

Wireless network configuration (Русский)



Эта страница нуждается в сопроводителе



Статья не гарантирует актуальность информации. Помогите русскоязычному сообществу поддержкой подобных страниц. См. [Команда переводчиков ArchWiki](#)

Ссылки по теме

- [Настройка сети](#)
- [Программная точка доступа](#)
- [Сеть ad-hoc](#)
- [Раздача интернета](#)

Настройка беспроводного соединения в Archlinux (или в любом другом Linux) состоит из 2-х частей. Первая часть это определение и установка правильного драйвера для вашего устройства (обычно они есть на установочном носителе, но устанавливаются вручную). Вторая - выбор метода управления беспроводным соединением. Эта статья описывает обе части, и содержит необходимые ссылки на утилиты управления беспроводными соединениями.

Contents

[hide]

- [1 Драйвер устройства](#)
 - [1.1 Проверка состояния драйвера](#)
 - [1.2 Установка драйвера/прошивки](#)
- [2 Управление беспроводными соединениями](#)
 - [2.1 Ручная настройка](#)
 - [2.1.1 Получение некоторой полезной информации](#)
 - [2.1.2 Активация интерфейса](#)
 - [2.1.3 Поиск точки доступа](#)
 - [2.1.4 Режим работы](#)
 - [2.1.5 Привязка](#)
 - [2.1.6 Получение IP-адреса](#)
 - [2.1.7 Собственные стартовые скрипты/службы](#)
 - [2.1.7.1 Ручное беспроводное подключение при загрузке при помощи systemd и dhcpcd](#)
 - [2.1.7.2 Systemd с wpa_supplicant и статическим IP](#)
 - [2.2 Автоматическая настройка](#)
 - [2.2.1 Connman](#)
 - [2.2.2 netctl](#)
 - [2.2.3 Wicd](#)
 - [2.2.4 NetworkManager](#)
 - [2.2.5 WiFi Radar](#)
- [3 Решение проблем](#)
 - [3.1 Предостережения Rfkill](#)
 - [3.2 Уважение управляющего домена](#)
 - [3.3 Просмотр логов](#)
 - [3.4 Энергосбережение](#)
 - [3.5 Не удалось получить IP адрес](#)
 - [3.6 Connection always times out](#)
 - [3.6.1 Lowering the rate](#)
 - [3.6.2 Понижение txpower](#)
 - [3.6.3 Setting rts and fragmentation thresholds](#)
 - [3.7 Внезапные отключения](#)

- [3.7.1Причина #1](#)
 - [3.7.2Причина #2](#)
 - [3.7.3Причина #3](#)
- [4Решение проблем с драйверами и прошивками](#)
 - [4.1wlan-ng](#)
 - [4.2rt2x00](#)
 - [4.3RT2500](#)
 - [4.4RT61](#)
 - [4.5RT73](#)
 - [4.6madwifi](#)
 - [4.7ath5k](#)
 - [4.8ath9k](#)
 - [4.9rtl8723bu](#)
 - [4.10ipw2100 and ipw2200](#)
 - [4.11ipw3945 and ipw4965](#)
 - [4.12ipw3945 \(Альтернативный метод\)](#)
 - [4.13orinoco](#)
 - [4.14ndiswrapper](#)
 - [4.15prism54](#)
 - [4.16ACX100/111](#)
 - [4.17BCM43XX](#)
 - [4.18b43](#)
 - [4.19rtl8187](#)
 - [4.20zd1211rw](#)
 - [4.21Тестирование установки](#)
- [5Смотрите также](#)

Драйвер устройства

По-умолчанию ядро Arch Linux *модульное*, это означает, что многие драйверы для компьютера расположены на жёстком диске и доступны как [модули](#). При загрузке [udev](#) составляет опись вашего оборудования и загружает соответствующие модули (драйверы) для конкретного оборудования, что позволит создать сетевой *интерфейс*.

Некоторые беспроводные чипсеты в дополнение к нужному драйверу также требуют прошивку. Многие образы прошивок присутствуют в пакете [linux-firmware](#) и установлены по умолчанию, однако, проприетарные модули прошивок не включены и должны быть установлены отдельно. Это описано в разделе [#Установка драйвера/прошивки](#).

Примечание: Udev не идеален. Если нужный модуль не был загружен udev'ом при старте системы, просто [загрузите его вручную](#) [[broken link: invalid section](#)]. Также заметьте, что иногда udev может загружать более одного драйвера для устройства, что приведёт к конфликту и помешает успешному конфигурированию. Убедитесь, что вы [запретили загрузку](#) [[broken link: invalid section](#)] ненужного модуля

Совет: Это не обязательно, но лучше сперва установить пользовательские инструменты, описанные в разделе [#Ручная настройка](#), особенно если ожидается возникновение какой-либо проблемы

Проверка состояния драйвера

Чтобы проверить загрузился ли драйвер вашей сетевой карты, посмотрите на вывод команд `lspci -k` или `lsusb -v` в зависимости от того, подключена ли карта по PCI(e) или USB. Вы должны увидеть что используются некоторые драйверы ядра, например:

```
$ lspci -k
```

```
06:00.0 Network controller: Intel Corporation WiFi Link 5100
        Subsystem: Intel Corporation WiFi Link 5100 AGN
```

```
Kernel driver in use: iwlwifi
Kernel modules: iwlwifi
```

Примечание: Если ваша карта является USB-устройством, выполнение `dmesg | grep usbcore` должно выдать что-то похожее на `usbcore: registered new interface driver rtl8187` в выводе.

Также проверьте вывод команды `ip link`, чтобы убедиться, что сетевой интерфейс (**обычно** название начинается с буквы "w", например, `wlp2s1`) был создан. Затем поднимите интерфейс командой `ip link set интерфейс up`. Например, если интерфейс называется `wlan0`:

```
# ip link set wlan0 up
```

Если вы получаете такое сообщение об ошибке: `SIOCSIFFLAGS: Нет такого файла или каталога` (No such file or directory), это несомненно означает что ваш беспроводной чипсет требует прошивки для функционирования.

