AR9001 Linux Driver User Guide

November 2008

ATHEROS™
COMMUNICATIONS

Subject to change without notice.

Notice

The information in this document has been carefully reviewed and is believed to be accurate. Nonetheless, this document is subject to change without notice, and Atheros Communications, Inc. (Atheros) assumes no responsibility for any inaccuracies that may be contained in this document, and makes no commitment to update or to keep current the contained information, or to notify a person or organization of any updates. Atheros reserves the right to make changes, at any time, to improve reliability, function or design and to attempt to supply the best product possible. Atheros does not represent that products described herein are free from patent infringement or from any other third party right. No part of this document may be reproduced, adapted or transmitted in any form or by any means, electronic or mechanical, for any purpose, except as expressly set forth in a written agreement signed by Atheros. Atheros or its affiliates may have patents or pending patent applications, trademarks, copyrights, maskwork rights or other intellectual property rights that apply to the ideas, material and information expressed herein. No license to such rights is provided except as expressly set forth in a written agreement signed by Atheros.

ATHEROS MAKES NO WARRANTIES OF ANY KIND WITH REGARD TO THE CONTENT OF THIS DOCUMENT. IN NO EVENT SHALL ATHEROS BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL SPECULATORY OR CONSEQUENTIAL DAMAGES ARISING FROM THE USE OR INABILITY TO USE THIS PRODUCT OR DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBLITY OF SUCH DAMAGES. IN PARTICULAR, ATHEROS SHALL NOT HAVE LIABILITY FOR ANY HARDWARE, SOFTWARE, OR DATA TRANSMITTED OR OTHERWISE USED WITH THE PRODUCT,
INCLUDING THE COSTS OF REPAIRING, REPLACING, INTEGRATING, INSTALLING OR RECOVERING SUCH HARDWARE, SOFTWARE OR DATA. ATHEROS SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE AS THEY MIGHT OTHERWISE APPLY TO THIS DOCUMENT AND TO THE IDEAS, MATERIAL AND INFORMATION EXPRESSED HEREIN.

Revision History

Revision Date	**Release Version**	**Release Description**
11/17/08	1.0	Initial version
11/27/08	1.1	Update WPA section

Document Conventions

Text Conventions

bold

Bold type within paragraph text indicates commands, file names, directory names, paths, output, or returned values.

Example: The DK_Client package does not function, except when the **wdreg_install** batchfile is used.

italic

Within commands, italics indicate a variable that the user must specify.

Example: **mem_alloc** *size_in_bytes*

Titles of manuals or other published documents are also set in italics.

Courier

The Courier font indicates output or display.

Example: Error: Unable to allocate memory for transfer!

[ ]

Within commands, items enclosed in square brackets are optional parameters or values that the user can choose to specify or omit.

{ }

Within commands, items enclosed in braces are options from which the user must choose.

|

Within commands, the vertical bar separates options.

…

An ellipsis indicates a repetition of the preceding parameter.

>

The right angle bracket separates successive menu selections.

Example: Start > Programs > DK > wdreg_install**.**

Notices

NOTE: This message denotes neutral or positive information that calls out important points to the text. A note provides information that may apply only in special cases.

Table of Contents

1

Overview

Due to the increased popularity of the Linux operating system for desktops and notebooks, Atheros provides a Linux solution for the AR9001 802.11n Wireless LAN Card. AR9001 provides 802.11n wireless solutions for the USB interface. Our AR9001 solution can run in the Infrastructure (Managed) and Ad-hoc modes, which may be easily changed. This document is intended to describe the setup and use of the AR9001 in a Linux operating system.

Requirements

1. Kernel 2.4 or 2.6.
2. To build AR9001 driver, users will need to configure the kernel source code for the kernel being used. Ideally, configured implies that 'make config', 'make menuconfig', or 'make xconfig' has been run previously. If your platform is not an SMP system, please do not configure SMP support due to unresolved symbols made when the module is loaded.
3. Make sure kernel USB 2.0 support is running

   Use lsmod to check if the "ehci-hcd" module is loaded.

2

Driver Installation

This section contains information describing how to install the AR9001 Linux driver in the Linux operating system.

RPM Installation

The installation can be done by double clicking the RPM package, or by manually installing it using the following RPM command:

```
rpm -ivh otusdriver-3.2.0.X.i386.rpm
# where X is the version number
```

During the RPM installation process, the package will check whether the kernel source, GNU toolchain and awk have been installed. A message will be prompted and the installation will stop if those tools have not yet been installed. Those tools and the kernel header can be installed into the **Fedora** distribution from the installation CD or by choosing the add/remove program. In most cases, they have already been installed.

After the RPM has been installed, the source code will be copied to /tmp/srcOtusLinux_3_2_0_X.tar.gz.

