* [Home](https://www.archlinux.org/)
* [Packages](https://www.archlinux.org/packages/)
* [Forums](https://bbs.archlinux.org/)
* [Wiki](https://wiki.archlinux.org/)
* [Bugs](https://bugs.archlinux.org/)
* [AUR](https://aur.archlinux.org/)
* [Download](https://www.archlinux.org/download/)

**Wireless network configuration (简体中文)**

(Redirected from [Wireless Setup (简体中文)](https://wiki.archlinux.org/index.php?title=Wireless_Setup_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)&redirect=no))

**相关文章**

* [**Network configuration**](https://wiki.archlinux.org/index.php/Network_configuration)
* [**Software access point (简体中文)**](https://wiki.archlinux.org/index.php/Software_access_point_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87))
* [**Ad-hoc networking**](https://wiki.archlinux.org/index.php/Ad-hoc_networking)
* [**Internet sharing**](https://wiki.archlinux.org/index.php/Internet_sharing)
* [**Wireless bonding**](https://wiki.archlinux.org/index.php/Wireless_bonding)

**翻译状态：**本文是英文页面 [**Wireless\_network\_configuration**](https://wiki.archlinux.org/index.php/Wireless_network_configuration) 的[**翻译**](https://wiki.archlinux.org/index.php/ArchWiki_Translation_Team_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87))，最后翻译时间：2016-06-12，点击[**这里**](https://wiki.archlinux.org/index.php?title=Wireless_network_configuration&diff=0&oldid=437081)可以查看翻译后英文页面的改动。

配置无线网络一般分两步：第一步是识别硬件、安装正确的驱动程序并进行配置，安装盘中已经包含驱动，但是通常需要额外安装；第二步是选择一种管理无线连接的方式。这篇文章涵盖了这两方面，并提供了无线管理工具的链接地址。

**Contents**

 [[hide](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87))]

* [1 设备驱动](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E8.AE.BE.E5.A4.87.E9.A9.B1.E5.8A.A8)
  + [1.1 检查设备状态](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E6.A3.80.E6.9F.A5.E8.AE.BE.E5.A4.87.E7.8A.B6.E6.80.81)
  + [1.2 安装 driver/firmware](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E5.AE.89.E8.A3.85_driver.2Ffirmware)
* [2 无线网络管理](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E6.97.A0.E7.BA.BF.E7.BD.91.E7.BB.9C.E7.AE.A1.E7.90.86)
  + [2.1 手动设置](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E6.89.8B.E5.8A.A8.E8.AE.BE.E7.BD.AE)
    - [2.1.1 获取有用信息](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E8.8E.B7.E5.8F.96.E6.9C.89.E7.94.A8.E4.BF.A1.E6.81.AF)
    - [2.1.2 激活内核接口](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E6.BF.80.E6.B4.BB.E5.86.85.E6.A0.B8.E6.8E.A5.E5.8F.A3)
    - [2.1.3 查看接入点](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E6.9F.A5.E7.9C.8B.E6.8E.A5.E5.85.A5.E7.82.B9)
    - [2.1.4 运行模式](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E8.BF.90.E8.A1.8C.E6.A8.A1.E5.BC.8F)
    - [2.1.5 关联](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E5.85.B3.E8.81.94)
    - [2.1.6 获取 IP 地址](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E8.8E.B7.E5.8F.96_IP_.E5.9C.B0.E5.9D.80)
    - [2.1.7 示例](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E7.A4.BA.E4.BE.8B)
    - [2.1.8 自动设置](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E8.87.AA.E5.8A.A8.E8.AE.BE.E7.BD.AE)
    - [2.1.9 Connman](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Connman)
    - [2.1.10 Netctl](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Netctl)
      * [2.1.10.1 Wicd](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Wicd)
      * [2.1.10.2 NetworkManager](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#NetworkManager)
      * [2.1.10.3 Wifi Radar](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Wifi_Radar)
  + [2.2 节电](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E8.8A.82.E7.94.B5)
* [3 排错](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E6.8E.92.E9.94.99)
  + [3.1 Temporary internet access](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Temporary_internet_access)
  + [3.2 Rfkill 警告](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Rfkill_.E8.AD.A6.E5.91.8A)
  + [3.3 Respecting the regulatory domain](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Respecting_the_regulatory_domain)
  + [3.4 Observing Logs](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Observing_Logs)
  + [3.5 Failed to get IP address](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Failed_to_get_IP_address)
  + [3.6 Valid IP address but cannot resolve host](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Valid_IP_address_but_cannot_resolve_host)
  + [3.7 Setting RTS and fragmentation thresholds](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Setting_RTS_and_fragmentation_thresholds)
  + [3.8 Random disconnections](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Random_disconnections)
    - [3.8.1 Cause #1](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Cause_.231)
    - [3.8.2 Cause #2](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Cause_.232)
    - [3.8.3 Cause #3](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Cause_.233)
    - [3.8.4 Cause #4](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Cause_.234)
* [4 驱动与固件排错Troubleshooting drivers and firmware](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E9.A9.B1.E5.8A.A8.E4.B8.8E.E5.9B.BA.E4.BB.B6.E6.8E.92.E9.94.99Troubleshooting_drivers_and_firmware)
  + [4.1 Ralink](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Ralink)
    - [4.1.1 rt2x00](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#rt2x00)
    - [4.1.2 rt3090](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#rt3090)
    - [4.1.3 rt3290](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#rt3290)
    - [4.1.4 rt3573](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#rt3573)
    - [4.1.5 rt5572](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#rt5572)
  + [4.2 Realtek](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Realtek)
    - [4.2.1 rtl8192cu](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#rtl8192cu)
    - [4.2.2 rtl8192e](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#rtl8192e)
    - [4.2.3 rtl8188eu](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#rtl8188eu)
    - [4.2.4 rtl8723ae/rtl8723be](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#rtl8723ae.2Frtl8723be)
  + [4.3 Atheros](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Atheros)
    - [4.3.1 ath5k](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#ath5k)
    - [4.3.2 ath9k](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#ath9k)
    - [4.3.3 ath9k](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#ath9k_2)
      * [4.3.3.1 Power saving](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Power_saving)
  + [4.4 Intel](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Intel)
    - [4.4.1 ipw2100 与 ipw2200](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#ipw2100_.E4.B8.8E_ipw2200)
    - [4.4.2 iwlegacy](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#iwlegacy)
    - [4.4.3 iwlwifi](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#iwlwifi)
      * [4.4.3.1 禁用 LED 闪烁](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E7.A6.81.E7.94.A8_LED_.E9.97.AA.E7.83.81)
  + [4.5 Broadcom](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Broadcom)
    - [4.5.1 Tenda w322u](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Tenda_w322u)
    - [4.5.2 orinoco](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#orinoco)
    - [4.5.3 prism54](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#prism54)
    - [4.5.4 ACX100/111](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#ACX100.2F111)
    - [4.5.5 zd1211rw](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#zd1211rw)
    - [4.5.6 hostap\_cs](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#hostap_cs)
  + [4.6 ndiswrapper](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#ndiswrapper)
  + [4.7 backports-patched](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#backports-patched)
* [5 参见](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E5.8F.82.E8.A7.81)
* [6 其他资源](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E5.85.B6.E4.BB.96.E8.B5.84.E6.BA.90)

设备驱动

默认的 Arch Linux 内核是**模块化**的，，硬件的设备驱动作为[**内核模块**](https://wiki.archlinux.org/index.php/Kernel_modules)保存在硬盘上。启动时 [**udev**](https://wiki.archlinux.org/index.php/Udev) 会根据硬件加载不同的驱动模块，这就创建了需要的网络接口。

有些无线芯片需要额外的固件，默认安装的 [**linux-firmware**](https://www.archlinux.org/packages/?name=linux-firmware) 提供了很多固件。如果缺失需要的固件，请查看 [**#安装 driver/firmware**](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E5.AE.89.E8.A3.85_driver.2Ffirmware).

