



SUPPORT GUIDE FOR NOKIA D211 LINUX DEVICE DRIVER

NOKIA





Contents

- 1. INTRODUCTION 1
- 2. REQUIREMENTS 1
- 3. COMPILING AND INSTALLING THE DEVICE DRIVER 1
- 4. CONFIGURING THE DEVICE DRIVER..... 2
 - 4.1 GENERAL CONFIGURATION VARIABLES 3
 - 4.2 WLAN AD HOC NETWORK 3
 - 4.3 WLAN INFRASTRUCTURE NETWORK..... 3
 - 4.4 GSM CONFIGURATION 4
- 5. USING THE DEVICE DRIVER..... 4
 - 5.1 NOKIA_CTL 5
- 6. UNINSTALLING THE DEVICE DRIVER 6
- 7. WEP KEYS 6
- 8. COMPILING THE USER INTERFACE..... 6
- 9. SIM SERVICES 7
- 10. TROUBLESHOOTING 8
- 11. GLOSSARY 8

Legal Notice

Copyright © 2002 Nokia. All rights reserved.

Reproduction, transfer, distribution or storage of part or all of the contents in this document in any form without the prior written permission of Nokia is prohibited.

Nokia and Nokia Connecting People are registered trademarks of Nokia Corporation. Other product and company names mentioned herein may be trademarks or tradenames of their respective owners.

Nokia operates a policy of continuous development. Nokia reserves the right to make changes and improvements to any of the products described in this document without prior notice.

Under no circumstances shall Nokia be responsible for any loss of data or income or any special, incidental, consequential or indirect damages howsoever caused.

The contents of this document are provided "as is". Except as required by applicable law, no warranties of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose, are made in relation to the accuracy, reliability or contents of this document. Nokia reserves the right to revise this document or withdraw it at any time without prior notice.

The availability of particular products may vary by region. Please check with the Nokia dealer nearest to you.





1. INTRODUCTION

This document is an installation and user's guide for the Linux device driver of the Nokia D211. The user should be familiar with installing and compiling device drivers in source code form.

For further information on the use of the Nokia D211, please refer to the *User's Guide* found on the product CD-ROM or visit the Nokia Web site at www.forum.nokia.com.

2. REQUIREMENTS

The Linux device driver of the Nokia D211 is provided partly in binary and partly in source code form. The code must be compiled to a loadable kernel module, which can then be inserted into a running Linux kernel. A binary userspace daemon is used for the kernel module management. The included command line utility can be used for configuring the device driver. The device driver may not work with all Linux kernel versions. See the section for tested kernel versions.



Note: Although there are Linux versions for many different architectures, the Nokia D211 device driver has been tested with x86 compatible single processor Linux only.

To compile and install the device driver, there must be kernel sources available in the system. In the 2.4 kernel series, the needed PCMCIA support is included in the kernel. A working PCMCIA support is required in order to make the device driver work. For more information on this matter, please refer to your Linux distribution documentation or the documentation provided with the PCMCIA package.

Support for /proc file system in the kernel is required due to the card initialisation scripts.



Note: The system must have kernel sources available because the device driver compilation requires them. The kernel must have /proc file system support. You should also have working PCMCIA support.

3. COMPILING AND INSTALLING THE DEVICE DRIVER

First the device driver must be unpacked.

1. Copy the driver packages to the appropriate directory (e.g. /usr/src).
2. Unpack the packages with, for example, the command “tar -zxvf *package_name*”. The device driver sources and binaries are now located in the d211 subdirectory.

After you have unpacked the device driver package, you can begin the compilation.

1. Copy *config.mk.orig* to *config.mk*.
2. Edit the *config.mk* to match your system. It is important that all the directives in *config.mk* are correct. Otherwise the device driver may not work or may not even compile.

Syntax of the file:

LINUX = /usr/src/linux	This is the location of the Linux source code
OS_RELEASE=2.4.16	Kernel version for module installation
ROOTDIR=/	Root directory for kernel module and tool installation
SMAC2=d211fw.bin	Firmware for the radio card

3. When the *config.mk* is correct, compile the device driver by issuing the command 'make all'. If the compilation succeeds, *nokia_cs.o* appears in the *src* directory. If the compilation is not successful, check again the *config.mk* for possible errors. It is possible that the device driver does not compile to your system. In that case you should use one of the setups that Nokia has used in developing the device driver.
4. If you are installing the device driver for the first time, issue the command 'make config_install' to install the default configuration. If you have previously installed the device driver and do not want to override the previous setup, you need not do this.
5. Issue the command 'make install' to install the device driver. It installs the kernel module and the userspace utilities. You may need to restart your PCMCIA system before using the device driver.

