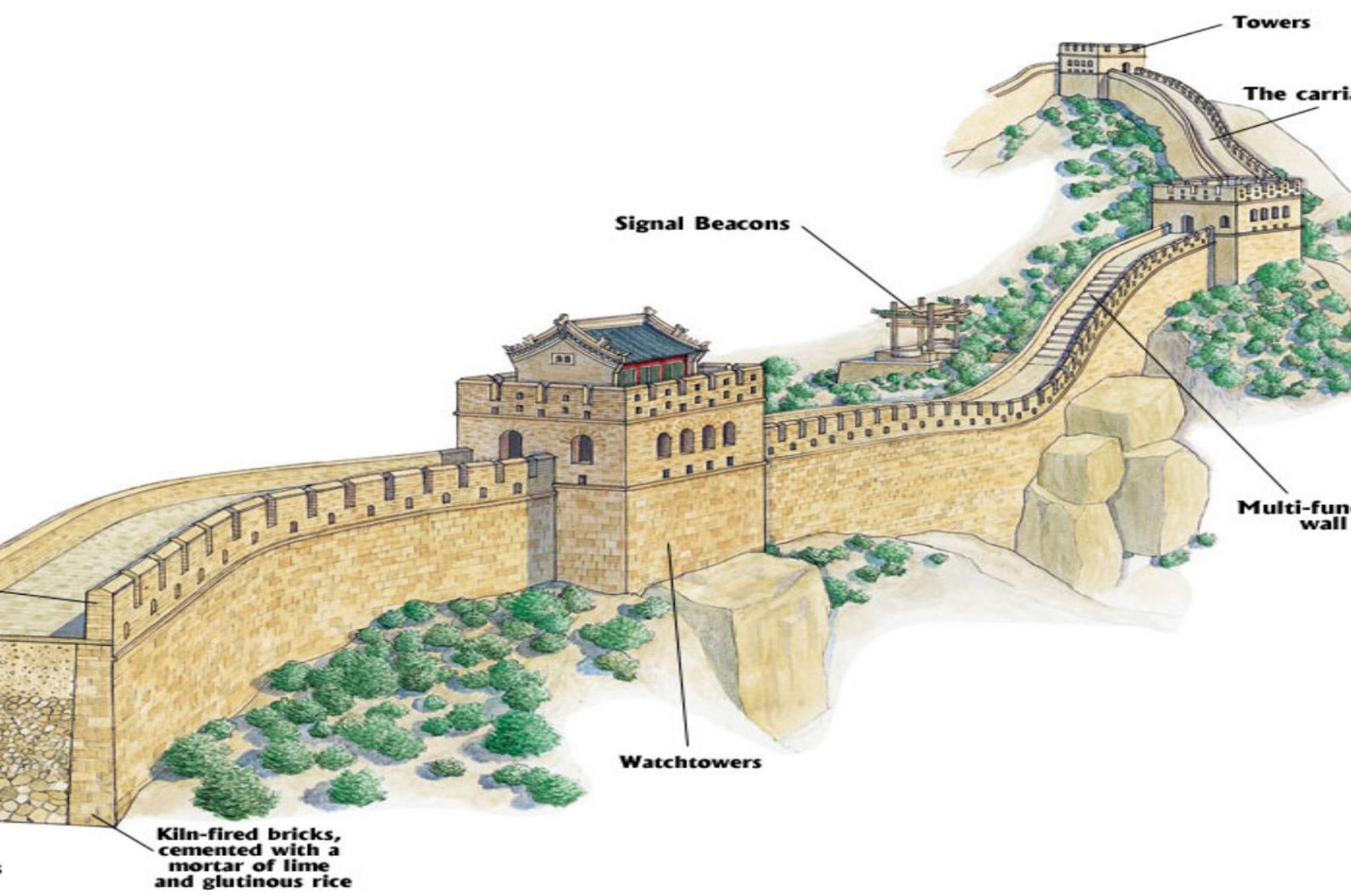


# 穿越长城



# Table of Contents

---

1. [Introduction](#)
2. [无线路由器刷OpenWrt固件的准备工作](#)
  - i. [什么是无线路由器固件](#)
  - ii. [支持OpenWrt的路由器](#)
  - iii. [备份原厂路由器配置文件](#)
3. [路由器怎样刷OpenWrt固件 \(WR2543N为例\)](#)
  - i. [怎样下载OpenWrt固件](#)
  - ii. [网页界面刷OpenWrt教程](#)
  - iii. [网页界面OpenWrt拨号上网设置教程](#)
  - iv. [网页界面OpenWrt无线\(Wifi\)设置图文教程](#)
  - v. [网页界面怎样备份OpenWrt设置](#)
  - vi. [网页界面升级OpenWrt固件](#)
  - vii. [怎样进入OpenWrt安全恢复模式](#)
  - viii. [命令行 OpenWrt sysupgrade 更新固件](#)
  - ix. [命令行 uci设置 OpenWrt 上网参数](#)
4. [OpenWrt+shadowsocks-libev自动翻墙](#)
  - i. [什么是shadowsocks-libev翻墙软件](#)
  - ii. [翻墙软件Shadowsocks-libev服务端设置](#)
  - iii. [OpenWrt路由器运行shadowsocks-libev客户端](#)
  - iv. [史上最通俗易懂的OpenWrt翻墙路由器解释](#)
  - v. [配置OpenWrt路由器智能自动翻墙](#)
  - vi. [OpenWrt自动更新设置和屏蔽广告](#)
5. [OpenWrt编译翻墙固件教程](#)
  - i. [编译shadowsocks-libev for OpenWrt ipk安装包](#)
  - ii. [下载和设置翻墙配置文件](#)
  - iii. [使用Image Builder编译自动翻墙OpenWrt固件](#)
  - iv. [如何使用别人预编译的OpenWrt翻墙固件](#)
6. [应用 : D-Link DIR-505刷OpenWrt翻墙教程](#)
  - i. [如何进入 DIR-505 恢复模式](#)
  - ii. [DIR-505刷OpenWrt固件过程](#)
  - iii. [DIR-505启用工作模式开关](#)
  - iv. [DIR-505 Router 模式](#)
  - v. [DIR-505 AP 模式](#)
  - vi. [DIR-505 编译OpenWrt全自动翻墙固件](#)
  - vii. [DIR-505 刷预编译OpenWrt翻墙固件](#)
  - viii. [登陆并设置 DIR-505 OpenWrt 翻墙固件](#)
7. [附录](#)
  - i. [翻墙软件、教程汇总](#)
  - ii. [本机阅读本教程的方法](#)
  - iii. [知识若不分享，实在没有意义](#)
  - iv. [如何贡献本项目](#)

# OpenWrt智能、自动、透明翻墙路由器教程

---

手把手教你路由器刷OpenWrt固件，自动穿越万里长城。

史上最简易、详细的路由器安装OpenWrt智能翻墙教程。

## 知识若不分享，实在没有意义

---

2014年6月 Dropbox壮烈被墙。

查资料发现，著名的开源路由器固件OpenWrt支持家里的路由器 TP-Link WR2543N V1，于是就给路由器安装了OpenWrt并设置为自动智能翻墙。

自由的感觉真好： youtube, hulu, twitter, facebook, google...

什么是圣人，圣人就是得到和付出比较均衡的人。天地生我，我敬天地；父母育我，我亦养父母；网上获得知识，也要在网上分享知识。于是，花了许多天，查资料，写教程，调试固件，不知不觉一天就过去了。

希望你应用本教程后，也把你的过程写下来，合并到这个项目中来：

<https://github.com/softwaredownload/openwrt-fanqiang>

2014-07-14

## 相关资源

---

- OpenWrt自动翻墙教程电子书下载  
<https://software-download.name/2014/fanqiang-jiaocheng/>
- TP-Link TLWR2543 预编译翻墙固件(2014-09-01):  
<https://software-download.name/2014/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade-bin-with-shadowsocks/>
- D-Link DIR-505 预编译翻墙固件(2014-09-01):  
<https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>
- shadowsocks-libev-polarssl\_ar71xx.ipk (2014-08-18):  
<https://software-download.name/2014/shadowsocks-libev-polarssl-ar71xx-ipk-latest/>

## 关于预编译固件（2014-09-24）

---

目前的预编译固件是启用了计划任务的，这会有潜在的不确定性，如果更新时下载的文件存在错误，导致dnsmasq无法读取，可能造成dnsmasq无法启动，翻墙功能自然失效。

如果你刷了预编译固件后，某一天突然不能翻墙了，这时设置客户端的IP地址为和路由器同网段，登陆路由器，用ps命令查看dnsmasq进程是否启动了，如果没有启动，就重刷固件或者用原来的文件代替路由器里/etc/dnsmasq.d/下的文件。

第一次刷预编译固件后，建议立即禁用自动更新 dnsmasq.d 目录下的文件，运行稳定一段时间后，需要时再加上计划任务。

下次发布固件时，将默认不启用计划任务。

## 授权许可

---

除特别声明外，本书中的内容使用[CC BY-SA 3.0 License](#)（创作共用 署名-相同方式共享3.0许可协议）授权，代码遵循[BSD 3-Clause License](#)（3项条款的BSD许可协议）。

在线阅读**OpenWrt翻墙路由器教程**:

---

- <https://github.com/softwaredownload/openwrt-fanqiang/blob/master/SUMMARY.md>
- <http://softwaredownload.gitbooks.io/openwrt-fanqiang/>

# 无线路由器刷OpenWrt固件的准备工作

---

在给你的路由器刷新固件之前，有必要先了解：

1. 什么是无线路由器固件
2. 准备支持OpenWrt路由器
3. 如何备份路由器配置

# 什么是无线路由器固件

---

网络的本质是知识的开放与共享。人类社会进步速度，如果原来是自行车速，加上网络后，就坐上了火箭。

一个热爱学习的人，必然要查找一些英文学习资料，在某个国家的某个阶段必然会遇到一个问题：怎么Google搜索这么料，经常打不开，YouTube真差劲，加载半天还在打转...

后来，可能会发现，不是人家烂，而是有人故意为之。

怎么办呢？有很多种办法解决这个问题，其中一个较好的方案是从家用无线路由器上解决，然后全部有线和无线设备都可以无障碍上网了。

路由器的原厂固件限制了用户自行开发功能，我们必须给路由器刷上特定的固件，并进行一些设置才可以翻墙。

无线路由器就好比是一台小电脑。电脑上安装了Windows XP, Windows 7, Windows 8, 或者Ubuntu等操作系统就可以使用了。固件就是给路由器使用的操作系统，是固化在路由器芯片内的操作系统。

## 常用的开源第三方无线路由器固件

---

### 1. 开源OpenWRT路由器固件：部署复杂、灵活性高

这也是本文系列所用的固件。发展成熟，支持的硬件多。

### 2. 开源DD-WRT路由器固件：支持广泛、功能全面 DD-WRT比较实用，通过网页对固件进行配置的功能强大，但是定制和扩展比较困难。

### 3. 开源Tomato路由器固件：衍生版本众多 原始版本固件代码自2010年后就再没有更新。

本系列教程使用OpenWrt来讲解路由器翻墙方法。

# 支持OpenWrt的路由器

现在3G手持设备已经普及，一般情况下读者家里都已经有无线路由器了，到底能不能刷上OpenWrt固件呢？到OpenWrt官方网站查一下就知道了。

打开[支持OpenWrt无线路由器列表](#)这个页面，搜索一下。比如我家原来的无线路由器型号是TP-LINK TL-WR2543N，同时按下Ctrl+F，输入**WR2543**就可以找到，如下图：

Model	Version	Status	Target(s)	Platform	CPU Speed (MHz)	Flash (MB)	RAM (MB)	Vendor	Arch
<a href="#">TL-WR2543ND</a>	1.0	12.09	ar71xx	Atheros AR7242	400	8	64	ASUS	ARM (cortex-A9)

从上图可以看出，OpenWrt支持WR2543N无线路由器版本1。此外，还可以看出更多信息，比如芯片类型是ar71xx，芯片型号是Atheros AR7242，CPU频率是400 MHz，原厂带8MB Flash, 64MB RAM内存。

目前[京东](#)已经不再销售WR2543N，淘宝有可能还有。如果你购买其他品牌，建议Flash在8 MB或以上，RAM在64MB以上。

如果你准备买新路由器，可以在上面列表中[查找OpenWrt推荐路由器型号](#)，并到[京东](#)或[淘宝](#)搜索确认能否买到，能买到的话，再以关键词 型号 **OpenWrt** 在搜索引擎搜索相关信息，确保你想购买的型号能比较容易地刷上OpenWrt固件。

作为新手来说，推荐使用D-Link DIR-505，可能是最便宜的适合学习OpenWrt的路由器，如果你的应用场景要求不高，也可以用来作为日常使用的路由器。

# 备份原厂路由器配置文件

---

提示，刷机有风险，如果不当操作，或者有其他意外发生，路由器可能变成砖头，本文系列旨在提供参考，刷机风险由读者自负，作者不承担任何责任，也没有义务提供个别指导。

本文作者给WR2543N刷OpenWrt固件不下10次，因为完全没有经验，有几次刷了后不能进入管理界面，只能用手机3G上网查找解决方案，还好WR2543N非常容易进入安全模式，然后重新刷固件，解决了问题。作为初学者，一定要购买容易进入安全模式的路由器。

对于本文作者来说，现在已经不需要原厂固件了，但是在第一次刷OpenWrt前，我还是把原厂固件的配置文件作备份，建议读者也是如此。

怎样备份原厂固件，WR2543N的原厂说明书说得很详细，建议找出来详细阅读。

## LAN 和WAN的区别

---

什么是LAN和WAN,第一次听到这种专业名词容易让人头大。

LAN并不是一个单词，而是三个英文单词的缩写：Local Area Network，查出这三个单词的意思，就比较好理解了，就是本地区域网络的意思。本地，比如是室内，公司内，办公室内都是本地，也就是LAN是用来连接本地电脑的。

WAN，Wide Area Network，广泛区域网络，也就是连向更广泛的外部的网络，一般家用就是通向ADSL modem，再通过ADSL modem连接互联网。

路由器通常有多个LAN口，一个WAN口。

在WR2543N路由器的后背，有并排4个的网线插口，叫LAN口，单独的一个网线插口叫WAN口，WAN口旁边还有个USB插口。把ADSL的线插在WAN口。备好一根网线，一头插路由器的任意一个LAN口，另一头插电脑。

## 设置电脑LAN口IP地址

---

路由器和电脑都处在本地网络里面，为了互相区分，本地网络的每台设备都需要有不同的IP地址。

本路由器默认 LAN 口 IP 地址是 192.168.1.1， 默认子网掩码是 255.255.255.0

电脑的IP地址要和路由器的不同，我们可以设置电脑的本地IP地址为动态获取。如果手动设置IP地址，那么计算机IP地址必须为192.168.1.X（X是2到254之间的任意整数），子网掩码须设置为255.255.255.0，默认网关须设置为192.168.1.1

以Windows XP 系统为例，介绍计算机参数的设置步骤。

右键单击桌面上的 网上邻居 图标，选择 属性，在打开的 网络连接 页面中，右键单击“本地连接”，选择 状态，打开“本地连接状态”进行操作。详细步骤请见购机时附带的手册。

