Intel(R) PRO/Wireless 2915ABG Driver for Linux

Support for:

- Intel(R) PRO/Wireless 2200BG Network Connection
- Intel(R) PRO/Wireless 2915ABG Network Connection

Note: The Intel(R) PRO/Wireless 2915ABG Driver for Linux and Intel(R) PRO/Wireless 2200BG Driver for Linux is a unified driver that works on both hardware adapters listed above. In this document the Intel(R) PRO/Wireless 2915ABG Driver for Linux will be used to reference the unified driver.

README.ipw2200

Version: 1.1.2
Date: March 30, 2006

0. IMPORTANT INFORMATION BEFORE USING THIS DRIVER

Important Notice FOR ALL USERS OR DISTRIBUTORS!!!!

Intel wireless LAN adapters are engineered, manufactured, tested, and quality checked to ensure that they meet all necessary local and governmental regulatory agency requirements for the regions that they are designated and/or marked to ship into. Since wireless LANs are generally unlicensed devices that share spectrum with radars, satellites, and other licensed and unlicensed devices, it is sometimes necessary to dynamically detect, avoid, and limit usage to avoid interference with these devices. In many instances Intel is required to provide test data to prove regional and local compliance to regional and governmental regulations before certification or approval to use the product is granted. Intel's wireless LAN's EEPROM, firmware, and software driver are designed to carefully control parameters that affect radio operation and to ensure electromagnetic compliance (EMC). These parameters include, without limitation, RF power, spectrum usage, channel scanning, and human exposure.

For these reasons Intel cannot permit any manipulation by third parties of the software provided in binary format with the wireless WLAN adapters (e.g., the EEPROM and firmware). Furthermore, if you use any patches, utilities, or code with the Intel wireless LAN adapters that have been manipulated by an unauthorized party (i.e., patches, utilities, or code (including open source code modifications) which have not been validated by Intel), (i) you will be solely responsible for ensuring the regulatory compliance of the products, (ii) Intel will bear no liability, under any theory of liability for any issues associated with the modified products, including without limitation, claims under the warranty and/or issues arising from regulatory non-compliance, and (iii) Intel will not provide or be required to assist in providing support to any third parties for such modified products.

Note: Many regulatory agencies consider Wireless LAN adapters to be modules, and accordingly, condition system-level regulatory approval upon receipt and review of test data documenting that the antennas and system configuration do not cause the EMC and radio operation to be non-compliant.

The drivers available for download from SourceForge are provided as a part of a development project. Conformance to local regulatory requirements is the responsibility of the individual developer. As such, if you are interested in deploying or shipping a driver as part of solution intended to be used for purposes other than development, please obtain a tested driver from Intel Customer Support at:

http://support.intel.com

1. Introduction

The following sections attempt to provide a brief introduction to using the Intel(R) PRO/Wireless 2915ABG Driver for Linux.

This document is not meant to be a comprehensive manual on understanding or using wireless technologies, but should be sufficient to get you moving without wires on Linux.

For information on building and installing the driver, see the INSTALL file.

1.1. Overview of Features

The current release (1.1.2) supports the following features:

- BSS mode (Infrastructure, Managed)
- IBSS mode (Ad-Hoc)
- WEP (OPEN and SHARED KEY mode)
- 802.1x EAP via wpa_supplicant and xsupplicant
- Wireless Extension support
- Full B and G rate support (2200 and 2915)
- Full A rate support (2915 only)
- Transmit power control

- S state support (ACPI suspend/resume)

The following features are currently enabled, but not officially supported:

- WPA
- long/short preamble support
- Monitor mode (aka RFMon)

The distinction between officially supported and enabled is a reflection on the amount of validation and interoperability testing that has been performed on a given feature.

1.2. Command Line Parameters

Like many modules used in the Linux kernel, the Intel(R) PRO/Wireless 2915ABG Driver for Linux allows configuration options to be provided as module parameters. The most common way to specify a module parameter is via the command line.

