Link-sys WPC11 Mini-HOWTO

Gerardo Arnaez

dude@mung.net

Raintree I.T.™ (http://www.raintree.org)
Department of Advanced Skunk Works

Revision History

Revision 2.2.4 2004-06-03 Revised by: gea
Cleaning up this HOW-TO. I have since moved on and offer this document for adoption or at least consider it obs
Revision 2.2.3 2003-07-31 Revised by: gea
WPC11 CARD DRIVERS HAVE CHANGED. Thanks to Bill Atkins for providing information and solution
Revision 2.2.2 2003-07-05 Revised by: gea
Most likely last update for this manual. The new 2.5 (to be 2.6) kernel appears to work fine with respect to wirele
Revision 2.2 2003-04-07 Revised by: gea
Made a few typo corrections. Publish it on Freshmeat
Revision 2.0.2 2003-02-24 Revised by: gea
Thanks to Justin Stockton for helping me eliminate a confusing bit of reading.
Revision 2.0 2003-01-15 Revised by: gea
I have decided to generalize this document to other distributions then just Debian. The redhat section is due to t
Revision 1.2 2003-01-11 Revised by: gea
Made clear where to get most up-to-date documents
Revision 1.1.4 2002-12-22 Revised by: gea
I forgot what i did here
Revision 1.1.3 2002-06-09 Revised by: gea
Made clear what version on linux-wlan I actually used.
Revision 1.1.2 2002-05-26 Revised by: gea
In part 2 of the HOWTO, the last item says make-kpkg --revision-custom.1.0 kernel_image when later on you us
Revision 1.1.1 2002-04-27 Revised by: gea
some more grammar improvements, and highlighting to make things clearer
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 i
Revision 1.1.4 2002-8-10 Revised by: gea
I attempted to follow my own instruction on re-installing on the same laptop and found my how-to a little lacking.

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. Other systems are addressed.

1. Preliminaries

1.1. Where to Get Most Recent Updates

The most recent updates to this mini-doc are at the mung[dot]net (http://www.mung.net). If you mirror this document, please try to keep it the most recent one.

2. WARNING ABOUT WPC11 VERSION 4

Caution

The NEW WPC11 CARDS HAVE Realtek 8180 CHIPSET INSTEAD OF THE RTL8180. Thanks to Juan Natera for clarification

Caution

It had come to my attention that the new version, Version 4 are not compatible with my old instructions. I include a set of instruction provided by bill atkins

Until I can clean this up, I include Bill atkins email for sake of urgency

Quote from Bill Atkins

OK.

First of all, make sure you have a V4 card. Type

```
cardctl ident
```

as root. If one of the entries shown is a

```
RealTek RTL8180L
```

or something similar, then you have a version 4 card.

Caution

If not, then you can probably just follow the rest of the instructions in the HOWTO.

Now you need to get drivers for the card. Go to RealTek's download AND do a search fo 8180 from the downloads section

or you can download the driver that works with Bill's email at

```
ftp://152.104.125.40/cn/wlan/rtl8180l/rtl8180_24x_suse82.zip
```

and pick up the drivers for SuSE (you don't need to be running SuSE for the drivers to work - I used them with Gentoo. However, the other divers don't seem to work at all).

Unpack the incoming tarball. As of this writing, there is a minor bug in the driver code that must be repaired in order to make the card work.

Open up r8180_type.h.

Caution

On line 128, you'll see a line with two slashes before the text. Remove these slashes. Now you're ready to build.

At the shell prompt, type make. The drivers will build themselves. If there are any problems making the drivers, open up the Makefile and check the kernel version settings on the first few lines.

Now open up the wlanup file.

Uncomment line 5 (remove the #) and change the SSID to the SSID of your network.

Uncomment line 8 and set the ssid2scan to your network's SSID. Uncomment line 9 and set the networktype to infra (unless you really are using adhoc). Save your changes.

Now eject the card

```
cardctl eject
```

and plug it in again.

From the directory where you unpacked the drivers, type

```
insmod -f rtl8180_24x.o
```

You will get a warning - ignore it.

Now run the wlanup script found in the driver package. Your card should now appear when you type ifconfig. You should configure your IP address at this point. If you use DHCP, just type "dhcpd wlan0". Try pinging google.com. You should get replies back. If so, your card is working!

Now copy rtl8180_24x.o to /lib/modules/YOURKERNELNAME, where YORUKERNELNAME is the name of the directory in /lib/modules.

Then copy the wlanup and wlandown scripts to /sbin.