## 登陆路由器管理界面

---

打开网页浏览器，在浏览器的地址栏中输入路由器的 IP地址：192.168.1.1，可以看到下图



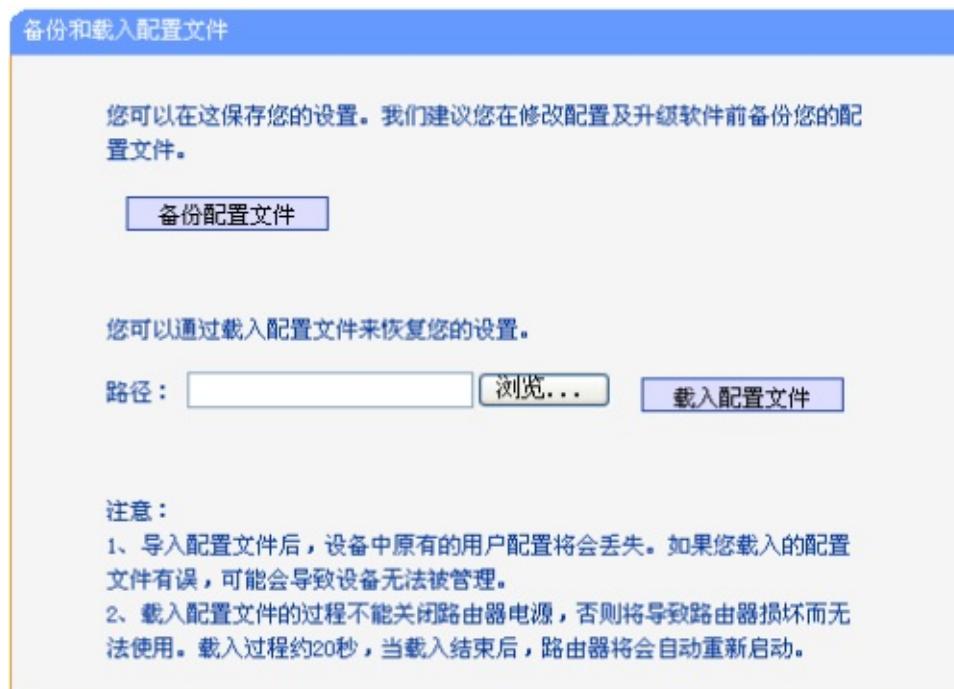
所示登录界面，输入用户名和密码（用户名和密码的出厂默认值均为admin），单击确定按钮。

## 备份原厂路由器固件配置文件

登陆路由器管理界面后，选择菜单，系统工具→备份和载入配置，可以在如下图所示备份或载入路由器配置文件。

配置备份功能可以将路由器的设置以文件形式保存到电脑中，以备下次使用；在升级路由器软件或在载入新的配置文件前备份路由器的原有配置，可以有效防止升级软件或载入新配置文件过程中丢失原有配置的问题。

配置载入功能则可以将先前保存的或已编辑好的配置文件重新载入。



# 怎样刷第三方路由器固件OpenWrt

---

经过前面的准备，终于要给亲自给路由器刷OpenWrt固件了。有可能失败，有可能成功。一连嘴里念叨FGW (=fuck great wall)，一边给自己打气。

有几个问题有必要提一下：

## OpenWrt有必要装中文管理界面吗？

---

我认为不需要。网上最新最全面的信息都是英文的。GFW在不断进步，我们也要不停地学习。我们要感谢GFW，让我们每天多记几个单词。一些步骤的操作，我特意截图并加上了步骤标识，实在记不住就每次打开这个教程照着图示来。

## 在开源的Linux类操作系统里连接OpenWrt进行操作

---

我认为有必要从现在开始切换到Linux类操作系统了。Windows已经开始走向没落，开源操作系统渐渐赶上闭源商业操作系统。

为什么呢？随着技术的不断进化，开源的技术合作越来越方便。我打个比方，如果佛教老大释迦牟尼，基督教创始人耶稣，不开源恐怕也会穷途末路。

再说OpenWrt就是微型的Linux操作系统，熟悉了Linux，学习OpenWrt就很容易了。

在以后的教程里，都是在Ubuntu下对OpenWrt进行管理。如果有两台电脑，建议一台装Ubuntu，如果只有一台电脑，可以装Ubuntu和Windows双启动。

# 怎样下载OpenWrt固件

---

## 下载最新版的OpenWRT固件

---

进入下面的页面下载：

<http://downloads.openwrt.org/snapshots/trunk/>

打开页面后，选择你的路由器的芯片型号进入，再选择你的路由器型号。有两个文件供下载，一个文件结尾是 `factory.bin`,适合原厂固件下刷，另一个文件名结尾是 `sysupgrade.bin`,适合已经是OpenWrt系统下刷。

## 如何下载OpenWrt固件 for WR2543

---

OpenWrt官方网页上有WR2543N的专页,详细介绍了刷机步骤及注意事项.

打开官方Wiki页面 [TP-Link TL-WR2543ND](#)

上面列出了支持的版本: v1.0和v1.2。我的路由器是v1.0的，可以刷，你的版本如果不是这两个，不能确保能刷成功。

这两个固件都带LuCI 网页管理界面。有时候，如果你升级了不带LuCI的固件，命令行方式又无法搞定OpenWRT上网参数设置,就需要先在电脑里下载带LuCI的固件，scp复制到路由器升级，再通过网页设置。

有两个固件供下载:

- [openwrt-ar71xx-generic-tl-wr2543n-v1-squashfs-factory.bin](#) - Installing OpenWRT from factory
- [openwrt-ar71xx-generic-tl-wr2543n-v1-squashfs-sysupgrade.bin](#) - Upgrading an existing OpenWRT install

一定要注意:

- 在原厂固件上刷OpenWrt, 要用固件文件名带 **factory** 的.bin文件.
- 已经刷了OpenWrt固件, 再升级 OpenWrt固件时就要用文件名带 **sysupgrade** 的 .bin文件.

现在我们是在原厂固件基础上刷 OpenWrt, 自然是下载第一个文件,也就是 `openwrt-ar71xx-generic-tl-wr2543n-v1-squashfs-factory.bin`

要确保下载下来的文件完整,下载过程没有中断,如果下载下来的文件不完整,并把这个不完整的文件刷进机器,恢复起来很麻烦,有可能变砖.

## 下载预编译的翻墙固件 for WR2543

- <https://software-download.name/2014/openwrt-ar71xx-generic-tl-wr2543n-v1-squashfs-sysupgrade-bin-with-shadowsocks/>

# 通过网页界面WR2543刷OpenWrt教程

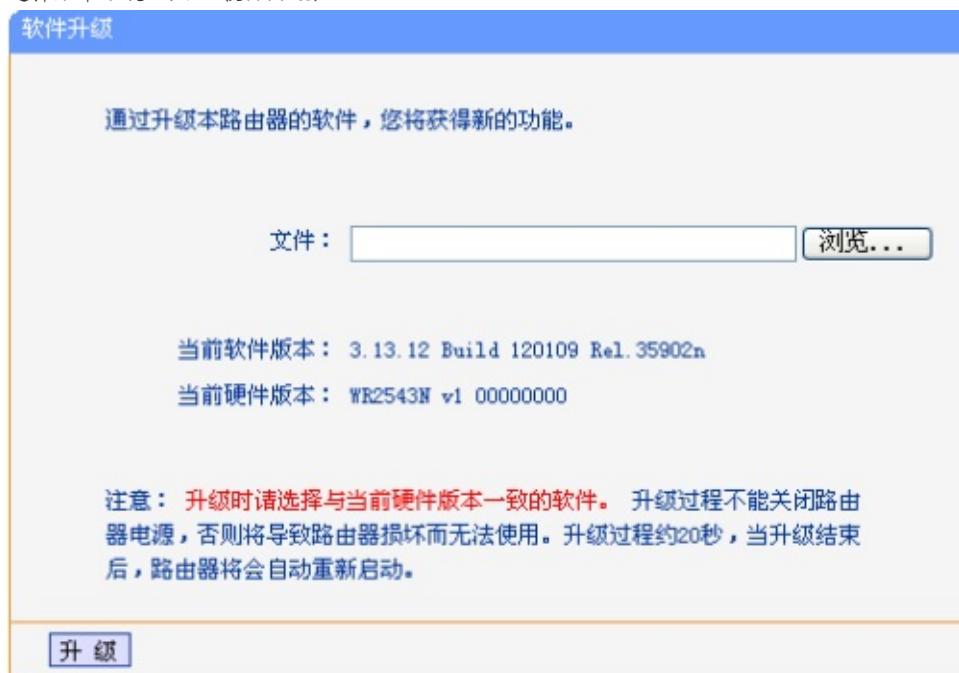
## 通过有线或无线连上WR2543路由器

打开浏览器,输入路由器的IP地址: **192.168.1.1**

回车,在密码验证框,输入用户名: **admin** 密码也是 **admin**

## 路由器固件升级

选择菜单系统工具 → 软件升级,



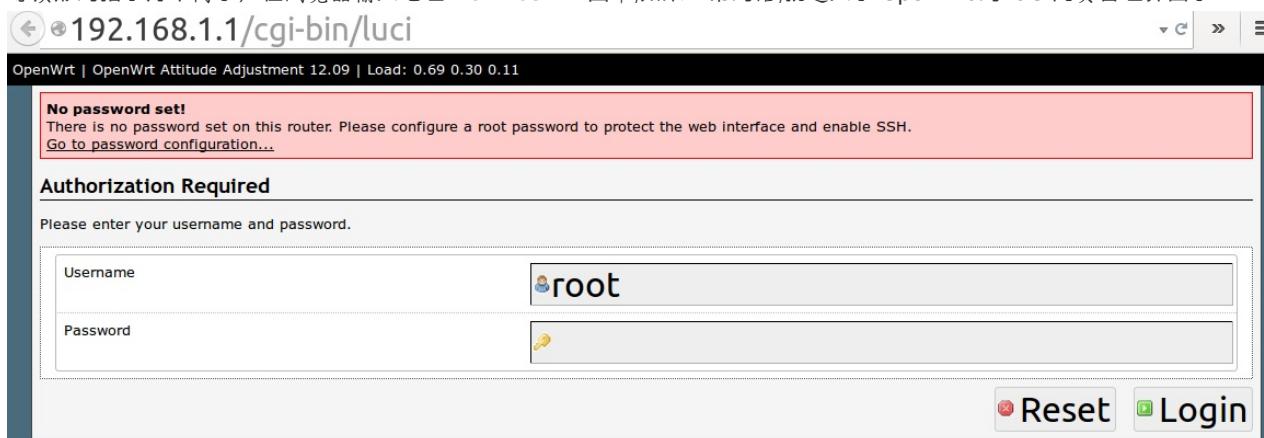
点击 浏览 按钮选择下载的文件 **openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-factory.bin**

注意，文件名必须是...factory.bin。

再单击 升级 进行软件升级。要注意，在刷固件过程中不可停电或其他原因造成中断，否则路由器就变砖了。

等待几分钟

等锁形的指示灯不闪了，在浏览器输入地址: **192.168.1.1** 回车,如果正常的话,就进入了 OpenWrt 的LuCI网页管理界面了.



默认用户名是root,默认密码是空。点 **Login** 直接登陆。



# 网页界面OpenWrt拨号上网设置教程

见界面登陆路由器后，就可以设置上网参数了。

## 编辑OpenWrt WAN设置

选择上面的 Network，在 Interface里，WAN右边，选择Edit。WAN和ADSL modem相连，设置拨号上网自然是在WAN而不是LAN。

The screenshot shows the OpenWrt web interface with the Network tab selected. Under the Interfaces section, the WAN interface is highlighted. Its status shows it is connected. The LAN interface is also listed with its status. Below the interfaces is a button to add a new interface.

## 配置OpenWrt拨号上网密码

进去后，在协议 Protocol 下拉列框里，选择拨号上网的协议，也就是 PPPoE，再点击下面的 Switch Protocol切换协议。

### Interfaces - WAN

This screenshot shows the WAN configuration page. The protocol dropdown is set to PPPoE (marked with a blue circle 1). The 'Really switch protocol?' checkbox is checked and highlighted with a blue circle 2. At the bottom are buttons for Reset, Save, and Save & Apply.

## OpenWrt拨号上网用户名密码设置

1. PAP/CHAP username: 拨号上网用户名
2. PAP/CHAP password: 拨号上网密码
3. 点击 Save & Apply 保存并应用设置

**Common Configuration**

General Setup Advanced Settings Physical Settings Firewall Settings

Status RX: 0.00 B (0 Pkts.)  
pppoe-wan TX: 0.00 B (0 Pkts.)

Protocol PPPoE

PAP/CHAP username admin (1)

PAP/CHAP password (2) [REDACTED]

Access Concentrator auto  
(?) Leave empty to autodetect

Service Name auto  
(?) Leave empty to autodetect

(3)

这时，连接LAN的电脑应该已经可以上网了，但无线设备还不行。

# 网页界面OpenWrt无线(Wifi)设置图文教程

登陆OpenWrt路由器后：

## 选择 Network, Wifi, Edit

No password set!  
There is no password set on this router. Please configure a root password to protect the web interface and enable wireless.

Go to password configuration...

### Wireless Overview

**Generic MAC80211 802.11abgn (radio0)**

**SSID:** OpenWrt | **Mode:** Unknown  
0% Wireless is disabled or not associated

**Enable** **Edit** **Re**

点击**Enable**按钮，这时无线设备已经可以连上Wifi

## Wireless Network: Unknown "OpenWrt" (radio0.network1)

The *Device Configuration* section covers physical settings of the radio hardware such as channel, transmit power or antenna selection which is shared among all defined wireless networks (if the radio hardware is multi-SSID capable). Per network settings like encryption operation mode are grouped in the *Interface Configuration*.

**Device Configuration**

**General Setup** **Advanced Settings**

**Status**

**SSID:** OpenWrt | **Mode:** Unknown  
0% Wireless is disabled or not associated

**Wireless network is disabled**

**Enable**

**Channel**

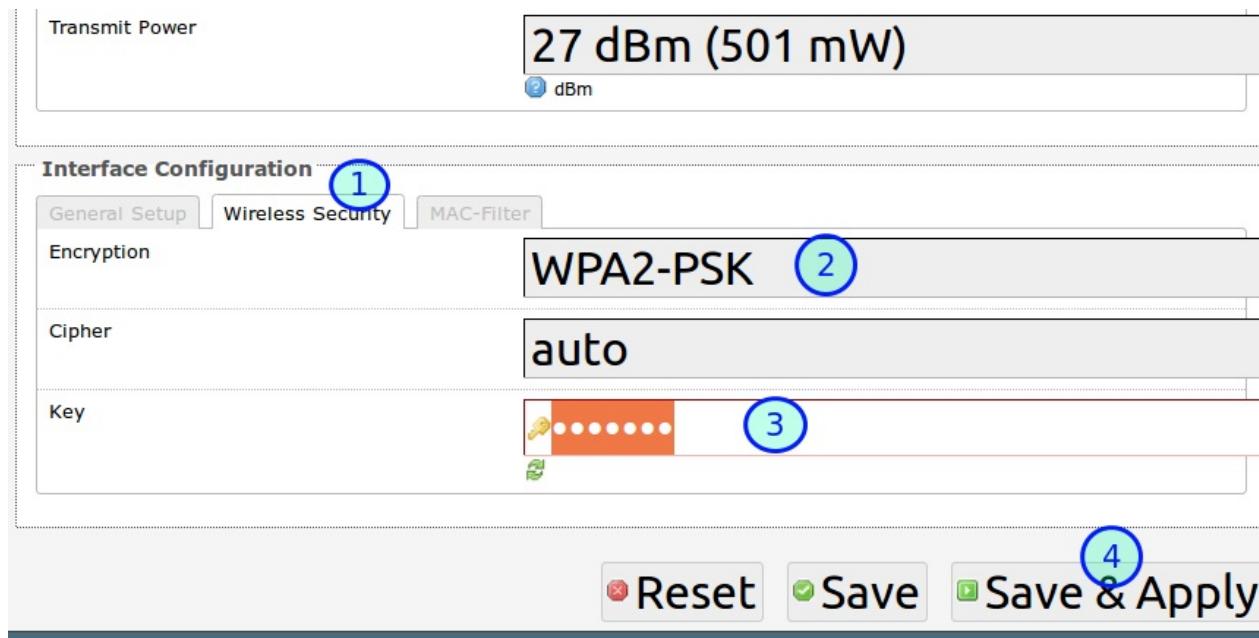
**11 (2.462 GHz)**

默认ESSID就是OpenWrt，没有密码。不想做活雷锋的加个密码吧。

## OpenWrt Wifi密码设置

把ESSID改成 eastking-wr2543,然后：

- 点击Wireless Security进入OpenWrt无线安全设置
- Encryption加密方式，WPA2-PSK
- Key密码：wsjdw,8181
- Save & Apply 保存并应用设置