Проверьте сообщения ядра насчёт загрузки прошивки:

```
$ dmesg | grep firmware
```

```
[ 7.148259] iwlwifi 0000:02:00.0: loaded firmware version 39.30.4.1 build
35138 op_mode iwldvm
```

Если там нет интересующей вас информации, проверьте сообщения подробного вывода, относящихся к модулю, который вы определили ранее (в данном примере `iwlwifi`), чтобы найти интересное сообщение или дальнейшие ошибки:

```
$ dmesg | grep iwlwifi
```

```
[ 12.342694] iwlwifi 0000:02:00.0: irq 44 for MSI/MSI-X
[ 12.353466] iwlwifi 0000:02:00.0: loaded firmware version 39.31.5.1 build
35138 op_mode iwldvm
[ 12.430317] iwlwifi 0000:02:00.0: CONFIG_IWLWIFI_DEBUG disabled
...
[ 12.430341] iwlwifi 0000:02:00.0: Detected Intel(R) Corporation WiFi Link
5100 AGN, REV=0x6B
```

Если модуль ядра загружен успешно и интерфейс поднялся, можете пропустить следующий раздел.

Установка драйвера/прошивки

Проверьте в следующих списках, поддерживается ли ваше устройство (вы можете узнать, какая у вас карточка, по выводу `hwdetect --show-net` или `lshwd`):

- В [Ubuntu Wiki](#) есть хороший список беспроводных карт и информация об их поддержке ядром Linux или же user-space драйвером (включая название драйвера).
- Проверьте на [Linux Wireless Support](#) ваше устройство или на The Linux Questions' [hardware compatibility list](#) (HCL), которое также содержит список поддерживаемого ядром оборудования.

- Также на [странице ядра](#) есть таблица поддерживаемого оборудования.

Если ваша беспроводная карта есть в списке выше, перейдите в раздел [#Решение проблем с драйверами и прошивками](#) этой статьи, в которой содержатся инструкции по установке драйверов и прошивок на некоторые экзотические беспроводные карты. Затем [проверьте состояние драйвера](#) снова.

Если вашей беспроводной карты нет в списках, возможно она поддерживается только в Windows (некоторые Broadcom, Zcom и т.д.). Для них вы можете попробовать использовать [#ndiswrapper](#).

Управление беспроводными соединениями

Допустим, что ваш драйвер найден и прекрасно работает, вам необходимо выбрать метод управления беспроводными соединениями. Следующая подсекция поможет вам найти подходящий метод работы.

Arch Linux обладает несколькими решениями для управления беспроводными соединениями. Процедура и необходимые инструменты зависят от нескольких факторов:

- От личных предпочтений в конфигурировании; начиная от полностью ручного метода через командную строку и заканчивая автоматическими решениями с графическими оболочками.
- От типа шифрования (или его отсутствия), защищающего беспроводную сеть.
- От нужности сетевых профайлов, если компьютер часто будет менять сети (например, на ноутбуке).

Совет:

- Каков бы ни был ваш выбор, **сначала вы должны попытаться подключиться через ручной способ**. Это позволит вам понять различные действия, которые требуется выполнить и устранить возможные проблемы.
- По возможности (если вы владеете вашей Wi-Fi точкой доступа), попробуйте подключиться без шифрования, чтобы проверить, что всё работает. Затем попробуйте включить шифрование, либо WEP (легко в настройке, но элементарно взламывается), WPA или WPA2.

Эта таблица показывает различные способы активации и управления беспроводным соединением, в зависимости от шифрования и типа управления и требуемых утилит. Могут быть и другие способы, но эти используются чаще всего:

Способ управления	Активация интерфейса	Управление беспроводным соединением (/=alternatives)	Назначение IP адреса (/=alternatives)
Управляется вручную , без шифрования или с WEP шифрованием	ip	iw / iwconfig	ip / dhcpcd / dhclient
Управляется вручную , с WPA или WPA2 PSK шифрованием	ip	iw / iwconfig + wpa_supplicant	ip / dhcpcd / dhclient
Управляется автоматически , с поддержкой сетевых	netctl , Wicd , NetworkManager , и т.д. Эти утилиты докачивают необходимые зависимости из списка пакетов в		

профилей	ручной настройке.
----------	-------------------

Ручная настройка

Беспроводные интерфейсы, также как и другие сетевые интерфейсы, контролируются с помощью *ip* из пакета [iproute2](#).

Вы должны установить базовый набор утилит для управления беспроводными соединениями, либо:

- [iw](#) - текущий nl80211 стандарт, но не все модули беспроводных чипов поддерживают iw
либо
- [wireless_tools](#) - уже устарел, но до сих пор широко поддерживается.

А для WPA/WPA2 шифрования вам ещё понадобится:

- [wpa_supplicant](#)

Эта таблица показывает аналоги команд для *iw* и *wireless_tools* (см. ещё примеры на [\[1\]](#)). Эти утилиты пользовательского окружения работают чрезвычайно хорошо и позволяют полностью контролировать беспроводные соединения.

Примечание:

- Примеры в этом разделе предполагают, что ваш беспроводной интерфейс называется *wlan0* и вы подключаетесь к точке доступа под названием *ваш_essid*. Заменяйте их на свои значения
- Заметьте, что большинство этих команд должно быть запущено с [правами суперпользователя](#). Исполнение некоторых команд с правами обычного пользователя (напр. *iwlist*), завершится без ошибок, однако не выполнит правильный вывод, что может сбить вас с толку

Команда утилиты <i>iw</i>	Команда утилиты <i>wireless_tools</i>	Описание
<i>iw dev wlan0 link</i>	<i>iwconfig wlan0</i>	Получение статуса соединения
<i>iw dev wlan0 scan</i>	<i>iwlist wlan0 scan</i>	Сканирование доступных точек доступа
<i>iw dev wlan0 set type ibss</i>	<i>iwconfig wlan0 mode ad-hoc</i>	Установка режима <i>ad-hoc</i> .
<i>iw dev wlan0 connect ваш_essid</i>	<i>iwconfig wlan0 essid ваш_essid</i>	Подключение к открытой сети
<i>iw dev wlan0 connect ваш_essid 2432</i>	<i>iwconfig wlan0 essid ваш_essid freq 2432M</i>	Подключение к открытой сети, с уточнением канала

iw dev wlan0 connect <i>ваш_essid</i> key 0: <i>ваш_ключ</i>	iwconfig wlan0 essid <i>ваш_essid</i> key <i>ваш_ключ</i>	Подключение к сети с WEP шифрованием с использованием шестнадцатеричного ключа
iw dev wlan0 connect <i>ваш_essid</i> key 0: <i>ваш_ключ</i>	iwconfig wlan0 essid <i>ваш_essid</i> key s: <i>ваш_ключ</i>	Подключение к сети с WEP шифрованием с использованием ASCII ключа
iw dev wlan0 set power_save on	iwconfig wlan0 power on	Включение режима энергосбережения

