Link−sys WPC11 Mini−HOWTO

Gerardo Arnaez

garnaez_((AT))_yahoo.com

Revision History

Revision 1.1	2002−04−13	Revised by: gea
Correct grammar, made things a little more clearer, made software requirements more explicit.		
Revision 1.0	2002−03−24	Revised by: gea
Written because I spent enough figuring this out that I wanted to store "how I did it" somewhere I wouldn't lose it and figure you all might like it too.		

This is a Cookbook on how to set up a Wireless Link−Sys WPC11 card using a Link−SYS Wireless Access Point/DSL/Switch on a Debian system .

Table of Contents

1. Link−sys WPC11 install on Debian

1.1. Why Debian and why just this card?

I have been trying for months to get wireless working on Debian and after reading far and wide and getting help from irc.debian.org, I realized that there really is no Cookbook in getting wireless set up. Thus having just done it I want to commit to 'paper' so that you all can use it and I can refer to it knowing it is safe somewhere. :)

1.2. Required Hardware

By required I mean, here is what I used to get this to work, and may serve as guide to anyone who wants to know what really works.

BEFW11S4− EtherFast? Wireless AP + Cable/DSL Router w/4−Port Switch. I really really like this WAP (Wireless Access Point). It is OS independent (read, linux friendly) and is configured using a browser so no need to touch Microsoft software at all, even to configure it. And if you don't know what a switch is, let me tell ya, they rock. Essentially they allow the NIC to communicate in both directions at the same time. I highly recommend one.

Link−sys WPC11. I have a version 3.0 and don't recommend any thing less than a version 2.5 Cost about 80 dollars

---

1.3. Software Requirements

Debian Distribution. I find the 'testing' distribution to work well

go to Absolute Systems and download the 11Mbps linux−wlan(tm) Project

you need the pcmcia−cs. Get this by

```
apt-get install pcmcia-cs
```

This is different from pcmcia−source which I also asked you apt−get. Thus also do

```
apt-get install pcmcia-source
```

You need the wireless tools so

```
apt-get install wireless-tools
```

You will also need some way to setup you IP address on your wireless card, I recommend either

```
apt-get install DHCP-client
```

to install the DHCP−client that will automatically configure your IP address, if you have a DHCP server. The

WAP−11 hardware does provide DHCP server capabilities

Or at least have the *pump* application, which also will query a DHCP server and get you an IP address from the DHCP server. Note I tend to use *pump* when I am trying out new hardware to see if there is a connection, since to test a particular device, say *eth0* I would type

```
pump −i eth0
```

where the option *−i* tells *pump* what device to try to get an IP address. In this particular case, when I could not get Debian to automatically set up my wireless card, which was device *wlan0*, I would type

```
pump −i wlan0
```

and *pump* would try to set up the device. Anyway, the point is, that it is a good trouble shooting command, and you should know about it, and I talk more about it later.

I also recommend you use the "kernel−package" package when you want to build your new kernel, which I will get to. This tool is very good and you should be using it anyway when you are building new kernel for the Debian distribution. You can install it by typing

```
apt−get install kernel−package
```

Also, be sure to read the documentation it comes with, in case I don't do a good job explaining how to use it, later in this document

1.4. Kernel Configuration

1.4.1. What TO enable

In order to use the wireless tools, like *iwconfig*, which will allow you tell how good your connection is, you need to enable support for *Wireless LAN (Non−Ham Rasio)*.

You can do this by:

```
Go to:

Network Device support -->

then Select:

  Wireless LAN (non-hamradio)  --->
```

Then Choose the options, so that it looks like below, or something as close to this. Note I am uing 'make menuconfig' to configure my kernel

```
[*] Wireless LAN (non-hamradio)
< >   STRIP (Metricom starmode radio IP)
< >   ATT WaveLAN & DEC RoamAbout DS support
< >   Aironet Arlan 655 & IC2200 DS support
< >   Aironet 4500/4800 series adapters
< >   Cisco/Aironet 34X/35X/4500/4800 ISA and PCI cards
<*>   Hermes chipset 802.11b support (Orinoco/Prism2/Symbol)
< >   Hermes in PLX9052 based PCI adaptor support (Netgear MA301 et
```

1.4.2. What *NOT* to Enable

One of the main stumbling blocks was to realize that the *pcmcia support in the kernel is not as good as the pcmcia−source support* that one gets when you build it from pcmcia–source.

