

HOWTO build the RaLink RT3090 driver on Ubuntu 10.04 LTS (Lucid Lynx)

initial version: Sun, 06 Jun 2010 23:06:27 -0400

last modified: Fri, 11 Jun 2010 23:09:15 -0400

- * This HOWTO document is placed under the public domain.
- * You are free to copy, mirror, link to, or duplicate it.

This HOWTO is mirrored in two places:

- * http://www.halibutdepot.org/how_to_build_rt3090_for_ubuntu_lucid/
- * http://stat.case.edu/~jrt32/how_to_build_rt3090_for_ubuntu_lucid/

I have an MSI Wind U230 laptop. The built-in wireless NIC identifies itself as an "RaLink RT3090 Wireless 802.11n 1T/1R PCIe". This NIC is not supported out-of-the-box under Ubuntu 10.04 LTS (Lucid Lynx) because there is no driver for it in the 2.6.32-22-generic kernel tree.

I searched for a solution and found that other people who are running Lucid Lynx and using the same unsupported rt3090 NIC are getting it to work by using the old version of the rt3090 driver packaged by Markus Heberling in his PPA (<https://launchpad.net/~markus-tisoft/+archive/rt3090>).

I'm grateful to Markus Heberling for doing all of the hard work to package RaLink's driver for Ubuntu. But because his PPA for the rt3090 has not been updated since Ubuntu 9.10 (Karmic Koala), and because RaLink has released a newer version 2.3.1.4 that no one has packaged, I don't feel comfortable relying on the old rt3090 package built for Karmic, even though it does work fine on Lucid.

Below I describe the procedure I used in order to build an rt3090-dkms package just like the one that Markus Heberling built for Karmic Koala. The process involves downloading the newest rt3090 driver from RaLink (version 2.3.1.4 at the time of this writing), applying Markus Heberling's patches against it, and building a package from the source.

Here is the procedure:

- (1) Install packages required to build packages.
 - (a) `sudo apt-get install build-essential`
 - (b) `sudo apt-get install devscripts`
 - (c) `sudo apt-get install cdb`
- (2) Define a root directory in which to download files.
 - (a) `cd /path/to/some/download/directory/`
 - (b) `export DOWNLOAD=`pwd``
- (3) Get the rt3090 drivers from the manufacturer's website.
 - (a) Option 1: Use the pre-saved RaLink tarball from http://stat.case.edu/~jrt32/how_to_build_rt3090_for_ubuntu_lucid/RT3090_LinuxSTA_V2.3.1.4_20100222.tar.bz2
 - (b) Option 2: Download the tarball directly from RaLink: Go to <http://eng.ralinktech.com.tw/support.php?s=2>
 - (c) Download the file "RT3090PCIe". (RT3090_LinuxSTA_V2.3.1.4_20100222.tar.bz2)
- (4) Get Markus Heberling's Ubuntu-friendly patches against RaLink's driver tarball.
 - (a) Option 1: Download a mirrored copy of Markus Heberling's patch: http://stat.case.edu/~jrt32/how_to_build_rt3090_for_ubuntu_lucid/rt3090_2.3.1.3-0ubuntu0-ppa1.diff.gz
 - (b) Option 2: Download the patches from Markus Heberling's PPA: Browse to <https://launchpad.net/~markus-tisoft/+archive/rt3090/+packages>
 - (c) Expand the release "rt3090 - 1:2.3.1.3-0ubuntu0-ppa1".
 - (d) Download the patch "rt3090_2.3.1.3-0ubuntu0-ppa1.diff.gz"
- (5) Get my patch to bump the packaged version number to 2.3.1.4 .
 - (a) Download this patch: http://stat.case.edu/~jrt32/how_to_build_rt3090_for_ubuntu_lucid/rt3090-2.3.1.3-2.3.1.4.diff

- (6) Apply patches against RaLink's drivers.
 - (a) Create a staging directory:

```
mkdir -p /path/to/some/staging/directory
cd /path/to/some/staging/directory
export STAGING=`pwd`
```
 - (b) Extract RaLink's drivers to the staging directory.

```
cd $STAGING
tar xjf $DOWNLOAD/RT3090_LinuxSTA_V2.3.1.4_20100222.tar.bz2
cd $STAGING/RT3090_LinuxSTA_V2.3.1.4_20100222/
```
 - (c) Apply Markus' patches to the extracted sources.

```
cd $STAGING/RT3090_LinuxSTA_V2.3.1.4_20100222/
gzip -dc $DOWNLOAD/rt3090_2.3.1.3-0ubuntu0~ppa1.diff.gz | patch -p1
```
 - (d) Update the version number from 2.3.1.3 to 2.3.1.4 .

```
cd $STAGING/RT3090_LinuxSTA_V2.3.1.4_20100222/
patch -p1 < $DOWNLOAD/rt3090-2.3.1.3-2.3.1.4.diff
```
- (7) Build a DKMS-friendly .deb package.
 - (a) `cd $STAGING/RT3090_LinuxSTA_V2.3.1.4_20100222/`
 - (b) `dpkg-buildpackage -A -sa -us -uc`
 - (c) An installable .deb package gets created and stored as `$STAGING/rt3090-dkms_2.3.1.4-0ubuntu0~ppa1_all.deb`
 - (d) Optional: You may download my pre-built rt3090-dkms package here:
http://stat.case.edu/~jrt32/how_to_build_rt3090_for_ubuntu_lucid/rt3090-dkms_2.3.1.4-0ubuntu0~ppa1_all.deb
- (8) Install the newly-built package.
 - (a) `sudo dpkg -i $STAGING/rt3090-dkms_2.3.1.4-0ubuntu0~ppa1_all.deb`
 - (b) `sudo reboot`
- (9) Appendix: Automatically rebuilding the module via DKMS.

Sometimes I switch to a different kernel (such as the one provided in the package "linux-rt"). This breaks the wireless NIC because the module hasn't yet been built and installed in the new kernel. Because we built a DKMS module, we can have it rebuilt automatically after we reboot into a new kernel:

 - (a) `sudo dkms build -m rt3090 -v 2.3.1.4`
 - (b) `sudo dkms install -m rt3090 -v 2.3.1.4`