这时，所有无线设备都可以通过OpenWrt路由器上网了。

## OpenWrt管理界面登陆密码设置

你注意到没有，网页上方有一个红色的框框(No password set!)一直在提示我们：小人不得不防，OpenWrt叫你设一个路由器管理界面登陆密码呢！

1. 点击最上面的System进入系统设置
2. 再点击Administration进入管理员设置
3. 密码Password: fanqiang
4. 确认密码Confirmation: fanqiang

5. 其他设置：下面的：Gateway ports，勾选 Allow remote hosts to connect to local SSH forwarded ports（允许远程主机连接本地SSH转发端口），这样我们就可以用SSH命令行的方式管理路由器。最后点击右下角 Save & Apply保存并应用设置。

# 网页界面怎样备份OpenWrt设置

现在有线和无线上网都正常了。应该把现有的OpenWrt设置备份一下，因为我们还要经常折腾OpenWrt，有时一个设置错误，可能就上不了网，有了备份，就可以快速恢复。

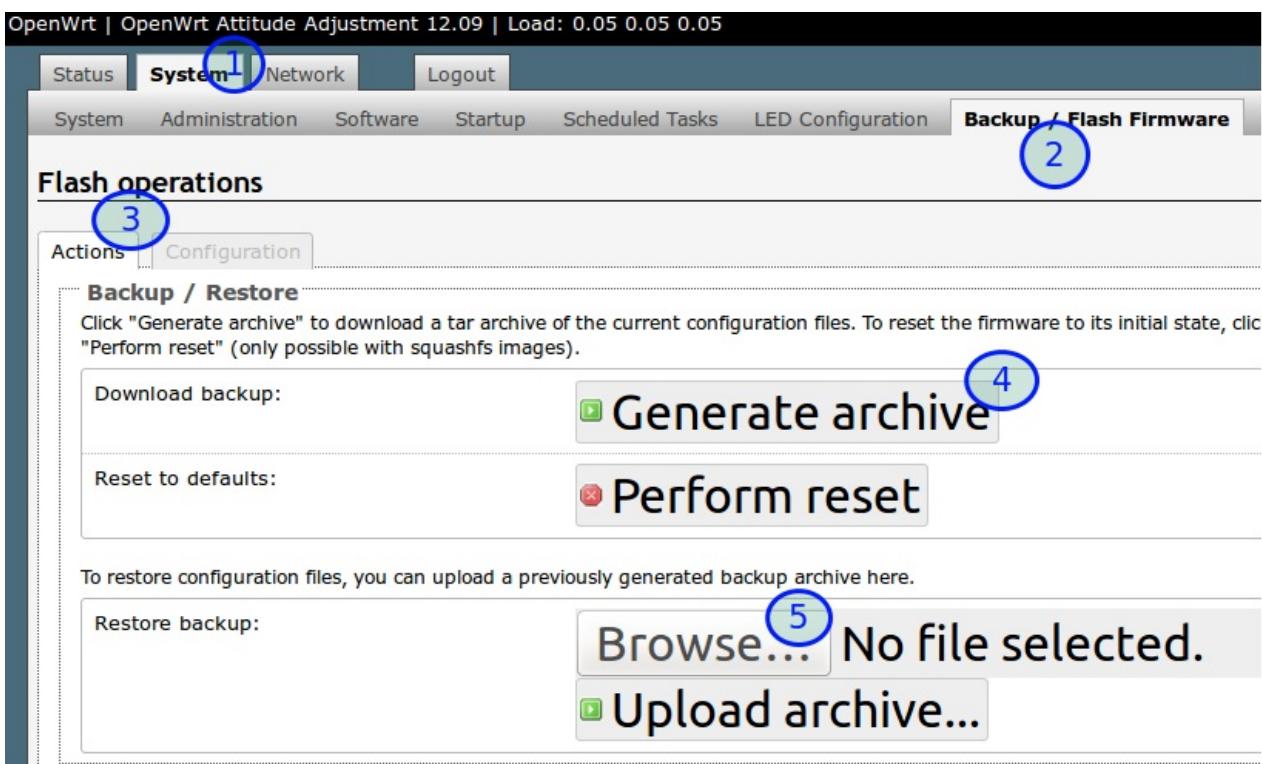
## 选择System系统设置

## 选择Backup / Flash Firmware备份恢复固件

### Actions动作

### Generate生成备份文件并保存到电脑

如果以后你要恢复备份，就点击Browse...浏览并选择先前备份的文件来恢复



# 通过LuCI网页界面升级OpenWrt固件

---

我们现在已经给TP-Link WR2543N刷上了OpenWrt固件，并且可以正常上网了。如果要升级OpenWrt固件，又该怎么做呢？

有两个途径升级固件：

- LuCI web界面升级
- SSH命令行登录路由器升级

本节就讲 web界面升级固件的方法。

## 下载OpenWrt升级用固件sysupgrade.bin

---

下载用于WR2543N路由器的升级固件，升级用固件文件名中有sysupgrade字样。

还是到OpenWrt Wiki 页面 [TP-Link TL-WR2543ND](#)

点击 [openwrt-ar71xx-generic-tl-wr2543n-v1-squashfs-sysupgrade.bin](#) 下载。

其实这个固件的核心和我们先前安装的...factory.bin一样，我们是出于实验目的，演示升级固件的方法，

## 用前文讲过的方法从网页登录OpenWrt路由器

---

### 开始升级OpenWrt固件

---

1. System系统
2. Backup / Flash Firmware备份或刷新固件
3. Flash new firmware, Browse...选择我们刚下载下来的固件
4. Flash image...刷新固件

注：如果Keep settings保持勾选，升级固件后，原来的设置就会保留，不用重新设置拨号上网参数。

1

2

3

4

Status System Network Logout

System Administration Software Startup Scheduled Tasks LED Configuration Backup / Flash Firmware Reboot

## Flash operations

Actions Configuration

### Backup / Restore

Click "Generate archive" to download a tar archive of the current configuration files. To reset the firmware to its initial state, click "Perform reset".

Download backup:

Reset to defaults:

To restore configuration files, you can upload a previously generated backup archive here.

Restore backup:

### Flash new firmware image

Upload a sysupgrade-compatible image here to replace the running firmware. Check "Keep settings" to retain the current configuration (req image).

Keep settings:

Image:

# 怎样进入OpenWrt 安全恢复模式(WR2543N为例)

有时候，我们可能操作失误，无法进入LuCI网页界面管理恢复固件，这时就需要进入安全模式来恢复了。

不同的路由器，进入安全模式的方法可能有所差别，本文系列适用于 TP-LINK WR2543N。

安全模式是玩OpenWrt的救命仙丹。能熟练进入安全模式来恢复设置，是OpenWrt已经上手的一个标志。

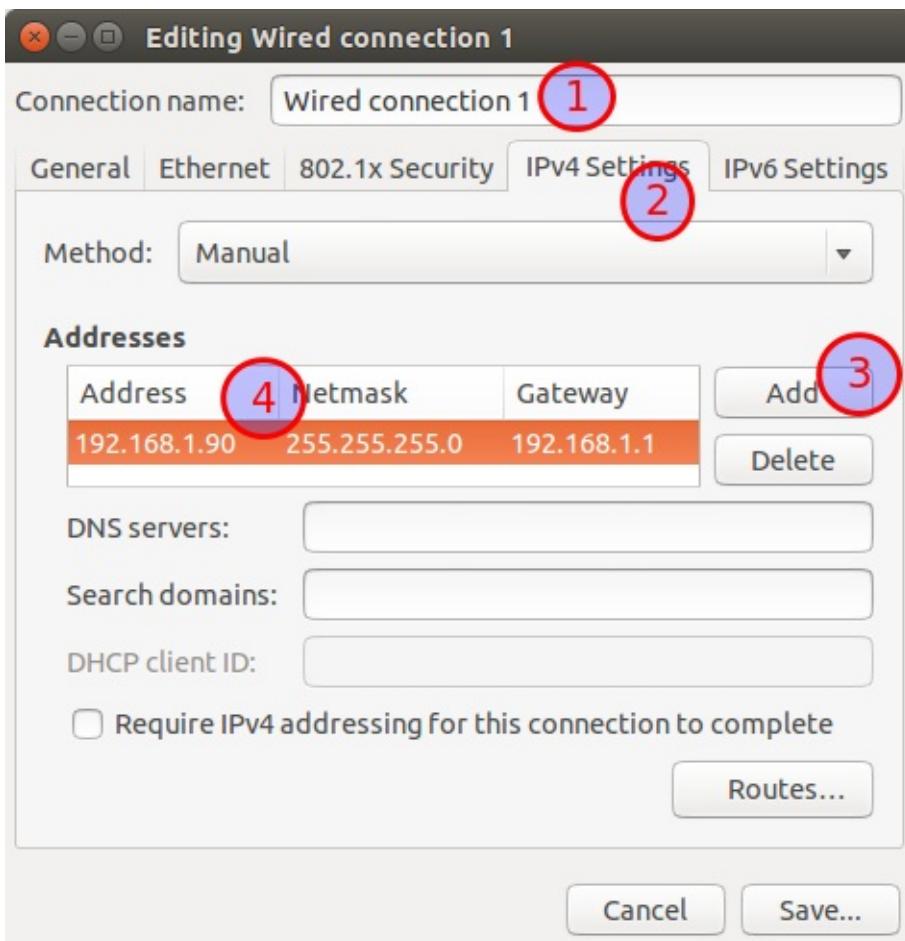
进入安全模式时，没有无线连接可用，所以我们要有线的方式登陆OpenWrt。OpenWrt默认的IP地址是192.168.1.1，我们要设置电脑有线连接的IP地址类似于192.168.1.x，其中x是2至255的数字。

WR2543N无线路由器进入OpenWrt安全模式的方法：

1. 用网线把路由器和电脑连接起来，设置电脑网卡的IPv4地址

以Ubuntu为例，点击桌面右上角连接符号，选择 **Edit Connections**，再选择 Ethernet连接，点击 **Edit** 按钮，在弹出的窗口中选择 **IPv4 Settings**，Method选择Manual，Address栏点击Add，设置如下：

- Address: 192.168.1.90
- Netmask: 255.255.255.0
- Gateway: 192.168.1.1



2. 在Ubuntu运行命令：

```
sudo tcpdump -A -i eth0 port 4919 and udp
```

3. 重启路由器，当WR2543N的锁形指示灯刚一开始闪烁时，立即按路由器背面的wps按钮3次

4. Ubuntu命令行界面出现：

```
| Please press button now to enter failsafe
```

```
donald@Software-Download:/ $ sudo tcpdump -Ani eth0 port 4919 and udp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
20:08:41.122133 IP 192.168.1.1.46561 > 192.168.1.255.4919: UDP, length 1001
E.....@. ....7..!...Please press button now to enter failsafe.....
.
.
.
.
.
```

5. Ubuntu命令行执行(有时可以不需tcpdump直接telnet)：

```
telnet 192.168.1.1
```

这时就成功登陆了OpenWrt，如下图：

```
donald@Software-Download:/ $ telnet 192.168.1.1
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.

==== IMPORTANT =====
Use 'passwd' to set your login password
this will disable telnet and enable SSH
-----
-----
```

BusyBox v1.19.4 (2013-03-14 11:28:31 UTC) built-in shell (ash)
Enter 'help' for a list of built-in commands.

```
-----
```

-----

\_\_\_\_	.-. .-. .-.	\_\_\_\_					.-.	\_											
-		-	-\_																
\_\_\_\_																			
		W I R E L E S S F R E E D O M																	
-----

ATTITUDE ADJUSTMENT (12.09, r36088)

```
-----
```

\* 1/4 oz Vodka      Pour all ingredients into mixing  
\* 1/4 oz Gin        tin with ice, strain into glass.  
\* 1/4 oz Amaretto  
\* 1/4 oz Triple sec  
\* 1/4 oz Peach schnapps  
\* 1/4 oz Sour mix  
\* 1 splash Cranberry juice

```
-----
```

root@(none):/#

1. 设置登陆OpenWrt SSH登陆密码：

```
passwd  
#输入密码 fanqiang
```

如果出现：

```
passwd: /etc/passwd: Read-only file system  
passwd: can't update passwd file /etc/passwd
```

就输入 `mount_root` 再重新passwd设置管理员密码。如下图：

```
-----  
root@(none):/# passwd  
Changing password for root  
New password:  
Retype password:  
passwd: /etc/passwd: Read-only file system  
passwd: can't update password file /etc/passwd  
root@(none):/# mount_root  
switching to jffs2  
root@(none):/# passwd  
Changing password for root  
New password:  
Retype password:  
Password for root changed by root  
root@(none):/# █
```

telnet登陆路由器后，可以用vi命令修改设置。

这时如果你试图用浏览器登陆192.168.1.1进入管理界面的话，可能失败。

重启路由器，路由器锁形指示灯先是慢闪，到变成常亮时，你又可以登陆 192.168.1.1管理界面。一切恢复正常。

# OpenWrt sysupgrade 命令行更新固件到最新版

下面我们要使用 sysupgrade 更新固件到trunk最新版。

要注意的是，trunk包含试验的功能，可能不稳定，刷机风险自己承担。

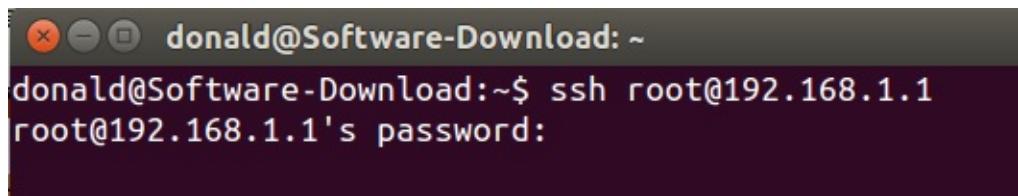
在浏览器里登陆 192.168.1.1 进行固件升级是比较简单的。今天我们要尝试的的是命令行刷机升级。命令行的方式更强大。

## SSH登录路由器

在Ubuntu里，按Ctrl+Alt+T打开命令行终端，输入：

```
ssh root@192.168.1.1
```

输入密码，登录成功。



```
donald@Software-Download: ~
donald@Software-Download:~$ ssh root@192.168.1.1
root@192.168.1.1's password:
```

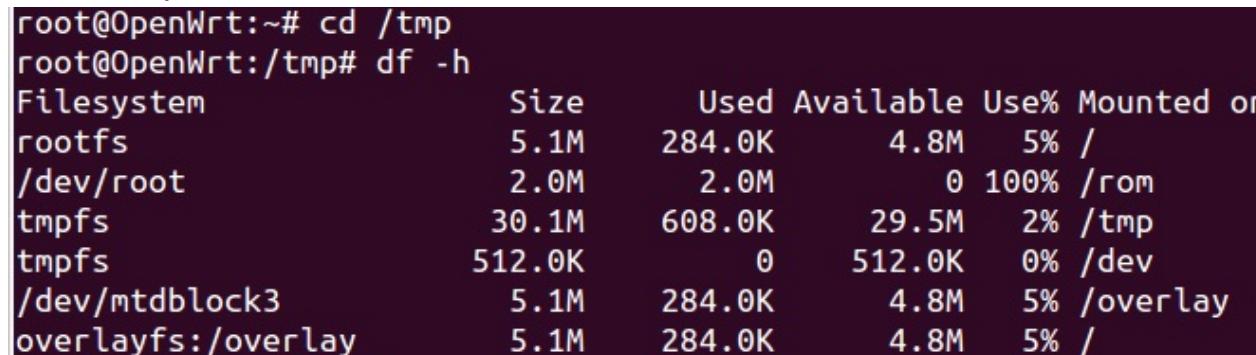
## 进入OpenWrt /tmp目录

```
cd /tmp
```

## 检查OpenWrt路由器是否有足够的内存

```
df -h
```

可以看出，/tmp 还有29.5MB可用空间，而升级固件在3MB左右，足够了。



Filesystem	Size	Used	Available	Use%	Mounted on
rootfs	5.1M	284.0K	4.8M	5%	/
/dev/root	2.0M	2.0M	0	100%	/rom
tmpfs	30.1M	608.0K	29.5M	2%	/tmp
tmpfs	512.0K	0	512.0K	0%	/dev
/dev/mtdblock3	5.1M	284.0K	4.8M	5%	/overlay
overlayfs:/overlay	5.1M	284.0K	4.8M	5%	/

