWD line in `/etc/sudoers`.

Many people also prefer to have a more standard Linux login, asking for username and password. This post explains how to do it. It is relatively trivial, editing `/etc/rc.d/rc.S`, however, running Acer's Linpus Live Update will overwrite the file, so the steps will have to be repeated after an update.

One of the first things most people do is switch to the advanced mode. This post explains how it's done. Once you have switched to advanced mode, right clicking anywhere on the desktop gives a more standard xfce menu.

There is an older version of NetworkManager. It works well enough with WPA2 personal encryption, but doesn't work with WPA2 corporate, common at many companies and universities. A forum member has provided instructions to update it to work with WPA2 corporate. The instructions can be found on the forums in the third post in the thread.

There are various posts on the forums about how to switch to a more standard xfce desktop. This one has a straightforward explanation.

One forum member has a script to make more major changes to xfce and a few other programs. Some people like this, others find that it breaks other things. The reader is advised to skim the thread before doing the upgrade.

Acer ties the xfce desktop to some other startup scripts. If one uses the login change method mentioned above and decides to use fluxbox as desktop, for example, wireless won't work. The system won't even recognize the wireless card. One forum poster has a script to handle turning wireless on and off. I've found it just as easy to add a line to `/etc/rc.local`, calling Acer's custom script. Acer's script is called `add_driver`, so one can add a line to `/etc/rc.local` reading

```
/usr/bin/add_driver.sh
```

After doing this, regardless of desktop choice, wireless will be working.

Onelinux.org has produced an Ubuntu based iso specially designed for the Aspire One. It worked quite well for me if I stuck to the default Gnome desktop. However, I prefer fluxbox, so I installed and used that. To my disappointment, fluxbox ran much more slowly in the onelinux installation than it did in the default Linpus one. In the end, I went back to Linpus, leaving user as is, since the laptop may be used by others who aren't experienced with Linux. I then added another user.

```
sudo adduser -G wheel user2
```

This created a second user, myself. Although the user was in the wheel group, as the default boot for user2 was fluxbox, neither wireless nor sound would work properly.

The next step was to edit `/etc/security/console.perms.d/50-default.perms`. Absurd as this sounds, ever since Fedora made the decision to tie sound in with ConsoleKit, it's the easiest way to get sound working for all users in a Fedora based distribution. I cover this in my page on sound in Fedora.

In a nutshell, at the top of the file I add

```
<sound>=/dev/dsp* /dev/snd/*
```

There are several similar lines that are already there. Towards the bottom of the file, where permissions are defined--again, there are several similar lines there--I add

```
<console> 0666 <sound>    0600 root
```

The second user can then use sound.

For wireless, I found that I didn't need the entire add driver shell. Usually, I use this at home, so I created a default `wpa.conf` file in my home directory. I used `wpa_passphrase`. This procedure is described on my page about `wpa_supplicant` with Fedora. I created a simple script.

```
#!/bin/sh
sudo modprobe ath_pci
sudo wpa_supplicant -B -Dwext -iath0 -c /home/user2/wpa.conf
sudo dhclient ath0
```

The wireless can be started with this script. If one uses it in different places, one can use a more advanced howto which is linked from the page above, that gives information on using multiple networks.

Although I've tried Ubuntu, Onelinux and Fedora, in the end, I found that the Linpus install, despite all its drawbacks, is noticeably faster. Again, my needs for this netbook are rather simple. If I were using it as my main laptop, I'd be more frustrated by the relatively older packages and spend more effort trying to install, for example, firefox 3. However, I prefer opera as my browser, and the 9.60 beta opera rpm for Fedora installs without problems.

One person on the forums highly recommends Mandriva for the system, but I haven't tried it. I did make a quick test with Linux Mint's fluxbox edition, but after 15 minutes, it was still trying to boot, so I gave up.

In retrospect, I should have been slightly less frugal and gotten the XP edition with a 120 gig hard drive and a gig of RAM, wiped XP and installed a distribution of my choice. However, as I said

at the beginning, I decided to spend as little as possible, with the idea of upgrading--especially as the competition seems to make the manufacturers continue to drop prices.

At time of writing, mid September, 2008, considering the price, it's good for what it is. The 8 gig SSD version lacks some configurability, but for a Linux beginner seems quite easy to use. The more experienced Linux user can either install something else or can modify most of the more aggravating aspects of the Linpus system, depending upon their needs. The relatively simple modifications mentioned above were sufficient for me.

The three biggest drawbacks are the difficulty in upgrading RAM, the short battery life and the fact that the Linpus version of NetworkManager can't connect to WPA2 corporate networks. At time of writing, Acer is already beginning to ship machines with 6 cell batteries, but seem to be ignoring the other two issues. Whether these are showstoppers for the user or not, depend upon their own situation. The 512MB of RAM is adequate for the limited demands I place on this machine.