Примечание: В зависимости от вашего оборудования и метода шифрования, некоторые из этих шагов могут не требоваться. Некоторым картам требуется активация интерфейса и/или сканирование точек доступа перед тем, как они смогут подключиться к точке доступа и получить IP адрес. Может потребоваться поэкспериментировать. Например, WPA/WPA2 пользователи могут попробовать непосредственно активировать свою беспроводную сеть с шага [#Привязка](#)

Получение некоторой полезной информации

Совет: Для просмотра большего количества примеров использования утилиты *iw* обратитесь к [официальной документации](#)

- Первым делом вы должны узнать название беспроводного интерфейса. Вы можете сделать это выполнив команду:

```
$ iw dev

phy#0
    Interface wlan0
        ifindex 3
        wdev 0x1
        addr 12:34:56:78:9a:bc
        type managed
        channel 1 (2412 MHz), width: 40 MHz, center1: 2422 MHz
```

- Чтобы проверить статус соединения используйте следующую команду. Пример вывода когда нет подключения к точке доступа:

```
$ iw dev wlan0 link

Not connected.
```

Когда вы подключены к точке доступа, вы увидите что-то вроде этого:

```
$ iw dev wlan0 link

Connected to 12:34:56:78:9a:bc (on wlan0)
    SSID: MyESSID
    freq: 2412
```

```
RX: 33016518 bytes (152703 packets)
TX: 2024638 bytes (11477 packets)
signal: -53 dBm
tx bitrate: 150.0 MBit/s MCS 7 40MHz short GI

bss flags:      short-preamble short-slot-time
dtim period:    1
beacon int:     100
```

- Статистическую информацию (такую как количество бит tx/rx, сила сигнала и т.д.) можно получить, введя команду:

```
$ iw dev wlan0 station dump

Station 12:34:56:78:9a:bc (on wlan0)
    inactive time: 1450 ms
    rx bytes:      24668671
    rx packets:    114373
    tx bytes:      1606991
    tx packets:    8557
    tx retries:    623
    tx failed:     1425
    signal:        -52 dBm
    signal avg:    -53 dBm
    tx bitrate:    150.0 MBit/s MCS 7 40MHz short GI
    authorized:    yes
    authenticated: yes
    preamble:      long
    WMM/WME:       yes
    MFP:           no
    TDLS peer:     no
```

Активация интерфейса

Совет: В большинстве случаев выполнять эти действия не требуется

Некоторые карты требуют чтобы ядерный интерфейс был активирован до того, как вы сможете воспользоваться *iw* или *wireless_tools*:

```
# ip link set wlan0 up
```

Примечание: Если вы получаете ошибки вида `RTNETLINK answers: Operation not possible due to RF-kill`, убедитесь, что аппаратный переключатель находится в положении *on*. Для получения дополнительной информации смотрите раздел [#Предостережения Rfkill](#)

Чтобы удостовериться, что интерфейс поднят, проверьте вывод следующей команды:

```
# ip link show wlan0
```

```
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state
DOWN mode DORMANT group default qlen 1000
    link/ether 12:34:56:78:9a:bc brd ff:ff:ff:ff:ff:ff
```

О том, что интерфейс поднят говорит надпись `UP` в `<BROADCAST,MULTICAST,UP,LOWER_UP>`, а не надпись `state DOWN`.

Поиск точки доступа

Посмотреть какие точки доступа доступны:

```
# iw dev wlan0 scan | less
```

Примечание: Если вы увидите сообщение `Интерфейс не поддерживает сканирование` (Interface doesn't support scanning), значит, вы, наверное, забыли установить прошивку. В некоторых случаях это сообщение отображается, когда `iw` запущен не от имени суперпользователя

Совет: В зависимости от вашего местоположения, вам может понадобиться установить правильное имя [управляющего домена](#) чтобы надлежащим образом увидеть все доступные сети.

Обратите внимание на следующие поля:

- **SSID:** название сети
- **Signal:** измеряется в беспроводных единицах мощности dbm (например, от -100 до 0). Чем это отрицательное число больше (ближе к нулю), тем сигнал лучше. Наблюдая за отображаемой мощностью можно получить представление о покрытии вашей сети.
- **Security:** не сообщается прямо; проверьте строку, начинающуюся с `capability`. Если там `Privacy`, например, `capability: ESS Privacy ShortSlotTime (0x0411)`, значит сеть как-то защищена.
 - Если вы видите информационный блок `RSN`, значит сеть защищена протоколом [Robust Security Network](#), также известным под именем WPA2
 - Если вы видите информационный блок `WPA`, значит сеть защищена протоколом [Wi-Fi Protected Access](#)
 - В блоках `RSN` и `WPA` вы можете найти следующую информацию:
 - **Group cipher:** принимает значения TKIP, CCMP, both, others
 - **Pairwise ciphers:** принимает значения TKIP, CCMP, both, others. Не обязательно такое же значение, как в Group cipher
 - **Authentication suites:** принимает значения PSK, 802.1x, others. Для домашнего роутера вы обычно увидите PSK (то есть пароль). В университетах вы скорее всего увидите 802.1x, что требует логин и пароль. Тогда вам нужно узнать какое используется управление ключами (например, EAP) и какую инкапсуляцию оно использует (например, PEAP). Ищите более подробную информацию в статье [Протокол аутентификации](#) и статьях, связанных с ней
 - Если вы не видите ни `RSN`, ни `WPA` блоки, но есть `Privacy`, значит используется WEP.

Режим работы

Совет: Выполнять эти действия не обязательно, но они могут быть необходимы

На этом шаге вы должны задать подходящий режим работы беспроводной карты. Например, если вы собираетесь подключить [ad-hoc сеть](#), вы должны установить режим `ibss`:


```
# iw dev wlan0 set type ibss
```

Примечание: На некоторых картах изменение режима работы может потребовать *опустить* беспроводной интерфейс (`ip link set wlan0 down`).