If you need to compile the device driver more than once, it is recommended that you issue the command 'make clean' before another build is made. This procedure ensures that the object files from a previous build are recompiled and no version mismatch will occur.



Note: If the device driver does not compile to your system, try one of the setups listed in the section.

4. CONFIGURING THE DEVICE DRIVER

The configuration file that needs to be edited is */etc/pcmcia/nokia_cs.opts*. It defines what the radio card should do at startup. Currently only the WLAN connection can be activated when the radio card is inserted. For GSM functionality, see chapter 4.4.



Note: GSM and WLAN functionality cannot be used at the same time.



4.1 GENERAL CONFIGURATION VARIABLES

The configuration file contains three general configuration variables. They configure the actions that are performed after the card is inserted.

```
MODE=WLAN
PROFILE=My_profile
NETWORK_START=yes
```

MODE selects the operating mode. It can be set to WLAN or GSM to enable WLAN or GSM functionality after startup. PROFILE is used to specify the used profile for the selected operating mode. If NETWORK_START is set to “yes”, the network is started after the profile startup. If it is set to “no”, no network is started. [m2]

After these general configuration variables [m3], multiple profiles can be described consecutively. Each profile starts with a profile name, which is placed in brackets, and is followed by profile parameters.

The profile can also be set manually by using the nokia_ctl utility. See chapter 5.1.

4.2 WLAN AD HOC NETWORK

For a simple ad hoc network, MODE is set to WLAN and the other configuration variables are located in the selected profile (PROFILE). For example, edit the nokia_cs.opts as follows:

```
MODE=WLAN
PROFILE=My_profile
NETWORK_START=yes
[My_profile]
mode=adhoc
ssid=<put here the desired ssid>
```

This configuration joins an existing ad hoc network with the desired ssid. If there is no ad hoc network with the desired ssid available, the configuration creates one. The profile must contain: country (0), mode (adhoc), and the desired ssid (network name, e.g. “Nokia ADHOC”).

4.3 WLAN INFRASTRUCTURE NETWORK

For a simple infrastructure network, MODE is set to WLAN and the other configuration variables are located in the selected profile (PROFILE). For example, edit the nokia_cs.opts as follows:

```
MODE=WLAN
PROFILE=Infra_profile
NETWORK_START=yes
[Infra_profile]
mode=infra
```

ssid=Nokia WLAN
shared_key1=0x123456789a

In this case the network name is Nokia WLAN and the network has one shared WEP key. The card will automatically find the best matching WLAN access point with the desired SSID and matching WEP key. The profile must contain: mode (infra) and the desired ssid (network name).

4.4 GSM CONFIGURATION



Note: When referring to GSM data or GSM connection, the information also applies to GPRS connections.

To enable a GSM connection, set MODE to GSM:

```
MODE=GSM
```

This setting alone will ensure that the WLAN connection is not established at device driver startup. GSM connection requires that you have a working PPP setup in your device. Check from your Linux distribution documentation how to install PPP support on you device.

When the radio card is inside the PCMCIA slot and WLAN connection is not established, in order to use the GSM functionality, it is necessary to first give the PIN code for the SIM card. PIN code can be given with the `nokia_ctl` program. The GSM radio also needs to be enabled, and this can be done with `nokia_ctl`, too.

The device driver creates a serial port `/dev/ttyNC0`. This serial port can be used for GSM connections. The serial port supports AT commands that are listed in the *Developer Manual for the Nokia D211* found at www.forum.nokia.com. In scripts directory there is a sample `wvdial` script for different connection types. A sample PPP options file is also in the same directory. You can copy this into your PPP configuration directory (usually `/etc/ppp`).

5. USING THE DEVICE DRIVER

If the device driver has been installed correctly, it should load automatically when the Nokia D211 is inserted into the PC card slot of the device. You can check the status from your system log messages which typically are located in `/var/log/messages`. Other means of checking the status is to give the command `'dmesg'` and reading `/proc/net/nokia_cs`.