Udev 不是完美的，有些内核模块需要[**手动安装**](https://wiki.archlinux.org/index.php/Kernel_modules#Loading). 有些时候 Udev 会同时加载相互冲突的多个模块，就需要[**屏蔽**](https://wiki.archlinux.org/index.php/Kernel_modules#Blacklisting) 不需要的模块。

**检查设备状态**

根据设备是 PCI 还是 USB 连接，执行 lspci -k 或 lsusb -v 检查设备驱动是否已经加载：

$ 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

如果是 USB 设备，执行 dmesg | grep usbcore 可以看到类似下面的输出 usbcore: registered new interface driver rtl8187。

通过 ip link 查看无线 ([**设备名**](https://wiki.archlinux.org/index.php/Network_configuration#Device_names)，通常是类似 wlp2s1) 的设备。启用设备：

# ip link set <设备名> up

如果设备加载，接口正常启用，说明不需要安装额外的驱动和固件。

**安装 driver/firmware**

错误信息SIOCSIFFLAGS: No such file or directory 说明需要固件才能工作,

检查内核中的固件信息：

$ dmesg | grep firmware

[ 7.148259] iwlwifi 0000:02:00.0: loaded firmware version 39.30.4.1 build 35138 op\_mode iwldvm

如果没有类似的输出，先执行命令，例如iwlwifi，然后查找对应的错误信息：

$ 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

根据获得的信息，在下面网址查找硬件支持：

* 查看[**Linux 支持的无线驱动**](https://wireless.wiki.kernel.org/en/users/drivers)表格并查看对应的Driver页面，此外还有一个 [**Linux Wi-Fi 设备 IDs 列表**](https://wikidevi.com/wiki/List_of_Wi-Fi_Device_IDs_in_Linux).
* [**Ubuntu Wiki**](https://help.ubuntu.com/community/WifiDocs/WirelessCardsSupported) 维护了一个设备被内核和用户空间驱动支持状况的列表。
* [**Linux 无线支持页面**](http://linux-wless.passys.nl/) 和[**硬件兼容性列表**](http://www.linuxquestions.org/hcl/index.php?cat=10)(HCL)也维护了一个内核友好的设备列表。

注意有些厂商的产品即使有相同的名称，实际使用的芯片却是不同的。必须通过usb-id (USB设备) 或 pci-id (PCI设备) 进行判断。

如果列表中没有，可能你的设备只提供了 Windows 驱动(比如 Broadcom, 3com 等)。这时需要用 [**ndiswrapper**](http://ndiswrapper.sourceforge.net/wiki/index.php/List).

Ndiswrapper 可以在 Linux 中使用 Windows 驱动。兼容性列表在 [**这里**](http://ndiswrapper.sourceforge.net/mediawiki/index.php/List). 需要 Windows 中安装的 .inf 和 .sys 文件。如果有更新的网卡，请通过互联网搜索型号名称 + 'linux' 以获取更多信息。

无线网络管理

为了管理已经安装好的无线驱动，并且使无线能正常工作，需要安装一个无线连接管理工具。下面章节将帮助您确定一个最佳管理方法。

过程和需要使用的工具，将依赖于下面几个因素:

* 配置方式，从完全手动执行每一步到软件自动管理、自动启动
* 是否使用加密及加密类型
* 是否需要区分网络配置,是否经常切换不同网络（比如手提电脑）。
* 如果要在不同网络间切换，使用工具会更方便。

无论选的那个方案，最好先尝试手动方法。这将有助于您了解不同步骤的意义，并在出问题时解决之。 如果可以的话（比如说你在管理你自己的无线接入点），尝试连接一个开放的无线网络来检查是否所有的配置都在正常工作。然后再尝试加密的无线接入点，比如WEP（更易于配置）或者WPA。

此表列出可以使用的激活和管理无线网络的方法，按照加密和管理方式分类，给出了需要的工具。虽然还有其他办法，但这些是最常使用的:

|  |  |  |  |
| --- | --- | --- | --- |
| **管理方法** | **接口激活** | **无线连接管理  (/=alternatives)** | **IP 地址分配  (/=alternatives)** |
| [**手动设置**](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E6.89.8B.E5.8A.A8.E8.AE.BE.E7.BD.AE),  无加密或 WEP 加密 | [**ip**](https://wiki.archlinux.org/index.php/Core_utilities_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#ip) | [**iw**](https://www.archlinux.org/packages/?name=iw)/[**iwconfig**](https://www.archlinux.org/packages/?name=wireless_tools) | [**ip**](https://wiki.archlinux.org/index.php/Core_utilities_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#ip)/[**dhcpcd**](https://wiki.archlinux.org/index.php/Dhcpcd)/[**dhclient**](https://www.archlinux.org/packages/?name=dhclient)/[**networkd**](https://wiki.archlinux.org/index.php/Networkd) |
| [**手动管理**](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E6.89.8B.E5.8A.A8.E8.AE.BE.E7.BD.AE),  WPA 或 WPA2 PSK 加密 | [**ip**](https://wiki.archlinux.org/index.php/Core_utilities_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#ip) | [**iw**](https://www.archlinux.org/packages/?name=iw)/[**iwconfig**](https://www.archlinux.org/packages/?name=wireless_tools) + [**wpa\_supplicant**](https://wiki.archlinux.org/index.php/WPA_supplicant_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)) | [**ip**](https://wiki.archlinux.org/index.php/Core_utilities_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#ip)/[**dhcpcd**](https://wiki.archlinux.org/index.php/Dhcpcd)/[**dhclient**](https://www.archlinux.org/packages/?name=dhclient) |
| [**自动管理**](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E8.87.AA.E5.8A.A8.E8.AE.BE.E7.BD.AE),  支持网络配置 | [**netctl**](https://wiki.archlinux.org/index.php/Netctl_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)), [**Wicd**](https://wiki.archlinux.org/index.php/Wicd_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)), [**NetworkManager**](https://wiki.archlinux.org/index.php/NetworkManager_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)), etc.  这些工具会自动安装手动配置需要的工具。 | | |

**手动设置**

软件包 [**wireless\_tools**](https://www.archlinux.org/packages/?name=wireless_tools) 提供了建立无线连接的基础工具。如果你需要使用 WPA/WPA2 加密，还需要 [**wpa\_supplicant**](https://www.archlinux.org/packages/?name=wpa_supplicant)。 这些强大的用户空间终端工具提供了完全的控制手段。

