

Linux Reef Controller – NSLU2 Sensor Server

The Linksys NSLU2 Network Storage Link device can easily be hacked with a modified firmware and used as a sensor server for your LRC. Here's how!

- First you gotta find one! Since they are not manufactured anymore this can be tough or expensive!
 - A good place to start looking would be here: [Cisco-Linksys Storage Link for USB 2.0 Disk Drives NSLU2](#)



- Download the latest Unslung firmware from [here](#).
- Follow the firmware flashing instructions provided in the Unslung Firmware [README](#)
 - Make sure to follow the instructions **TO THE LETTER!!**
- After you have successfully flashed your NSLU2, ssh into it and perform the following commands to update it:
- After the device reboots install some necessary packages
 - Follow the guide [here](#) to fix an issue with the hardware clock and ntpclient
- Setup OWFS
 - Add the following to the owfs-feed.conf file:
 - Save the file and run the following commands:
 - You should be able to access the owfs server from

the built in web interface.

- <http://DEVICEIP:3001/>

- Your Linksys NSLU2 is now ready to serve as a OWFS sensor server
 - Get yourself one of [these](#) and some [sensors](#), plug them in and reboot the NSLU2 for the correct modules to load on boot.
 - Once the device is rebooted you should be able to see the sensor Id's listed in the web interface.
 - You can also view all of the devices by using the following owfs commands in an ssh session
 - List the one wire directory tree:
 - List a sensor directory:
 - Read sensor data:
 - A temperature sensor read example in Fahrenheit
 - Your Slug is now ready to be polled by the webserver via cron jobs and perl scripts!
-

[Linux Reef Controller - Example Script & Cron Entry](#)

This page will run through a perl script that is used on the [Web Server](#) in order to process polled data from sensors attached to [Sensor Servers](#) on your sensor network.

The script can be stored anywhere as long as it can be called by cron. It can also be reused for each sensor on your network by just making a few small modifications. This script is the

heart of the project and provides sensor polling, website updates, graph generation and updating, email alerts, and logging.

To have the script run every 5 mins setup a cron entry:

- Append the following line to your crontab to have the script run every 5 mins:
- Restart Cron
- This will create a cron job that runs the script at every 5 mins and logs any runtime errors to

[Linux Reef Controller – Web Server Setup](#)

This guide will explain how to setup a web server to poll sensor servers that are installed on your network, and use that data to generate meaningful graphs and output to a webpage for central data collection and viewing of all of your sensor information.

In all of the following examples I use Gentoo Linux as my Operating System. Any flavor of Linux will do as long as you can install the required owfs packages, either via source or a package manager. Check your particular distro's documentation for further details.

This document assumes you have a fully configured Gentoo or other Linux environment with Apache, PHP, Perl, Sendmail, and Cron properly configured. Any specific configuration requirements will be outlined. Other than that default

installations will suffice.

Notice: The installation and configuration of Gentoo Linux, Sendmail, Cron, Perl, Apache, and PHP are beyond the scope of this document. Please visit the [Gentoo Website](#) for a wealth of documentation to help you along. You may also check out my [Gentoo Guide](#).

- Before you begin verify package settings
 - Add the following to /etc/portage/package.keywords
 - Add the following to /etc/portage/package.use
- Make sure your kernel has fuse support
 - Should output:
 - If not rebuild your kernel with fuse support
- Setup your web directory and cgi-bin
 - On Gentoo these are both setup in /var/www/localhost by default after installing Apache
 - Setting up userdir support and vhosts are beyond the scope of this document, however they will work fine if used.
- Install owfs and rrdtool
- Create the necessary directories
 - Get root privileges
 - Setup the directory that will hold the generated web file for each sensor that you will include in your main page.
 - Setup the graphs directory
 - Setup the RRD database directory
 - Setup the log directories
- All the prereqs should now be in place! Your server now has the ability to poll your devices as well as serve as a sensor server itself.

- Check out my [Script Example](#) page to see scripts you can use to get things going!