## 下载OpenWrt最新trunk版本固件

1. 在Ubuntu里浏览器打开 <http://downloads.openwrt.org/snapshots/trunk/>
2. TP-LINK WR2543N路由器的芯片类型是ar71xx，就点击 ar71xx 目录进入。要注意，路由器的芯片类型千万不能搞错，不同路由器很可能是不同的。



downloads.openwrt.org/snapshots/trunk/

# Index of /snapshots/trunk/

<a href="#">..</a>	
<a href="#">adm5120/</a>	10-Jul-2014 02:25
<a href="#">adm8668/</a>	22-May-2014 15:58
<a href="#">ar7/</a>	10-Jul-2014 22:06
<a href="#">ar71xx/</a>	09-Jul-2014 21:10
<a href="#">ar71xx.mikrotik/</a>	10-Jul-2014 08:19
<a href="#">ar71xx_nand/</a>	10-Jul-2014 07:38

- 按Ctrl+F查找自己的路由器型号。比如我输入的是 **wr2543**, 有两个固件，升级用的是 **sysupgrade.bin**文件。右键点击该链接，复制下载地址。在FireFox里是 **Copy Link Location**复制链接地址。
- 回到Ubuntu命令行终端， 下载固件到 **/tmp** 目录。TP-LINK wr2543路由器是这样的：

```
root@OpenWrt:/tmp# wget http://downloads.openwrt.org/snapshots/trunk/ar71xx/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs
```

## md5校验， 确保下载的固件完整：

```
root@OpenWrt:/tmp# wget http://downloads.openwrt.org/snapshots/trunk/ar71xx/md5sums
root@OpenWrt:/tmp# md5sum -c md5sums 2> /dev/null | grep OK
openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin: OK
```

输出结尾是OK， 说明固件是完整的。

## OpenWrt sysupgrade命令升级OpenWrt固件

```
root@OpenWrt:/tmp# sysupgrade -v openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin
...
Upgrade completed
Rebooting system...
```

过约2分钟， 等路由器重启成功， 如果没有意外， 会发现有线和无线上网都正常。但浏览器192.168.1.1无法登陆， 因为 snapshots版本固件是不带LuCI网页管理界面的。没有也好， 可以节省路由器的存储空间， 也可以学习一下命令行管理OpenWrt路由器。

## 参考：

- [generic.sysupgrade](#)
- [sysupgrade source code](#)
- [sysupgrade doc](#)

# 命令行 uci 设置 OpenWrt 上网参数

如果路由器可以正常上网的前提，我们可以ssh登陆路由器，直接在路由器的/tmp目录wget下载最新版固件并sysupgrade命令进行固件升级。

有时候，路由器无法上网，这时候，可以在电脑里下载好固件，再把固件复制到路由器，再sysupgrade升级或设置其他参数。

只要能进入路由器的安全模式，并ssh登陆路由器，一切都不是问题。

## Ubuntu 下载 OpenWrt for TP-LINK wr2543N trunk 版固件

```
cd /opt  
wget http://downloads.openwrt.org/snapshots/trunk/ar71xx/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin
```

## scp 复制固件到 OpenWrt 路由器 /tmp 目录

```
scp openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin root@192.168.1.1:/tmp/
```

## ssh 登陆 OpenWrt 路由器

```
ssh root@192.168.1.1  
cd /tmp
```

## sysupgrade 升级固件并取消保留原来配置文件

注意，升级后将无法上网，也没有LuCI网页界面可以设置，必须以命令行方式设置好上网参数。

如果在下面的实验中，命令行方式无法搞定路由器上网，就只能在电脑里下载好带LuCI的固件，scp复制固件到路由器升级固件，然后以网页方式设置上网。

在进行这一步前，确保你熟练掌握以前部分教程。

```
root@OpenWrt:/tmp# sysupgrade -n openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin
```

参数 -n 表示升级时不保留原来的配置文件。固件刷好后会自动重启，这时要用前文教程讲过的方法进入OpenWrt安全模式，登陆路由器并重新设置root密码。

下面假设你已经登陆了路由器并设好了root密码。

## OpenWrt uci 命令行设置拨号上网：

```
root@OpenWrt: uci set network.wan.proto='pppoe'  
root@OpenWrt: uci set network.wan.username='adslname'  
root@OpenWrt: uci set network.wan.password='adslpassword'  
root@OpenWrt: uci set network.wan.peerdns=0
```

adslname 替换成你自己的拨号上网用户名，adslpassword 替换成你自己的密码。

## OpenWrt uci命令行设置无线上网：

```
root@OpenWrt: uci set wireless.@wifi-device[0].channel=11
root@OpenWrt: uci set wireless.@wifi-device[0].txpower=17
root@OpenWrt: uci set wireless.@wifi-device[0].disabled=0
root@OpenWrt: uci set wireless.@wifi-device[0].country='CN'
root@OpenWrt: uci set wireless.@wifi-iface[0].mode='ap'
root@OpenWrt: uci set wireless.@wifi-iface[0].ssid='eastking-wr2543'
root@OpenWrt: uci set wireless.@wifi-iface[0].encryption='psk2'
root@OpenWrt: uci set wireless.@wifi-iface[0].key='wsjdw,8181'
```

### uci设置说明：

- channel 信道
- txpower 功率
- disabled 是否启用无线
- ssid 名称，推荐后面以路由器型号结尾，这样调试多个路由器时不会混淆。
- encryption 加密方式
- key 无线密码，如果你照上文的设置不动，好处是忘记密码时可以上 <http://www.github.com/softwaredownload/openwrt-fanqiang> 来查看。

## 允许远程主机用ssh的方式登陆路由器及设置时区

```
root@OpenWrt: uci set dropbear.@dropbear[0].GatewayPorts='on'
root@OpenWrt: uci set system.@system[0].zonename='Asia/Shanghai'
root@OpenWrt: uci set system.@system[0].timezone='CST-8'
```

## ssh登陆OpenWrt相关高级设置（你可能暂时用不到）

```
root@OpenWrt: uci set dropbear.@dropbear[0].Port=22
root@OpenWrt: uci set dropbear.@dropbear[0].PasswordAuth=off
root@OpenWrt: uci set dropbear.@dropbear[0].RootPasswordAuth=off
```

### 说明（不懂千万别乱设）：

- Port ssh默认端口就是22,可以改成其他的提高安全性
- PasswordAuth ssh是否启用密钥登陆。如果你改成off，又没有设置好ssh私钥和安装好LuCI，你将无法ssh方式登陆路由器，唯一的办法就是安全恢复模式登陆重新开始设置。
- RootPasswordAuth 是否允许root用密码登陆，如果已经设置好了ssh私钥就可以改成off增加安全性。

## 启用新的网络和无线设置

```
root@OpenWrt: /etc/init.d/dropbear restart
root@OpenWrt: /etc/init.d/system restart
root@OpenWrt: /etc/init.d/network restart
```

怎么样，有线和无线上网又都回来了吧！

注意，有的人在网上贴出了他的完整配置文件/etc/config/network 和/etc/config/wireless，如果你复制他的文件覆盖你的文件，再修改用户名和密码，可能会出问题，因为不同路由器的硬件配置可能不同。

# OpenWrt + shadowsocks-libev自动翻墙

---

相信经过前面的教程，大家对OpenWrt和Linux Ubuntu有一定的熟悉了。如果还不熟悉Ubuntu，就安装Ubuntu，实际使用一个月。

前面的文章都是技术准备，有基础的读者可以略过。在本章中，我们要OpenWrt路由器安装 shadowsocks-libev来实践翻墙。

本教程已经在**github**开源，欢迎提交改进，报告bug：

<https://github.com/softwaredownload/openwrt-fanqiang>

# 什么是shadowsocks-libev翻墙软件

---

shadowsocks-libev 是一个 shadowsocks 协议的轻量级实现，是 shadowsocks-android, shadowsocks-ios 以及 shadowsocks-openwrt 的上游项目。其具有以下特点：

1. 体积小巧。静态编译并打包后只有 100 KB。
2. 高并发。基于 libev 实现的异步 I/O，以及基于线程池的异步 DNS，同时连接数可上万。
3. 低资源占用。几乎不占用 CPU 资源，服务器端内存占用一般在 3MB 左右。
4. 跨平台。适用于所有常见硬件平台，已测试通过的包括 x86, ARM 和 MIPS。也适用于大部分 POSIX 的操作系统或平台，包括 Linux, OS X 和 Cygwin 等。
5. 协议及配置兼容。完全兼容 shadowsocks 协议，且兼容标准实现中的 JSON 风格配置文件，可与任意实现的 shadowsocks 客户端或服务端搭配使用。

shadowsocks-libev 包括服务端和客户端两部分，一共三个模块。

1. ss-server : 服务器端，部署在远程服务器，提供 shadowsocks 服务。
2. ss-local : 客户端，提供本地 socks5 协议代理。
3. ss-redir : 客户端，提供本地透明代理，需要与 NAT 配合使用
4. ss-tunnel: 客户端，本地端口转发

官网地址：

<https://github.com/ciowwindy/shadowsocks-libev>

参考：

- [shadowsocks-libev 1.2 发布](#)

# 翻墙软件Shadowsocks-libev服务端设置

要利用 shadowsocks-libev翻墙，首先要有一台国外的服务器安装并运行shadowsocks 服务端。如果还没有服务器,可以到[digitalocean](#) 购买一台虚拟服务器，价格非常便宜。

注册DO并绑定支付方式后，登陆，然后Create Droplet:

- Hostname 主机名，任意填，比如 shadowsocks
- Select Site 默认的512MB那款就够了
- Select Region 选择San Francisco 1
- Select Image 选择 Ubuntu的第一个也是最新的版本
- Create Droplet 创建虚拟服务器

创建服务器后如何设置ssh登陆，请自行搜索相关教程。

## Ubuntu安装 shadowsocks-libev服务端

```
sudo add-apt-repository "deb http://shadowsocks.org/debian wheezy main"
sudo apt-get update
sudo apt-get install shadowsocks
```

上述命令的效果：

- 把 deb <http://shadowsocks.org/debian> wheezy main 加到 /etc/apt/sources.list
- 安装ss-local ss-redir ss-server ss-tunnel到 /usr/bin
- 启动文件 /etc/init.d/shadowsocks
- 配置文件 /etc/shadowsocks/config.json
- 一些默认启动配置 /etc/default/shadowsocks

## Ubuntu源码编译 shadowsocks-libev 最新版

```
sudo apt-get install build-essential autoconf libtool libssl-dev git
cd /opt
git clone https://github.com/ciowwindy/shadowsocks-libev.git
cd shadowsocks-libev
./configure && make
```

## 使用最新版shadowsocks-libev服务端软件 ss-server

```
sudo /etc/init.d/shadowsocks stop
sudo cp /opt/shadowsocks-libev/src/ss-server /usr/bin/
```

## 编辑shadowsocks-libev配置文件

```
sudo vi /etc/shadowsocks/config.json
```

改成类似如下：

```
{  
    "server":"107.89.0.12",  
    "server_port":8345,  
    "password":"killgfw",  
    "method":"aes-256-cfb"  
}
```

必须把server IP地址改成你自己的，其他可以不改

## 控制shadowsocks-libev的方法

```
sudo service shadowsocks start  
sudo service shadowsocks stop  
sudo update-rc.d shadowsocks defaults 99  
sudo update-rc.d -f shadowsocks remove
```

上面第三行设置随机启动 Shadowsocks-libev, 第四行取消shadowsocks-libev随机启动

## 启动shadowsocks-libev服务端

```
sudo service shadowsocks start
```

参考：

- [shadows-libev on github](#)
- [create a simple linux daemon](#)

# OpenWrt路由器运行shadowsocks-libev客户端

## 下载shadowsocks-libev OpenWrt客户端(ar71xx为例)

打开 <http://shadowsocks.org/en/download/clients.html>

如果这个页面打不开，可能是被墙了，要用代理上，或者在下面网址下载：

<https://software-download.name/2014/shadowsocks-libev-polarssl-ar71xx-ipk-latest/>

本文系列以TP-LINK wr2543n路由器为例，用的是ar71xx芯片，就下载 [shadowsocks-libev-polarssl\\_1.x.x\\_ar71xx.ipk](https://software-download.name/2014/shadowsocks-libev-polarssl_ar71xx_ipk_later)

在Ubuntu下载：

```
cd /opt
wget http://shadowsocks.org/nightly/shadowsocks-libev-polarssl_1.4.6_ar71xx.ipk
```

## 安装shadowsocks-libev客户端到OpenWrt路由器

```
/opt$ scp shadowsocks-libev-polarssl_1.*.*_ar71xx.ipk /opt$ root@192.168.1.1:/tmp/
/opt$ ssh root@192.168.1.1
root@OpenWrt:~# cd /tmp
root@OpenWrt:~# opkg install shadowsocks-libev-polarssl_1.*.*_ar71xx.ipk
```

## 修改shadowsocks-libev客户端配置

```
root@OpenWrt:~# vi /etc/shadowsocks.json
```

改成类似如下：

```
{
  "server": "107.89.0.12",
  "server_port": 8345,
  "local_port": 8024,
  "password": "killgfw",
  "method": "aes-256-cfb"
}
```

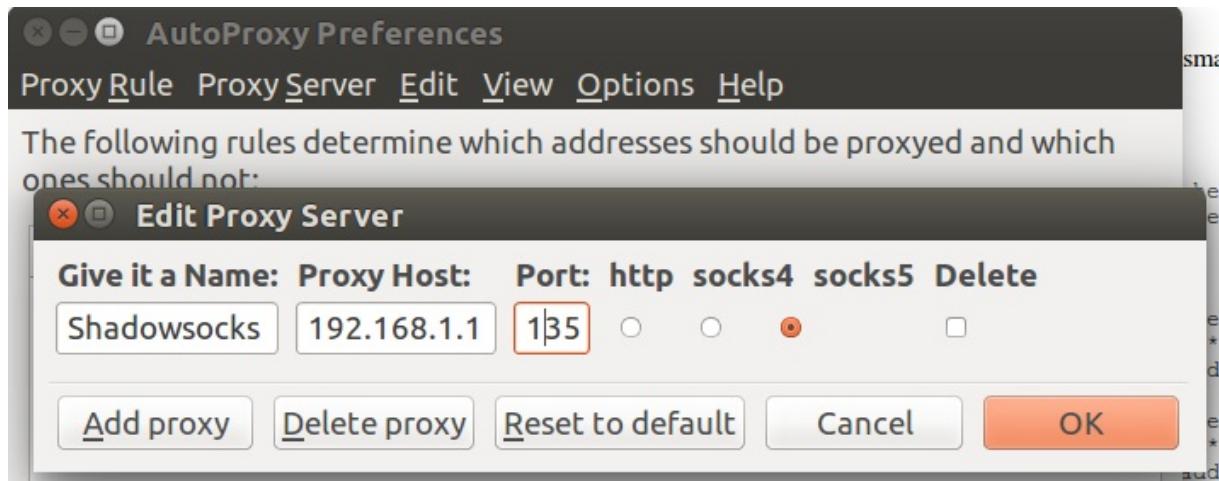
注意，server IP必须修改你的实际IP。

## shadowsocks代理上网测试

- 启动shadowsocks客户端:

```
root@OpenWrt:~# ss-local -c /etc/shadowsocks.json
```

- Ubuntu浏览器代理上网设置，以FireFox配合AutoProxy为例，增加Proxy Server, Proxy Host填192.168.1.1,Port是8024，勾选Sock5.如下图：



Ubuntu设置AutoProxy的默认代理是shadowsocks,就可以打开被墙的网站如[YouTube.com](https://www.youtube.com)

以前我在每台电脑上都运行一个shadowsocks客户端，每台电脑都要像上面这样配置浏览器代理上网翻墙。

现在路由器里安装了shadowsocks，所有有线和无线上网设备都不用分别安装shadowsocks了，是比以前方便了。但是还是太复杂，如果家里有十台上网设备，所有要连国外网站的软件都可能要配置代理访问，有些软件还根本没有设置代理的接口。有没有更简单的方法呢？

# 史上最通俗易懂的OpenWrt翻墙路由器解释

## 什么是域名和IP地址

每个网站都可以有两个唯一标识：域名和IP地址。域名相当于人的名字，IP地址相当于该人使用的电话号码。（不同之处：域名是唯一的，人的名字会有重名）

网站为什么要有两个标识？域名是为了方便人类记忆的，比如[YouTube.com](https://www.youtube.com)，而电脑处理却喜欢处理数字，纯数字格式的IP地址就是为了让电脑查找计算方便些。

## 通过域名查询IP的那些事情

我们在浏览器地址栏里输入 [www.youtube.com](https://www.youtube.com) 并回车，到底会发生哪些不可思议的事情呢：

1. 浏览器问就近的某台电脑（叫域名服务器）：Hi, youtube.com的IP地址是什么？
2. 域名服务器：不就是 74.125.239.98
3. 浏览器：谢谢。我就到你给我的地址去找内容了

还有种情况，浏览器第一次问的域名服务器不知道某域名的IP地址：

1. 浏览器问就近的域名服务器：Hi, youtube.com的IP地址是什么？
2. 域名服务器：这个我不知道哇，我帮你问问我的上线
3. 上线服务器：我也不知道哇，我也只好问我的上线，等等，别挂掉
4. 某域名服务器：这么简单还来问我，不就是 74.125.239.98
5. 浏览器：谢谢。我就到你给我的地址 74.125.239.98 去找内容

## 白脸很忙，不看YouTube(看不懂？)

在中国，YouTube为什么被封？YouTube有几千万，上亿个视频，如果某几个视频让某些人看了不爽，就来个宁可错杀百万，不可放过一个，把整个YouTube给封了，全国人民都无法正常访问YouTube了。

这个时候，又发生了哪些不可告人的事情呢？

1. 浏览器问就近的域名服务器：喂, youtube.com的IP地址是什么？
2. 中国的某域名服务器：这我知道，44.44.44.44，（心里嘀咕，我给你的是太平洋海底的地址，你能找到内容才怪呢，白脸（领导）很忙，天朝很好，访问这种破网站干啥，满屏洋文，我怎么看得懂，哼）
3. 浏览器：谢谢。我这就去找主人需要的内容。。。找了好久，还是什么也没找到，我的命怎么这么苦。。。

阳光底下，每时每刻每秒，这样龌龊的事情在发生千次，万次，亿次...