这些例子假设无线设备是 wlan0, 请将其替换为正确的设备名。

**注意:**根据硬件和加密方式的不同，下面一些步骤可以省略。有些设备需要在建立关联时激活接口或扫描访问点，并提供 IP 地址。需要进行一些尝试，例如 WPA/WPA2 用户可以直接到第三步激活无线网络。

和其它网络接口一样，无线设备也是通过 [**iproute2**](https://www.archlinux.org/packages/?name=iproute2) 软件包提供的 ip 命令进行。

基本的工具如下，这些用户空间工具可以对无线连接进行完整控制。

* [**iw**](https://www.archlinux.org/packages/?name=iw) - 仅支持 nl80211 标准，不支持老的 WEXT (Wireless EXTentions) 标准. 如果 *iw* 没有显示网卡，可能是这个原因。
* [**wireless\_tools**](https://www.archlinux.org/packages/?name=wireless_tools) - 已经过时，但是依然广泛使用。WEXT 设备使用此工具.
* [**wpa\_supplicant**](https://www.archlinux.org/packages/?name=wpa_supplicant) - 提供 WPA/WPA2 加密支持,同时支持 nl80211 和 WEXT。

下面表格给出了 iw 和 wireless\_tools 命令的对比(更多示例参阅 [**这里**](http://wireless.kernel.org/en/users/Documentation/iw/replace-iwconfig)).

**Note:**

* 安装介质上包含 [**iw**](https://www.archlinux.org/packages/?name=iw) 和 [**wpa\_supplicant**](https://www.archlinux.org/packages/?name=wpa_supplicant).
* 示例中使用网络接口 wlan0 和热点 *your\_essid*.
* 大部分命令需要以 [**root 权限**](https://wiki.archlinux.org/index.php/Users_and_groups)执行，否则会无输出就退出。

|  |  |  |
| --- | --- | --- |
| ***iw* 命令** | ***wireless\_tools* 命令** | **描述** |
| iw dev wlan0 link | iwconfig wlan0 | 获取连接状态 |
| iw dev wlan0 scan | iwlist wlan0 scan | 扫描可用热点 |
| iw dev wlan0 set type ibss | iwconfig wlan0 mode ad-hoc | 设置操作模式为 *ad-hoc*. |
| iw dev wlan0 connect *your\_essid* | iwconfig wlan0 essid *your\_essid* | 连接到开放网络 |
| iw dev wlan0 connect *your\_essid* 2432 | iwconfig wlan0 essid *your\_essid* freq 2432M | 连接到开放网络的一个频道 |
| iw dev wlan0 connect *your\_essid* key 0:*your\_key* | iwconfig wlan0 essid *your\_essid* key *your\_key* | 用16进制加密密码访问 WEP 加密网络 |
| iw dev wlan0 connect *your\_essid* key 0:*your\_key* | iwconfig wlan0 essid *your\_essid* key s:*your\_key* | 用 ASCII 密码访问 WEP 加密网络. |
| iw dev wlan0 set power\_save on | iwconfig wlan0 power on | 启用省电模式 |

**注意:**根据硬件和加密设备的不同，有些步骤可以跳过。一些网卡需要在关联到热点前先激活或扫描热点，需要一些实验才能确定。WPA/WPA2 用户可以按照[**#关联**](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#.E5.85.B3.E8.81.94)中的步骤激活网络。

**获取有用信息**

[**iw 官方文档**](http://wireless.kernel.org/en/users/Documentation/iw) 包含更多示例。

* 获取接口名:

$ 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

* 检查连接状态，未连接时，可以看到：

$ iw dev wlan0 link

Not connected.

连接到 AP 后可以看到：

$ 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

* 获取统计数据:

$ 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

**激活内核接口**

(可能需要) 一些无线网卡在使用 [**wireless\_tools**](https://www.archlinux.org/packages/?name=wireless_tools)前需要激活内核接口:

# ip link set wlan0 up

如果出现错误 RTNETLINK answers: Operation not possible due to RF-kill, 请确保硬件开关已经打开。参阅 [**#Rfkill caveat**](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Rfkill_caveat)。

要验证接口确实打开：

# 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

<BROADCAST,MULTICAST,UP,LOWER\_UP> 中的UP 显示接口已经打开。

**查看接入点**

# iw dev wlan0 scan |less

**注意:**如果显示 "Interface doesn't support scanning"，可能是忘了安装固件。有时不以 root 运行 iwlist 也会产生这个问题。同样无线网络可能被软禁于，请安装 [**rfkill**](https://www.archlinux.org/packages/?name=rfkill) 并运行rfkill list all 进行检查。

需要关注的信息:

* **SSID:** 网络的名称.
* **Signal:** 用 dbm (-100 to 0) 报告的无线信号强度。数值越接近零，信号越好。观察高质量连接和低质量连接的数值差异可以了解设备的信号范围。
* **Security:** 没有直接报告, 检查 capability 开头的行，如果有 Privacy 信息，例如 capability: ESS Privacy ShortSlotTime (0x0411), 表示网络具有某种程度的保护，
  + 如果有 RSN 信息，网络被 [**Robust Security Network**](https://en.wikipedia.org/wiki/IEEE_802.11i-2004)(WPA2) 协议保护。
  + 如果有 WPA 信息，网络被 [**Wi-Fi Protected Access**](https://en.wikipedia.org/wiki/Wi-Fi_Protected_Access) 协议保护。
  + 在 RSN 和 WPA 信息块中，可能看到如下信息：
    - **Group cipher:** 数值包括 TKIP, CCMP, both, others.
    - **Pairwise ciphers:** 数值包括 TKIP, CCMP, both, others. 可能和 Group cipher 数值不同.
    - **Authentication suites:** 数值包括 PSK, 802.1x, others. 家用路由器通常可以看到 PSK (*i.e.* 密码). 在大学中，通常会链接到需要登录名和密码的 802.1x 网络。需要知道其使用的密码管理方式(例如 EAP), 封装方法 (例如 PEAP). 详情请参考 [**Wikipedia:Authentication protocol**](https://en.wikipedia.org/wiki/Authentication_protocol).
  + 如果没有看到 RSN 或 WPA，但是看到了 Privacy, 表示使用的是 WEP.