Привязка

В зависимости от типа шифрования вы должны привязать своё беспроводное устройство к точке доступа, а также передать ключ шифрования:

- **Без шифрования**

```
# iw dev wlan0 connect ваш_essid
```

- **WEP**

- используя шестнадцатеричный или ASCII ключ (формат определяется автоматически, так как WEP ключ имеет фиксированную длину):

```
# iw dev wlan0 connect ваш_essid key 0:ваш_ключ
```

- используя шестнадцатеричный или ASCII ключ, определяя третий набор ключей по умолчанию (ключи считаются с нуля, всего возможно до четырёх ключей):

```
# iw dev wlan0 connect ваш_essid key d:2:ваш_ключ
```

- **WPA/WPA2**

В зависимости от того, что вы получили в [#Поиск точки доступа](#), выполните эту команду:

```
# wpa_supplicant -i wlan0 -c <(wpa_passphrase ваш_SSID ваш_ключ)
```

Это зависит от того, использует ли ваше устройство `wext` драйвер. Если это не работает, вы должны подобрать следующие опции. Если подключение прошло успешно, продолжите в новом терминале (или завершите `wpa_supplicant` с помощью `Ctrl+C`, добавив при этом опцию `-B` к предыдущей команде, чтобы выполнять её в фоновом режиме). [WPA supplicant](#) содержит более подробную информацию об опциях и о том, как создать постоянный конфигурационный файл для беспроводной точки доступа.

Вне зависимости от использованного метода, вы можете проверить, удалось ли подключиться:

```
# iw dev wlan0 link
```

Получение IP-адреса

Примечание: Для просмотра дополнительных примеров обратитесь к разделу [Настройка сети#Настройка IP-адреса](#). Информация, представленная здесь, идентична

Наконец, предоставьте IP адрес сетевому интерфейсу. Вот простые примеры:

```
# dhcpcd wlan0
```

или

```
# dhclient wlan0
```

для DHCP, или

```
# ip addr add 192.168.0.2/24 dev wlan0
# ip route add default via 192.168.0.1
```

для статической IP адресации.

Совет: В [dhcpcd](#) содержится хук (включенный по умолчанию) для беспроводных интерфейсов, который автоматически запускает [WPA supplicant](#). Он запускается, только если существует файл `/etc/wpa_supplicant/wpa_supplicant.conf` и нет прослушивающего `wpa_supplicant` процесса на этом интерфейсе. В большинстве случаев вам не надо создавать каких-либо [собственных служб](#), просто включите `dhcpcd@интерфейс`

Собственные стартовые скрипты/службы

Несмотря на то, что ручной способ настройки поможет решить проблемы беспроводных подключений, вам нужно будет перенабирать каждую команду после каждой перезагрузки. Вы, конечно, можете быстренько написать shell script, чтобы автоматизировать этот процесс, что, кстати говоря, вполне подходит для управления сетевыми соединениями, оставляя полный контроль над вашей конфигурацией. Но здесь вы можете найти более правильные примеры.

Ручное беспроводное подключение при загрузке при помощи systemd и dhcpcd

В этом примере для запуска используется [systemd](#), для соединения - [WPA supplicant](#), а для получения IP-адреса - [dhcpcd](#).

Примечание: Убедитесь, что у вас установлен пакет [wpa supplicant](#), и создайте файл `/etc/wpa_supplicant/wpa_supplicant.conf`. Для получения дополнительной информации смотрите статью [WPA supplicant \(Русский\)](#)

Создайте юнит `systemd`, например, `/etc/systemd/system/network-wireless@.service`:

```
/etc/systemd/system/network-wireless@.service

[Unit]
Description=Wireless network connectivity (%i)
Wants=network.target
Before=network.target
BindsTo=sys-subsystem-net-devices-%i.device
After=sys-subsystem-net-devices-%i.device

[Service]
Type=oneshot
RemainAfterExit=yes
```

```
ExecStart=/usr/bin/ip link set dev %i up
ExecStart=/usr/bin/wpa_supplicant -B -i %i -c
/etc/wpa_supplicant/wpa_supplicant.conf
ExecStart=/usr/bin/dhccpd %i

ExecStop=/usr/bin/ip link set dev %i down

[Install]
WantedBy=multi-user.target
```

Запустите и/или включите юнит, как это описано в разделе [systemd \(Русский\)#Использование юнитов](#), не забыв при этом указать название интерфейса:

```
# systemctl enable network-wireless@wlan0.service
# systemctl start network-wireless@wlan0.service
```

Systemd с wpa_supplicant и статическим IP

Примечание: Убедитесь, что у вас установлен пакет [wpa_supplicant](#), и создайте файл `/etc/wpa_supplicant/wpa_supplicant.conf`. Для получения дополнительной информации смотрите статью [WPA supplicant \(Русский\)](#)

Сначала создайте конфигурационный файл для службы [systemd](#), заменив *интерфейс* на имя вашего интерфейса:

```
/etc/conf.d/network-wireless@интерфейс

address=192.168.0.10
netmask=24
broadcast=192.168.0.255
gateway=192.168.0.1
```

Создайте файл юнита *systemd*:

```
/etc/systemd/system/network-wireless@.service

[Unit]
Description=Wireless network connectivity (%i)
Wants=network.target
Before=network.target
BindsTo=sys-subsystem-net-devices-%i.device
After=sys-subsystem-net-devices-%i.device

[Service]
Type=oneshot
RemainAfterExit=yes
EnvironmentFile=/etc/conf.d/network-wireless@%i
```

```

ExecStart=/usr/bin/ip link set dev %i up
ExecStart=/usr/bin/wpa_supplicant -B -i %i -c
/etc/wpa_supplicant/wpa_supplicant.conf
ExecStart=/usr/bin/ip addr add ${address}/${netmask} broadcast
${broadcast} dev %i
ExecStart=/usr/bin/ip route add default via ${gateway}

ExecStop=/usr/bin/ip addr flush dev %i
ExecStop=/usr/bin/ip link set dev %i down

[Install]
WantedBy=multi-user.target

```

Включите его в автозагрузку и запустите, указав имя интерфейса:

```

# systemctl enable network-wireless@wlan0.service
# systemctl start network-wireless@wlan0.service

```

Автоматическая настройка

Существует несколько вариантов, которые вы можете выбрать, но учтите, что все они взаимно исключаемые. Вы не должны запускать два демона одновременно. Эта таблица сравнивает разных менеджеров соединений, дополнительные сведения в субсекциях ниже.