## 白脸不知道TCP？

人民要学习新知识，不能容忍这样的事情，于是想到了一些办法，国内的域名服务器要说谎，那我直接就问国外的服务器比如google的8.8.8或8.8.4.4，人家才不会这么卑鄙无耻下流无底线。直接问google的域名服务器的办法存活了一段时间，后来白脸（领导不晒太阳的）又知道了，又不高兴了，于是google的域名服务器IP地址也被封了。

网民的力量是无穷的，有人又发现了，查询域名IP地址的方式有udp和tcp两种，领导暂时只知道udp，我用tcp的方式去问国外域名服务器，就可以得到网站域名对应的真实IP了。

写这篇文章的时候，tcp的方式依然有效。但我怀疑，白脸迟早又会不高兴。8.8.8或8.8.4.4树大招风，而且白脸最容易记住的就是8和4这两个数字，白脸要发，白脸怕死。所以，我们索性不用8和4了，我们用加密的方式(shadowsockss ss-tunnel)向自己的服务器查询，领导这下该满意了吧！

# 太阳要升起， 网民要雄起

---

但是， 还有问题没有解决：

网站有两种， 国内的和国外的。如果不分国内国外全部都到国外去查询域名的IP， 访问国内的网站就会变慢。虽然有心逃离， 还是无法割断哪。

有几种解决方案：

1. 建国外重要网站名单， 简称外单， 外单上的域名都到国外去查询IP， 其他就在国内查询。

如果IP地址在外单上， 就加密访问， 领导不知道我访问了这个地址， 这样领导的心情可能会好些。

2. 同样是建立外单。不同的是， 我不想花费精力去区分某个IP是不是在外单上， IP地址可能经常在变， 这样做不怕累吗。我的办法是， 不是亚洲的IP， 全部加密访问。

3. 每个人的用途不同， 谁有本事建立通用的外单？

即使有人建立了包含很多域名的外单， 网站内容往往是互相引用的， 某外单上网站引用了不在外单上的被封网站， 导致打网站贼慢， 这个该怎么办？难道要手动查看网页源代码， 一个个地搜索查找， 逐个测试？

最简单有效的方法， 是给国内重要网站建立名单， 简称内单。内单上的网站都在国内dns， 其他网站全部到国外dns。访问非亚洲的IP都流量加密。

第三种方案， 就是本教程使用的方案， 是目前来说比较好的方案。

## OpenWrt翻墙路由器内部发生的故事(千万别告诉白脸)：

1. 浏览器：喂， 谁知道YouTube.com的IP， 主人要用
2. 路由器：稍等， 我查下主人设置的内单， 稍等。。。不在内单， 我通过秘密通道查
3. 浏览器：喂， 告诉我baidu.com的IP
4. 路由器：哇， 内单， 马上就给你
5. 浏览器：请给我IP地址60.188.5.6的内容
6. 路由器：等下， 立即就好。。。亚洲IP， 该那就那去取内容。不是亚洲IP， 借道主人的秘密通道去取内容

# 配置OpenWrt路由器智能自动翻墙

## OpenWrt路由器用dnsmasq转发国内重要域名查询

OpenWrt默认自带dnsmasq，我们只要配置一下就好了。ssh登陆OpenWrt路由器后：

- 建立dnsmasq.d目录：

```
root@OpenWrt:~# mkdir /etc/dnsmasq.d
root@OpenWrt:~# echo "conf-dir=/etc/dnsmasq.d" >> /etc/dnsmasq.conf
```

- OpenWrt安装GNU wget以支持https下载，下载国内重要网站名单，用国内域名服务器查询IP地址

```
root@OpenWrt:~# cd /etc/dnsmasq.d
root@OpenWrt:/etc/dnsmasq.d# opkg install wget
root@OpenWrt:/etc/dnsmasq.d# wget -4 --no-check-certificate -O /etc/dnsmasq.d/accelerated-domains.china.conf https://raw.githubusercontent.com/linrongtian/accelerated-domains-china/master/accelerated-domains.china.conf
root@OpenWrt:/etc/dnsmasq.d# wget -4 --no-check-certificate -O /etc/dnsmasq.d/bogus-nxdomain.china.conf https://raw.githubusercontent.com/linrongtian/accelerated-domains-china/master/bogus-nxdomain.china.conf
```

注：accelerated-domains.china.conf文件中的条目举例：

```
server=/10010.com/114.114.114.114
server=/115.com/114.114.114.114
```

意思是，访问10010.com这个结尾的域名时，dnsmasq会转发到国内的域名服务器114.114.114.114进行dns查询。

## 其他域名，转发到shadowsocks-libev ss-tunnel指定的端口dns查询

```
root@OpenWrt:/etc/dnsmasq.d# echo "server=/#/127.0.0.1#7913" > gfwlist.conf
```

上面#是通配符，代表所有域名。dnsmasq匹配域名的特点是详细特征优先匹配，因此会先匹配accelerated-domains.china.conf上的域名，如果不匹配，再匹配这条规则：转发到本地端口7913进行域名查询。

后面我们会配置shadowsocks-libev的本地客户端ss-tunnel转发本地端口7913的查询到远程自建服务器。

## 配置shadowsocks本地客户端ss-redir启动和停止函数

```
root@OpenWrt:/etc/dnsmasq.d# vi /etc/init.d/shadowsocks
```

其中start和stop函数修改成如下：

```
start() {
    service_start /usr/bin/ss-redir -c /etc/shadowsocks.json
    service_start /usr/bin/ss-tunnel -c /etc/shadowsocks.json -l 7913 -L 8.8.8.8:53 -u
    /usr/bin/shadowsocks-firewall
}

stop() {
    service_stop /usr/bin/ss-redir
    service_stop /usr/bin/ss-tunnel
    /etc/init.d/firewall restart
}
```

**shadowsocks**本地客户端配置文件**start**函数说明：

- **service\_start /usr/bin/ss-tunnel -c /etc/shadowsocks.json -l 7913 -L 8.8.8.8:53 -u** 监听本地7913端口，转发到自己的服务器的53端口向8.8.8.8查询
- **/usr/bin/shadowsocks-firewall** dnsmasq只是负责域名查询分配转发，查询到IP地址后，是否需要通过shdowsocks加密请求内容，要在shdowsocks-firewall里进行设置

## 配置**iptables**防火墙转发IP和端口

```
root@OpenWrt:~# cd /usr/bin
touch shadowsocks-firewall
chmod +x shadowsocks-firewall
vi shadowsocks-firewall
```

shdowsocks-firewall内容如下：

```

#!/bin/sh

#create a new chain named SHADOWSOCKS
iptables -t nat -N SHADOWSOCKS

# Ignore your shadowsocks server's addresses
# It's very IMPORTANT, just be careful.

iptables -t nat -A SHADOWSOCKS -d 107.89.0.12 -j RETURN

# Ignore LANs IP address
iptables -t nat -A SHADOWSOCKS -d 0.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 10.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 127.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 169.254.0.0/16 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 172.16.0.0/12 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 192.168.0.0/16 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 224.0.0.0/4 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 240.0.0.0/4 -j RETURN

# Ignore Asia IP address
iptables -t nat -A SHADOWSOCKS -d 1.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 14.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 27.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 36.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 39.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 42.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 49.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 58.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 59.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 60.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 61.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 101.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 103.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 106.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 110.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 111.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 112.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 113.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 114.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 115.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 116.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 117.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 118.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 119.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 120.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 121.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 122.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 123.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 124.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 125.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 126.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 169.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 175.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 180.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 182.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 183.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 202.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 203.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 210.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 211.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 218.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 219.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 220.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 221.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 222.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 223.0.0.0/8 -j RETURN

# Anything else should be redirected to shadowsocks's local port
iptables -t nat -A SHADOWSOCKS -p tcp -j REDIRECT --to-ports 8024
# Apply the rules
iptables -t nat -A PREROUTING -p tcp -j SHADOWSOCKS

```

## iptables防火墙设置含义

- 如果本地发出请求到shadowsocks服务端所在的服务器,就返回, 不作任何特殊处理。
- 如果本地发出请求到局域网, 也立即返回

- 如果发出请求到亚洲的IP地址，也立即返回
- 剩下的IP内容请求，全部转发到shadowsocks-libev本地客户端ss-redir监听的端口，由ss-redir负责和服务端进行加密通讯。（手下报告访问youtube的屁民为个位数，领导心里那个高兴啊。可惜经过加密，内容传输速度会有下降）

## 防火墙设置重要说明：

- 你必须把上面的107.89.0.12换成你服务器真实的IP地址
- `iptables -t nat -A SHADOWSOCKS -p tcp -j REDIRECT --to-ports 8024` 这里的8024必须和OpenWrt路由器 /etc/shadowsocks.json里的 `local_port`一样，也就是说，如果 /etc/shadowsocks.json里 "`local_port":1090`, 那这里的8024也要改成1090
- 其他可以保持默认

## 控制shadowsocks本地客户端的方法

```
root@OpenWrt:~# /etc/init.d/shadowsocks stop
root@OpenWrt:~# /etc/init.d/shadowsocks start
root@OpenWrt:~# /etc/init.d/shadowsocks enable
root@OpenWrt:~# /etc/init.d/shadowsocks disable
```

### 说明：

- stop: 停止shadowsocks
- start: 运行shadowsocks
- enable: 设置shadowsocks在OpenWrt路由器启动时自动启动
- disable: 取消shadowsocks随机启动

## 启动并测试shadowsocks-libev本地客户端

确保所有设置无误后，可以启动测试一下：

```
root@OpenWrt:~# /etc/init.d/dnsmasq restart
root@OpenWrt:~# /etc/init.d/shadowsocks stop
root@OpenWrt:~# /etc/init.d/shadowsocks start
root@OpenWrt:~# /etc/init.d/shadowsocks enable
```

然后在Ubuntu电脑，手机等设备上打开[youtube.com](https://www.youtube.com), [twitter.com](https://www.twitter.com)

## 下载配置文件的最新版

```
git clone https://github.com/softwaredownload/openwrt-fanqiang
```

git clone 项目到本地后，可以进入 openwrt 目录查看文件。

如果所有设置都正确，应该可以较快速度打开被墙网站。

# OpenWrt自动更新设置和屏蔽广告

## OpenWrt路由器自动更新国内重要网站名单

登陆路由器后：

```
root@OpenWrt:~# cd /usr/bin  
root@OpenWrt:~# touch updatednsmasq  
root@OpenWrt:~# chmod +x updatednsmasq  
root@OpenWrt:~# vi updatednsmasq
```

增加如下内容：

```
#!/bin/sh  
cnlist() {  
    wget -4 --no-check-certificate -O /etc/dnsmasq.d/accelerated-domains.china.conf https://raw.githubusercontent.com/felixonmars/  
    wget -4 --no-check-certificate -O /etc/dnsmasq.d/bogus-nxdomain.china.conf https://raw.githubusercontent.com/felixonmars/dnsmasq  
}  
cnlist
```

## 国内广告屏蔽

dnsmasq支持基于域名的广告屏蔽方式，这种屏蔽方式比较粗糙，屏蔽效果会比浏览器广告屏蔽插件差很多，并且可能导致有些网站不能顺利打开。

在updatednsmasq文件中加入：

```
adblock() {  
    wget -4 --no-check-certificate -O - https://easylist-downloads.adblockplus.org/easylistchina+easylist.txt |  
    grep ^\|\|[^\*]*\^$ |  
    sed -e 's:||:address\|=V:' -e 's:\^:/127\.0\.0\.1:' | uniq > /etc/dnsmasq.d/adblock.conf  
  
    wget -4 --no-check-certificate -O - https://raw.githubusercontent.com/kcschan/AdditionalAdblock/master/list.txt |  
    grep ^\|\|[^\*]*\^$ |  
    sed -e 's:||:address\|=V:' -e 's:\^:/127\.0\.0\.1:' >> /etc/dnsmasq.d/adblock.conf  
}
```

最后完整的updatednsmasq就是这样：

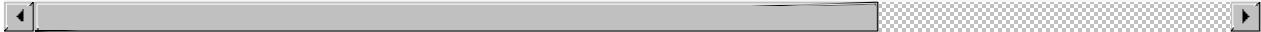
```
#!/bin/sh  
cnlist() {  
    wget -4 --no-check-certificate -O /etc/dnsmasq.d/accelerated-domains.china.conf https://raw.githubusercontent.com/felixonmars/  
    wget -4 --no-check-certificate -O /etc/dnsmasq.d/bogus-nxdomain.china.conf https://raw.githubusercontent.com/felixonmars/dnsmasq  
}  
adblock() {  
    wget -4 --no-check-certificate -O - https://easylist-downloads.adblockplus.org/easylistchina+easylist.txt |  
    grep ^\|\|[^\*]*\^$ |  
    sed -e 's:||:address\|=V:' -e 's:\^:/127\.0\.0\.1:' | uniq > /etc/dnsmasq.d/adblock.conf  
  
    wget -4 --no-check-certificate -O - https://raw.githubusercontent.com/kcschan/AdditionalAdblock/master/list.txt |  
    grep ^\|\|[^\*]*\^$ |  
    sed -e 's:||:address\|=V:' -e 's:\^:/127\.0\.0\.1:' >> /etc/dnsmasq.d/adblock.conf  
}  
cnlist  
adblock
```

## 计划任务：定时更新dnsmasq配置文件和自动重启 ss-redir

```
root@OpenWrt:~# crontab -e
```

输入以下内容：

```
*/30 * * * * isfound=$(ps | grep "ss-redir" | grep -v "grep"); if [ -z "$isfound" ]; then echo "$(date): restart ss-redir...">>/tmp/log/ss-mo  
* 12 * * * /usr/bin/updatednsmasq
```



### 计划任务说明：

- 每半小时检查shadowsocks-libev 客户端，如果退出就自动重启
- 每天中午12点更新dnsmasq配置文件

## 计划任务注意事项（2014-09-24）

目前的预编译固件是启用了计划任务的，这会有潜在的不确定性，如果更新时下载的文件存在错误，导致dnsmasq无法读取，可能造成dnsmasq无法启动，翻墙功能自然失效。