**运行模式**

设置无线网卡的操作模式，如果连接到漫游网络，需要设置操作模式为 **ibss**

# iw wlan0 set type ibss

**注意:**有些网卡需要先关闭无线接口(ip link set wlan0 down)才能修改模式。

**关联**

根据加密方式不同，需要使用密码将无线设备关联到接入点。

假设要使用的接入点 ESSID 为 MyEssid:

* 无加密

# iw wlan0 connect MyEssid

* WEP

使用十六进制或 ASCII 密码(格式是自动识别出来的，因为 WEP 密码长度是固定的):

# iw dev wlan0 connect *your\_essid* key 0:*your\_key*

使用十六进制或 ASCII 密码，第三个是默认 (从0计数，共四个):

# iw dev wlan0 connect *your\_essid* key d:2:*your\_key*

* **WPA/WPA2**

# wpa\_supplicant -i interface -c <(wpa\_passphrase *your\_SSID* *your\_key*)

假设设备使用 wext 驱动。如果无法工作，可能需要调整选项，参见 [**WPA supplicant**](https://wiki.archlinux.org/index.php/WPA_supplicant)。

如果连接成功，在新终端中执行后续命令或(或者通过 Ctrl+c 退出并使用 -B 参数在后台再次执行上述命令。[**WPA supplicant**](https://wiki.archlinux.org/index.php/WPA_supplicant) 页面包含更多参数和配置文件的信息。

通过下面命令确认是否连接成功：

# iw dev wlan0 link

**获取 IP 地址**

使用 DHCP：

# dhcpcd wlan0

静态 IP：

# ip addr add 192.168.0.2/24 dev wlan0

# ip route add default via 192.168.0.1

**Tip:**[**dhcpcd**](https://wiki.archlinux.org/index.php/Dhcpcd) 提供了 [**钩子**](https://wiki.archlinux.org/index.php/Dhcpcd#10-wpa_supplicant), 可以使用它自动在无线接口上启动 [**WPA supplicant**](https://wiki.archlinux.org/index.php/WPA_supplicant)。

**示例**

下面是一个用 WPA supplicant 和 DHCP 设置无线网络的完整示例.

# ip link set dev wlp13s1 up

# wpa\_supplicant -B -i wlp13s1 -c /etc/wpa\_supplicant/wpa\_supplicant.conf

# dhcpcd wlp13s1

要关闭连接，可以禁用接口：

# ip link set dev wlp13s1 down

要使用静态 IP，将 dhcpcd 命令替换为：

# ip addr add 192.168.0.10/24 broadcast 192.168.0.255 dev wlp13s1

# ip route add default via 192.168.0.1

禁用接口前，需要先刷新 IP 地址和网关:

# ip addr flush dev wlp13s1

# ip route flush dev wlp13s1

**自动设置**

有许多可选方法，但是注意它们是互斥的，不能同时运行两个守护进程。下面是比较表格：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **连接管理器** | **profiles 支持** | **漫游  (自动连接和重连)** | [**PPP**](https://en.wikipedia.org/wiki/point-to-point_protocol)**支持  (3G modem)** | **官方  GUI** | **控制台工具** |
| [**Connman**](https://wiki.archlinux.org/index.php/Connman) | Yes | Yes | Yes | No | connmanctl |
| [**Netctl**](https://wiki.archlinux.org/index.php/Netctl) | Yes | Yes | Yes | No | netctl,wifi-menu |
| [**NetworkManager**](https://wiki.archlinux.org/index.php/NetworkManager) | Yes | Yes | Yes | Yes | nmcli,nmtui |
| [**Wicd**](https://wiki.archlinux.org/index.php/Wicd) | Yes | Yes | No | Yes | wicd-curses |
| [**Wifi Radar**](https://wiki.archlinux.org/index.php/Wifi_Radar) | Yes | ? | ? | Yes | wifi-radar |

**Connman**

ConnMan 可以替代 NetworkManager 或 Wicd, 设计上考虑低资源消耗，以上网本和其它移动设备为目标。模块化设计并利用 dbus API，提供了 wpa\_supplicant 抽象.

参阅: [**Connman**](https://wiki.archlinux.org/index.php/Connman)

**Netctl**

*netctl* 替代了 *netcfg* 和 systemd 一起工作。使用基于 profile 的配置，可以检查和连接多种网络类型。使用简单，并不比图形工具难。 参阅： [**Netctl (简体中文)**](https://wiki.archlinux.org/index.php/Netctl_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87))

**Wicd**

Wicd 是可以同时处理无线和有线网络的管理器。用 Python 和 Gtk 写成，依赖关系比 NetworkManager 少，所以是轻量级桌面的理想选择。位于[**官方软件仓库**](https://wiki.archlinux.org/index.php/%E5%AE%98%E6%96%B9%E8%BD%AF%E4%BB%B6%E4%BB%93%E5%BA%93).

参见: [**Wicd (简体中文)**](https://wiki.archlinux.org/index.php/Wicd_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)).

**NetworkManager**

NetworkManager 是高级网络管理工具，在大部分流行发行版中使用。除了能管理有线链接，NetworkManager还提供了一个易于使用的图形界面程序来选择想要的无线移动链接。

详情请见 [**NetworkManager (简体中文)**](https://wiki.archlinux.org/index.php/NetworkManager_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87))。

**Wifi Radar**

WiFi Radar是 一个Python/PyGTK2 的管理无线配置的程序（**只有**无线）。它能够扫描可用的网络,为选择的网络创建新的配置。

详情请见[**Wifi Radar (简体中文)**](https://wiki.archlinux.org/index.php/Wifi_Radar_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87))。

**节电**

参阅 [**Power management#Network interfaces**](https://wiki.archlinux.org/index.php/Power_management#Network_interfaces).

排错

This section contains general troubleshooting tips, not strictly related to problems with drivers or firmware. For such topics, see next section [**#Troubleshooting drivers and firmware**](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Troubleshooting_drivers_and_firmware).

**Temporary internet access**

If you have problematic hardware and need internet access to, for example, download some software or get help in forums, you can make use of Android's built-in feature for internet sharing via USB cable. See [**Android tethering#USB tethering**](https://wiki.archlinux.org/index.php/Android_tethering#USB_tethering) for more information.

**Rfkill 警告**

Many laptops have a hardware button (or switch) to turn off wireless card, however, the card can also be blocked by kernel. This can be handled by [**rfkill**](https://www.archlinux.org/packages/?name=rfkill). Use*rfkill* to show the current status:

# rfkill list

0: phy0: Wireless LAN

Soft blocked: yes

Hard blocked: yes

If the card is *hard-blocked*, use the hardware button (switch) to unblock it. If the card is not *hard-blocked* but *soft-blocked*, use the following command:

# rfkill unblock wifi

**Note:**It is possible that the card will go from *hard-blocked* and *soft-unblocked* state into *hard-unblocked* and *soft-blocked* state by pressing the hardware button (i.e. the *soft-blocked* bit is just switched no matter what). This can be adjusted by tuning some options of the rfkill [**kernel module**](https://wiki.archlinux.org/index.php/Kernel_module).

切换无线网卡的硬件按钮是厂商专用的 [**内核模块**](https://wiki.archlinux.org/index.php/Kernel_module) 处理的, 通常是 [**WMI**](https://lwn.net/Articles/391230/) 模块。新硬件模块可能还没有被最新内核支持，这时可能需要查看内核 bug 系统并将硬件信息汇报给对应内核模块的维护者。

More info: [**http://askubuntu.com/questions/62166/siocsifflags-operation-not-possible-due-to-rf-kill**](http://askubuntu.com/questions/62166/siocsifflags-operation-not-possible-due-to-rf-kill)

**Respecting the regulatory domain**

The [**regulatory domain**](https://en.wikipedia.org/wiki/IEEE_802.11#Regulatory_domains_and_legal_compliance), 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**](https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2). 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**](https://en.wikipedia.org/wiki/List_of_WLAN_channels) for more detailed information.