Менеджер подключения	Поддержка сетевых профилей	Роуминг (автоподключение упало или изменилось местоположение)	Поддержка PPP (например, 3G-модемов)	Официальный графический интерфейс	Консольные утилиты
ConnMan	Да	Да	Да	Нет	connmanctl
netctl	Да	Да	Да	Нет	netctl, wifi-menu
NetworkManager	Да	Да	Да	Да	nmcli
Wicd	Да	Да	Нет	Да	wicd-curses

Connman

ConnMan - это альтернатива *NetworkManager* и *Wicd*, разработанная так, чтобы быть нетребовательной к ресурсам, что делает ее идеальной для нетбуков и других мобильных устройств. Он модульный, что даёт преимущество перед dbus API и предоставляет требуемую абстракцию над *wpa_supplicant*.

Смотрите статью [ConnMan](#).

netctl

netctl - это замена *netcfg*, созданная для работы совместно с *systemd*. Он использует настройку, основанную на профилях, и имеет возможности обнаружения и подключения к широкому кругу типов сетей. Использовать его не сложнее, чем графические инструменты.

Смотрите статью [netctl](#).

Wicd

Wicd - это сетевой менеджер, способный управлять как беспроводными, так и проводными подключениями. Он написан на Python и Gtk и требует меньше количество зависимостей, чем *NetworkManager*, что делает его идеальным решением для пользователей легковесных окружений.

Смотрите статью [Wicd](#).

Примечание: В случае использования некоторых драйверов [wicd](#) может вызывать частые разрывы соединения, тогда как [NetworkManager](#), возможно, будет работать лучше

NetworkManager

NetworkManager - это улучшенный инструмент управления сетью, который включен по умолчанию в большинство популярных дистрибутивов GNU/Linux. В дополнение к управлению проводными соединениями, *NetworkManager* предоставляет простой и беззаботный способ управления беспроводными подключениями при помощи лёгкой в использовании GUI-программы для выбора нужной сети.

Смотрите статью [NetworkManager \(Русский\)](#).

Примечание: [network-manager-applet](#) из окружения GNOME также работает в [Xfce](#), если сперва установить пакет [xfce4-xfapplet-plugin](#)^{aur}^{[ссылка](#)}
[недействительна](#): сохранено в [aur-mirror](#). Также существуют апплеты для [KDE](#)

WiFi Radar

WiFi Radar - это утилита управления беспроводными (и **только** беспроводными) профилями, написанная на Python/PyGTK2. Она позволяет осуществлять сканирование на наличие доступных сетей и создавать для них профили.

Смотрите статью [Wifi Radar](#).

Решение проблем



Эта статья или раздел нуждается в [перевode](#)



Примечания: Английский текст в секциях (обсуждение: [Talk:Wireless network configuration \(Русский\)#](#))

В этом разделе содержатся основные рекомендации по решению проблем, не связанных непосредственно с драйверами и прошивками. Для получения такой информации смотрите следующий раздел [#Решение проблем с драйверами и прошивками](#).

Предостережения Rfkill

Многие лэптопы имеют аппаратный переключатель (или кнопку) для выключения беспроводной карты, однако, она может быть также заблокирована и ядром. Этим можно управлять через `rkill`. Используйте `rfkill`, чтобы посмотреть текущий статус:

```
# rfkill list

0: phy0: Wireless LAN
    Soft blocked: yes
    Hard blocked: yes
```

Если карта *заблокирована аппаратно*, используйте переключатель (кнопку), чтобы разблокировать её. Если же карта заблокирована не *аппаратно*, а *программно*, используйте следующую команду:

```
# rfkill unblock wifi
```

Примечание: Возможно, при нажатии аппаратной кнопки карта из состояния *hard-blocked* и *soft-unblocked* перейдёт в состояние *hard-unblocked* и *soft-blocked* (i.e. the *soft-blocked* bit is just switched no matter what). Это можно исправить, отрегулировав некоторые опции [модуля ядра rfkill](#)

Дополнительная информация: <http://askubuntu.com/questions/62166/siocsiflags-operation-not-possible-due-to-rf-kill>

Уважение управляющего домена

The [regulatory domain](#), or "regdomain", is used to reconfigure wireless drivers to make sure that wireless hardware usage complies with local laws set by the FCC, ETSI and other organizations. Regdomains use [ISO 3166-1 alpha-2 country codes](#). For example, the regdomain of the United States would be "US", China would be "CN", etc.

Regdomains affect the availability of wireless channels. In the 2.4GHz band, the allowed channels are 1-11 for the US, 1-14 for Japan, and 1-13 for most of the rest of the world. In the 5GHz band, the rules for allowed channels are much more complex. In either case, consult [this list of WLAN channels](#) for more detailed information.

Regdomains also affect the limit on the [effective isotropic radiated power \(EIRP\)](#) from wireless devices. This is derived from transmit power/"tx power", and is measured in [dBm/mBm \(1dBm=100mBm\) or mW \(log scale\)](#). In the 2.4GHz band, the maximum is 30dBm in the US and Canada, 20dBm in most of Europe, and 20dB-30dBm for the rest of the world. In the 5GHz band, maximums are usually lower. Consult the [wireless-regdb](#) for more detailed information (EIRP dBm values are in the second set of brackets for each line).

Misconfiguring the regdomain can be useful - for example, by allowing use of an unused channel when other channels are crowded, or by allowing an increase in tx power to widen transmitter range. However, **this is not recommended** as it could break local laws and cause interference with other radio devices.

To configure the regdomain, install [crda](#) and [wireless-regdb](#) and reboot (to reload the `cfg80211` module and all related drivers). Check the boot log to make sure that CRDA is being called by `cfg80211`:

```
$ dmesg | grep cfg80211
```

The current regdomain can be set to the United States with:

```
# iw reg set US
```

And queried with:

```
$ iw reg get
```

Note: Your device may be set to country "00", which is the "world regulatory domain" and contains generic settings. If this cannot be unset, CRDA may be misconfigured.

However, setting the regdomain may not alter your settings. Some devices have a regdomain set in firmware/EEPROM, which dictates the limits of the device, meaning that setting regdomain in software **can only increase restrictions**, not decrease them. For example, a CN device could be set in software to the US regdomain, but because CN has an EIRP maximum of 20dBm, the device will not be able to transmit at the US maximum of 30dBm.

