Dado que el Linux que lleva de serie no explota todas las capacidades de este pequeño pero gran gadget, en helpdeskweb hemos decidido trastear con él (sin usar windows, soporte para el cual acer tiene completo en su web) y meterle ubuntu (en un principio probamos fedora)

fuente https://help.ubuntu.com/community/AspireOne

- AspireOne

Install Ubuntu Hardy Heron (8.04.1) on the Acer Aspire One

Fully functional:

- Suspend / Resume
- Video (with desktop effects)
- Wireless Networking - *** the hardware indicator light is not functional ***
- Wired Networking
- Webcam
- USB

Partial Function:

- Card Readers (only work if card in when booting, researching fix)
- Audio - there is sound, issues detailed below

Not Functional:

- Hibernate

Pre-Requs:

Before you begin you will need the following:

-

A copy of 8.04.1 Hardy Heron.
This is the latest version. If you downloaded the ISO a while ago, it may be 8.04.0. The original 8.04.0 release will not even boot.

- A 1+ GB memory key that can be formatted.
-

A copy of the liveUSB DEB.
(Website: http://klik.atekon.de/liveusb and the DEB: http://ppa.launchpad.net/probono/ubuntu ... u1_all.deb).

- A wired network connection.
- Another computer, with network access.

Note on using this guide

This guide was written with Ubuntu in mind. Generally everything works the same with the other varients of the distrobution (Kubuntu, Xubuntu). For ease use gedit is used as the text editor. If you are using a varient please substitute a different text editor (kate for Kubuntu, mousepad for Xubuntu or just use vim/emacs/nano).

Step One: Preparation

So the first thing we need to do is create a bootable copy of the Ubuntu LiveCD. This will be our install media for the Aspire One. Its actually pretty similar to creating recovery media for Linpus Lite (the included version of Linux).

Download and copy the LiveUSB .deb file from the link above to a memory key (you can use the same one we we are going to format to create the install media, we only need this file once), and remove the memory key. The LiveUSB software should only be installed when booted off the LiveCD environment, so we do that next.

Boot to the 8.04.1 Live CD on your other computer, and choose the first option off the boot menu (Try Ubuntu). When you reach the desktop insert the memory key with the LiveUSB software one it and double click. Follow the prompts to install it. Now we can create our bootable USB key. Start the LiveUSB application from System -> Administration -> Install LiveUSB. Select the target device (my two keys were listed by a human readable name, so it was easy to find) and click Execute. It will give you a warning about data loss, so double check that you selected the correct device and continue.

Step Two: Install Ubuntu

Once that is done, you can shut down the other computer and remove the memory key. Shut down your Aspire One and insert the memory key that we just used. Turn it on and tap F12 to bring up the boot menu. Choose the boot the USB HDD option. This will boot you to the LiveUSB stick, and allow you to install Ubuntu. Install it like normal, except for good performance, and to increase the life of the SSD use a non-journaled (I suggest EXT2) filesystem. If you have already installed with EXT3 then follow this post: http://www.aspireoneuser.com/forum/viewtopic.php?f=5&t=164&st=0&sk=t&sd=a&start=10#p1177 to convert to EXT2. Installation (file copy) will take a LONG time (hour +). If your not currently connected to the internet on a wired connection, you may get an error about setting up a mirror. Don't worry about it if you do however, its non-fatal.

Step 3: Tweak / Fix

So now we should have an installed Ubuntu system. At this point wire your One into the network. If you use DHCP it should pick up an IP address automatically. The first thing you will want to do is updates, since the wireless driver needs to be reinstalled after every kernel update. Open a terminal (Applications -> Accessories -> Terminal). Do the updates:

```
sudo apt-get update
sudo apt-get upgrade
```

WIRELESS:

Now we need to disable the hardware drivers that Ubuntu tries to use before the ones we make will function. So go to System -> Administration -> Hardware Drivers and uncheck everything. It should prompt us to reboot, so lets do it now.

We need to grab the wireless driver, and the things we need to build it, from a terminal:

```
mkdir source
cd source
wget http://snapshots.madwifi.org/madwifi-hal-0.10.5.6-current.tar.gz
tar xzvf madwifi-hal-0.10.5.6-current.tar.gz
cd madwifi-hal-0.10.5.6-r3816-20080724
sudo apt-get install build-essential linux-headers-$(uname -r)
```

And we build, install, and make it persistent...

```
make
sudo make install
sudo echo ath_pci >> /etc/modules
sudo modprobe ath_pci
```

You should now have working wireless. If you get a symbol mismatch when loading the driver, just reboot as that has corrected it everytime for me.

Every time there is a kernel update you will need to perform the following steps to make the wireless work. Go to the directory (madwifi-hal-0.10.5.6-r3816-20080724) and run:

```
make clean
make
sudo make install
```

USB MOUNT:

If you insert a memory key, you may notice an error and that it cannot be mounted. This is due to the CD-ROM entry in the fstab. Since we don't have an optical drive on the One we will comment that out. From a terminal again:

```
sudo gedit /etc/fstab
```

You should see a line that looks like:

```
/dev/sdb        /media/cdrom0   udf,iso9660 user,noauto,exec 0       0
```

add a hash in front:

```
#/dev/sdb       /media/cdrom0   udf,iso9660 user,noauto,exec 0       0
```

Reboot, and automount should work.

REDUCING SSD WEAR:

Frequent writes to the SSD will cause failure eventually. We can reduce the number of writes to the SSD my moving our logs to a temporary filesystem in RAM that gets destroyed at ever reboot. Now this means your logs will not be persistent across reboots making debugging difficult in some cases. This step is optional of course, so if you need the logs for an extended period of time do not follow these steps.

Open your fstab again, and add the following lines:

```
sudo gedit /etc/fstab
tmpfs      /var/log/apt    tmpfs       defaults          0    0
tmpfs      /var/log        tmpfs       defaults          0    0
tmpfs      /tmp            tmpfs       defaults          0    0
tmpfs      /var/tmp        tmpfs       defaults          0    0
```

AUDIO:

Out of the box there are various issues with the sound. These range from headphone detection not functioning correctly, to the internal MIC not working. There are solutions to these problems. Currently, however, there is no known way to get everything working at once. All of the steps begin the same way, rebuilding ALSA:

sudo apt-get install module-assistant
sudo m-a update
sudo m-a prepare
sudo m-a a-i alsa
sudo alsa force-unload
sudo depmod -ae
sudo modprobe snd-hda-intel

Now we need to make a choice. To have the internal MIC non-functional (external works), but sound working after suspend and resume, we edit /etc/modprobe.d/alsa-base (sudo gedit /etc/modprobe.d/alsa-base) and add the following line to the bottom:

options snd-hda-intel model=toshiba

Reboot for that to take effect.

To have the internal MIC function correctly, but no sound after suspending and resuming the computer add or change the following to the /etc/modprobe.d/alsa-base as before:

options snd-hda-intel model=auto

Again, reboot for this to take effect.

You can switch between the two models at any time to get the desired function.

If you experience crackling sound after rebooting, insert the following line in /etc/modprobe.d/blacklist:

blacklist snd_pcsp