如果你刷了预编译固件后，某一天突然不能翻墙了，这时设置客户端的IP地址为和路由器同网段，登陆路由器，用ps命令查看dnsmasq进程是否启动了，如果没有启动，就重刷固件或者用原来的文件代替路由器里/etc/dnsmasq.d/下的文件。

第一次刷预编译固件后，建议立即禁用自动更新 dnsmasq.d目录下的文件，运行稳定一段时间后，需要时再加上计划任务。

下次发布固件时，将默认不启用计划任务。

### 参考：

- [install-shadowsocks-on-hg255d-openwrt-and-config-nat](#)

# OpenWrt编译翻墙固件教程

---

实践前面的教程，翻墙已经不是问题，白脸也很happy。在这一章中，我们要定制自己OpenWrt固件，刷上定制的固件，不用任何设置就自动翻墙并自动更新规则。

# 编译shadowsocks-libev for OpenWrt ipk安装包

网上也许没有适合自己路由器的shadowsocks-libev for OpenWrt ipk 安全包，即使有也不一定是最新的源码编译的。自己动手自力更生是个好办法。

下面是在Ubuntu 64bit下编译shadowsocks-libev for OpenWrt ipk安装包的步骤：

## 安装依赖库，不同的操作系统版本可能要作相应调整

```
sudo apt-get install build-essential subversion libncurses5-dev zlib1g-dev gawk gcc-multilib flex git-core gettext
```

## 下载OpenWrt源代码

```
cd /opt  
git clone git://git.openwrt.org/openwrt.git
```

## 下载shadowsocks-libev源码

```
cd /opt/openwrt/package  
git clone https://github.com/madeye/shadowsocks-libev.git
```

## 更新Feeds，使package在make menuconfig中可用，而不是真正安装或编译

```
cd /opt/openwrt  
.scripts/feeds update -a  
.scripts/feeds install -a  
make defconfig
```

## 先编译要用到的工具和库

```
make prereq && make tools/install && make toolchain/install
```

等待时间较长，可以先和大妈一起去跳个广场舞，制造更多噪音恶心一下别人。:)

## make menuconfig配置选项

有三个选项：

- y: 编译进固件
- m: 编译出安装包，但不打包进固件
- n: 排除

输入命令 make menuconfig 进入配置程序

- Target System: Atheros AR7xxx/AR9XXX (Default value, 不同的路由器，可能选择不同)

- Subtarget: Generic (Default value)
- Target Profile: TP-LINK TL-WR2543N/ND(因我们只是编译包, 这步可以不选)
- Network, 选择shadowsocks-libev-polarssl, 按m设置为编译独立ipk安装包
- Save && Exit

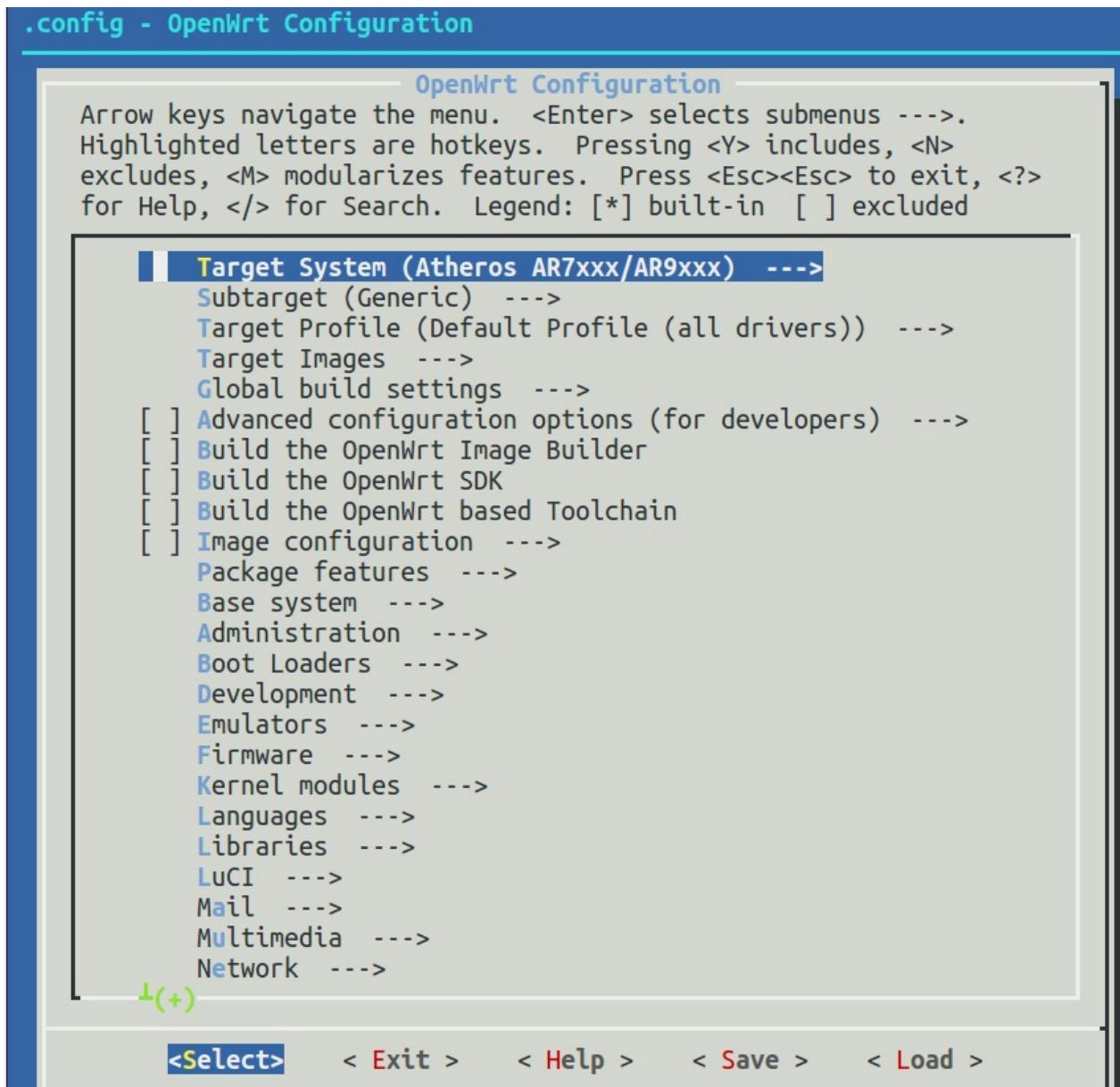


图 make menuconfig

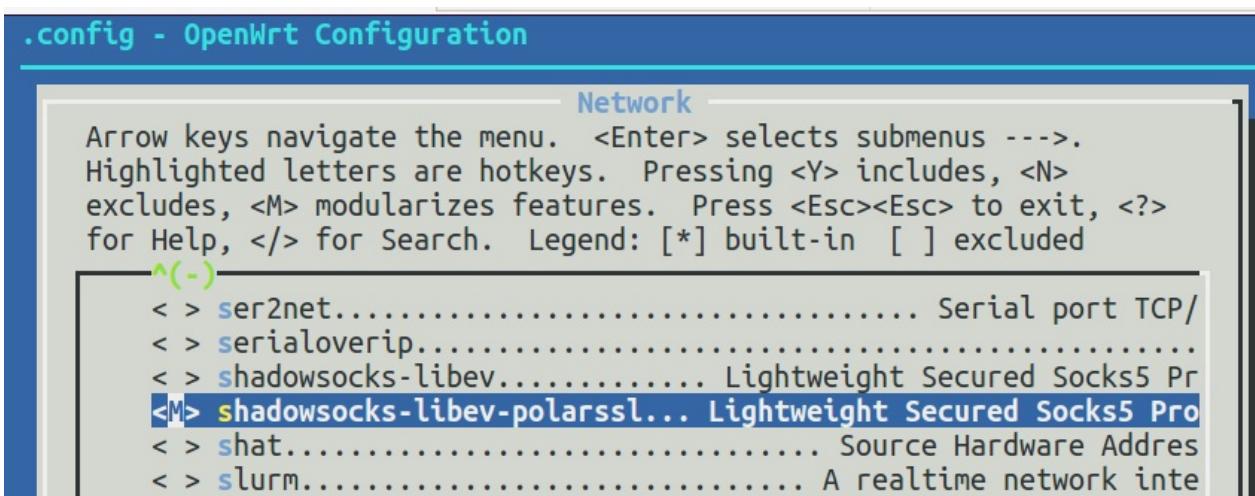


图 选择shadowsocks-libev-polarssl

# 编译shadowsocks-libev for OpenWrt

---

make V=99 package/shadowsocks-libev/openwrt/compile

## 编译结果文件

---

输出文件是 openwrt/bin/ar71xx/packages/base/shadowsocks-libev-polarssl\_1.4.6\_ar71xx.ipk

把文件scp复制到OpenWrt路由器/tmp, 就可以 opkg install shadowsocks-libev-polarssl\_1.4.6\_ar71xx.ipk 安装了。

参考：

- <http://wiki.openwrt.org/doc/howto/buildroot.exigence>
- <http://wiki.openwrt.org/doc/howto/build>
- <https://github.com/madeye/shadowsocks-libev>
- <https://0066.in/archives/312>

# 下载和设置翻墙配置文件

自己手工收集编辑翻墙所用到的配置文件是一件比较累的事情。最快的方法是 git clone 本项目，修改其中某些选项。

## 下载翻墙配置文件

```
cd /opt  
git clone https://github.com/software/download/openwrt-fanqiang
```

配置文件目录就是 /opt/openwrt-fanqiang/openwrt 不要直接在其中改，而是复制到另外目录再修改：

```
cp -R /opt/openwrt-fanqiang/openwrt /opt/openwrt-wr2543
```

其中wr2543改成你的路由器型号。

/opt/openwrt-wr2543 就是下节编译固件要用到FILES参数的值。

## 配置文件目录树

```
etc  
├── dnsmasq.conf  
├── dnsmasq.d  
│   ├── accelerated-domains.china.conf  
│   ├── adblock.conf  
│   ├── bogus-nxdomain.china.conf  
│   └── gfwlist.conf  
└── init.d  
    └── shadowsocks  
        ├── shadow  
        └── shadowsocks.json  
    └── uci-defaults  
        └── defaults  
└── usr  
    └── bin  
        └── shadowsocks-firewall  
            └── updatednsmasq
```

## 文件用途说明

定制固件的前提是你要有一台服务器运行shadowsocks服务端ss-server。

- etc/dnsmasq.conf 设置dnsmasq配置文件目录
- etc/shadow 登陆路由器的密码 fanqiang
- etc/uci-defaults/defaults 默认上网设置及时区等设置

## 关于 /etc/uci-defaults 目录

uci-defaults目录下的文件会在路由器第一次启动时由/etc/init.d/boot执行,如果在文件末尾加上exit 0, 则执行就会删除此文件, 否则执行成功则删除, 不成功则在下次启动时继续执行直到成功。

我们在这个目录下创建一个defaults文件, 在这个文件中设置上网参数, 时区等。

To set some system defaults the first time the device boots, create a script in the folder

All scripts in that folder are automatically executed by /etc/init.d/boot and if they exited with code 0 deleted afterwards

(scripts that did not exit with code 0 are not deleted and will be re-executed during the next boot until they also successfully exit).

## 编译固件前必须修改的三个文件

- /etc/shadowsocks.json
  - server必须改成你的服务器实际IP
- /etc/uci-defaults/default
  - adslname 和 adslpassword必改
- /usr/bin/shadowsocks-firewall
  - 107.89.0.12必须改成你的服务器实际IP

默认端口及修改方法（可以不改）：

- shadowsocks服务端监听端口：8345
  - 文件位置：服务器/etc/shadowsocks/config.json
  - 如更改，路由器里 /etc/shadowsocks.json也相应更改
- 路由器shadowsocks ss-redir 监听端口：8024
  - 文件位置：路由器/etc/shadowsocks.json
  - 如更改，路由器/usr/bin/shadowsocks-firewall也相应更改
- 路由器shadowsocks ss-tunnel 监听端口: 7913
  - 文件位置: 路由器/etc/init.d/shadowsocks
  - 如更改, 路由器 /etc/dnsmasq.d/gfwlist.conf也相应更改

以上端口建议不改。程序运行稳定后，相关密码可以改掉。

端口关联的理解：

- shadowsocks-firewall 负责把非亚洲流量转发到本地端口8024
- ss-redir监听端口8024，该端口流量都加密走自己的服务器通道
- dnsmasq把非国内重要域名的dns查询转发本地7913端口
- ss-tunnel监听本地端口7913,把该端口的dns查询转发到自己服务器向8.8.8.8查询

## 设置权限

```
chmod +x usr/bin
chmod +x usr/bin/*
chmod +x etc/uci-defaults
chmod +x etc/uci-defaults/default
```

参考：

\*<http://wiki.openwrt.org/doc/uci>

# 使用Image Builder编译自动翻墙OpenWrt固件

Image Builder又叫Image Generator，利用它我们可以方便地定制适合自己无线路由器的固件。

## 下载适合自己无线路由器的Image Builder

从 <http://downloads.openwrt.org/snapshots/trunk/> 选择适合自己的目录，比如我选的是：<http://downloads.openwrt.org/snapshots/trunk/ar71xx/>

```
cd /opt
wget http://downloads.openwrt.org/snapshots/trunk/ar71xx/OpenWrt-ImageBuilder-ar71xx_generic-for-linux-x86_64.tar.bz2
tar -xjf OpenWrt-ImageBuilder-ar71xx_generic-for-linux-x86_64.tar.bz2
```

## 确定OpenWrt无线路由器的PROFILE值

```
cd OpenWrt-ImageBuilder-ar71xx_generic-for-linux-x86_64
make info
```

找到自己固件的型号，比如我的是 TP-LINK TL-WR2543N/ND，它的PROFILE值是TLWR2543。如下图：

TLWR2543:  
TP-LINK TL-WR2543N/ND  
Packages: kmod-usb-core kmod-usb2 kmod-ledtrig-usbdev

## 找出默认应该包含进OpenWrt固件的包

对于TP-LINK WR2543无线路由器来说，可以这样获取：

```
echo $(wget -qO - http://downloads.openwrt.org/snapshots/trunk/ar71xx/config | sed -ne 's/^CONFIG_PACKAGE_\([a-z0-9-]*\)=y/\1/ip')
```

```
base-files busybox dnsmasq dropbear firewall ftools jsonfilter libc libgcc mtd netifd opkg procd swconfig ubox
ubus ubusd uci kmod-crypto-aes kmod-crypto-arc4 kmod-crypto-core kmod-ledtrig-usbdev kmod-lib-crc-ccitt kmod-
nls-base kmod-ip6tables kmod-ipt-contrack kmod-ipt-core kmod-ipt-nat kmod-ipt-nathelper kmod-ipv6 kmod-ppp
kmod-pppoe kmod-pppox kmod-slhc kmod-gpio-button-hotplug kmod-usb-core kmod-usb-ohci kmod-usb2 kmod-
ath kmod-ath9k kmod-ath9k-common kmod-cfg80211 kmod-mac80211 libip4tc libip6tc libxtables libblobmsg-json
libiwinfo libjson-c libnl-tiny libubox libubus libuci ip6tables iptables hostapd-common iw odhcp6c odhcpd ppp ppp-
mod-pppoe wpad-mini iwninfo jshn libjson-script uboot-envtools
```

以上默认包，我们要包含在PACKAGES命令行参数中，并再加上必要的包：wget shadowsocks-libev-polarssl，如果你需要网页管理界面，可以再加上luci-ssl。注意，在编译前要把 shadowsocks-libev-polarssl\_1.x.x\_ar71xx.ipk 放到 OpenWrt-
ImageBuilder-ar71xx\_generic-for-linux-x86\_64/packages/base/ 目录下。

如果你的openWrt版本是 ATTITUDE ADJUSTMENT，可能要加上iptables-mod-nat-extra包，如果没安装的话iptables的端口转发会不支持。