The general form is:

```
% modprobe ipw2200 parameter=value
```

Where the supported parameter are:

associate

Set to 0 to disable the auto scan-and-associate functionality of the driver. If disabled, the driver will not attempt to scan for and associate to a network until it has been configured with one or more properties for the target network, for example configuring the network SSID. Default is 0 (do not auto-associate)

Example: % modprobe ipw2200 associate=0

auto_create

Set to 0 to disable the auto creation of an Ad-Hoc network matching the channel and network name parameters provided. Default is 1.

channel

channel number for association. The normal method for setting the channel would be to use the standard wireless tools (i.e. *iwconfig eth1 channel 10*), but it is useful sometimes to set this while debugging. Channel 0 means 'ANY'

debug

If using a debug build, this is used to control the amount of debug info is logged. See the 'dvals' and 'load' script for more info on how to use this (the dvals and load scripts are provided as part of the ipw2200 development snapshot releases available from the SourceForge project at http://ipw2200.sf.net)

led

Can be used to turn on experimental LED code. 0 = Off, 1 = On. Default is 1.

mode

Can be used to set the default mode of the adapter. 0 = Managed, 1 = Ad-Hoc, 2 = Monitor

1.3. Wireless Extension Private Methods

As an interface designed to handle generic hardware, there are certain capabilities not exposed through the normal Wireless Tool interface. As such, a provision is provided for a driver to declare custom, or private, methods. The Intel(R) PRO/Wireless 2915ABG Driver for Linux defines several of these to configure various settings.

The general form of using the private wireless methods is:

```
% iwpriv $IFNAME method parameters
```

Where $IFNAME is the interface name the device is registered with (typically eth1, customized via one of the various network interface name managers, such as ifrename)

The supported private methods are:

get_mode

Can be used to report out which IEEE mode the driver is configured to support. Example:

% iwpriv eth1 get_mode eth1 get_mode:802.11bg (6)

set_mode

Can be used to configure which IEEE mode the driver will support.

Usage:

```
% iwpriv eth1 set_mode {mode}
```

Where {mode} is a number in the range 1-7:

1	802.11a (2915 only)
2	802.11b
3	802.11ab (2915 only)
4	802.11g
5	802.11ag (2915 only)
6	802.11bg
7	802.11abg (2915 only)

get_preamble

Can be used to report configuration of preamble length.

set_preamble

Can be used to set the configuration of preamble length:

Usage:

```
% iwpriv eth1 set_preamble {mode}
```

Where {mode} is one of:

1	Long preamble only
0	Auto (long or short based on connection)

1.4. Sysfs Helper Files

The Linux kernel provides a pseudo file system that can be used to access various components of the operating system. The Intel(R) PRO/Wireless 2915ABG Driver for Linux exposes several configuration parameters through this mechanism.

An entry in the sysfs can support reading and/or writing. You can typically query the contents of a sysfs entry through the use of cat, and can set the contents via echo. For example:

```
% cat /sys/bus/pci/drivers/ipw2200/debug_level
```

Will report the current debug level of the driver's logging subsystem (only available if CONFIG_IPW2200_DEBUG was configured when the driver was built).

You can set the debug level via:

```
% echo $VALUE > /sys/bus/pci/drivers/ipw2200/debug_level
```

Where $VALUE would be a number in the case of this sysfs entry. The input to sysfs files does not have to be a number. For example, the firmware loader used by hotplug utilizes sysfs entries for transferring the firmware image from user space into the driver.

The Intel(R) PRO/Wireless 2915ABG Driver for Linux exposes sysfs entries at two levels -- driver level, which apply to all instances of the driver (in the event that there are more than one device installed) and device level, which applies only to the single specific instance.