Regdomains also affect the limit on the [**effective isotropic radiated power (EIRP)**](https://en.wikipedia.org/wiki/Equivalent_isotropically_radiated_power) from wireless devices. This is derived from transmit power/"tx power", and is measured in [**dBm/mBm (1dBm=100mBm) or mW (log scale)**](https://en.wikipedia.org/wiki/DBm). 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**](http://git.kernel.org/cgit/linux/kernel/git/linville/wireless-regdb.git/tree/db.txt) 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**](https://www.archlinux.org/packages/?name=crda) 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**](http://wiki.openwrt.org/doc/howto/wireless.utilities#iw), 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**](http://wireless.kernel.org/en/developers/Documentation/cfg80211) kernel module to use a specific regdomain by adding, for example, options cfg80211 ieee80211\_regdom=EU as [**module options**](https://wiki.archlinux.org/index.php/Kernel_modules#Setting_module_options). However, this is part of the [**old regulatory implementation**](http://wireless.kernel.org/en/developers/Regulatory#The_ieee80211_regdom_module_parameter).

For further information, read the [**wireless.kernel.org regulatory documentation**](http://wireless.kernel.org/en/developers/Regulatory/).

**Observing Logs**

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**](http://www.aboutcher.co.uk/2012/07/linux-wifi-deauthenticated-reason-codes/) might give a first hint. Maybe it also helps you to look at the control message [**flowchart**](https://wireless.wiki.kernel.org/en/developers/documentation/mac80211/auth-assoc-deauth), the journal messages will follow it.

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.

**Failed to get IP address**

* If getting an IP address repeatedly fails using the default [**dhcpcd**](https://www.archlinux.org/packages/?name=dhcpcd) client, try installing and using [**dhclient**](https://www.archlinux.org/packages/?name=dhclient) instead. Do not forget to select *dhclient* as the primary DHCP client in your [**connection manager**](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Automatic_setup)!
* If you can get an IP address for a wired interface and not for a wireless interface, try disabling the wireless card's [**power saving**](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Power_saving) features (specify offinstead of on).
* If you get a timeout error due to a *waiting for carrier* problem, then you might have to set the channel mode to auto for the specific device:

# iwconfig wlan0 channel auto

Before changing the channel to auto, make sure your wireless interface is down. After it has successfully changed it, you can bring the interface up again and continue from there.

**Valid IP address but cannot resolve host**

If you are on a public wireless network that may have a [**captive portal**](https://en.wikipedia.org/wiki/Captive_portal), make sure to query an HTTP page (not an HTTPS page) from your web browser, as some captive portals only redirect HTTP. If this is not the issue, it may be necessary to remove any custom DNS servers from [**resolv.conf**](https://wiki.archlinux.org/index.php/Resolv.conf).

**Setting RTS and fragmentation thresholds**

Wireless hardware disables RTS and fragmentation by default. These are two different methods of increasing throughput at the expense of bandwidth (i.e. reliability at the expense of speed). These are useful in environments with wireless noise or many adjacent access points.

Packet fragmentation improves throughput by splitting up packets with size exceeding the fragmentation threshold. The maximum value (2346) effectively disables fragmentation since no packet can exceed it. The minimum value (256) maximizes throughput, but may carry a significant bandwidth cost.

# iw phy0 set frag 512

[**RTS**](https://en.wikipedia.org/wiki/IEEE_802.11_RTS/CTS) improves throughput by performing a handshake with the access point before transmitting packets with size exceeding the RTS threshold. The maximum threshold (2347) effectively disables RTS since no packet can exceed it. The minimum threshold (0) enables RTS for all packets, which is probably excessive for most situations.

# iw phy0 set rts 500

**Note:**phy0 is the name of the wireless device as listed by $ iw phy.

**Random disconnections**

**Cause #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[**[1]**](http://us.generation-nt.com/answer/gentoo-user-wireless-deauthenticating-by-local-choice-help-204640041.html). Try disabling the wireless card's [**power saving**](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#Power_saving) features (specify off instead of on).

If your card does not support enabling/disabling power save mode, check the BIOS for power management options. Disabling PCI-Express power management in the BIOS of a Lenovo W520 resolved this issue.

**Cause #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.

**Cause #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.

**Cause #4**

Another cause for frequent disconnects or a complete failure to connect may also be a sub-standard router, incomplete settings of the router, or interference by other wireless devices.

To troubleshoot, first best try to connect to the router with no authentication.

If that works, enable WPA/WPA2 again but choose fixed and/or limited router settings. For example:

* If the router is considerably older than the wireless device you use for the client, test if it works with setting the router to one wireless mode
* Disable mixed-mode authentication (e.g. only WPA2 with AES, or TKIP if the router is old)
* Try a fixed/free channel rather than "auto" channel (maybe the router next door is old and interfering)
* Disable 40Mhz channel bandwidth (lower throughput but less likely collisions)
* If the router has quality of service settings, check completeness of settings (e.g. Wi-Fi Multimedia (WMM) is part of optional QoS flow control. An erroneous router firmware may advertise its existence although the setting is not enabled)

驱动与固件排错Troubleshooting drivers and firmware

This section covers methods and procedures for installing kernel modules and *firmware* for specific chipsets, that differ from generic method.

See [**Kernel modules**](https://wiki.archlinux.org/index.php/Kernel_modules) for general informations on operations with modules.

**Ralink**

**rt2x00**

Ralink 芯片组的统一驱动，代替了 rt2500, rt61, rt73 等。Linux 内核从 2.6.24 开始包含此驱动，但是有些设备可能需要额外固件。可以使用标准 [**wpa\_supplicant**](https://www.archlinux.org/packages/?name=wpa_supplicant) 和 iwconfig 工具配置。

有些芯片组需要固件文件，可以安装软件包 [**linux-firmware**](https://www.archlinux.org/packages/?name=linux-firmware)。

参见: [**Using the new rt2x00 beta driver**](https://wiki.archlinux.org/index.php/Using_the_new_rt2x00_beta_driver)

* Since kernel 3.0, rt2x00 includes also these drivers: rt2800pci, rt2800usb. rt2860sta 被主分支驱动 rt2800pci 替代，rt2870sta 被 rt2800usb 替代。
* 通过 iwpriv 可以配置很多参数，文档在 [**Ralink 源代码包**](http://web.ralinktech.com/ralink/Home/Support/Linux.html) 中。

**rt3090**

For devices which are using the rt3090 chipset it should be possible to use rt2800pci driver, however, is not working with this chipset very well (e.g. sometimes it's not possible to use higher rate than 2Mb/s).

The best way is to use the [**rt3090-dkms**](https://aur.archlinux.org/packages/rt3090-dkms/)AUR[[**broken link**](https://wiki.archlinux.org/index.php/ArchWiki:Requests#Broken_package_links): archived in[**aur-mirror**](http://pkgbuild.com/git/aur-mirror.git/tree/rt3090-dkms)] driver from [**AUR**](https://wiki.archlinux.org/index.php/AUR). Make sure to [**blacklist**](https://wiki.archlinux.org/index.php/Kernel_modules#Blacklisting) the rt2800pci module and setup the rt3090stamodule to [**load**](https://wiki.archlinux.org/index.php/Kernel_modules#Loading) at boot.