I use either

```
make xconfig
```

or

```
make menuconfig
```

to configure my kernel, so when you configure your kernel, be sure to not have pcmcia support enabled under

```
General setup
```

Nor do you want to select any particular pcmcia card under

```
Network device support
```

. I repeat you do not want this under the kernel and you will be building it when you download pcmcia–source.

Be sure to download all the necessary components before you take pcmcia support out of the kernel, otherwise, if you were using a pcmcia card for net access, you will not be able to connect to the Internet using the new kernel, until you have built both pcmcia support and module drivers for the wireless card

2. Using make−kpkg to build kernels

Note: *makepine −kpkg* was installed when you installed *kernel−package*

Once you have configured your kernel just save and exit, and don't use make−kpkg yet! I will show you a good way to make sure sure everything is builds well.

Follow the instruction in the make−kpkg documentation but essentially:

You need to be in the

```
/usr/src/linux
```

directory, then

```
make−kpkg clean
```

to clean all the binaries out and prepare to build a new kernel, then:

```
make−kpkg −−revision−custom.1.0 kernel_image
```

Don't Do this Yet!

We need to get the pcmcia−source ready to be built at the same time as the kernel is built. This is a very neat feature of make−kpkg!

---

3. Building PCMCIA−SOURCE

First, be sure to download pcmcia−source, by typing

```
apt-get install pcmcia-source
```

This will download the source into

```
/usr/src
```

as

```
pcmcia-source.tar.gz
```

You now need to gunzip the file by

```
gunzip pcmcia-source.tar.gz
```

and then untar the file by

```
tar xvf pcmcia-source.tar
```

You should see pcmcia−source unpacked into the directory

```
/usr/src/modules/pcmcia-cs
```

4. Using make−kpkg to build the new kernel and pcmcia−source modules

Be sure the pcmcia−source is under /usr/src/modules.

Go ahead and configure your kernel and be sure that pcmcia support IS NOT compiled in as an option in the kernel.

To build the kernel and pcmcia−source, be sure you are under the

```
/usr/src/linux
```

or have a symbolic link from /usr/src/linux to whatever kernel source you have set up. Then type

```
make-kpkg --revision=custom.1.0 kernel_image modules_image
```

The *kernel_image* option will build the kernel while the *modules_image* option will build all modules located under

```
/usr/src/modules/
```

so be sure that you it is ok to build any other modules there.

After some chugging, go up one level to

```
/usr/src
```

and you should see two new Debian packages that should look something like this:

`kernel−image−2.4.19−pre4_custom.1.0_i386.deb`

`pcmcia−modules−2.4.19−pre4_3.1.31−7+custom.1.0_i386.deb`

You first want to install the kernel image so you would type

```
dpkg -i kernel-image-etc....
```

Now install the modules by typing

```
dpkg -i pcmcia-modules.etc...
```

There are a couple of assumptions that make−kpkg makes about your lilo.conf file is that you have not radically changed it. the new kernel will be installed over the older kernel image and the older kernel image will be moved to linux.old. Anyway if you have any questions ask me.

5. Wlan Drivers for You Link−Sys Card

You have downloaded the 11 Wlan project. Go a head and read the instruction, and put it under modules. Follow the instructions when you

```
make config
```

The one key is to make sure you specify the pcmcia−source as under

```
/usr/src/modules/pcmcia−cs
```

and not choose the default it gives you.

Go ahead and

```
make all
```

and

```
make install
```

I suggest you read the documentation that comes with it, but essentially, if you have a WAP that is connected to your DSL or cable modem then you have a infrastructure set up. I found that it was best to edit the

```
networks.opt
```

under the

```
/etc/pcmcia
```

directory.

To make things easier edit the option

```
# Use DHCP (via /sbin/dhcpcd, /sbin/dhclient, or /sbin/pump)? [y/n]

DHCP="y"
```

to what I have, i.e., set it to yes.