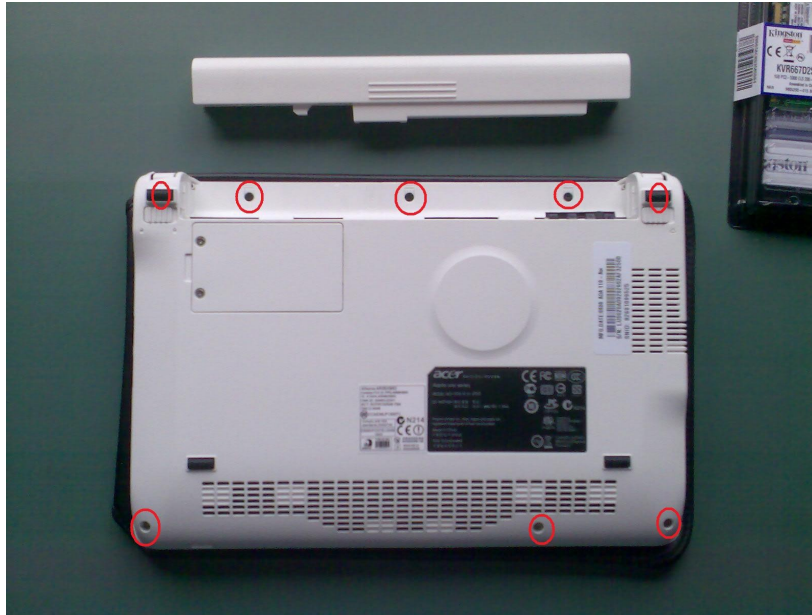
How to dismantle the One and install new RAM module

Acer Aspire One 110-AW and 110-AB comes with a 512MB mainboard-soldered RAM but it has a SODIMM slot which supports up to 1GB module RAM. (1GB, not 2GB). The maximum amount of RAM we can get is 1.5GB (512MB mainboard-soldered + 1GB in slot). Laptops, often have a "gate" to the RAM slot in the bottom of the unit but this is not the case of the Aspire One. In the Aspire One we need to dismantle all the unit (Screen not included) to access the RAM slot.

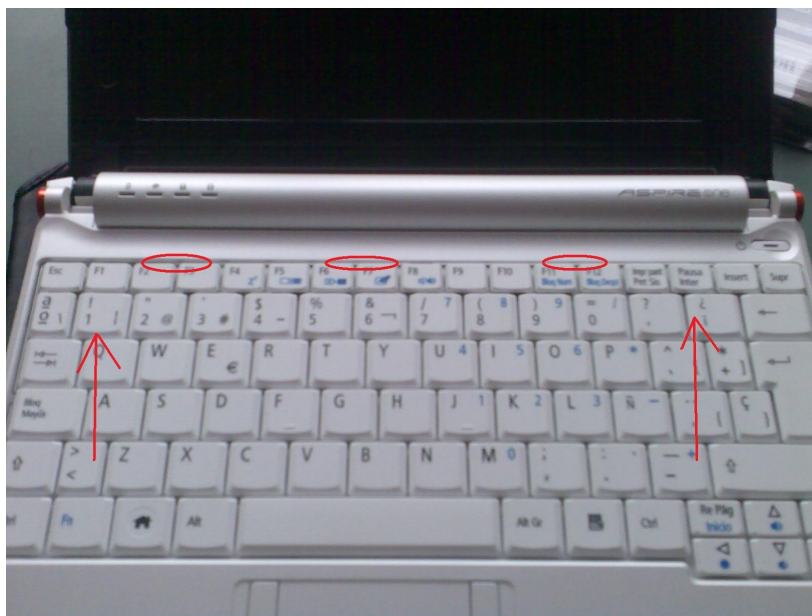


To dismantle and install fresh RAM we will need:

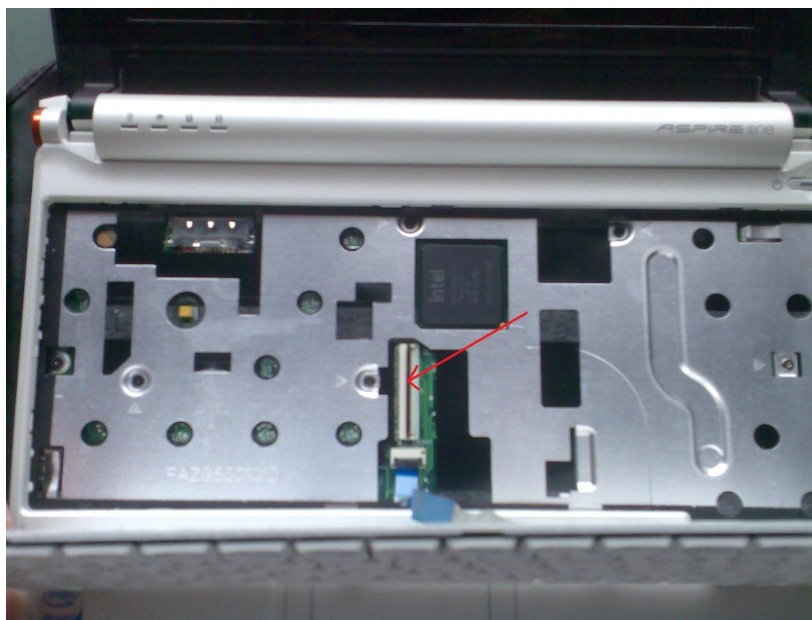
- RAM Module: SODIMM DDR2 up to 1GB, 533Mhz (667Mhz modules also works, but not all, is better to be secure and get 533Mhz), 1.5v. Check this thread.
- Screwdriver
- An Acer Aspire One
- A lot of patience