There is an additional action needed to install this driver in the **Ubuntu 6.1** distribution. The installation program needs to connect to the internet and issue the following command to install the building environment:

```
apt-get install build-essential
apt-get install
```

RPM Un-Installation

To remove the driver, or if a problem occurs during the RPM package installation process, the RPM package can be removed using the following command:

```
rpm -e otusdriver-3.2.0.X
```

Source Code Installation

Users need to initially decompress the source code package by using the tar program. After extracting this package, users can see the source files. The directory should be changed into this directory to proceed to the next step.

```
tar zxvf srcOtusLinux_3_2_0_X.tar.gz
# where X is the version number
```

Enter the makefile folder:

```
cd srcOtusLinux_3_2_0_X/OAL/Otus/Linux
# where X is the version number
```

Build the driver module:

```
make
```

Install driver module:

```
make install
```

3

Loading the Driver

Generally, the driver is automatically loaded when the AR9001 dongle is inserted. If not, Users may use the `modprobe -v arusb_lnx` to load the driver. To check whether the driver has loaded successfully, one can use the "*lsmod*" for this check. If the driver was loaded successfully, the following message should be seen.

```
arusb_lnx 183576 0
```

NOTE: Please note that the driver number, 183576, may not be the same as the one in your system.

Opening the Network Interface

In the driver, all commands will be stopped until the network interface assigned is opened. Users may open the network interface by issuing the following command.

```
ifconfig ath0 up
```

or

```
ifconfig ath0 up <IP address>
```

Configuring the Wireless Settings

This driver supports wireless extension commands to control the driver.

scan:

Scans the wireless network.

Example:

```
iwlist ath0 scan
```

essid:

Sets the ESSID (or Network Name – it may be called the Domain ID in some products). The ESSID is used to identify cells which are part of the same virtual network.

Example:

```
iwconfig ath0 essid <ESSID>
```

mode:

Sets the operation mode of the device.

Example:

```
iwconfig ath0 <mode>
```

Possible modes include:

Managed (Infrastructure Station mode)
Ad-hoc (Ad-hoc mode)

channel:

Sets the channel of the device.

Example:

```
iwconfig ath0 channel <channel>
```

NOTE: Note that the set channel command will **NOT** work under Managed (infrastructure mode) because in this mode, the channel should change to the channel of the AP with which to associate

rts[_threshold]:

Sets the RTS threshold.

Example:

```
iwconfig ath0 rts 250
```

frag[_threshold]

Sets the fragmentation threshold.

Example:

```
iwconfig ath0 frag 512
```

key/enc[ryption]

Used to manipulate the encryption or scrambling keys and the encryption mode. To set the current encryption key, just enter the key in hex digits as *XXXX-XXXX-XXXX-XXXX* or *XXXXXXXX*. To set a key other than the current key, append *[index]* to the key itself. You can also enter the key as an ASCII string by using the *s:* prefix. To change which key is the current active key, just enter *[index]* (without entering any key value). *off* and *on* disable and re-enable encryption, *open* sets the system in open mode (accept non-encrypted packets) and *restricted* discards non-encrypted packets.

Examples:

```
iwconfig ath0 key 0123-4567-89 [1]
```

```
iwconfig ath0 key [1] open
iwconfig ath0 wlan0 key off
```

power:

Used to manipulate the power management scheme mode.

Examples:

```
iwconfig ath0 power on #Turn on power saving mode
```

```
iwconfig ath0 power off #Turn off power saving mode
```

Private Commands

Except for command support for wireless extensions, we also define some commands in which to set parameters for the driver. One can use the ***“iwpriv”*** command for this purpose.

Set authentication mode:

Used to set the authentication of the driver.

Command:

```
iwconfig ath0 set_auth <Auth Type>
```

0: Open system	1: Shared key

NOTE: Be aware that the shared key authentication requires a WEP key

Setting Up an IP Address

When using the Linux RedHat distribution, users may edit the `/etc/sysconfig/network-scripts/ifcfg-ethX` or edit the `/etc/network/interfaces` under the Debian to set up an IP address during the boot up process. The ***netconfig*** command for setting IP addresses may also be used.

Two type settings in the following examples may be used. One is to assign a fixed IP address, netmask, and default gateway. Another is to get the IP configuration from a DHCP server.

Fixed Setting

Example:

```
DEVICE=’eth0'
IPADDR='192.168.2.98'
NETMASK='255.255.255.0'
NETWORK='192.168.2.0'
BROADCAST='192.168.2.255'
```

```
ONBOOT='yes'
GATEWAY='192.168.2.254'
```

Get the IP Setting from a DHCP Server

Example:

```
DEVICE=’eth0’
BOOTPROTO=’dhcp’
ONBOOT='yes
```

Working With the Linux WPA Supplicant

The following procedure should work flawlessly in Linux Fedora Care2 distribution. For other distribution packages, users may need to install additional libraries (ex. openssl) required to build the WPA supplicant.