## OpenWrt Image Builder的三个命令行参数

- PROFILE 指定设备类型，此处是 TLWR2543
- PACKAGES 指定要编译进固件的包
- FILES 指定要编译进固件的自定义文件，如网络有关配置文件，/opt/openwrt-wr2543

# 开始编译OpenWrt自动翻墙固件

```
cd /opt/OpenWrt-ImageBuilder-ar71xx_generic-for-linux-x86_64  
make image PROFILE=TLWR2543 PACKAGES="base-files busybox dnsmasq dropbear firewall fstools jsonfilter libc libgcc mtd netifd op
```



编译好的的固件在 /opt/OpenWrt-ImageBuilder-ar71xx\_generic-for-linux-x86\_64/bin/ar71xx/目录，升级固件要用到的是 openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin

然后把这个固件刷进TP-LINK WR2543N，重启路由器后就能免设置智能翻墙，并能定时自动更新规则。

## 管理员登陆OpenWrt

刷好固件并重启路由器后，电脑连上无线网络 eastking-wr2543，然后就可用密码 fanqiang 登陆路由器。

- ssh登陆openwrt管理路由器：

```
ssh root@192.168.1.1
```

- 浏览器打开192.168.1.1登陆

以后玩OpenWrt出问题，可以重新刷上这个翻墙固件就又可以在网上畅行无阻了。

## 参考：

- <http://wiki.openwrt.org/doc/howto/obtain.firmware.generation>

# 如何使用别人预编译的OpenWrt翻墙固件 for TP-LINK wr2543N (包含shadowsocks-libev)

如果你的无线路由器和我的一样，也是 TP-LINK wr2543N v1，你不想自己编译固件，那么可以下载我预先编译好的固件，刷好固件后，稍微设置下，就可以自动翻墙。

在下载和刷OpenWrt固件前，确保熟悉本教程的前面部分，已经配置好shadowsocks-libev服务端，并能自由进入路由器的安全模式。再次强调，刷机有风险，风险自承担。

该固件只是在OpenWrt trunk版加上：luci-ssl wget shadowsocks-libev的最新版，还有翻墙要用到的配置，没有添加其他任何内容。

## 翻墙默认配置

- 教程用到的OpenWrt翻墙配置文件](<https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt>)
- 教程中用到的shadowsocks服务端配置文件

## 下载OpenWrt固件 for TP-LINK wr2543N

到下面的网址下载：

<https://software-download.name/2014/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade-bin-with-shadowsocks/>

下载后保存在Ubuntu: /opt/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin

## 复制OpenWrt固件到路由器

```
scp /opt/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin root@192.168.1.1:/tmp/
```

## 登陆OpenWrt路由器，并查看文件大小是否正确

```
ssh root@192.168.1.1
root@OpenWrt: cd /tmp/
ls
```

## 升级OpenWrt固件(不留原来配置)

```
root@OpenWrt:/tmp# sysupgrade -n openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin
```

路由器重启后，电脑连接到无线网络 **eastking-wr2543**

**ssh**登陆并修改设置：

```
ssh root@192.168.1.1
```

输入密码 fanqiang 登陆

有时会提示错误：

```
@@@@@@@WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! @
@@@@@@@IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
cf:c5:12:34:56:0b:4d:1c:56:48:6a:87:04:cf:b8:83.
Please contact your system administrator.
Add correct host key in /home/openwrt-fanqiang/.ssh/known_hosts to get rid of this message.
Offending RSA key in /home/openwrt-fanqiang/.ssh/known_hosts:3
remove with: ssh-keygen -f "/home/openwrt-fanqiang/.ssh/known_hosts" -R 192.168.1.1
RSA host key for 192.168.1.1 has changed and you have requested strict checking.
Host key verification failed.
```

解决办法就是复制并运行提示中的清理命令：

```
ssh-keygen -f "/home/openwrt-fanqiang/.ssh/known_hosts" -R 192.168.1.1
```

以下设置必须修改：

- /etc/shadowsocks.json
  - server必须改成你的服务器实际IP
- /etc/config/network
  - adslname 和 adslpassword必改
- /usr/bin/shadowsocks-firewall
  - 107.89.0.12必须改成你的服务器实际IP

如果你还改了其他默认值，请自行修改相应文件。不建议修改其他默认值，以提高一次成功率。

## 执行以下命令使修改生效

```
root@OpenWrt:~# /etc/init.d/shadowsocks stop
root@OpenWrt:~# /etc/init.d/shadowsocks start
root@OpenWrt:~# /etc/init.d/network restart
```

测试一下是否可以在网上畅行无阻了。

本教程已经在**github**开源，欢迎提交改进，报告**bug**：

<https://github.com/softwaredownload/openwrt-fanqiang>

# D-Link DIR-505路由器刷OpenWrt固件翻墙教程

前面的教程用结合 TP-LINK TL-WR2543N 来讲解翻墙原理与方法，并不是我特别推荐TP-LINK TL-WR2543N，而是因为手头正好有这个路由器。毫无疑问，初学者使用教程同款路由器比较容易上手。但此型号趋向退市，价格也不便宜，网上有二手货，如果功能正常倒也可以考虑。

另外的选择，是使用 D-Link DIR-505 便携式路由器。配置高，价格便宜。

## D-Link DIR 505 硬件信息

```
Architecture: MIPS 24Kc
Vendor: Atheros
Bootloader: UBoot 1.1.4
System-On-Chip: SoC: Atheros AR9330 rev 1
CPU-Speed Atheros AR9330 400.000MHz
Flash-Chip: NANYA NTSTU32M16DG-AC
Flash size: 8192 KiB
RAM: 64 MiB
Wireless: 802.11b/g/n
Ethernet: 10/100 full duplex
USB: Yes 1 x 2.0 ar7240-ehci
Serial: Yes - tested working over TTL converter (3.3V!)
JTAG: Nope
```

与之同价格档次的TP-LINK TL-WR706N 150M迷你型无线路由器，AR9331 SOC 2MB Flash/16MB RAM, 相比之下简直是垃圾。我花数百元购买的TP-LINK TL-WR2543N，也不过是8MB Flash, 64MB RAM内存。

还有，D-Link DIR-505 自带不死恢复模式，调试OpenWrt系统出现问题时我们既可以进D-Link的恢复模式刷新固件，也可以进入OpenWrt的恢复模式刷新固件，可谓是最安全的路由器。

## 如何购买 D-Link DIR 505 A1

我不是D-Link的员工，也无意为其做广告。DIR-505是我购买的第一款D-Link路由器。

我是2014年8月从淘宝 D-Link官方旗舰店买的 D-Link DIR 505 A1，69元，固件版本号：1.03CN。买了后，看了下手机淘宝，只要59元。准备再入一个，都刷上OpenWrt，方便随时随地无障碍上网。

# 如何进入 DIR-505 恢复模式

在学习OpenWrt可能要测试很多配置，有时会出现错误，需恢复或补救，这时就需要进入路由器的恢复模式。

有两种方法进入 DIR-505 的恢复模式。

## 进入D-Link 恢复模式

把 DIR-505 和计算机用网线连接起来，设置计算机网卡的IPv4地址为192.168.0.12，子网掩码255.255.255.0，在路由器启动时顶住reset孔，当红色指示灯开始缓慢闪烁时，松开reset孔。然后浏览器打开 192.168.0.1，这里你可以上传原厂固件或刷 OpenWrt 固件。

Plug in your computer to the unit, assign it an ip address of 192.168.0.12, and boot the unit up while holding down the reset. Once the red light starts to blink slowly, release the reset, and go to 192.168.0.1 on your web browser. From there you can upload a new image. After successful flashing, you'll see a "Success" page in your browser.

刷新固件完成后，重新改回自动获取IP地址。

## 进入 OpenWrt 恢复模式

用网线将路由器和电脑连接起来，将电脑网卡的IPv4地址设置成 192.168.1.23

路由器插上电源重新开机，在启动时多次按压路由器侧面的圆形 WPS 按钮直到 LED 指示灯开始快速闪烁。

For the generic failsafe mode you can follow <http://wiki.openwrt.org/doc/howto/generic.failsafe> You can use the WPS button for that. While booting up, just press it several times until the LED flashes very quick. If you're still not able to telnet it on 192.168.1.1 maybe there's something wrong on the client-side.

接下来就是ubuntu 里 telnet 进入 OpenWrt 并设置 root 密码。

```
telnet 192.168.1.1
```

telnet连上后就设置root密码，自动启用 ssh:

```
root@OpenWrt:/# passwd
Changing password for root
New password:
Retype password:
Password for root changed by root
root@OpenWrt:/#
```

这里，可以在 Ubuntu 里 Ctrl + Shift + t新开一个命令行窗口，尝试 ssh 连接OpenWrt:

```
ssh root@192.168.1.1
```

如果 ssh 连上了，则后面设置的内容和前面 TLWR-2543N 翻墙教程一样了。

要注意的是，D-Link DIR-505 使用接口名称 eth1 而不是通常的 eth0.

Other than similar routers (e.g., the TP-Link TL-WR703N), the D-Link DIR-505 uses the interface eth1 rather than eth0. This means that if you build your own firmware, you must configure /etc/config/network accordingly (option ifname 'eth1'), or you will not be able to connect later on via Ethernet.

## 设置无线连接

在没有设置无线连接前，要登陆OpenWrt，必须用网线把电脑和路由器连接起来，不太方便。设置无线连接后，电脑就可以通过无线方式连上路由器，再登陆 DIR-505 OpenWrt进行设置。

```
uci set wireless.@wifi-device[0].disabled=0;
uci set wireless.@wifi-iface[0].ssid='eastking-dir505';
uci set wireless.@wifi-iface[0].encryption='psk2+ccmp';
uci set wireless.@wifi-iface[0].key='wsjdw,8181';
uci commit wireless;
wifi
```

设置好无线连接后，就可以拔掉电脑的有线连接，连接无线，再ssh登陆路由器。

参考：

- <https://forum.openwrt.org/viewtopic.php?id=38742&p=8>
- <http://wiki.openwrt.org/toh/d-link/dir-505#debricking.>
- <http://my.oschina.net/umu618/blog/271630>

# D-Link DIR-505 A1 刷 OpenWrt固件过程

D-Link 路由器是锁区的，不能直接刷OpenWrt 固件。要先到D-Link 官方国际站下载原厂固件，用16进制编辑器把DEF改成CN，升级固件，再刷OpenWrt固件。

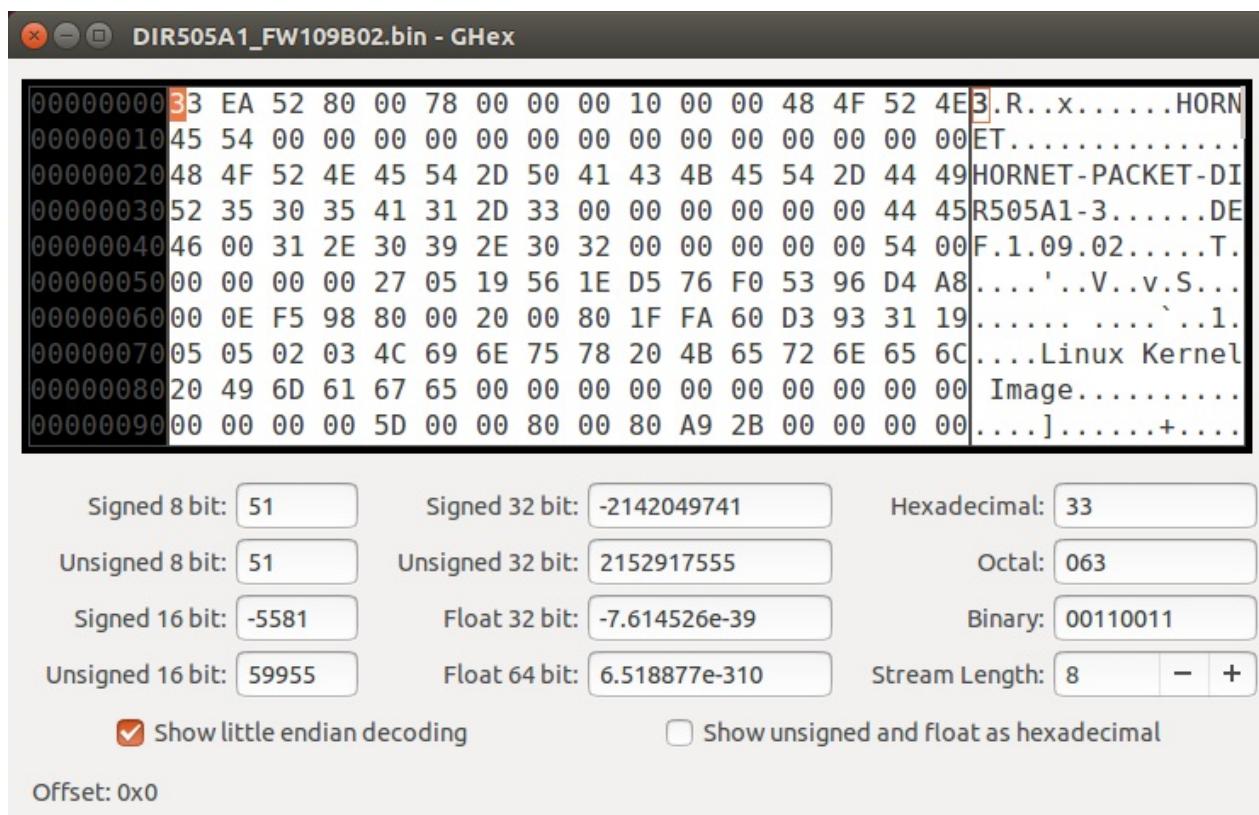
## 下载D-Link DIR-505 A1 国际版官方固件

下载地址：

- <http://support.dlink.com.au/download/download.aspx?product=DIR-505>
- [ftp://files.dlink.com.au/products/DIR-505/REV\\_A/Firmware/v1.09b02/DIR-505A1\\_FW109B02.bin](ftp://files.dlink.com.au/products/DIR-505/REV_A/Firmware/v1.09b02/DIR-505A1_FW109B02.bin)

## 用16进制编辑器修改固件的国家代码， DEF改成CN

准备一个16进制编辑器，在本文中，我用的是Ubuntu下的轻量级16进制编辑器GHex,把固件拖到GHex打开固件。



修改后变成如下：

DIR505A1\_FW109B02.bin - GHex

00000000	33 EA 52 80 00 78 00 00 00 00 10 00 00 48 4F 52 4E	3.R..x.....HORN
00000010	45 54 00 00 00 00 00 00 00 00 00 00 00 00 00 00	ET.....
00000020	48 4F 52 4E 45 54 2D 50 41 43 4B 45 54 2D 44 49	HORNET-PACKET-DI
00000030	52 35 30 35 41 31 2D 33 00 00 00 00 00 43 4E	R505A1-3.....CN
00000040	00 00 31 2E 30 39 2E 30 32 00 00 00 00 00 54 00	.1.09.02.....T.
00000050	00 00 00 27 05 19 56 1E D5 76 F0 53 96 D4 A8	....'..V..v.S...
00000060	00 0E F5 98 80 00 20 00 80 1F FA 60 D3 93 31 19	..... ....`..1.
00000070	05 05 02 03 4C 69 6E 75 78 20 4B 65 72 6E 65 6C	....Linux Kernel
00000080	20 49 6D 61 67 65 00 00 00 00 00 00 00 00 00 00	Image.....
00000090	00 00 00 00 5D 00 00 80 00 80 A9 2B 00 00 00 00	....].....+....