The documentation talks about setting ESSID but when you edit the

```
wlan−ng.opts
```

you will only see

```
#=======INFRASTRUCTURE STATION START===================

# SSID is all we have for now

AuthType="opensystem"           # opensystem | sharedkey (requires WEP)

DesiredSSID="howardnet"
```

From what I can gather,DesiredSSID means ESSID and it works when the WAP and link−sys pcmcia card share the same name.

At this point, you should reboot and should have a working link−sys card that gets its address via DHCP.

6. Checking things in case they don't work

1. Be sure to type

```
ifconfig
```

You should something like this

```
lo         Link encap:Local Loopback

           inet addr:127.0.0.1  Mask:255.0.0.0

           UP LOOPBACK RUNNING  MTU:16436  Metric:1

           RX packets:0 errors:0 dropped:0 overruns:0 frame:0

           TX packets:0 errors:0 dropped:0 overruns:0 carrier:0

           collisions:0 txqueuelen:0

           RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)


wlan0      Link encap:Ethernet  HWaddr 00:06:25:A8:AE:64

           inet addr:192.168.1.104  Bcast:192.168.1.255  Mask:255.255.255.0

           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

           RX packets:35197 errors:0 dropped:0 overruns:0 frame:0

           TX packets:57676 errors:0 dropped:0 overruns:0 carrier:0

           collisions:0 txqueuelen:100

           RX bytes:43386657 (41.3 MiB)  TX bytes:2670811 (2.5 MiB)

           Interrupt:3 Base address:0x100
```

The keys point here are that have a real IP address, and that Bcast and Netmask are set up such that they are on the same "wave−length" as your Wireless Access Point.

2. If you don't, you might have had the same problem i did which was that there was no easy script to initiate the wlan0 device setup. That is to say, if the card was recognized but you still did not get a connection and say that ifconfig showed wlan0 present but with no IP address. In other words, you might see something like this:

```
text:/home/dude# ifconfig


lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:700 (700.0 b)  TX bytes:700 (700.0 b)

wlan0     Link encap:Ethernet  HWaddr 00:06:25:A8:AE:64
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:46 (46.0 b)  TX bytes:0 (0.0 b)
          Interrupt:3 Base address:0x100
```

As you can see, the interface device, the Wireless pcmcia card, is noted, but there is no *inet addr*. The pcmcia software recognized the card, but it has not successfully connected with the Wireless Access Point.

I used the command,*pump* to send a simple DHCP request to the DHCP server for the device in question. I used

```
pump −i wlan0
```

which essentially runs a simple DHCP request to set up that card, wlan0, in this case.

You can get the *pump* by

```
apt−get install pump
```

While I needed to use

```
pump −i wlan0
```

on my laptop, I did not need this when I set up the link−sys wireless WPC11 card on my girlfriend's laptop. She has a Link−Sys WPC11 version 2.5 pcmcia card.

---

7. Wireless tools

While it is not necessary to include this in your kernel configuration, you can enable Wireless tool extensions by going (i assume you use xconfig or menuconfig) to

```
Network device support
```

and then go to

```
Wireless LAN (non-hamradio)
```

and enable support for the

```
Hermes chipset 802.11b support (Orinoco/Prism2/Symbol)
```

. This will let you use the Wireless Tools like

```
iwconfig
```

,

```
iwspy
```

and such.

The one thing I found this good for is that by repeated typing iwconfig, you can see your Link Quality. Its quite good

---

8. Wireless Access Point

Perhaps its it missing the forest for the trees, but I did not spend any discussion setting up the actual Wireless Access Point. The reason is that the documentation that comes with the WAP is actually really good and well written. The only thing I haven't spoken about is enabling Wireless Encryption Protocol in the WAP (Wireless Access Point) which I really don't suggest as I don't think WEP has been properly set up in the drivers for the Pcmcia Wireless Cards. However, let me know if you have any problems and I will be glad to help.

9. Request for comments

I will be glad to help anyone out and if things are a bit confusing in this quite mini how to, please tell me how I can fix it to make it better.

Thanks!