For example, to see if the regdomain is being set in firmware for an Atheros device:

```
$ dmesg | grep ath:
```

For other chipsets, it may help to search for "EEPROM", "regdomain", or simply the name of the device driver.

To see if your regdomain change has been successful, and to query the number of available channels and their allowed transmit power:

```
$ iw list | grep -A 15 Frequencies:
```

A more permanent configuration of the regdomain can be achieved through editing `/etc/conf.d/wireless-regdom` and uncommenting the appropriate domain. `wpa_supplicant` can also use a regdomain in the `country=` line of `/etc/wpa_supplicant.conf`.

It is also possible to configure the [cfg80211](#) kernel module to use a specific regdomain by adding, for example, `options cfg80211 ieee80211_regdom=EU` to `/etc/modprobe.d/modprobe.conf`. However, this is part of the [old regulatory implementation](#).

For further information, read the [wireless.kernel.org regulatory documentation](https://wireless.kernel.org/regulatory_documentation).

Просмотр логов

A good first measure to troubleshoot is to analyze the system's logfiles first. In order not to manually parse through them all, it can help to open a second terminal/console window and watch the kernels messages with

```
$ dmesg -w
```

while performing the action, e.g. the wireless association attempt.

When using a tool for network management, the same can be done for systemd with

```
# journalctl -f
```

Frequently a wireless error is accompanied by a deauthentication with a particular reason code, for example:

```
wlan0: deauthenticating from XX:XX:XX:XX:XX:XX by local choice
(reason=3)
```

Looking up [the reason code](#) might give a first hint.

The individual tools used in this article further provide options for more detailed debugging output, which can be used in a second step of the analysis, if required.

Энергосбережение

Смотрите раздел [Энергосбережение#Сетевые интерфейсы](#).

Не удалось получить IP адрес

- Если получение IP адреса неоднократно не удаётся при использовании клиента по умолчанию [dhcpcd](#), попробуйте установить и использовать [dhclient](#) вместо него. Не забудьте выбрать *dhclient* как первичный DHCP клиент в вашем [менеджере соединений](#)!
- Если вы можете получить IP для проводного интерфейса, но не можете для беспроводного, попробуйте отключить энергосберегающие функции вашей беспроводной карты:

```
# iwconfig wlan0 power off
```

- Если вы получаете timeout ошибку из-за *waiting for carrier* проблемы, возможно вам понадобится установить канал в `auto` для конкретного устройства:

```
# iwconfig wlan0 channel auto
```

Перед тем как изменить канал на автоматический, убедитесь что вы опустили беспроводной интерфейс. После того, как поменяете канал, можете опять поднять интерфейс.

Connection always times out

The driver may suffer from a lot of tx excessive retries and invalid misc errors for some unknown reason, resulting in a lot of packet loss and keep disconnecting, sometimes instantly. Following tips might be helpful.

Lowering the rate

Try setting lower rate, for example 5.5M:

```
# iwconfig wlan0 rate 5.5M auto
```

Fixed option should ensure that the driver does not change the rate on its own, thus making the connection a bit more stable:

```
# iwconfig wlan0 rate 5.5M fixed
```

Понижение txpower

Вы можете попробовать понизить мощность передатчика. Это может сберечь энергию:

```
# iwconfig wlan0 txpower 5
```

Установите значение от 0 до 20, `auto` или `off`.

Setting rts and fragmentation thresholds

Default `iwconfig` options have rts and fragmentation thresholds off. These options are particularly useful when there are many adjacent APs or in a noisy environment.

The minimum value for fragmentation value is 256 and maximum is 2346. In many windows drivers the maximum is the default value:

```
# iwconfig wlan0 frag 2346
```

For rts minimum is 0, maximum is 2347. Once again windows drivers often use maximum as the default:

```
# iwconfig wlan0 rts 2347
```

Внезапные отключения

Причина #1

If `dmesg` says `wlan0: deauthenticating from MAC by local choice (reason=3)` and you lose your Wi-Fi connection, it is likely that you have a bit too aggressive power-saving on your Wi-Fi card [2]. Try disabling the wireless card's power-saving features:

```
# iwconfig wlan0 power off
```

See [Power saving](#) for tips on how to make it permanent (just specify `off` instead of `on`).

If your card does not support `iwconfig wlan0 power off`, check the **BIOS** for power management options. Disabling PCI-Express power management in the BIOS of a Lenovo W520 resolved this issue.

Причина #2

If you are experiencing frequent disconnections and `dmesg` shows messages such as `ieee80211 phy0: wlan0: No probe response from AP xx:xx:xx:xx:xx:xx after 500ms, disconnecting`

try changing the channel bandwidth to `20MHz` through your router's settings page.

Причина #3

On some laptop models with hardware rfkill switches (e.g., Thinkpad X200 series), due to wear or bad design, the switch (or its connection to the mainboard) might become loose over time resulting in seemingly random hardblocks/disconnects when you accidentally touch the switch or move the laptop. There is no software solution to this, unless your switch is electrical and the BIOS offers the option to disable the switch. If your switch is mechanical (most are), there are lots of possible solutions, most of which aim to disable the switch: Soldering the contact point on the mainboard/wifi-card, glueing or blocking the switch, using a screw nut to tighten the switch or removing it altogether.

Решение проблем с драйверами и прошивками

Здесь описаны подробности о том, как можно получить драйверы для вашего устройства. Вы можете обнаружить, что для вас есть несколько вариантов, помните, что вы можете найти здесь [HCL](#) помощь в выборе лучшего драйвера.

Здесь, возможно, описаны не все драйвера. Смотрите англоязычную версию статьи для получения информации по другим картам.

wlan-ng

```
pacman -S wlan-ng26 wlan-ng26-utils
```

Для wlan-ng вам не нужна утилита wireless-tools как сказано выше. Вместо них вам нужны утилиты из пакета wlan-ng26-utils: wlancfg и wlanctl-ng.

rt2x00

Для чипсетов Ralink (как rt2500, rt61, rt73 др.). Совместимы с wpa_supplicant, используют wext как интерфейс драйвера. Этот драйвер сейчас (в 2.6.24) является частью ядра и может быть загружен вручную например так...

```
modprobe rt2500pci
```

(замените при необходимости на rt2500pci например, т.е. rt2400pci, rt2500usb, rt61pci, rt73usb)

Для некоторых чипов необходимы прошивки (firmware). Смотри [rt2x00 статью wiki](#).