Syntax of the information in `/proc/net/nokia_cs`:

WLAN NetDev	name of the card ethernet device
WLAN MAC	MAC address of the Nokia D211
WLAN ChrDev	name of the controlling character device
WLAN ChMajor	major number of the character device

WLAN ChMinor	minor number of the character device
GSM TTYDev	name of the GSM serial port
GSM ChMajor	major number of the serial port
GSM ChMinor	minor number of the serial port
GSM status	GSM radio status, possible states are: on, off
WLAN status	WLAN radio status, possible states are: off, initialized, on
WLAN state	WLAN state, possible values are: disabled, initialized, reset, hidden scan, scanning, joining, authenticating, associating, closed, creating BSS, infra, adhoc
WLAN BSSID	BSSID of the created or joined network
WLAN SSID	SSID of the created or joined network
WLAN Channel	radio channel used
WLAN RadioSS	radio strength of the current WLAN access point
WLAN Mode	current operating mode (infrastructure/ad hoc)

WLAN Known BSSs lists all the BSSs the device driver knows.

The syntax:

ID	sequence number
CAP	capability information as in the IEEE 802.11 specification. For example, bit0=1 means that the BSS is an infrastructure network and if 0, it is an ad hoc network.
MAC	MAC address of the WLAN access point
CH	channel
MX	maximum rate (in 500kbit/s intervals)
RSSI	radio strength of the WLAN access point (as seen by the radio card)
SSID	SSID of the network

5.1 NOKIA_CTL

Nokia_ctl is a simple user interface to control the radio card. Running nokia_ctl without any arguments will tell the supported commands. Top level commands or command classes are:

enable_wlan	Command enables WLAN
disable_wlan	Command disables WLAN
wlan_mibset	WLAN MIB write commands
wlan_mibget	WLAN MIB read commands
profile	WLAN profile

<code>enable_gsm</code>	Command gives a PIN code and enables GSM
<code>disable_gsm</code>	Command disables GSM and stops any connections
<code>gsm</code>	GSM commands

Most important command is `enable_gsm` which will ask for the PIN code and then give it to the radio card. Giving the PIN code is necessary if GSM is going to be used. One of the command classes is 'gsm'. You can give commands in the 'gsm' set by writing 'nokia_ctl gsm enableGSMradio,' for example. This command enables GSM radio access and it is also necessary if GSM data connection is desired. Also the WLAN connection can be controlled with the `nokia_ctl`: for example, it can be disabled with the 'nokia_ctl disable_wlan' command before using GSM functionality.

The WLAN profile in use can also be selected and changed with the `nokia_ctl`. When changing profiles, the command 'nokia_ctl disable_wlan' is used to disable the old profile. The new profile is selected with the command 'nokia_ctl profile <profile_name>'.

6. UNINSTALLING THE DEVICE DRIVER

Type 'make uninstall' in the main directory to uninstall the device driver. This removes everything else of the driver except the directory where the device driver was first unpacked (e.g. `/usr/src/d211`)

7. WEP KEYS

WEP keys consist of a secret key and a 24-bit Initialization Vector. For example, the 128-bit WEP key has a 104-bit secret key that the user can set, and a 24-bit Initialization Vector that cannot be controlled by the user. Many manufacturers refer to this 128-bit key as a 128-bit key, whereas others refer to it as a 104-bit key (104+24). Both keys offer the same level of encryption and are therefore interoperable.




Regardless of how the WEP keys have been named, all keys are compatible as long as their length is the same. For example, keys that are 40 bits long, always consist of 5 alphanumeric characters or 10 hexadecimal characters.

The Nokia D211 supports three key lengths: 40 (40+24), 128 (104+24), and 152 (128+24) bits. The 40-bit key is Wi-Fi (Wireless Fidelity) compatible.

8. COMPILING THE USER INTERFACE

The Nokia D211 package comes with a simple user interface for which the source code is also supplied. Normally it is not necessary to compile it since it is also supplied as a binary. If the user desires to make changes to the user interface it must be compiled. The compilation



of the user interface is also an example of how to develop software for the Nokia D211 radio card.

In order to compile the user interface, the programmer needs to obtain the C++ headers from <http://www.forum.nokia.com>. The headers can be found in the D211 SDK package and they should be copied from there to control/public/ directory in the Linux driver main directory. After that you can compile the user interface with the command 'make -f Makefile.sdk' in the user interface directory.

9. SIM SERVICES

SIM services refer to a service which enables you to access the Internet via your service provider's or network operator's own public network. Your service provider may, for example, offer you the possibility to check data from your company intranet, send and receive e-mail, and save documents. SIM services are usually available in public places such as hotels, airports, railway stations, business centres, and corporate buildings.



Note: Before you can take advantage of the SIM services, you must subscribe to these services from your service provider or network operator and obtain instructions for use.

