ATHEROS™
COMMUNICATIONS

AR9001U Linux Driver User Guide

REVISION 0.01

(02/04/08)

Revision History

Date	Revision		Comments
02/04/08	0.01	YunGu Wei	Initial version

1. Introduction

The document describes how to build and install the WLAN driver AR9170 on Linux platform, and also how to enable WPA supplicant and WPS feature. The AR9170 Linux driver is released in both RPM and source code package. The installation can be installed by RPM by just clicking the icon or execute the rpm command. In case of some Linux distribution can not provide RPM. The source code is provided for installation.

2. Driver Installation

2.1 RPM Installation

The installation requires kernel source header and GNU toolchain, awk. Following steps to check whether your machine contain those tools needed for building the driver.

Kernel source header

```
]$ ls –al /lib/modules/`uname –r`/build
```

GNU toolchain

```
]$ which gcc
]$ which make
```

awk

```
]$ which awk
```

The installation can use the file browser to double click the RPM we provide, or manually install by the following command

```
]$ rpm –ivh otusdriver-3.2.0-X.i386.rpm       (X means the release version for our driver)
```

During the RPM installation process, the package will check whether kernel source, GNU toolchain and awk are installed. It will show message and stop installing if those tool are not installed. Those tools and kernel header could be installed from the installation CD or by choosing add/remove program to add for these tools in **Fedora** distribution. In most of cases, they are installed.

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

There is an additional action is needed to be installed in the **Ubuntu 6.1** distribution. It needs to connect to the internet and issue the following command to install building environment.

```
]$ apt-get install build-essential
]$ apt-get install
```

2.2 RPM Un-Installation

To remove drive or in case of problems happened in the installation procedure of rpm package, the rpm package can be removed by following commands.

]$ rpm –e otusdriver-3.2.0-X

2.3 Source code Installation

In case of the Linux distribution did support RPM. The installation can be done by source code installation. The following steps are used for manually compile for Otus.

```
]$ tar zxvf srcOtusLinux_3_2_0_X.tar.gz
]$ cd srcOtusLinux_3_2_0_X/OAL/Otus/Linux
]$ make RELEASE=<DIST>   TARGET=<ARCH>
]$ make RELEASE=<DIST>   TARGET=<ARCH> install
```

The <DIST> in RELEASE option is used to specify the Linux distribution that can use different compile rule in different distribution. There are 4 different <DIST> options.

```
FC_1 : Fedora Core 1 only
MDV  : Mandriva
RH_9 : RedHat 9 only
FC     : Others
```

The <ARCH> in TARGET is used to specify the CPU type. It could be either x86 or x86_64 which means 32bit or 64bit platform. There are 2 different <ARCH> options.

```
x86:    32 Bit x86 platform
x86_64: 64 bit x86 platform
```

Note that, x86_64 only support RELEASE=FC now.

2.4 Make WLAN connection

Here list an example to show how make a connection when driver is installed.

Open the network interface

```
]$ ifconfig ath0 up
```

or

```
]$ ifconfig ath0 <IP address>
```

Site survey

```
]$ iwlist ath0 scan
```

Configure the Wireless settings

The driver support the wireless extension commands to control devices. So the WLAN parameters can be set via iwconfig command.

For example:

```
]$iwconfig ath0 essid <ESSID>            : Set ESSID
]$iwconfig ath0 channel <channel>        : Set channel
]$iwconfig ath0 key 0123-4567-89 [1]     : Set WEP key index 1
```

For more, please reference to help of iwconfig

Setup IP address for DHCP client

```
]$ dhclient
```

After acquiring the IP from DHCP server, sometimes, the resolver configuration file /etc/resolv.conf might be overwritten by dhclient. If the connection to internet are forbidden, this file should be checked whether the file is empty. Usually, the file should be looked at following:

```
################## /etc/resolv.conf ###################
nameserver      xx.xx.xx.xx
```

where xx.xx.xx.xx should be the name server IP address of yours.

3. WPA Supplicant Installation

AR9170 driver can be controlled by the open source supplicant project, wpa_supplicant. The driver control interface did not integrated into this project, so the it need used the wpa_supplicant we provide.

```
]$ tar zxvf wpa_supplicant-0.X.Y_otus.tar (X, and Y are the release version)
]$ cd wpa_supplicant-0.X.Y
]$ make
```

After the build process, there is a tool named wpa_supplicant in the folder. The wpa_supplicant is driven by a configuration file, so the configuration file is needed to be modified to make connection . Following command for starting the connecting.

```
]$ ./wpa_supplicant –Dotus –iathX –c <configuration file>
```

To modify configuration file, there is example in the wpa_supplicant-0.X.Y/examples or wpa_supplicant-0.X.Y/wpa_supplicant.conf

4. WPS Installation

WPS is a new feature for the Wireless LAN. It can bring a very convenient way for those users which don't know too much about the WLAN. There is an open source code project for WPS. One can get further information by going into the following URL
http://softwarecommunity.intel.com/articles/eng/2662.htm

It might be some modification to the source code, so just untar the source code we provide.

```
]$ tar zxvf WscRefImpl_Intel_1_0_X_otus.tar.gz          (X is the release version)
```

After untar the tarball, there will be a folder named WscRefImpl_Intel_1_0_X generated. The build and usage can refer to the README in the WscRefImpl_Intel_1_0_X/src/linux.

The Openssl 0.9.8a or higher version for openssl is needed, it need to be installed before building this source code. The building procedure is as follows:

```
]$ cd WscRefImpl_Intel_1_0_X/src/linux
```

```
]$ make
```

Then the WPS program **“wsccmd”** will be generated.

The Intel’s WPS reference design **“wsccmd”** is also a configuration file wsc_config.conf driven program. Then copy the wsc_config_X_cli.conf in the scRefImpl_Intel_1_0_X/src/common to WscRefImpl_Intel_1_0_X/src/linux

Note that the reference design is not a standalone program, it should also need wpa_supplicant to be its front-end process. The wpa_supplicant we provide should be built firstly and copy it into WscRefImpl_Intel_1_0_X/src/linux. Program **“wpa_supplicant”** and **“wpa_cli”** are both needed to be copied.

To invoke the wsccmd by issuing the following command

```
]$ ./wsccmd
```

Note that wsccmd should be under the X Windows environment, it will create a xterm for running wpa_supplicant.