**Note:**This driver also works with rt3062 chipsets. Also the [**rt3090**](https://aur.archlinux.org/packages/rt3090/)AUR[[**broken link**](https://wiki.archlinux.org/index.php/ArchWiki:Requests#Broken_package_links): archived in[**aur-mirror**](http://pkgbuild.com/git/aur-mirror.git/tree/rt3090)] package is not supported by the latest kernel and has been orphaned [**rt3090-dkms**](https://aur.archlinux.org/packages/rt3090-dkms/)AUR[[**broken link**](https://wiki.archlinux.org/index.php/ArchWiki:Requests#Broken_package_links): archived in[**aur-mirror**](http://pkgbuild.com/git/aur-mirror.git/tree/rt3090-dkms)] should be used instead.

**rt3290**

The rt3290 chipset is recognised by the kernel rt2800pci module. However, some users experience problems and reverting to a patched Ralink driver seems to be beneficial in these [**cases**](https://bbs.archlinux.org/viewtopic.php?id=161952).

**rt3573**

2012年新出的芯片组，需要 Ralink 的闭源驱动，有不同的厂商使用他们，参阅[**Belkin N750 示例**](https://bbs.archlinux.org/viewtopic.php?pid=1164228#p1164228) 。

**rt5572**

支持 5 Gh 频率，需要 Ralink 的闭源驱动，编译指令位于[**这里**](http://bernaerts.dyndns.org/linux/229-ubuntu-precise-dlink-dwa160-revb2)

**Realtek**

**rtl8192cu**

The driver is now in the kernel, but many users have reported being unable to make a connection although scanning for networks does work.

Package [**8192cu-dkms**](https://aur.archlinux.org/packages/8192cu-dkms/)AUR in the AUR includes many patches, try this if it doesn't work fine with the driver in kernel.

**rtl8192e**

The driver is part of the current kernel package. 启动时可能装入模块失败，错误信息是:

rtl819xE:ERR in CPUcheck\_firmware\_ready()

rtl819xE:ERR in init\_firmware() step 2

rtl819xE:ERR!!! \_rtl8192\_up(): initialization is failed!

r8169 0000:03:00.0: eth0: link down

一个暂时的解决方法是卸载模块：

# modprobe -r r8192e\_pci

等一会后，重新装入模块：

# modprobe r8192e\_pci

**rtl8188eu**

Some dongles, like the TP-Link TL-WN725N v2 (not sure, but it seems that uses the rtl8179 chipset), use chipsets compatible with this driver. In Linux 3.12 the driver [**has been moved**](http://lwn.net/Articles/564798/) to kernel staging source tree. For older kernels use out-of-tree driver sources built with [**dkms**](https://wiki.archlinux.org/index.php/Dkms) - install [**8188eu-dkms**](https://aur.archlinux.org/packages/8188eu-dkms/)AUR. At the times of 3.15 kernel rtl8188eu driver is buggy and has many stability issues.

**rtl8723ae/rtl8723be**

The new rtl8723ae module is included in the mainline Linux kernel since version 3.6, the rtl8723be module since 3.15.

Some users may encounter errors with powersave on this card. This is shown with occasional disconnects that are not recognized by high level network managers ([**netctl**](https://wiki.archlinux.org/index.php/Netctl), [**NetworkManager**](https://wiki.archlinux.org/index.php/NetworkManager)). This error can be confirmed by running dmesg -w or journalctl -f and looking for output related to powersave and the rtl8723ae/rtl8723bemodule. If you are having this issue, use the fwlps=0 kernel option, which should prevent the WiFi card from automatically sleeping and halting connection.

/etc/modprobe.d/rtl8723ae.conf

options rtl8723ae fwlps=0

or

/etc/modprobe.d/rtl8723be.conf

options rtl8723be fwlps=0

**Atheros**

[**MadWifi team**](http://madwifi-project.org/) 开发组维护了三个模块：

* madwifi 是最老的驱动, Arch kernel 从 2.6.39.1 开始已经不再包含。
* [**ath5k**](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#ath5k) 将逐步替代 ath\_pci，有些芯片组使用效果很好，但有些还不能很好工作(后面有介绍)
* [**ath9k**](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#ath9k) 是新的官方驱动，适用于新 Atheros 硬件。

There are some other drivers for some Atheros devices. See [**Linux Wireless documentation**](http://wireless.kernel.org/en/users/Drivers/Atheros#PCI_.2F_PCI-E_.2F_AHB_Drivers) for details.

**ath5k**

参考:

* [**http://wireless.kernel.org/en/users/Drivers/ath5k**](http://wireless.kernel.org/en/users/Drivers/ath5k)
* [**http://wiki.debian.org/ath5k**](http://wiki.debian.org/ath5k)

If you find web pages randomly loading very slow, or if the device is unable to lease an IP address, try to switch from hardware to software encryption by loading the ath5k module with nohwcrypt=1 option. See [**Kernel modules#Setting module options**](https://wiki.archlinux.org/index.php/Kernel_modules#Setting_module_options) for details.

有些笔记本的 LED 指示灯有问题，可以：

echo none > "/sys/class/leds/ath5k-phy0::tx/trigger"

echo none > "/sys/class/leds/ath5k-phy0::rx/trigger"

**ath9k**

ath9k 是 Atheros 官方支持的驱动，支持所有带 802.11n 功能的芯片组，最大传输速度 180 Mbps. [**这个页面**](http://wireless.kernel.org/en/users/Drivers/ath9k) 列出了所有支持的硬件。

工作模式：Station, AP and Adhoc.

ath9k 是官方内核的一部分。如果在极个别情况下遇到稳定性问题，可以使用 [**compat-wireless**](http://wireless.kernel.org/en/users/Download) 软件包。[**ath9k 邮件列表**](https://lists.ath9k.org/mailman/listinfo/ath9k-devel) 提供了支持和开发的相关信息。

参考:

* [**http://wireless.kernel.org/en/users/Drivers/ath9k**](http://wireless.kernel.org/en/users/Drivers/ath9k)
* [**http://wiki.debian.org/ath9k**](http://wiki.debian.org/ath9k)

**ath9k**

External resources:

* [**http://wireless.kernel.org/en/users/Drivers/ath9k**](http://wireless.kernel.org/en/users/Drivers/ath9k)
* [**http://wiki.debian.org/ath9k**](http://wiki.debian.org/ath9k)

As of Linux 3.15.1, some users have been experiencing a decrease in bandwidth. In some cases this can fixed by editing /etc/modprobe.d/ath9k.conf and adding the line:

options ath9k nohwcrypt=1

**Note:**Check with the command lsmod what module(-name) is in use and change it if named otherwise (e.g. ath9k\_htc).

In the unlikely event that you have stability issues that trouble you, you could try using the [**backports-patched**](https://aur.archlinux.org/packages/backports-patched/)AUR package. An [**ath9k mailing list**](https://lists.ath9k.org/mailman/listinfo/ath9k-devel) exists for support and development related discussions.

**Power saving**

Although [**Linux Wireless**](http://wireless.kernel.org/en/users/Documentation/dynamic-power-save) says that dynamic power saving is enabled for Atheros ath9k single-chips newer than AR9280, for some devices (e.g. AR9285) [**powertop**](https://www.archlinux.org/packages/?name=powertop)might still report that power saving is disabled. In this case enable it manually.