A SIM card is used as a means for user identification: the data stored on the SIM card is read, and if valid, you are allowed to connect to the Internet and intranets. The SIM card is provided by the service provider or network operator.

Accounting data, such as used access time and/or transferred data, is passed from the network on to the service provider for billing purposes. Accounting begins when the wireless station is authenticated and ends when the wireless station logs off.



Note: The actual invoice for services from your service provider may vary, depending upon network features, rounding-off for billing, taxes and so forth.

A SIM services client has been included in the distribution but it is not installed by default. The client can be found in directory NAAP under the top directory of the distribution. To install the program, simply copy the binary nokia_ss to a directory in your PATH, e.g. /usr/bin or /usr/local/bin.

You can change the configuration by supplying a configuration file to the program as an argument, i.e. "nokia_ss /path/to/configurationfile".

The configuration file consists of *key = value* pairs as follows:

address	IP address of the interface to be used for authentication, e.g. 127.0.0.1. If not defined, the address of eth0 is used as a default.
realm	owlan.org by default



A sample configuration file "config.example" can be found in the same directory as the client.



Note: The client assumes that GSM has been enabled on the Nokia D211 radio card, i.e. the user has manually enabled it by issuing the command 'nokia_ctl enable_gsm' before running the client. See chapter 5.1 for further information on how to use the nokia_ctl utility.

10. TROUBLESHOOTING

I cannot compile the device driver.

Make sure you have the right kernel sources. The kernel must match the kernel in config.mk and the kernel as 'uname -r' reports.

If the kernels match, make sure you have the correct kernel versions (see below for more information).

I cannot load the module; it says undefined reference to some functions.

If you are sure you compiled the device driver against the correct kernel sources, then you probably have to obtain a known-to-work kernel. The developers of this device driver have used different 2.4 series kernels up to 2.4.16 with the kernel PCMCIA support. Because of the differences that may arise as the kernel is compiled, Nokia cannot guarantee that the device driver will work even with these kernels.

My PC crashes as soon as the device driver module is loaded or radio card is inserted.

There is probably some incompatibility between the device driver binary part and your system. Please try one of the recommended kernels.

I cannot make any connections although the kernel module loads correctly.

Check /proc/net/nokia_cs and kernel log messages for more information about the situation. You can try to get more debugging information by editing src/Makefile and uncommenting `OPTIONS = -DD_DEBUG`. After that you need to compile and install the device driver again. Check also that the nokia_cs userspace daemon is started after card insertion, e.g. with command 'ps aux' which shows all currently started userspace programs.

11. GLOSSARY

Ad hoc

One of the two WLAN operating modes that can be selected when using the Nokia D211. With this configuration option, users can set up a wireless network where wireless stations can send and receive data directly with each other without WLAN access points. This type of network is sometimes called a peer-to-peer network.

BSSID	Basic Service Set Identifier. A six-byte address that distinguishes one WLAN access point from another. Sometimes called <i>network ID</i> .
Channel	A specified frequency band for the transmission and reception of signals.
GSM	Global System for Mobile Communications. GSM is a digital telecommunications system used widely in Europe, Asia and the Pacific.
GPRS	General Packet Radio Service. GPRS is a technology that enables sending and receiving data over a mobile network. GPRS as such is a data bearer that enables wireless access to data networks like the Internet.
Infrastructure	One of the two WLAN operating modes that can be selected when using the Nokia D211. With this configuration option users can set up a network where wireless stations communicate with wired and wireless stations through a WLAN access point.
MAC	Media Access Control. A protocol that governs access to a shared transmission medium, such as a wireless LAN. In a local area network, MAC address is the computer's unique hardware address.
MIB	Management Information Base.
PCMCIA	Personal Computer Memory Card International Association. Standard architecture-independent extension device. Also called PC card.
Profile	Feature unique to the Nokia D211. A profile is a collection of settings needed establish a network connection.
RSSI	Received Signal Strength Indication.
SIM	Subscriber Identity Module. SIM card is a small plastic card with an embedded integrated circuit. It contains all the information the cellular network needs to identify the network user. The SIM card also contains security-related information.
SSID	Service Set Identifier.
WEP	Wired Equivalent Privacy. A security feature using the RC4 algorithm that performs wireless data encryption. The WEP algorithm uses up to 152-bit keys.
WLAN	A local area network in which radio links are used to connect devices instead of physical cables.
WLAN access point	Physical device that connects wired and wireless local area networks together.