3. Link-sys WPC11 install on Debian

3.1. 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. :)

3.1.1. Redhat installation

I have gotten a few requests for help on getting the card installed on other distributions. I will try to address the RedHat Installation in this mini-how-to

3.1.2. Suse Installation

My girlfriend got the WPC-11 card working on Suse. I am waiting for her to give me her how to

3.2. 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

3.2.1. What is the linksys card based on?

It is an Inersil Prism 3-based card

3.3. Software Requirements

3.3.1. Debian Software Requirements

Table 1. Debian Software Requirements

Software	**Version**	**URL link**	**notes**
Debian Distribution	Stable ("Woody")	www.debian.org (http://www.debian.org)	linux-2.4.20.tar, patched with patch-2.4.21-pre3.bz[a]
Absolute Systems	0.1.16-pre8	Absolute systems (http://www.linux-wlan.com/linux-wlan/)	Make sure you download the *11Mbps* version. Works for 0.1.16-pre8
pcmcia-source	Stable	`apt-get install pcmcia-source`	<--type this
pcmcia-cs	stable	`apt-get install pcmcia-cs`	<--type this
wireless-tools	Stable	`apt-get install wireless-tools`	<--type this
pump	stable	`apt-get install pump`	Useful to see if card works
kernel-package	stable	`apt-get install kernel-package`	Good way to build kernel and the one I describe.
Kernel	2.4.20, patched with patch-2.4.21-pre3.bz2	www.kernel.org (http://www.kernel.org)	You must know how to build and patch a kernel to do this. Its not hard and I will show you[b]

Software	Version	URL link	notes
Notes: a. This new patched kernel worked amazingly well b. Note to patch a kernel you type `bzip2 -dc patch-2.4.21-pre3.bz2 | patch -p0`			

3.3.2. RedHat Software Requirements

Table 2. RedHat Software Requirements

Software	Version	URL link	notes
Redhat 8.0	stock kernel	*	RedHat (http://www.redhat.com)

3.4. Notes on additional helpful software

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

4. Debian Kernel Configuration

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 Radio)*.

In these examples, I use

```
make menuconfig
```

to configure my kernel.

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 using 'make menuconfig' to configure my kernel

```
  [*] Wireless LAN (non-hamradio)
  < >   STRIP (Metricom starmode radio IP)
  < >   AT T 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
  <*>    Prism 2.5 PCI 802.11b adaptor support
```

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.

Caution

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

5. Building PCMCIA-SOURCE

Download pcmcia-source, by typing

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

This will download the source into

```
/usr/src
```

as

```
pcmcia-cs.tar.gz
```

You now need to gunzip the file by

```
gunzip pcmcia-cs.tar.gz
```

and then untar the file by

```
tar xvf pcmcia-cs.tar
```

You should see pcmcia-source unpacked into the directory

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

6. Using make-kpkg to build the new kernel and pcmcia-source modules

Steps to build the kernel

- 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 pointing to whatever kernel source you have set up.
- Type
  ```
  make-kpkg clean
  ```
  to clean
- 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 do indeed want to rebuild any other modules that are located in source when you are ready to build your new kernel.

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...
```

Caution

There are a couple of assumptions that make-kpkg makes about your lilo.conf file. One is that you have not radically changed it. Make-kpgk will make symbolics links from '/boot' where the actual kernel resides to 'vmlinuz' which is under '/'. In other words, under '/', you will see *vmlinuz* and *vmlinuz.old* which are symbolic links to the real kernel images under */boot/*. Anyway if you have any questions ask me.

7. 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.

8. 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 *inet addr:*has 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.

9. 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

10. RedHat Installation

This is a summary of what Mr.Tony Perrie of involution.com (http://involution.com) fame has wriiten. I include here with his permission. I encourage you to visit his site because he has a great "how-to" on IPtables (http://involution.com/iptables_demo/).

10.1. Drivers

The stock kernel driver to use with the WPC11 is the orinoco_cs. Make sure that it's loaded.

```
modprobe orinoco_cs
```

If you get some error messages after executing modprobe, insure that the wavelan_cs and wvlan_cs drivers are unloaded. By doing

```
lsmod | egrep lan
```

If they are loaded do the following:

```
rmmod wavelan_cs
```

and

```
rmmod wvlan_cs
```

10.2. Hermes.conf Hack

The hack is putting the following in

```
/etc/pcmcia/hermes.conf.
```

card "Instant Wireless Network PC Card"
manfid 0x0274,0x1613
bind "orinoco_cs"

10.3. Redhat PCMCIA Services

You'll then need to restart pcmcia service.

```
service pcmcia restart
```

10.4. Redhat System Tools

Go to the Redhat System Tools and hit Configure. Then add a wireless device in Managed mode if you have an access point.

Setup dhcp, and the WEP key. The channel autoconfigures to 6 in managed mode.

10.5. Restart the network.

```
service network restart
```

11. 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 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.

12. 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!