On some devices (e.g. AR9285), enabling the power saving might result in the following error:

# iw dev wlan0 set power\_save on

command failed: Operation not supported (-95)

The solution is to set the ps\_enable=1 option for the ath9k module:

/etc/modprobe.d/ath9k.conf

options ath9k ps\_enable=1

**Intel**

**ipw2100 与 ipw2200**

内核完全支持，但是需要安装额外的固件。根据芯片组型号，[**安装**](https://wiki.archlinux.org/index.php/Pacman_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)) 软件包 [**ipw2100-fw**](https://www.archlinux.org/packages/?name=ipw2100-fw) 或 [**ipw2200-fw**](https://www.archlinux.org/packages/?name=ipw2200-fw)。

**Tip:**You may use the following [**module options**](https://wiki.archlinux.org/index.php/Kernel_modules#Setting_module_options):

* use the rtap\_iface=1 option to enable the radiotap interface
* use the led=1 option to enable a front LED indicating when the wireless is connected or not

**iwlegacy**

[**iwlegacy**](http://wireless.kernel.org/en/users/Drivers/iwlegacy) is the wireless driver for Intel's 3945 and 4965 wireless chips. The firmware is included in the [**linux-firmware**](https://www.archlinux.org/packages/?name=linux-firmware) package.

[**udev**](https://wiki.archlinux.org/index.php/Udev) should load the driver automatically, otherwise load iwl3945 or iwl4965 manually. See [**Kernel modules#Loading**](https://wiki.archlinux.org/index.php/Kernel_modules#Loading) for details.

If you have problems connecting to networks in general or your link quality is very poor, try to disable 802.11n:

/etc/modprobe.d/iwl4965.conf

options iwl4965 11n\_disable=1

**iwlwifi**

[**iwlwifi**](http://wireless.kernel.org/en/users/Drivers/iwlwifi) is the wireless driver for Intel's current wireless chips, such as 5100AGN, 5300AGN, and 5350AGN. See the [**full list of supported devices**](http://wireless.kernel.org/en/users/Drivers/iwlwifi#Supported_Devices). The firmware is included in the [**linux-firmware**](https://www.archlinux.org/packages/?name=linux-firmware) package.

If you have problems connecting to networks in general or your link quality is very poor, try to disable 802.11n and enable software encryption:

/etc/modprobe.d/iwlwifi.conf

options iwlwifi 11n\_disable=1

options iwlwifi swcrypto=1

If you have a problem with slow uplink speed in 802.11n mode, for example 20Mbps, try to enable antenna aggregation:

/etc/modprobe.d/iwlwifi.conf

options iwlwifi 11n\_disable=8

Do not be confused with the option name, when the value is set to 8 it does not disable anything but re-enables transmission antenna aggregation.[**[2]**](http://forums.gentoo.org/viewtopic-t-996692.html?sid=81bdfa435c089360bdfd9368fe0339a9) [**[3]**](https://bugzilla.kernel.org/show_bug.cgi?id=81571)

In case this does not work for you, you may try disabling [**power saving**](https://wiki.archlinux.org/index.php/Power_saving#Network_interfaces) for your wireless adapter.

[**Some**](http://ubuntuforums.org/showthread.php?t=2183486&p=12845473#post12845473) have never gotten this to work. [**Others**](http://ubuntuforums.org/showthread.php?t=2205733&p=12935783#post12935783) found salvation by disabling N in their router settings after trying everything. This is known to have be the only solution on more than one occasion. The second link there mentions a 5ghz option that might be worth exploring.

**Note:**The [**linux-lts**](https://www.archlinux.org/packages/?name=linux-lts)-3.14 kernel may take several minutes to load the firmware and make the wireless card ready for use. The issue is reported to be fixed in[**linux**](https://www.archlinux.org/packages/?name=linux)-3.17 kernel.[**[4]**](https://bbs.archlinux.org/viewtopic.php?id=190757)

**禁用 LED 闪烁**

**Note:**This works with the iwlegacy and iwlwifi drivers.

默认设置中 LED 闪烁是开着的，有些人不喜欢，可以[**systemd-tmpfiles**](https://wiki.archlinux.org/index.php/Systemd#Temporary_files)禁止闪烁：

/etc/tmpfiles.d/phy0-led.conf

w /sys/class/leds/phy0-led/trigger - - - - phy0radio

Run systemd-tmpfiles --create phy0-led.conf for the change to take effect, or reboot.

To see all the possible trigger values for this LED:

# cat /sys/class/leds/phy0-led/trigger

**Tip:**If you do not have /sys/class/leds/phy0-led, you may try to use the led\_mode="1" [**module option**](https://wiki.archlinux.org/index.php/Kernel_modules#Setting_module_options). It should be valid for both iwlwifi and iwlegacy drivers.

# cat /sys/class/leds/phy0-led/trigger

**Broadcom**

See [**Broadcom wireless**](https://wiki.archlinux.org/index.php/Broadcom_wireless).

**Tenda w322u**

Treat this Tenda card as an rt2870sta device. See [**#rt2x00**](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)#rt2x00).

**orinoco**

这应当是内核的一部分，是已经被安装的。

Some Orinoco chipsets are Hermes II. You can use the wlags49\_h2\_cs driver instead of orinoco\_cs and gain WPA support. To use the driver, [**blacklist**](https://wiki.archlinux.org/index.php/Kernel_modules#Blacklisting) orinoco\_cs first.

**prism54**

The driver p54 is included in kernel, but you have to download the appropriate firmware for your card from [**this site**](http://linuxwireless.org/en/users/Drivers/p54#firmware) and install it into the /usr/lib/firmwaredirectory.

过时的 prism54 和新内核模块 p54pci 或 p54usb 同时装入造成冲突，使用 lsmod | grep prism54 查看是否装入了过时模块，如果是，那么就 [**屏蔽**](https://wiki.archlinux.org/index.php/Blacklisting) prism54 并根据上面方法修改固件名称。

**ACX100/111**

**Warning:**The drivers for these devices [**are broken**](https://mailman.archlinux.org/pipermail/arch-dev-public/2011-June/020669.html) and do not work with newer kernel versions.

Packages: tiacx tiacx-firmware (deleted from official repositories and AUR)

See [**official wiki**](http://sourceforge.net/apps/mediawiki/acx100/index.php?title=Main_Page) for details.

**zd1211rw**

[**zd1211rw**](http://zd1211.wiki.sourceforge.net/) 是ZyDAS ZD1211 802.11b/g USB WLAN芯片的驱动，最近的版本的内核已经包括了。[**zd1211rw**](http://zd1211.wiki.sourceforge.net/) [**[5]**](http://www.linuxwireless.org/en/users/Drivers/zd1211rw/devices%EF%BD%9C%E8%BF%99%E9%87%8C)有被支持的设备列表。 你只需要这样安装固件[**zd1211-firmware**](https://www.archlinux.org/packages/?name=zd1211-firmware)。

**hostap\_cs**

[**Host AP**](http://hostap.epitest.fi/) is a Linux driver for wireless LAN cards based on Intersil's Prism2/2.5/3 chipset. The driver is included in Linux kernel.

**Note:**Make sure to [**blacklist**](https://wiki.archlinux.org/index.php/Kernel_modules#Blacklisting) the orinico\_cs driver, it may cause problems.