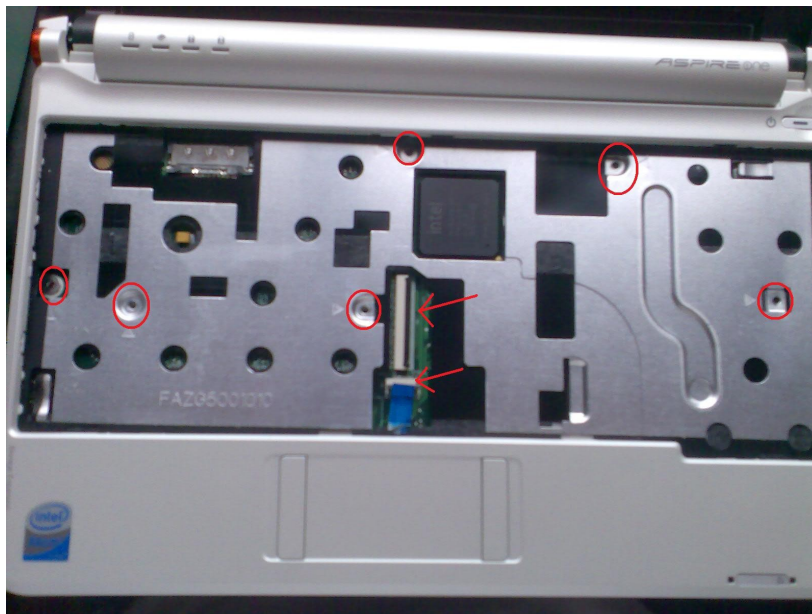
Put the Acer Aspire One over a table with some protective surface (the sleeve that comes in the netbook box can be used) with the bottom part exposed. Remove the battery as well to expose more screws. Remove all the circle-red marked screws including the ones under the rubbers.



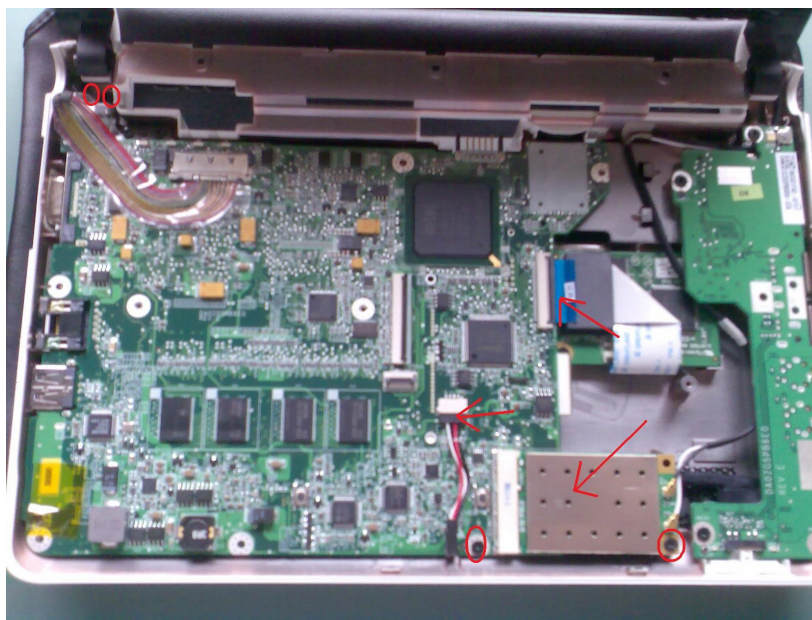
Now we must extract the keyboard because under it there are more screws to extract. We need to put attention in the front part of the keyboard (near the screen), we can see 3 small black tabs we need to press while lifting slowly the keyboard (the bars are already lift in the image). Once the small bars were gone we need to push back the keyboard (to the screen) to remove it.



Now we must disconnect the keyboard cable pushing up the brown tab.

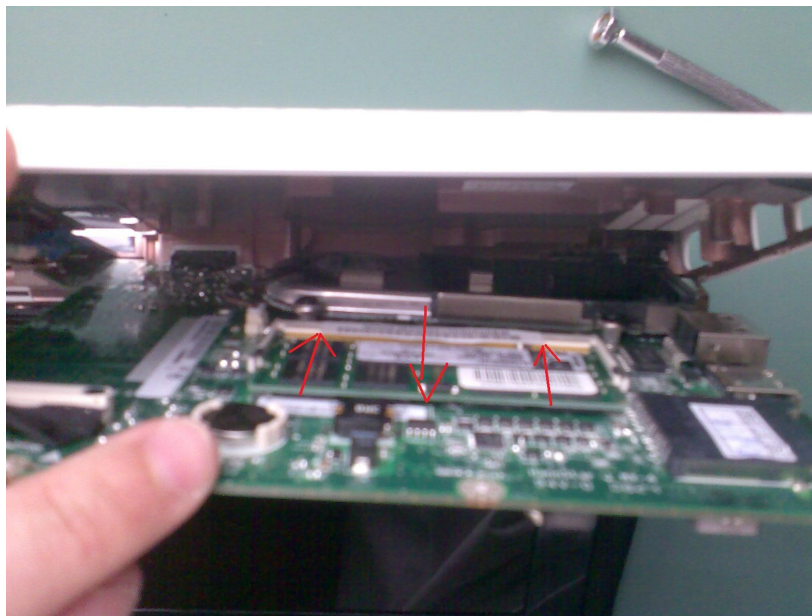


Now that we have a keyboard-free netbook, we need to remove the screws under it. Remove the touchpad cable as well like the keyboard one.