RT2500

Для чипсетов Ralink PCI/PCMCIA основанных rt2500 сериях (первое поколение чипов Ralink с поддержкой 802.11g).

```
pacman -S rt2500
```

Поддержка стандартной утилиты iwconfig для шифрования WEP соединений, также могут быть использованы другие стандартные утилиты. wpa_supplicant не поддерживает стандартный wext интерфейс. Драйвер поддерживает WPA (используя встроенное шифрование), но не стандартными способами. Разрабатываемая версия wpa_supplicant (0.6.x) включает в себя поддержку специальных технологий и это может негативно сказаться на WPA соединениях, устанавливаемых вручную через iwpriv команды. Смотрите [эти инструкции](#) для подробностей. Некоторые применимые методы для RT61 и RT73 ниже.

RT61

Для PCI/PCMCIA карт, основанных на чипе Ralink следующего поколения 802.11g (включена поддержка проприетарных MIMO функций). Смотри [RT61 статью wiki](#).

RT73

Для USB устройств, основанных на чипах Ralink следующих поколений 802.11g (включена поддержка проприетарных MIMO функций). Смотри [RT73 статью wiki](#).

madwifi

```
pacman -S madwifi
```

Модуль называется `ath_pci`. Чтобы его использовать, Вы должны в `rc.conf` убрать загрузку `ath5k` и добавить два модуля `madwifi`:

```
MODULES=(!ath5k ath_hal ath_pci ... ...)
```

Некоторым пользователям, возможно, при загрузке драйвера `madwifi` придется использовать код региона. Это связано с использованием каналов и частот, легальных для конкретной страны/региона. Для России, например, вы должны загрузить этот модуль так:

```
modprobe ath_pci countrycode=643
```

Вы можете проверить настройки, используя команду `iwlist`. Смотрите `man iwlist` и [CountryCode page on the MadWifi wiki](#). Для использования этих настроек при загрузке, добавьте следующую строку в `/etc/modprobe.d/modprobe.conf`:

```
options ath_pci countrycode=643
```

ATTENTION: Возможно, Вам придётся удалить код страны/региона, если устройство `ath0` не будет создано (kernel 2.6.21)!

Особенностью драйверов `madwifi` является то, что переключение в режим `ad-hoc` осуществляется двумя командами:

```
wlanconfig ath0 destroy  
wlanconfig ath0 create wlandev wifi0 wlanmode adhoc
```

ath5k

Планируется, что с течением времени `ath_pci` станет частью истории, и его заменит `ath5k`. Для использования этого открытого драйвера добавьте его загрузку в `rc.conf`:

```
MODULES=(ath5k ... ...)
```

Проверьте, что в секции `MODULES` отсутствуют параметры `ath_hal` и `ath_pci`.

ath9k

`ath9k` - это официальный драйвер компании Atheros для карт с новейшими 802.11n чипсетами (максимальная пропускная способность около 180 Мб/с). Чтобы просмотреть весь список поддерживаемого оборудования, проверьте [supported chipsets](#).

Доступные режимы: Station, AP и Adhoc.

`ath9k` включен в состав ядра, начиная с версии 2.6.27. Для дискуссий по поддержке и разработке создан [mailing list](#).

rtl8723bu

В текущем ядре драйвер для `rtl8723bu` не рабочий. Для решения проблемы требуется самостоятельная сборка модуля из исходников, либо установка из AUR. Исходники можете найти в [GitHub репозитории](#). Пакеты из AUR [rtl8723bu-git](#)^{AUR}, либо [rtl8723bu-git-dkms](#)^{AUR}

ipw2100 and ipw2200

Смотря какой чипсет у вас имеется, используйте следующее:

```
pacman -S ipw2100-fw
```

или:

```
pacman -S ipw2200-fw
```

Вам необходимо перезагрузиться, чтобы изменения были приняты.

ipw3945 and ipw4965

Новые драйверы Intel [iwlwifi project](#) работают с обоими чипсетами и включены в ядра v2.6.24 и выше. Просто установите прошивки:

```
pacman -S iwlwifi-3945-ucode
```

или:

```
pacman -S iwlwifi-4965-ucode
```

Если MOD_AUTOLOAD установлено в yes в /etc/rc.conf (так по умолчанию). Просто перезагрузитесь и проверьте, что драйверы работают с помощью **ifconfig** из терминала. Теперь можно сканировать сети через wlan0.

Если вы хотите, чтобы драйвера загружались вручную при загрузке добавьте их в строку MODULES:

```
nano /etc/rc.conf
```

в строке MODULES=(), добавьте **iwl3945** или **iwl4965** в список, в зависимости от вашего чипсета.

CTRL + X, Y для закрытия и сохранения.

Теперь драйверы должны быть загружены после перезагрузки и при запуске 'ifconfig' из терминала вы увидите, что там появился новый сетевой интерфейс **wlan0**.

Note: если драйверы iwlwifi, являющиеся "экспериментальными", не работают, знайте, что драйверы NETw4x32 работают отлично через ndiswrapper.

ipw3945 (Альтернативный метод)

Note: Этот драйвер ipw3945 должен входить в проект Intel's iwlwifi.

```
pacman -S ipw3945
```

Это должно установить ipw3945-ucode, ipw3945, и ipw3945d (daemon).

Для инициализации устройства при загрузке отредактируйте...

```
nano /etc/rc.conf
```

в строке `modules=()`, добавьте `ipw3945` в список

в строке `daemons=()`, добавьте `ipw3945d` в список (он должно быть ПЕРЕД `network` и `dhcdd/networkmanager` в списке)

CTRL + X, Y для закрытия и сохранения.

Модуль `ipw3945` должен быть загружен в процессе "Loading Modules.." и "Starting IPW3945d" должен появиться в ходе загрузки демона, и должен присутствовать интерфейс `ethX`.

Обновление: На моём HP nc6320 Bluetooth не соединяется, пока не выгрузишь модуль `ipw3945`.

orinoco

Часть, которая идёт с пакетом ядра и уже должна быть установлена.

ndiswrapper

Ndiswrapper не настоящий драйвер, но с ним вы можете использовать неродные Linux драйвера для ваших беспроводных устройств. Это очень помогает во многих ситуациях. Для использования его у вас должны быть *.inf файл из windows-драйверов (*.sys файл также должен присутствовать в этой же директории). Для установки `ndiswrapper` вам необходимо проделать следующие шаги:

Установить `ndiswrapper` используя `pacman`:

```
pacman -S ndiswrapper ndiswrapper-utils
```

Note: Beyond kernel-ядру необходим пакет `ndiswrapper-beyond` вместо `ndiswrapper`!