**ndiswrapper**

Ndiswrapper并不是一个真正的驱动,但是如果你无法找到适合你的无线网卡驱动的适合, 它就派上用场了.有的时候, 它是非常有用的.为了使用Ndiswrapper, 你需要Windows驱动中的\*.inf文件(\*.sys文件应该和\*.info在同一个目录中)。如果你需要从 \*.exe 文件解压缩，你可以使用 [**cabextract**](https://www.archlinux.org/packages/?name=cabextract).

**警告:**确保使用合适架构(也就是32/64位)的驱动。

下面是安装ndiswrapper的几个步骤:

1. 安装 [**ndiswrapper-dkms**](https://www.archlinux.org/packages/?name=ndiswrapper-dkms) 2. 安装驱动到 /etc/ndiswrapper/\*

ndiswrapper -i filename.inf

3. 列出所有的安装的驱动

ndiswrapper -l

4. 配置文件写到 /etc/modprobe.d/ndiswrapper.conf

ndiswrapper -m

depmod -a

现在基本上就要安装完ndiswrapper了; 依照 [**这里**](https://wiki.archlinux.org/index.php/Kernel_modules#Loading)设置启动时加载这个模块。

modprobe ndiswrapper

iwconfig

如果正常的话, 你应该可以看到wlan0接口了. 如果有问题的话, 你可以阅读： [**Ndiswrapper installation wiki**](http://ndiswrapper.sourceforge.net/joomla/index.php?/component/option,com_openwiki/Itemid,33/id,installation/). [**ndiswrapper howto**](http://sourceforge.net/p/ndiswrapper/ndiswrapper/HowTos/) 和 [**ndiswrapper FAQ**](http://sourceforge.net/p/ndiswrapper/ndiswrapper/FAQ/).

**backports-patched**

[**backports-patched**](https://aur.archlinux.org/packages/backports-patched/)AUR 为老内核提供了新内核中无线网卡驱动的移植，项目从 2007 年开始，以前名称是 compat-wireless, 现在重命名为 backports. 如果使用旧内核并且有无线问题，可以试试这个软件包。

参见

* [**Sharing PPP Connection**](https://wiki.archlinux.org/index.php/Sharing_PPP_Connection)
* [**Ad-hoc networking**](https://wiki.archlinux.org/index.php/Ad-hoc_networking)

其他资源

* [**NetworkManager**](http://www.gnome.org/projects/NetworkManager/) - NetworkManager官方网站
* [**WICD**](http://wicd.sourceforge.net/) - WICD官方网站
* [**Wifi Radar**](http://wifi-radar.systemimager.org/) - Wifi Radar官方网站
* [**一个很少能帮忙的废话连篇的howto**](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html)
* [**madwifi的安装方法，如果你在用Arch的方式安装时遇到了问题**](http://madwifi-project.org/wiki/UserDocs/FirstTimeHowTo)

[Categories](https://wiki.archlinux.org/index.php/Special:Categories):

* [简体中文](https://wiki.archlinux.org/index.php/Category:%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)
* [Wireless networking (简体中文)](https://wiki.archlinux.org/index.php/Category:Wireless_networking_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87))

Navigation menu

* [**page**](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87))

* [discussion](https://wiki.archlinux.org/index.php?title=Talk:Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)&action=edit&redlink=1)

* [view source](https://wiki.archlinux.org/index.php?title=Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)&action=edit)

* [history](https://wiki.archlinux.org/index.php?title=Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)&action=history)
* [create account](https://wiki.archlinux.org/index.php?title=Special:UserLogin&returnto=Wireless+network+configuration+%28%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87%29&type=signup)

* [log in](https://wiki.archlinux.org/index.php?title=Special:UserLogin&returnto=Wireless+network+configuration+%28%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87%29)

navigation

* [Main page](https://wiki.archlinux.org/index.php/Main_page)
* [Categories](https://wiki.archlinux.org/index.php/Table_of_contents)
* [Getting involved](https://wiki.archlinux.org/index.php/Getting_involved)
* [Wiki news](https://wiki.archlinux.org/index.php/ArchWiki:News)
* [Random page](https://wiki.archlinux.org/index.php/Special:Random)

search

窗体顶端

  

窗体底端

interaction

* [Help](https://wiki.archlinux.org/index.php/Category:Help)
* [Contributing](https://wiki.archlinux.org/index.php/ArchWiki:Contributing)
* [Recent changes](https://wiki.archlinux.org/index.php/Special:RecentChanges)
* [Recent talks](https://wiki.archlinux.org/index.php/Special:RecentChanges?namespace=0&invert=1)
* [New pages](https://wiki.archlinux.org/index.php/Special:NewPages)
* [Statistics](https://wiki.archlinux.org/index.php/ArchWiki:Statistics)
* [Reports](https://wiki.archlinux.org/index.php/ArchWiki:Reports)
* [Requests](https://wiki.archlinux.org/index.php/ArchWiki:Requests)

tools

* [What links here](https://wiki.archlinux.org/index.php/Special:WhatLinksHere/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87))
* [Related changes](https://wiki.archlinux.org/index.php/Special:RecentChangesLinked/Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87))
* [Special pages](https://wiki.archlinux.org/index.php/Special:SpecialPages)
* [Printable version](https://wiki.archlinux.org/index.php?title=Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)&printable=yes)
* [Permanent link](https://wiki.archlinux.org/index.php?title=Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)&oldid=442802)
* [Page information](https://wiki.archlinux.org/index.php?title=Wireless_network_configuration_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)&action=info)

in other languages

* [Čeština](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%C4%8Cesky))
* [Deutsch](https://wiki.archlinux.de/title/(W)LAN_und_Arch_Linux)
* [Ελληνικά](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%CE%95%CE%BB%CE%BB%CE%B7%CE%BD%CE%B9%CE%BA%CE%AC))
* [English](https://wiki.archlinux.org/index.php/Wireless_network_configuration)
* [Español](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(Espa%C3%B1ol))
* [Français](http://wiki.archlinux.fr/Wifi)
* [Italiano](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(Italiano))
* [日本語](https://wiki.archlinuxjp.org/index.php/%E3%83%AF%E3%82%A4%E3%83%A4%E3%83%AC%E3%82%B9%E8%A8%AD%E5%AE%9A)
* [Nederlands](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(Nederlands))
* [Română](http://wiki.archlinux.ro/index.php/Wireless)
* [Русский](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%D0%A0%D1%83%D1%81%D1%81%D0%BA%D0%B8%D0%B9))
* [ไทย](https://wiki.archlinux.org/index.php/Wireless_network_configuration_(%E0%B9%84%E0%B8%97%E0%B8%A2))
* [Türkçe](http://archtr.org/wiki/index.php?title=Kablosuz_ba%C4%9Flant%C4%B1)
* This page was last modified on 24 July 2016, at 15:27.

* Content is available under [GNU Free Documentation License 1.3 or later](http://www.gnu.org/copyleft/fdl.html) unless otherwise noted.

* [Privacy policy](https://wiki.archlinux.org/index.php/ArchWiki:Privacy_policy)

* [About ArchWiki](https://wiki.archlinux.org/index.php/ArchWiki:About)

* [Disclaimers](https://wiki.archlinux.org/index.php/ArchWiki:General_disclaimer)