Now we must be careful and lift the plastic case starting in one of the sides. Lift the plastic with you hands: there are tabs in both sides and in the front. Once removed you will be very close to the RAM slot. Remove

the marked screws in the exposed mainboard, remove SSD disc cable, wifi module cable and the small tricoloured cable.



Now we need to lift the mainboard in the part the wifi module was until we can see the RAM slot in the backside as show in the image. Be careful because they are components still connected in the backside. We insert our new RAM module as show in the image.

Now we will test if the RAM is correctly installed. Put back the mainboard and without connect anything except keyboard and the small tricoloured cable , connect battery or power supply and power on the netbook. When the "ACER" screen is shown , press F2 to enter CMOS BIOS Configuration menu. Once there. Press right arrow on the keyboard to get the next tab when Video memory and RAM memory are shown. If the RAM is correctly installed you will see a number bigger than 512MB (The soldered memory) indicating the total amount of the RAM including the new module you just added. If you get 512MB as memory amount, check if the module was well inserted and repeat the process. If you checked all and the memory is still 512Mb probably your memory module is not compatible. Try another and be sure to buy

SODIMM DDR2 not bigger than 1GB and 533Mhz (some 667Mhz modules will work but not all) 1.5v. Check this thread.

If all is OK mount again the netbook and you are ready to go. Boosted Acer Aspire ideal for installing Windows.

The Atheros 5007EG card and MadWifi

This section will soon become obsolete as more and more distributions provide built in support. The card worked, at least in 32 bit, with Mandriva Spring 2008. Some folks are having success with Ubuntu as well, though I wasn't one of them.

Support for this card seems to change rapidly. As of the end of October, Fedora's Rawhide (very soon to be F10) will support the card out of the box with the built in ath5k module. Mepis8 (Beta) does the same. With the exception of Ubuntu based distributions, it seems that almost any distribution that has already moved to the 2.6.27 kernel will be able to support the card more or less out of the box with the ath5k module.

However, Ubuntu's Intrepid does not support it by default. From the Intrepid release notes

The version of the ath5k driver for Atheros wireless devices included in Linux 2.6.27 interferes with the use of the madwifi driver for some wireless devices and as a result has been disabled by default. Many Atheros chipsets will work correctly with the madwifi driver, but some newer chipsets may not, and the madwifi driver may not work with WPA authentication. If you have an Atheros device that does not work with madwifi, you will want to install the linux-backports-modules-intrepid-generic package, which includes an updated version of the ath5k driver. While not installed by default, this linux-backports-modules-intrepid-generic package is included on the Ubuntu 8.10 CD and DVD images for ease of installation.

Different people have different results, no doubt, in part, because of the differing chipsets. Some, although they have some success with the ath5k module find that the card is more reliable with the MadWifi driver. Others have the opposite experience. The reader is advised to first try the

ath5k module (if your kernel supports it) and, if it doesn't seem to work very well, then try the MadWifi driver.

Note that many, even after installing the backports-modules package mentioned above, still find it necessary blacklist the ath_pci module before the card will work with the ath5k module. To do this in Ubuntu open /etc/modprobe.d/blacklist with your favorite text editor and add the line

```
blacklist ath_pci
```

Ubuntu's Hardy also seems to now support the card without using the MadWifi drivers. As of 26 October, 2008 doing

```
aptitude install linux-backports-modules-hardy
```

(or possibly hardy-generic rather than hardy), should enable the card to work with the ath5k module.

Although a reboot might be necessary, the card should then be seen as wlan0. (If you have the ath_pci module installed, remove and blacklist it. Add the line "blacklist ath_pci" without quotes to the end of /etc/modprobe.d/blacklist). In theory, one should be able to do

```
ifconfig ath0 down
modprobe -r ath_pci
modprobe ath5k
ifconfig wlan0 up
```

and have the card working, but in practice, it is usually simplest to add the blacklist line for the ath_pci and reboot.

Fedora 9, if one adds the livna-testing repo, may also work with this card. If one does yum install madwifi and gets the rpm with 3698 in its name, this card should just work. (You might have to modprobe ath_pci, but I don't believe that will be necessary.)

If you have the livna repo installed, then to enable testing

```
yum --enablerepo=livna-testing install madwifi
```

Livna is a private effort, and is merging with rpmfusion. Therefore, the livna driver may not always work. When it's installed, it will also install some kmod rpms. If those rpms don't match your kernel, then they probably won't work. One can check the running kernel by opening a terminal and typing

```
uname -r
```

Keep in mind that if you are running an update, your current kernel may also be updated--the idea is that the Livna (or rpmfusion) kmod numbers match your kernel. So, if the yum update is also updating the kernel, make sure that the livna kmod rpms match what the new kernel will be.

Many distributions are not yet supporting this card. At this point in time, I would recommend trying your distribution's version of MadWifi, for example, the Livna repo with Fedora. If that doesn't work, uninstall your distribution's package and use the instructions below. If you are using a 2.6.27.x kernel, you should also try the builtin ath5k module first.

