

Raspberry Pi wireless adapter

Hardware

This guide is about installing the [Edimax EW-7811Un](#) nano wireless adapter on a Raspberry Pi with a wireless router supporting DHCP and configured to use WPA2-PSK. The guide may be useful for installing other wireless adapters as well and for configuring WPA-PSK.

Software

The software used was the Debian squeeze package `debian6-19-04-2012.img` plus a replacement driver module (`8192cu.ko`) available from [here](#). The driver is a `tar.gz` file and needs to be unpacked twice. No other modifications were made to the software.

Power supply

There have been reports that wireless adapters take a lot of power, but the Raspberry Pi with an EW-7811Un adapter runs on a Samsung Micro USB mains charger, a standard Kindle power supply and even the Kindle lead attached to a USB hub on a laptop running on battery power.

Accessing RPi

It is possible to configure the adapter using `ssh` over Ethernet, but it is a lot easier to see error messages if a display screen is directly attached.

Configuration

All of this configuration needs to be done as a privileged user, so you can prefix all the editing commands with `sudo` or just run "`sudo su`" to turn yourself into an administrator.

`/etc/udev/rules.d/70-persistent-net.rules`

Delete anything in this file about `wlan0` or `wlan1`. This file is used to remember the physical characteristics of USB adapters and may well contain incorrect or misleading information, and should be empty apart from comments unless you have added other USB devices.

`/etc/network/interfaces`

Update this file to add the `wlan0` section, so that it looks like this.

```
# Used by ifup(8) and ifdown(8). See the interfaces(5) manpage or
# /usr/share/doc/ifupdown/examples for more information.

auto lo

iface lo inet loopback
iface eth0 inet dhcp

auto wlan0
iface wlan0 inet dhcp
pre-up wpa_supplicant -Dwext -i wlan0 -c /etc/wpa_supplicant.conf -B
```

/etc/wpa_supplicant.conf

Create this file looking like the following.

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
ap_scan=2
network={
    ssid="myssid"
    scan_ssid=1
    proto=RSN
    key_mgmt=WPA-PSK
    pairwise=CCMP
    group=CCMP
# choose one of the following
    psk="this is my secret phrase"
    psk=58cfe1c8a36ce8a1f3cd938e281ca1222707df58e27df4745148bfebca7c4197
}
```

This file is critical, and you should note the following points:

1. There must be quotes around the SSID.
2. These parameters relate to WPA2-PSK, even though it says WPA-PSK.
3. The key (psk) can be specified in quotes or as a hexadecimal string obtained from running `wpa_passphrase`. If you are concerned about security, you may prefer this, but it's not necessary.
4. You can make the file unreadable by ordinary users with the command `chmod 600 wpa_supplicant.conf`.

Installing the driver

To install the new wifi driver in place of the one supplied in the Debian image, you need to add the new driver to the operating system and block the old one.

1. Copy the file `8192cu.ko` to the directory `/lib/modules/3.1.9+/kernel/net/wireless`.
2. Add the line `blacklist rtl8192cu` to `/etc/modprobe.d/blacklist.conf`.
3. Add the line `8192cu` to `/etc/modules` so that wifi support is loaded at boot time.
4. Run the command `depmod -a`.

Testing the system

Reboot the Raspberry Pi and it should all just work! The blue light on the wifi adapter should come on, and should acquire an IP address from the router.

Check by running the command `ifconfig -a` on the Raspberry Pi to find the IP address and connect using `ssh` (PuTTY).

Troubleshooting

1. If the blue light on the adapter comes on, but the adapter does not acquire an IP address, the problem is in `wpa_supplicant.conf`. `ifconfig` will show the adapter, but not an IP address.
2. If the blue light does not come on, the problem is in the earlier stages.
3. Use the command line utility `dmesg` review kernel messages. It is helpful to save the output to file so that it can be edited and examined at leisure (use `dmesg >dmesg.log` and then `edit dmesg.log`).
4. Despite the fact that `rtl8192cu.ko` has been blocked, `dmesg` still gives the following message:
`usbcore: registered new interface driver rtl8192cu`. This does not indicate an error.
5. `dmesg` does not record error messages relating to errors in `wpa_supplicant.conf`. The only way to see these is to boot with a directly attached display and watch the messages going by. These are errors such as a failure to put quotes round the SSID.

Stability

Tomasz Milas in his very helpful posting [Raspberry Pi meets Edimax](#) recommends a large number of fixes to improve the stability of the Debian system. These are not required to make the Edimax adapter work, but may solve other problems.

The only problems discovered so far using the standard Debian image and the above configuration changes are that the wireless connection does not re-establish itself after the wireless router is rebooted, and `/etc/init.d/networking restart` brings down the network but does not re-establish it, either for Ethernet or wireless (possibly the same problem).

Removing the wifi adapter and reinserting it does not restart the network either.

Additional points

1. The above configuration file is for WPA2-PSK and DHCP. WPA-PSK and static IP addresses have not been tested.
2. The above process is cumbersome if you want to use different wireless access points, so you may want to consider one of the following (not tested).
 - a. [wicd](#) is an open source wired and wireless network manager that does not require graphical dependencies. The [Farnell site](#) also mentions `wicd-curses`, which may or may not be the same thing.
 - b. [Network Manager](#) is a network manager that does us a graphical interface.
3. You may want to configure your wireless router to give a fixed IP address to the Raspberry Pi Ethernet and wireless connections.
4. You can use the command `lsusb` to identify the chipset used by your wireless adapter.