1.4.1 Driver Level Sysfs Helper Files

For the driver level files, look in /sys/bus/pci/drivers/ipw2200/

debug_level

This controls the same global as the 'debug' module parameter

1.4.2 Device Level Sysfs Helper Files

For the device level files, look in:

```
/sys/bus/pci/drivers/ipw2200/{PCI-ID}/
```

For example::

```
/sys/bus/pci/drivers/ipw2200/0000:02:01.0
```

For the device level files, see /sys/bus/pci/drivers/ipw2200:

rf_kill

read -

0	RF kill not enabled (radio on)
1	SW based RF kill active (radio off)

2	HW based RF kill active (radio off)
3	Both HW and SW RF kill active (radio off)

write -

0	If SW based RF kill active, turn the radio back on
1	If radio is on, activate SW based RF kill

> **Note**
>
> If you enable the SW based RF kill and then toggle the HW based RF kill from ON -> OFF -> ON, the radio will NOT come back on

ucode

read-only access to the ucode version number

led

read -

0	LED code disabled
1	LED code enabled

write -

0	Disable LED code
1	Enable LED code

> **Note**
>
> The LED code has been reported to hang some systems when running ifconfig and is therefore disabled by default.

1.5. Supported channels

Upon loading the Intel(R) PRO/Wireless 2915ABG Driver for Linux, a message stating the detected geography code and the number of 802.11 channels supported by the card will be displayed in the log.

The geography code corresponds to a regulatory domain as shown in the table below.

Code	Geography	Supported channels	
		802.11bg	802.11a
---	Restricted	11	0
ZZF	Custom US/Canada	11	8
ZZD	Rest of World	13	0
ZZA	Custom USA & Europe & High	11	13
ZZB	Custom NA & Europe	11	13
ZZC	Custom Japan	11	4
ZZM	Custom	11	0
ZZE	Europe	13	19
ZZJ	Custom Japan	14	4
ZZR	Rest of World	14	0
ZZH	High Band	13	4
ZZG	Custom Europe	13	4
ZZK	Europe	13	24
ZZL	Europe	11	13

2. Ad-Hoc Networking

When using a device in an Ad-Hoc network, it is useful to understand the sequence and requirements for the driver to be able to create, join, or merge networks.

The following attempts to provide enough information so that you can have a consistent experience while using the driver as a member of an Ad-Hoc network.

2.1. Joining an Ad-Hoc Network

The easiest way to get onto an Ad-Hoc network is to join one that already exists.

2.2. Creating an Ad-Hoc Network

An Ad-Hoc networks is created using the syntax of the Wireless tool.

For Example: iwconfig eth1 mode ad-hoc essid testing channel 2

2.3. Merging Ad-Hoc Networks

3. Interaction with Wireless Tools

3.1 iwconfig mode

When configuring the mode of the adapter, all run-time configured parameters are reset to the value used when the module was loaded. This includes channels, rates, ESSID, etc.

3.2 iwconfig sens

The 'iwconfig ethX sens XX' command will not set the signal sensitivity threshold, as described in iwconfig documentation, but rather the number of consecutive missed beacons that will trigger handover, i.e. roaming to another access point. At the same time, it will set the disassociation threshold to 3 times the given value.

4. About the Version Numbers

Due to the nature of open source development projects, there are frequently changes being incorporated that have not gone through a complete validation process. These changes are incorporated into development snapshot releases.

Releases are numbered with a three level scheme:

> major.minor.development

Any version where the 'development' portion is 0 (for example 1.0.0, 1.1.0, etc.) indicates a stable version that will be made available for kernel inclusion.

Any version where the 'development' portion is not a 0 (for example 1.0.1, 1.1.5, etc.) indicates a development version that is being made available for testing and cutting edge users. The stability and functionality of the development releases are not know. We make efforts to try and keep all snapshots reasonably stable, but due to the frequency of their release, and the desire to get those releases available as quickly as possible, unknown anomalies should be expected.

The major version number will be incremented when significant changes are made to the driver. Currently, there are no major changes planned.

5. Firmware installation

The driver requires a firmware image, download it and extract the files under /lib/firmware (or wherever your hotplug's firmware.agent will look for firmware files)

The firmware can be downloaded from the following URL:

> http://ipw2200.sf.net/

6. Support

For direct support of the 1.0.0 version, you can contact http://supportmail.intel.com, or you can use the open source project support.

For general information and support, go to:

> http://ipw2200.sf.net/

7. License

The full GNU General Public License is included in this distribution in the file called LICENSE.

Contact Information:

James P. Ketrenos <ipw2100-admin@linux.intel.com>

Intel Corporation, 5200 N.E. Elam Young Parkway, Hillsboro, OR 97124-6497