The latest versions (including Livna's, at least) now work with 64 bit systems.

Many thanks to the MadWifi team.

A few warnings and disclaimers first.

For those revisiting this page, you'll note that the download snapshot has changed. This latest version should work with 2.6.26 kernels (the old one didn't) and 64 bit systems. The latest, and probably last, snapshot is the one from 3 September, 2008. (But see below)

One more warning--in Ubuntu, I've found that I often have to repeat the procedure, described below, of running `make uninstall`, `make clean`, and `make install` each time I reboot. I haven't yet figured out the reason for this--at first, I thought I'd upgraded my kernel without realizing it but that doesn't seem to be the case. Whether it's a bug in Ubuntu, or, far more likely, something that I'm missing, I don't know. So I will say that this usually works with Ubuntu as well as other distros, but with Ubuntu, you might have to do the whole uninstall clean and reinstall with each reboot. Not everyone has experienced that, but enough people beside myself have, so it's worth a mention here. In other cases, simply making sure that it did a `modprobe ath_pci` was sufficient, by, for example, adding the line, oddly enough, of

```
modprobe ath_pci
```

```
to /etc/rc.local.
```

I haven't had the problem often enough to be sure of the solution.

Lastly, if all else fails, Mandriva's Spring 2008 and later supports the card out of the box. I imagine this has to do with the fact that they have an official, working ASUS EEE PC version, since that uses this card for the 700 series. (The 900 may also use it, but the 901 uses a different card.) It's supported in their normal linux-one CD as well. Although they show it as an AR5006EG, it works perfectly.

If you are using a distribution with a 2.6.26 or older kernel, the steps below should enable you to use the card with the MadWifi drivers. Experienced users probably only need the steps written in boldface type. The explanations after each step are for the less experienced.

Most of these commands require root privilege. Fedora users will probably use `su`. Ubuntu users, you should probably preface all commands with `sudo`. That is if I write `make install`, please use `sudo make install`.

1. Determine if it is the AR5007EG card

Running `lspci` will, on a 32 bit system, show the card as AR5006EG. If you haven't uninstalled Windows, you can check with its Device Manager, which will show the card to be the AR5007EG. On my Acer 4720z, there is a sticker on the bottom, apparently present in most Acers, saying that it has AR5BXB63. With 2.6.24.x and newer kernels, `lspci` may identify it as AR242x 802.11 abg. Running the command `lspci -nn` should show, however a vendor ID of 168c:001c. On Fedora, use `/sbin/lspci -nn`.

2. Remove any `ndiswrapper` or other MadWifi cruft.

If you've previously installed `ndiswrapper` or MadWifi, there's a possibility that whatever is left might interfere with what we're going to do. With `ndiswrapper`, first uninstall whatever driver you installed. (It will probably be either the `net5211` or `net5416`.) As root or with root privilege (in other words, if using Ubuntu, use `sudo` for all these commands)

```
ndiswrapper -l
```

That will tell you what's installed. If it was the `net5211` then

```
ndiswrapper -e net5211
```

If it's the `5416` then change that to `net5416`.

Then remove `ndiswrapper`. If you installed it from your distribution's packages, then use `yum remove` or `apt-get remove` or whatever your distro uses. In Ubuntu, I believe it will also put a file in `/etc/modprobe.d`. If you see such a file (it will be probably be called `ndiswrapper`), remove it with the `rm` command.

If you installed `ndiswrapper` from source, then `cd` into the `ndiswrapper` directory that was created when you untarred the `tar.gz` file and type

```
make uninstall
```

It will remove the various files it installed and suggest that you run `make uninstall` a couple of times till you're sure there's nothing left.

If you've installed a version of MadWifi the procedure is similar. If you used your distribution's package use the distribution's package remove tool. If you installed from source, `cd` into the directory created when you untarred the source and run `make uninstall`.

3. Download the patched snapshot. The ticket concerning the build that works with the card is [here](#).

The ticket has a link to the snapshot page. Get the latest version. Often, there will be a problem with one distro or another's latest version, or a problem with the newest kernel, and fixes go into

the latest snapshot. For example, an older snapshot required patching before working with the latest alpha version of Ubuntu's Intrepid Ibex. Therefore, it is always better to go to the snapshots page linked above and make sure that you have the latest version available.

Save it somewhere. If you're in Gnome, the default will be your Desktop.

4. Make sure you have the necessary tools to compile source code.

In Fedora, I always install several packages that are necessary to compile source code, build rpms and the like. For a bare minimum, I'd suggest

```
yum -y install gcc gcc-c++ make kernel-devel
```

For Ubuntu, you can probably get by with

```
sudo apt-get install build-essential linux-headers-$(uname -r)
```

This first time, you probably won't need the headers. However it's useful to have them, and this way, when your kernel updates, hopefully that package will update as well. As you'll have to recompile this driver each time you update your kernel, it's good to have the package installed. Ubuntu and Fedora are the only distributions I've used that didn't have these tools installed by default. ArchLinux, Zenwalk and PCLinuxOS installed the necessary files with their default installation.

5. Untar the file, cd into the directory and run make; make install. If using a system based on RedHat, make sure you have root's path.

Make sure you are in the directory where you downloaded the file. For example, if it downloaded to the Desktop directory, when you open your terminal, do a cd Desktop. (Then run the ls command to make sure you are in the same directory as the file.)