Note: Если у вас на машине нет доступа в интернет, вы можете скачать эти пакеты заранее к себе на компьютер с одного из зеркал, таких

как <http://www2.cddc.vt.edu/linux/distributions/archlinux/extra/os/i686/> . (Note: это устаревшее зеркало, лучше использовать <ftp://ftp.archlinux.org/core/os/i686/>)

Вам необходим пакет `ndiswrapper` (или `ndiswrapper-beyond` как было сказано выше) и пакет `ndiswrapper-utils`. Также вы можете скачать последнее ядро `kernel26` (или `beyond`), т.к. на CD не всегда последнее ядро.

Когда установка завершена, выполните следующие шаги для настройки `ndiswrapper`.

```
ndiswrapper -i filename.inf
ndiswrapper -l
ndiswrapper -m
depmod -a
```

Сейчас установка `ndiswrapper` полностью завершена; вам только необходимо отредактировать `/etc/rc.conf` для загрузки модуля при старте системы (ниже приведён мой простейший конфиг; у вас может немного отличаться):

```
MODULES=(ndiswrapper snd-intel8x0 !usbserial)
```

Важно убедиться, что `ndiswrapper` присутствует в этом списке, также добавить другие необходимые модули. Лучший способ проверить, что `ndiswrapper` загружен:

```
modprobe ndiswrapper
```

```
iwconfig
```

и wlan0 должен присутствовать. Посмотрите следующую страницу при обнаружении проблем: [Установка Ndiswrapper](#).

prism54

Скачайте файлы прошивки (firmware) для вашей карточки [с этого сайта](#). Переименуйте файл прошивки в 'isl3890'. Если не существует, создайте директорию /lib/firmware и поместите файл 'isl3890' туда. Это должно быть сделано. ([forum source](#))

ACX100/111

Установите пакеты 'tiacx' и 'tiacx-firmware' из репозитория core.

```
pacman -S tiacx tiacx-firmware
```

Драйвер должен сказать, какая прошивка (firmware) ему необходима; проверьте /var/log/messages.log или через команду dmesg. Переместите прошивку в '/lib/firmware'. Я делаю так:

```
ln -s /usr/share/tiacx/acx111_2.3.1.31/tiacx111c16 /lib/firmware
```

Hint: Если драйвер захламляет лог ядра, например потому, что запущен Kismet, вы должны добавить следующее в /etc/modprobe.d/modprobe.conf:

```
options acx debug=0
```

BCM43XX

Пользователи, у которых чипсет из серии Broadcom 43xx имеют альтернативу ndiswrapper'у. В Ядре версии 2.6.17, драйвер bcm43xx представлен.

1. Запустите

```
iwconfig
```

или

```
hwd -s
```

для того, чтобы удостовериться, что драйвер загружен. Мой вывод hwd -s выглядит примерно так:

```
Network      : Broadcom Corp.|BCM94306 802.11g NIC module:
unknown
```

Список поддерживаемого оборудования можно найти здесь [here](#).

1. Запустите

```
pacman -S bcm43xx-fwcuttner
```

для установки прошивки.

2. Скачайте драйвера для Windows для вашей карточки откуда вы скачивали прошивку.
3. Распаковать драйвера с страницы Dell можно через Windows или под WINE (это .exe файл который распаковывается в C:\Dell\[driver numbers]). Или можете попробовать скачать [\[3\]](#) или [\[4\]](#). Я просто сохранил файлы на рабочий стол; вам это не надо после следующего шага.
4. Запустите

```
bcm43xx-fwcuttner -w /lib/firmware  
/home/<user>/Desktop/wl_apsta.o
```

Сначала необходимо сначала создать директорию /lib/firmware.

5. Перезагрузитесь, и нормально настройте соединение. Вы можете добавить модуль bcm43xx в секцию modules в вашем rc.conf. Удачи!

b43

Данный драйвер плохо работает с BCM4312 (возможны зависания при загрузке системы), для данной карты лучше использовать [broadcom-wl](#) из aur

Этот драйвер - преемник драйвера bcm43xx и он включен в ядро 2.6.24.

1. Запустите

```
hwd -s
```

для определения вашей карты. Мой вывод hwd -s выглядит примерно так:

```
Network      : BCM4318 [AirForce One 54g] 802.11g Wireless LAN  
Controller module: unknown
```

Список поддерживаемого оборудования находится [здесь](#).

1. Установите fwcuttner из репозитория core:

```
pacman -S b43-fwcuttner
```

2. Скачайте проприетарную версию драйверов Broadcom:

```
wget http://downloads.openwrt.org/sources/broadcom-wl-  
4.150.10.5.tar.bz2
```

3. Далее:

```
tar xjf broadcom-wl-4.150.10.5.tar.bz2
cd broadcom-wl-4.150.10.5/driver
b43-fwcutter -w /lib/firmware/ wl_apsta_mimo.o
```

4. Перезагрузитесь, и нормально настройте ваше оборудование. Вы также можете добавить модуль b43 в секцию modules в ваш rc.conf. Удачи!

rtl8187

Смотри [rtl8187 wiki page](#).

zd1211rw

[zd1211rw](#) драйвер для ZyDAS ZD1211 802.11b/g USB WLAN чипсетов и он включен в ядро, в настоящее время. Смотри список поддерживаемого оборудования [здесь](#). Только вам необходимо сначала установить файлы прошивки:

```
pacman -S zd1211-firmware
```

Тестирование установки

После загрузки вашего драйвера запустите

```
iwconfig
```

и посмотрите, появился ли интерфейс беспроводного соединения (wlanX)

Смотрите также

- [NetworkManager](#) - Официальная страница NetworkManager
- [WICD](#) - Официальная страница для WICD
- [Wifi Radar](#) - Официальная страница Wifi Radar
- [An overly wordy howto that rarely helps](#)
- [The madwifi project's method of installing, good if you're having trouble doing it the Arch way](#)

Categories:

- [Русский](#)
- [Wireless networking \(Русский\)](#)