

# Rigging a mobile 2.4 Ghz logger

by Dipl.-Ing. Frank Siegert, eMail frank@wizards.de, All rights reserved 2008.

THIS DOCUMENTATION IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL FRANK SIEGERT OR ANY OTHER CONTRIBUTOR BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THIS DOCUMENTATION OR THE PROCEDURES DESCRIBED WITHIN.

YOU ARE WORKING ENTIRELY AT YOUR OWN RISK.

Released under the *GNU Free Documentation License 1.2*, see <http://www.gnu.org/licenses/fdl.txt>

Basic knowledge of Linux installation procedures and command line use is needed. If you don't know about it, ask someone who does.

## 1. Preamble

With the rising popularity of 2.4 Ghz wireless devices there is often the need for a mobile logging solution to get field data over a long time period or in harsh environments. The WiSpy-2.4 USB spectrum analyser is a very helpful tool but it is tied to a PC or notebook. In some environments, especially close to Radio Control receivers in model airplanes, the logger must be much more lightweight and small. This document describes how to build such a small logger with standard components and without soldering.



## 2. Prerequisites

You need

- 1 - WiSpy or WiSpy-2.4 USB based spectrum analyser, see [www.metageek.net](http://www.metageek.net)
- 2 - Linksys NAS NSLU2
- 3 - USB Stick (larger than 2 GB recommended), the smaller the better
- 4 - Network cable (a small patch cable do nicely)

Furthermore a Windows or Linux system and Internet access is needed to install the base system. To setup and control the system in the field a notebook and a host-to-host network cable is needed.

## 3. System setup



(Stripped NSLU2 removed from its case with attached WiSpy-2.4 and Sony 4 GB USB Stick, onboard 266 Mhz ARM CPU and 32 MB memory)

The Linksys NSLU2 is a cheap, small Linux based device that runs from a single 5V power source. It features 2 powered USB host connections. Total weight of our setup is less than 200g (incl. WiSpy and memory).

You can read about the NSLU2 here:

<http://en.wikipedia.org/wiki/NSLU2> (English)

or

<http://de.wikipedia.org/wiki/NSLU2> (German)

It is normally used as a NAS, to connect USB hard drives to a network. In our application we use it as a full Linux system and we install Debian 4.0r3 on it:

<http://www.cyrius.com/debian/nslu2/>

<http://www.cyrius.com/debian/nslu2/install.html>

## 4. Installation

I used "upslug2" on Fedora Linux to download the 'unofficial' Debian image to the NSLU2. You can find this image "[debian-4.0r3.zip](#)" here:

**<http://www.slug-firmware.net/d-dls.php>**

You really want to use the 'unofficial' image as it contains support for the network card of the NSLU2.

Alternative methods - also using a Windows system - to upgrade are described in above installation description.

In case your NSLU2 locks up during upgrade (like mine) when the Debian installer is setting the clock you can try to remove the CR2032 battery briefly from the NSLU2 for a few minutes and then plug it in again to reset the NSLU2 hardware clock. This procedure is only needed if the hardware clock of the NSLU2 is locked and it takes ages to boot or install.

We use the Spectools to access the WiSpy over USB.

To setup the Spectools for getting access to the WiSpy you need to fetch the archive:

[http://www.wizards.de/~frank/RC/spectools\\_NSLU2\\_wsr.tar.bz2](http://www.wizards.de/~frank/RC/spectools_NSLU2_wsr.tar.bz2)

This is a ready-made and enhanced version of the Spectools from

<http://kismetwireless.net/spectools/>

Now copy over the archive to the NSLU2 and unpack it:

```
tar xjvf spectools_NSLU2_wsr.tar.bz2
```

You will find a folder 'spectools'. Inside you find a program 'spectool\_wsr', you can start it (become root first):

```
spectool_wsr logfile.wsr
```

Logging will start immediately. You can stop the recording by sending a SIGINT to the process (press CTRL-C).

To detach the logging process press CTRL-Z and enter 'bg' in the console. The logging process will go into the background. You can now log off.

To stop the recording after flight log into the system again and identify the process

```
ps aux | grep spectool_wsr
```

then send it a SIGINT signal

```
kill -2 <PID>
```

Recording will stop and the file dumped.

The resulting logfile.wsr is compatible to the 'Chanalyser' application from Metageek.net, so you can use it to visualize the data.

Frank Siegert,  
March 2008