Fedora users, and others who use distributions based on RedHat, please note that some of the commands in make install, e.g., depmod are either in /usr/sbin/ or /sbin. It's necessary to be sure that those directories are in your path. In RedHat and its offshoots, only root's path has that. So, if you did an su, rather than an su -, you might wind up with some command not found errors. For a more in depth explanation, see my page about paths in RH based distros. For now, if using Fedora or another RH based distribution, make sure to do su - and not just su. (This shouldn't be an issue for people using Ubuntu, PCLOS, etc.)

To untar, build and install the source file we just downloaded, do,

```
tar zxvf madwifi-<version>tar.gz
cd madwifi-<version>
make
make install
```

When I write <version> it refers to whatever the latest file name will be, for example madwifi-hal-0.10.5.6-r3816-20080724.tar.gz

6. It may be necessary to remove and/or blacklist conflicting Atheros modules.

In Fedora, you should be able to do this without rebooting.

```
rmmod ath5k
```

Add the module to blacklisted modules. With your favorite text editor, open the file /etc/modprobe.d/blacklist and add this line to it.

```
blacklist ath5k
```

In Ubuntu, I used System, Hardware from the menu. It showed some Atheros modules and I disabled them. (That was in Hardy--in Gutsy, it might be System, Restricted Drivers or something like that.) Once in Hardy, I had to do `rmmod ath_hal` as well. That should be added to the blacklist as shown above. In other words, in Hardy, add the line to /etc/modprobe.d/blacklist

```
blacklist ath_hal
```

In Arch, Zenwalk and PCLinuxOS, I didn't have to disable or blacklist anything.

7. Insert the ath_pci module

```
modprobe ath_pci
```

Hopefully, you then see a command prompt, meaning that the module installed without error. If you get an error message, it still might work after a reboot. For the moment, let's assume you didn't get any error messages. Now, doing `lsmod |grep ath` should show ath_pci and a few other things.

8. Bring the card up with ifconfig.

```
ifconfig ath0 up
```

That step isn't always necessary. However, it doesn't hurt anything, and sometimes things won't work until you do it.

In Fedora 8, the system may see it as wlan0 rather than ath0. If, when doing `ifconfig up` you get an error that there's no such device, try doing `ifconfig wlan0 up` instead. Hopefully, you'll get back a command prompt, indicating that the command succeeded. (If there are no errors, you don't get a message, you're just put back at the command prompt.)

9. Run `iwlist ath0 (or wlan0) scan`

```
iwlist ath0 scan
```

If this gives you a list of available wireless networks, you should be in good shape. From there, you can enable wireless the way you usually do. Most Ubuntu and Fedora users make use of Gnome's NetworkManager while many PCLinuxOS folks use the PCLinuxOS or KDE configuration tools. For Archers, there are good articles on the wiki, and for Fedora users who prefer the command line, go back to the beginning of this page which is about using `wpa_supplicant`.

With luck, you'll now be able to use your wireless card. Sometimes, a reboot might be necessary, but most of the time, I haven't found this to be the case.

The LED light doesn't work by default, but thanks to a posting on Fedora forums, this too is easily solved. As root or with root privilege

```
sysctl -w dev.wifi0.ledpin=3
sysctl -w dev.wifi0.softled=1
```

As of 30 August, 2008, these two `sysctl` values don't seem to be in the 2.6.27 kernels. (Judging from Fedora's Rawhide and Ubuntu's Intrepid--note, however, that both of these are in their alpha/beta stages.) On the other hand, as mentioned above, both of these have the card working with the kernel's builtin `ath5k` modules, so maybe it's a fair trade. :)

Some machines also have a slider. Mine doesn't, but the same poster added two more `sysctl` changes to get that working.

```
/usr/bin/setkeycodes e055 159
/usr/bin/setkeycodes e056 158
```

To make either or both of these permanent, add the lines to `/etc/rc.local`. Many laptops have a button or switch to turn the wireless on and off. On my acer laptop, there is such a button. I don't enable the LED light (I find its flashing a bit annoying) so I can't always tell if I've put wireless on or off. If wireless isn't working, and you have such a button, try pushing it a few times. (It didn't always work for me the first time.)

If you still have no luck, then try a reboot. After rebooting, run

```
dmesg |grep HAL
```

If it didn't work there might be some sort of HAL error, such as HAL error 3, Hardware didn't respond as expected, or unable to attach hardware: Hardware revision not supported (HAL status 13). You might get some other sort of error. If running the above command doesn't return anything, that's a good sign. If running Fedora 8 (or any other distribution that seems to use `wlan0` instead of `ath0`) try doing

```
dmesg |grep ath0
```

If you get back something like ath0 renamed to wlan0 you should be in good shape. Try running `iwlist wlan0 scan` again. Hopefully, this time it will be successful.

If it's still not working, you can run through this checklist. It's not a very good one, but it covers a few things that I missed in my early attempts. If you've followed all of the above instructions, you've already covered this.

If this wasn't the first time you installed madwifi, did you run `make uninstall` and then `make clean` before running `make`?

If you had used or tried using `ndiswrapper` first, did you uninstall that driver and `ndiswrapper`?