Setting Up the WPA Supplicant in Linux

To create a WPA-PSK connection, please modify the configuration file (for a detailed description, users may refer to the original sample configuration file, **wpa_supplicant.conf)** **wpa_supplicant_psk.conf** to meet the testing conditions for WPA-PSK.

Example for WPA-PSK:

```
network={
            ssid="Atheros"
            proto=WPA
            key_mgmt=WPA-PSK
            pairwise=CCMP TKIP
            group=CCMP TKIP WEP104 WEP40
            psk="12345678"
            priority=2
}
```

The settings are similar for WPA-EAP-TLS and WPA PEAP.

Here is an example for WPA-EAP-TLS:

```
network={
            ssid="atheros"
            proto=WPA
            key_mgmt=WPA-EAP
            pairwise=CCMP TKIP
            group=CCMP TKIP WEP104 WEP40
        eap=TLS
        identity="user@atheros.com"
        ca_cert="/etc/cert/fluffy.pem"
        client_cert="/etc/cert/id.pem"
        private_key="/etc/cert/id_key.pem"
        private_key_passwd="password"
            priority=2
}
```

For Non-WPA 802.1X networks, just change the key_mgmt field from WPA-EAP to IEEE8021X.

The fluffy.pem is created by:

```
openssl pkcs12
-in fluffy.pfx
-passin pass:password
-out fluffy.pem
-cacerts
-nokeys
```

The id_key.pem is created by:

```
openssl pkcs12
-in fluffy.pfx
-passin pass:password
-passout pass:password
-out id_key.pem
-nocerts
```

The id.pem is created by:

```
openssl pkcs12
-in fluffy.pfx
-passin pass:password
-out id.pem
-nokeys
```

You can run the openssl utility (Included in openssl.zip) in a Microsoft Windows OS.

For a detailed description, please refer to CertConvReadme.txt. (Located in lnx_wpa_supplicant.tar.gz)

Here is an example for WPA PEAP:

```
network={
            ssid="Atheros"
            key_mgmt=WPA-EAP
            eap=PEAP
            identity="jhsieh"
            password="jhsieh"
            ca_cert="/etc/cert/fluffy.pem"
            phase1="peaplabel=0"
            phase2="auth=MSCHAPV2"
            priority=10
}
```

4. After modifying, use the following commands to set up the WPA connection.

If the AR9001 is not open yet, open it first by using this command:

```
ifconfig ath0 up <IP address of the network interface>
```

After the network interface has been opened, enter this command to build the WPA-PSK connection:

```
./wpa_supplicant -Dwext -iath0 -c wpa_supplicant_psk.conf
```

To build the WPA EAP-TLS:

```
./wpa_supplicant -Dwext -iath0 -c wpa_supplicant_tls.conf
```

To build the WPA PEAP:

```
./wpa_supplicant -Dwext -iath0 -c wpa_supplicant_peap.conf
```

NOTE: @-i: interface name: ath0
@-c: Configuration file: wpa_supplicant_psk.conf
@-D: the name of the network interface

If the WPA-PSK connection is built successfully, the following message will be displayed

```
...
WPA: Sending EAPOL-Key 2/2   ---> The Group handshake is about to
finish.
...
EAPOL: SUPP_PAE entering state SUCCESS
EAP: EAP entering state SUCCESS
EAPOL: SUPP_PAE entering state AUTHENTICATED
EAPOL: SUPP_BE entering state IDLE
```

NOTE: There is an operational issue with the WPA supplicant. The WPA supplicant should be kept running during the connection. Pressing Ctrl-C to stop the WPA supplicant will also close the network interface card due to the call back routine. Therefore, users must issue the ifconfig ath0 up command again before using the network interface card.

Customization of the VID/PID Table

The AR9001 chipset may be used to drive just about any NIC card, but if the VID/PID isn't listed in the driver, the Linux kernel will not load the AR9001 driver to serve the device. Users may add the device's VID/PID to the driver source. Then, the kernel will load the driver the next time the USB dongle is inserted.

In the source file, zdusb.c, a static structure, and “struct usb_device_id”, stores the VID/PID list of supported products. Users may add the dongle’s VID/PID here. Then, remake, install the driver, plug in the dongle and it should function properly.

5

Conclusion

This document was not meant to explain how to setup the wireless LAN environment in detail. Problems may be incurred when setting up the wireless LAN environment. If you have any question about how to set up the environment, you may e-mail us or find the solution on the network.

The information in this document has been carefully reviewed and is believed to be accurate. Nonetheless, this document is subject to change without notice. Atheros assumes no responsibility for any inaccuracies that may be contained in this document, and makes no commitment to update or to keep current the contained information, or to notify a person or organization of any updates. Atheros reserves the right to make changes, at any time, to improve reliability, function or design and to attempt to supply the best product possible.

Atheros Communications, Incorporated
5480 Great America Parkway
Santa Clara, CA 95054
t: 408/773-5200
f: 408/773-9940
www.atheros.com