

Wi-Pi Installation Procedure for Raspbian on Raspberry Pi

This guide assumes you have a Raspberry Pi, and running the Raspbian “Wheezy” operating system on it. Raspbian is based on Debian Linux, and if you are running an older version of Debian, or some other distribution, then a more general guide is at the end of this article.

One obvious point. If you are using a USB keyboard and mouse, you are going to lose one of the USB sockets to the Wi-Pi device. That means you are going to need to use a USB hub. The hub is best accommodating the keyboard and mouse, leaving the Wi-Pi module to plug directly into the remaining USB socket on the Raspberry Pi.

Raspbian already has Wi-Pi’s drivers’ built in, so the process of downloading the driver and installing it is not necessary. However, there are still a few things to do before you can get Wi-Pi running on your Raspberry Pi.

If you aren’t familiar with editing text files, now is the time to try! It is better to use a command line editor like **vi** or **nano**,- both of which are already installed on Raspberry Pi, but whichever one you choose, it is important that you run them as root, otherwise any edits you make will not be saved. This is achieved by using the command **sudo** before starting the editor, which means “superuser do”, and is a way becoming the administrator for the computer, albeit for the duration of the command:

```
sudo vi <file to be edited>, or
```

```
sudo nano <file to be edited>
```

If you prefer to use an editor from the desktop like **Leafpad**, you can, but you must start it with the command line command

```
sudo leafpad <file to be edited>
```

Also note that if you haven’t already made sure you have changed directory to the place where the files to be edited are located, you need to include the full path to the file as well.

1. Configure Wireless Settings

So, knowing that the driver is already installed, the only thing we need to do is to configure the wireless settings. In other words, what the WiFi access point name is, what type of encryption is in use on that access point, and the password – all the normal things that you’d see from a Windows computer when you scan for WiFi networks. Unfortunately, there is no scan feature here, so we must know this detail before hand:

The file we need to edit is called **interfaces**, and it is located in folder **/etc/network**. So, if we were using Nano, the command we need to use would is

```
root@raspberrypi:~# sudo nano /etc/network/interfaces
```

The file will already have a network entry for the localhost, or loopback network interface, and the Ethernet socket, known as **eth0**. We’re going to add a new interface called **wlan0**,

There are two slightly different ways of editing this file, depending on which of type broad types of encryption is in use on the WiFi network you wish to connect to.

In the case of WPA/WPA2, add the following lines to the end of the **interfaces** document:

```
auto wlan0
iface wlan0 inet dhcp
wpa-ssid <name of your WiFi network>
wpa-psk <password of your WiFi network>
```

In the case of WEP, add the following instead

```
auto wlan0
iface wlan0 inet dhcp
wireless-essid <name of your WiFi network>
wireless-key <password of your WiFi network>
```

Make sure these files are typed exactly as they are presented here, and also note that you do not type the quotation marks.

2. Restart the Network Services

Even though you may have performed the above procedure exactly, the network services running on the Raspberry Pi need to be told to adopt the new settings. For that to happen, we have to stop and start network services on the Raspberry Pi, or even better, reboot your entire Raspberry Pi. Make sure your Wi-Pi module is plugged in to the USB socket before you perform the restart.

Restarting the network is achieved with the following command

```
root@raspberrypi:~# sudo /etc/init.d/networking restart
```

3. Verify everything is working

There are a few ways to check that everything is working the way it should.

- First of all, try to open a web page, for example using Midori. If that is successful, and you manage to see a public web page, everything is fine and the installation has succeeded and your Wi-Pi is working properly.
- If your browser did not work properly, you can look a bit more deeply. Try sending a ping to your local router by using the `ping` command. Using the command line, type something like `ping 192.168.1.1`, and the IP address will be unique to your own setup. If successful, you should see a series of responses that look a little like this

```
64 bytes from 192.168.1.1: icmp_req=1 ttl=254 time=6.86 ms
```

Press Ctrl-c to stop exit ping

- If that didn't work, its time to look a little closer to the wireless interface itself. Again, in the command line, type `ifconfig`. You should get a response made up of three main sections, which are the key interfaces – `eth0`, `lo`, and `wlan0`. If the `wlan0` section exists, that is good – it means part of your configuration worked – go and check your typing, and the WiFi access point information you have entered. If, within that section, you have an entry that reads

`inet addr: 192.168.1.43 (or some other address)`

that is good, and means you have succeeded. There will be a problem elsewhere related to your WiFi access point or router that needs to be resolved.

4. If you aren't using Raspbian

There are many Linux distributions, and it is beyond the scope of this guide to cover all of them. However, it is likely that a similar approach to the above can be used, but the drivers need to be manually installed beforehand. Please therefore follow steps 3 to 4 on the Wi-Pi User Manual included with your Wi-Pi, and then proceed with the steps above.

Revisions		
24-Aug-2012	Revision 1.0	Raspbian user guide released