If you did run it as root, (in Fedora) did you do `su -`, with the space and then the hyphen after the `su`? If not, there might have been an error somewhere, since Fedora only gives a path that includes `/sbin` and `/usr/sbin` to root's environment.

Did you remember to blacklist the kernel's builtin ath5k module?
Did you remember to `modprobe ath_pci`?

If your laptop has a switch or button to enable or disable wireless, did you remember to check that? On some, like my Acer, the only indication is an LED light that doesn't work in Linux, so one has to guess and possibly try a few times.

IMPORTANT NOTE

Each time you upgrade your kernel, you'll have to do the following. (The modules are tied into the kernel version, so when the kernel is updated, the modules no longer work.)

Go back into the Madwifi directory. (Hopefully, you haven't deleted it. If you have, download and untar it again.) Once you're in the directory

```
make uninstall
make clean
make
make install
modprobe ath_pci
```

At that point, you might get an error. If you do, then reboot and try again.

If you can't get it to work, you might wish to try `ndiswrapper`.

I've had success with the `net5416.inf` file, rather than the usual `net5211` one. This might be wrong, but I am getting the impression that the 5211 will not work with WPA2 encryption. There a thread on Ubuntu Forums about the Acer 4520, which also uses this card. That's where I found the suggestion to use the `net5416` driver, which, aside from working with my WPA2 network also turns on the wireless LED light.

In the same thread, another poster tells how he got it working with 64 bit. Unfortunately, there doesn't seem to be a 5416.inf 64 bit driver. I tried his instructions, including trying older versions of ndiswrapper, but it didn't work for me. I suspect it's because of the network being WPA2, but I am not sure. I've seen reports that the 5211 driver does work with WPA2, but in my searches, I've seen people post that it doesn't. I don't know, but it didn't work for me, and I never got around to trying it with plain WPA.

At one point, though I believe it's been fixed, the default Fedora kernel wouldn't work with ndiswrapper. This thread on Fedora Forums discusses ndiswrapper. One of the posters was kind enough to make an rpm of a recompiled kernel that can be used.

Hopefully, ndiswrapper won't be necessary. Judging from various posts on Fedora and other forums, the success rate seems to be improving. Even if you can't get it to work right now, there's reasonable hope that it will soon work, if not out of the box, then with very little effort. The hope is that soon the kernel's ath5k module will support this card. On Ubuntu forums, one person apparently got it working by using the very latest Hardy Atheros drivers. I wasn't able to duplicate his feat, but with any luck, it will, sooner or later, be one of those cards that just works.

Flashing the BIOS on the Acer Aspire One

While there is an excellent, often referenced article on the macles site about flashing the BIOS on the Aspire One, it strikes me as being a little less clear to the Linux newcomer than it might be. This brief article is an effort to fill in the gaps, and is completely based on his page, though any mistakes are, of course, my own.

Why flash the BIOS

There is always an element of risk in flashing the BIOS. However, many Aspire One owners have suffered from what has become known on the Aspire One forums as the black screen of death. This may happen for no reason, or might happen after modifications in hardware or software. The power LED will show green but the screen is completely black.

The first time it happened to me, I returned the machine. The second time, I thought of doing so again, but the other available netbooks are either much lower in specs or higher in price, so I decided to flash the BIOS (which does almost always fix the issue, as it did in my case) and keep it.

These are the steps. It is assumed that you have either prepared a USB stick in advance or that you have another working computer with a Linux installation. The instructions given are for Fedora and Ubuntu, but hopefully, the reader with another working distribution will be able to adapt it to their installation. It is also assumed that you have an extra USB stick.

Prepare the USB stick by creating a DOS partition

NOTE:A Windows user on the forums has suggested that I add a note suggesting that Windows users format the drive on the AA1. This of course, implies that you're going to prepare one before you need it, which is an excellent idea. He also found that it didn't work for him until he

formatted the drive with FAT32 rather than FAT. (You'll see that the link I give for preparing the drive suggests using FAT16 rather than FAT32.)

Make sure that the USB drive is not mounted. In many cases, especially if using Gnome, it will be mounted automatically at /media/Disk. You can check this by just typing "mount" (without the quotes) at a command prompt. If you see something like /dev/sdb or /dev/sdb1 on /media/Disk, unmount it with the umount command. For example

```
umount /media/Disk
```

(Note that the command is umount, not unmount. There is no n after the u.)

To format the drive, follow the instructions from the Fedora project's article on creating a live USB.

That link gives the commands as well as the expected output. Sometimes, people overlook the final step of initializing the partition. Make sure you follow all instructions, ending with `su -c 'mkdosfs -n usbdisk /dev/sdb1'`

(Assuming of course, that your USB stick was seen as /dev/sdb).

Download unetbootin

As the macles article says, the easiest way to get a DOS system on the USB stick is to use the excellent unetbootin program. If using Ubuntu, you can simply download the .deb package. If using Fedora, click on the link to download for Linux. It is an executable program. Save it somewhere on your hard drive where you can find it. Although macles says you can ignore the error messages about missing p7zip and vol_id, as I use the program frequently for other purposes (it's extremely useful, in many cases, for creating a bootable USB from an ISO image) I consider it worth installing the other programs it needs. In Ubuntu, one can install the necessary dependencies with

```
sudo apt-get install mtools p7zip-full
```

In Fedora, as root or with root privilege (in the example, I will also use sudo)

```
sudo yum -y install syslinux mtools p7-zip p7zip-plugins
```

In Fedora (and probably Linpus Lite, the default installation on the Linux version of the Aspire One), one more step is necessary. Unetbootin looks for vol_id and in Fedora, will complain that it doesn't find it. In Ubuntu, /lib/udev/vol_id is symlinked to /sbin/vol_id. Unetbootin is written to run on Ubuntu, and Fedora doesn't have this symbolic link. I link it to /usr/bin, since, prior to Fedora 10 /sbin isn't in a normal user's PATH. (See my article on paths in Fedora for a more complete explanation.).

```
cd /usr/bin
sudo ln -s /lib/udev/vol_id
```

This will prevent unetbootin from complaining about being unable to find vol_id.

Lastly, make unetbootin executable. (If one installs the .deb package, this may be done for you, I haven't tried it.)

```
chmod 755 unetbootin-linux-293
```

(293 is the latest version at time of writing. Your version may be have a different number at the end.)

Install FreeDOS on the USB stick

The USB partition must be mounted for unetbootin to use it. In Ubuntu, running Gnome, it should automatically mount if you unplug it and plug it back into the machine. In Fedora, (where I don't run Gnome) this may work in Gnome, but doesn't in Fluxbox, my window manager. If the drive isn't mounted, then do so.

```
sudo mount /dev/sdb1 /mnt
```

Now run unetbootin. Assuming you've saved it to your Desktop directory, at a command prompt (again, I'm using the 293 version, if your version has a different number, use that)

```
cd Desktop  
./unetbootin-linux-293
```

The macles page has a good screenshot of what you want to do. From the dropdown menu at the top, click the radio button for Distribution and choose FreeDOS. (The second box, reading 1.0 should be automatically selected for you.)

At the bottom, assuming you've formatted the USB drive as per the instructions in the Fedora article, it should show as in the macles screen shot, with Type being as USB Drive and Drive reading /dev/sdb1. Click OK and unetbootin will install FreeDOS to the USB.

Get the latest BIOS and unzip it to the USB drive.

The macles site usually has a link to the latest BIOS. (Aspire One users all owe a tremendous debt to macles for his articles and other work, such as making the BIOS available, posting extremely useful links on the forums, etc.) At time of writing, the link is here.

Download the zip file for the latest version. In this case, we'll assume that you're using Fedora and /dev/sdb1 is mounted on /mnt. We'll use the latest version at time of writing, ZG5_3305.zip and assume that you've saved it in your home directory. If using Linpus Lite, this would be /home/user, but in this case, we'll assume there is a user named john. Substitute /home/john with the location of the zip file

```
cd /mnt  
sudo unzip /home/john/ZG5_3305.zip
```

Once unzipped, you find a directory, probably called BIOS-3305. It may have a slightly different name, depending upon the version that you downloaded and the download site. (Different sites seem to have packed it slightly differently.)

It is possible that the files will be in the BIOS-3305 directory, or there may be another subdirectory, possibly called 3305, that contains them. Once you find them (it's not hard) :, follow the next steps.

As per macles' instructions, rename the ZG5_3305.fd file (or whatever file has the .fd suffix--this will change with BIOS version) to ZGIA32.FD.

```
mv ZG5_3305.fd ZG5IA32.FD
```

If your browser's font doesn't make it clear, that is an upper case letter "i" after the 5 and before the A, not the numeral one.

Run the ls command (by just typing ls) to make sure you renamed it correctly. It might show up as lower case, i.e., zg5ia32.fd, this is fine and seems to be a peculiarity of FreeDOS.

Copy both the renamed file and FLASHIT.EXE to the root directory of your USB drive. In other words, copy them out of whatever directory and/or subdirectory that they're in.

```
cp <path_to_directory>/FLASHIT.EXE /mnt  
cp <path_to_directory>/ZG5IA32.fd/mnt
```

For example, in the file that I am looking at as I write this, those two files are in /mnt/BIOS-3305/3305. So, my command in this case would be, assuming I was in the BIOS-3305/3305 directory (where I went to rename the .fd file)

```
cp FLASHIT.EXE /mnt  
cp ZG5IA32.fd /mnt
```

Now, your /dev/sdb1 should have those two files in there as well as in any directories or subdirectories created when you unzipped the file. Do a quick check with ls /mnt to make sure they are there. You should see them as well as the BIOS-3305 directory.

Flash the BIOS

The macles article, linked at the beginning of this page, covers this pretty thoroughly. I'll do a quick repeat here.

Turn off the Aspire One.

Make sure that both battery and AC power cord are connected. Insert the USB stick that you prepared.

Press Fn+Esc and press the power button to turn the machine on. Release Fn+Esc after a few seconds. The power light should be blinking.

Press the power button once and the machine will initiate the BIOS flash. DO NOT INTERRUPT THIS.

After a little while, the button should stop blinking. The machine should reboot itself shortly.

If it doesn't reboot, then wait at least 5 minutes to be sure that it isn't still flashing the BIOS. If it doesn't work, try again, making sure that you've followed the procedure correctly.

For the majority of people, judging from the forums, this solves the black screen of death.

Once again, many, many thanks to macles. I reiterate that all mistakes, however, are my own.