Signed 8 bit:  Signed 32 bit:  Hexadecimal:   
 Unsigned 8 bit:  Unsigned 32 bit:  Octal:   
 Signed 16 bit:  Float 32 bit:  Binary:   
 Unsigned 16 bit:  Float 64 bit:  Stream Length:  - +  
 Show little endian decoding  Show unsigned and float as hexadecimal  
 Offset: 0x40

Alt+S保存对固件的修改。

你也可以到下面网址直接下载修改好16进制值的固件：

<https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>

## 刷修改国家后的官方固件

按照路由器官方手册，电脑连上路由器。

在Ubuntu下电脑连接DIR-505路由器的方法： DIR-505路由器出厂默认设置没有开启DHCP，所以我们要给电脑手动设置和路由器同网段的IPv4地址才能连上路由器。路由器插上电源。右上角无线信号处，选择 Edit Connections, 选择dlink-xxxx,xxxx为路由器MAC ID的后4位，Edit..., IPv4 Setings, Method选择 Manula手动，Address选择Add, 设置Address为192.168.0.2，Netmask 255.255.255.0, Gateway 192.168.0.1。如此设置好后电脑就能连上无线网络dlink-xxxx了。

浏览器首次进入 <http://192.168.0.1> 会出现设置向导，点取消，然后会出现密码登陆页面：

产品页面 : DIR-505 硬件版本 : A1 固件版本 : 1.03CN

D-Link®

登录

登录到路由器:

用户名:  密码:

登入

WIRELESS

版权 © 2012 D-Link 企业/D-Link 系统股份有限公司

直接点击 登入 按钮，再点击界面上部的 维护，然后点击左侧栏的 固件 进入升级固件页面，点击 **Browse...** 上传我们修改好的固件：



然后点击 上传 按钮完成刷新固件，接下来就可以刷 OpenWrt 固件了。

## DIR-505A1 刷 OpenWrt 固件

### 下载 OpenWrt 固件 for DIR-505A1

- <http://downloads.openwrt.org/snapshots/trunk/ar71xx/>
- <http://downloads.openwrt.org/snapshots/trunk/ar71xx/openwrt-ar71xx-generic-dir-505-a1-squashfs-factory.bin>

### 刷OpenWrt固件

我们是在原厂固件上刷 OpenWrt,一定要下载 factory.bin. 上传后，等待150秒，DIR-505A1成功刷上了OpenWrt开源固件。



### 参考：

- <http://my.oschina.net/umu618/blog/268466>

# D-Link DIR-505 启用工作模式开关

## DIR-505 硬件开启四种应用模式

D-Link DIR-505 在全球销售多款型号，不同型号外观不一样，但内部硬件是一样的。在中国销售的 DIR-505 A1, 也就是本教程所用的型号，模式开关共有三档，在开关处动手动，就可以启用四种模式。

撕掉标贴，去掉螺丝，就可以打开DIR-505, 把开关剪短，剪掉挡住开关上推的底面，完工后如下图：



## Router模式和AP模式

便携式无线路由器常有Router模式和AP模式，有的路由器用两个档位对应这两种模式，拨到Router档就用Router模式，拨到AP档就用AP模式。DIR-505 原厂固件，Router和AP共用一个档位，需要用哪种，需要登陆路由器进行选择和设置。现在我们已经刷了OpenWrt，档位对应的模式需要自己定义设置。

在本教程中，把新开的第四档作为AP档，原来的Router/AP档作为Router档。

在Router模式时，DIR-505作为无线路由器使用，有线接口作为WAN口，连接到ADSL Modem。计算机通过无线的方式连接到路由器。在这种模式下一般需要设置拨号上网帐号。

在AP模式时，通常在DIR-505前端还有路由器，DIR-505的有线接口作为LAN口使用，前端路由器的LAN口引出网线连接到DIR-505. 在宾馆上网，把有线扩展为无线常应用此种模式。

## /etc/rc.local 利用 GPIO 读取开关位置

rc.local内容如下：

```
# Put your custom commands here that should be executed once
# the system init finished. By default this file does nothing.

if [ ! -f /etc/config/backup/network ]; then
    cp /etc/config/network /etc/config/backup/
    cp /etc/config/wireless /etc/config/backup/
    cp /etc/config/firewall /etc/config/backup/
    cp /etc/config/dhcp /etc/config/backup/
fi

read_gpio() {
(echo $1 > /sys/class/gpio/export) >& /dev/null
(echo "in" > /sys/class/gpio/gpio$1/direction) >& /dev/null
return `cat /sys/class/gpio/gpio$1/value`;
}
read_gpio 19;
v=$?;
read_gpio 20;
v=$v$?;
read_gpio 21;
v=$v$?;
read_gpio 22;
v=$v$?;
read_gpio 23;
v=$v$?;
case "$v" in
10001) v="router";;
11001) v="repeater";;
01001) v="hotspot";;
11000) v="ap";;
*) v="error";;
esac

/usr/bin/$v

logger working mode: $v

exit 0
```

原理：先备份原始的配置文件，不同模式的设置都是基于原始配置文件，以免出现混乱。

在/usr/bin目录下创建相应模式的bash文件，根据不同的GPIO值调用的不同的文件。在本教程中主要应用 /usr/bin/router 和 /usr/bin/ap这两个文件。

代码的最新版本，请查看：

- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/ebook/dir505/openwrt>

你使用时，可以直接下载整个项目到本地，所有配置文件自然在其中：

```
git clone https://github.com/softwaredownload/openwrt-fanqiang
```

参考：

- <http://my.oschina.net/umu618/blog/273945>

# DIR-505 Router 模式

## /usr/bin/router 代码：

```
#!/bin/sh

# Author: https://github.com/softwaredownload/openwrt-fanqiang
# Date: 2014-08-22

cp /etc/config/backup/* /etc/config/

uci delete network.lan.ifname
uci delete network.lan.type

uci add network interface
uci rename network.@interface[-1]='wan'
uci commit network

uci set network.wan.ifname='eth1'
uci set network.wan.peerdns=0
uci set network.wan.proto='pppoe'
uci set network.wan.username='adslname'
uci set network.wan.password='adslpassword'
uci set network.wan.peerdns=0

uci commit network

# default is no this option
#uci set dhcp.lan.ignore=0
#uci commit dhcp

uci set wireless.@wifi-device[0].channel=11
uci set wireless.@wifi-device[0].txpower=15
uci set wireless.@wifi-device[0].disabled=0
uci set wireless.@wifi-device[0].country='CN'
uci set wireless.@wifi-iface[0].mode='ap'
uci set wireless.@wifi-iface[0].ssid='eastking-dir505'
uci set wireless.@wifi-iface[0].encryption='psk2'
uci set wireless.@wifi-iface[0].key='wsjdw,8181'

uci commit wireless
wifi

/etc/init.d/network restart
```

## 代码说明：

先把备份的原始配置文件覆盖到配置文件目录，所有设置都基于原始配置文件。在使用Router模式时，有线接口为WAN口，这时wan的 interface name 是 eth1，默认lan的interface name 使用了 eth1，要删除。

代码的最新版本，请查看：

- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/ebook/dir505/openwrt>

# DIR-505 AP 模式

## /usr/bin/ap 代码：

```
#!/bin/sh

# Author: https://github.com/softwaredownload/openwrt-fanqiang
# Date: 2014-08-22

cp /etc/config/backup/* /etc/config/

uci set network.lan.gateway=192.168.1.1
uci set network.lan.dns=192.168.1.1
uci set network.lan.ipaddr=192.168.1.234

uci commit network

uci set dhcp.lan.ignore=1
uci commit dhcp

uci set wireless.@wifi-device[0].channel=11
uci set wireless.@wifi-device[0].txpower=15
uci set wireless.@wifi-device[0].disabled=0
uci set wireless.@wifi-device[0].country='CN'
uci set wireless.@wifi-iface[0].mode='ap'
uci set wireless.@wifi-iface[0].ssid='eastking-dir505'
uci set wireless.@wifi-iface[0].encryption='psk2'
uci set wireless.@wifi-iface[0].key='wsjdw,8181'

uci commit wireless
wifi

/etc/init.d/network restart
```

## 代码说明：

在AP模式下，DIR-505的有线接口作为LAN口使用，连接到前端路由器的LAN口。假设DIR-505前端路由器的IP地址是192.168.1.1，设置DIR-505的lan网关和dns都是192.168.1.1，再设置DIR-505的lan IP地址为192.168.1.234。

## 穿越功夫网翻墙方法

### 假设上级路由器没有设置翻墙

电脑设置无线连接eastking-dir505的IPv4地址是192.168.1.235（不同于路由器的地址），设置子网掩码为255.255.255.0，网关和DNS为路由器的地址即192.168.1.234，重启路由器后，电脑连上eastking-dir505即可自动翻墙

原理：以DIR-505作为DNS服务器，我们已经把DIR-505设置成翻墙路由器，自然可以打败功夫网了。

### 假设上级路由器已经翻墙

电脑设置无线连接eastking-dir505为DHCP即可。原理：以上级路由器作为DNS服务器，上级路由器已经翻墙，二级路由器就可以免设置自动翻墙了。

如果你想节省路由器资源，这时就可以禁用dir-505 dns及翻墙相关服务：

```
/etc/init.d/dnsmasq stop
/etc/init.d/dnsmasq disable
/etc/init.d/shadowsocks stop
/etc/init.d/shadowsocks disable
```

代码的最新版本，请查看：

- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/ebook/dir505/openwrt>

参考：

- <http://wiki.openwrt.org/doc/recipes/bridgedap>

# 编译OpenWrt全自动翻墙固件 for D-Link DIR-505 A1

除了增加模式转换开头，其他和编译 TP-LINK WR2543N翻墙固件一样。

## 下载适合D-Link DIR505无线路由器的Image Builder

Image Builder又叫Image Generator，利用它我们可以方便地定制适合自己无线路由器的固件。

从 <http://downloads.openwrt.org/snapshots/trunk/> 选择适合自己的目录，这里选：<http://downloads.openwrt.org/snapshots/trunk/ar71xx/>

```
cd /opt
wget http://downloads.openwrt.org/snapshots/trunk/ar71xx/OpenWrt-ImageBuilder-ar71xx_generic-for-linux-x86_64.tar.bz2
tar -xjf OpenWrt-ImageBuilder-ar71xx_generic-for-linux-x86_64.tar.bz2
```

## 确定OpenWrt无线路由器的PROFILE值

```
cd OpenWrt-ImageBuilder-ar71xx_generic-for-linux-x86_64
make info
```

找到自己固件的型号，D-Link DIR 505 A1的PROFILE值是DIR505A1。如下图：

```
DIR505A1:
    D-Link DIR-505 rev. A1
    Packages: kmod-usb-core kmod-usb2 kmod-ledtrig-usbdev
DIR600A1:
    D-Link DIR-600 rev. A1
```

## 找出默认应该包含进OpenWrt固件的包

对于D-Link DIR-505 A1 无线路由器来说，可以这样获取：

```
echo $(wget -qO - http://downloads.openwrt.org/snapshots/trunk/ar71xx/config | sed -ne 's/^CONFIG_PACKAGE_\([a-z0-9-]*\)=y/\1/ip')
```

```
base-files busybox dnsmasq dropbear firewall ftools jsonfilter libc libgcc mtd netifd opkg procd swconfig ubox
ubus ubusd uci kmod-crypto-aes kmod-crypto-arc4 kmod-crypto-core kmod-ledtrig-usbdev kmod-lib-crc-ccitt kmod-nls-base kmod-ip6tables kmod-ipt-contrack kmod-ipt-core kmod-ipt-nat kmod-ipt-nathelper kmod-ipv6 kmod-ppp
kmod-pppoe kmod-pppox kmod-slhc kmod-gpio-button-hotplug kmod-usb-core kmod-usb-ohci kmod-usb2 kmod-ath
kmod-ath9k kmod-ath9k-common kmod-cfg80211 kmod-mac80211 libip4tc libip6tc libxtables libblobmsg-json
libiwinfo libjson-c libnl-tiny libubox libubus libuci ip6tables hostapd-common iw odhcp6c odhcpd ppp ppp-
mod-pppoe wpad-mini iwnfo jshn libjson-script uboot-envtools
```

以上默认包，我们要包含在PACKAGES命令行参数中，并再加上必要的包：wget shadowsocks-libev-polarssl，如果你需要网页管理界面，可以再加上luci。注意，在编译前要把 shadowsocks-libev-polarssl\_1.x.x\_ar71xx.ipk 放到 /opt/OpenWrt-ImageBuilder-ar71xx\_generic-for-linux-x86\_64/packages/base/ 目录下。

如果你的openWrt版本是 ATTITUDE ADJUSTMENT，可能要加上iptables-mod-nat-extra包，如果没安装的话iptables的端口转发会不支持。

## 下载和设定配置文件

## 下载配置文件

```
cd /opt  
git clone https://github.com/softwaredownload/openwrt-fanqiang
```

配置文件目录就是：

```
/opt/openwrt-fanqiang/ebook/dir505/openwrt
```

不要直接在git项目修改添加你的个人网络配置，而是复制配置文件再修改。

```
cp -R /opt/openwrt-fanqiang/ebook/dir505/openwrt /opt/openwrt-dir505
```

然后修改 /opt/openwrt-dir505 中的相关配置文件。

## 设置可执行权限

```
cd /opt/openwrt-dir505  
chmod +x usr/bin  
chmod +x usr/bin/*  
chmod +x etc/uci-defaults  
chmod +x etc/uci-defaults/defaults
```

说明：etc/uci-defaults目录下的文件会在路由器第一次启动时执行一次。在这里我们设置一些常用值。

## 必须修改的配置文件：

- /opt/openwrt-dir505/etc/shadowsocks.json
  - server改成你的服务器实际IP
- /opt/openwrt-dir505/usr/bin/router
  - adslname 和 adslpassword改成实际值
- /opt/openwrt-dir505/usr/bin/shadowsocks-firewall
  - 107.89.0.12必须改成你的服务器实际IP

## 开始编译OpenWrt自动翻墙固件

```
cd /opt/OpenWrt-ImageBuilder-ar71xx_generic-for-linux-x86_64  
make image PROFILE=DIR505A1 PACKAGES="base-files busybox dnsmasq dropbear firewall fstools jsonfilter libc libgcc mtd netifd opk
```

编译好的的固件在 /opt/OpenWrt-ImageBuilder-ar71xx\_generic-for-linux-x86\_64/bin/ar71xx/目录，升级固件要用到的是 openwrt-ar71xx-generic-dir-505-a1-squashfs-sysupgrade.bin，如果在原厂固件上刷要用openwrt-ar71xx-generic-dir-505-a1-squashfs-factory.bin

然后把翻墙固件刷进D-Link DIR-505 A1，就能免设置智能翻墙，并定时自动更新规则。只要配置文件设置不出差错，编译固件一般都能成功，保存好这个固件，以后随便折腾路由器，出现问题大不了重刷一次，几分钟时间就一切都恢复正常。

## 参考：

- <http://wiki.openwrt.org/doc/howto/obtain.firmware.generate>

# D-Link DIR-505 A1 刷通用OpenWrt翻墙固件

照前面的教程自己编译翻墙固件，编译出来后刷进路由器，就能实现零设置自动翻墙。出于各种原因，有的朋友可能不想自己编译固件，又想用DIR-505实现智能翻墙，就要下载预编译的通用翻墙固件，刷好后，登陆路由器，用vi修改少数几个设置，就能实现智能翻墙，本教程就是针对这些朋友而写。

路由器的开关拨到刻有 Router/AP 字样的档位，如果你没有给路由器动过手术，就是从上往下数的第一档。

## DIR-505原厂固件刷翻墙固件的方法

适合购买了D-Link DIR-505 A1后没有刷过任何固件的朋友。

### 刷修改了16进制值的原厂固件

到下面地址下载已经修改了16进制值的原厂固件：

<https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>

照官方手册说明网页登陆路由器，刷新固件

### 刷DIR-505的翻墙固件 factory.bin

到下面地址下载用于 DIR-505的翻墙固件：

<https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>

下载 openwrt-ar71xx-generic-dir-505-a1-squashfs-factory.bin

按照官方手册的说明刷新固件。

## OpenWrt固件基础上升级到翻墙固件

注意，下面步骤适合于你已经在你的DIR-505上刷了OpenWrt固件，你想要升级到可以自己翻墙的openwrt固件。

### 下载翻墙固件 sysupgrade.bin

到下面地址下载用于 DIR-505的翻墙固件 openwrt-ar71xx-generic-dir-505-a1-squashfs-sysupgrade.bin：

<https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>

### 命令行上传固件到路由器

电脑通过网线或无线连接到路由器，然后：

```
scp openwrt-ar71xx-generic-dir-505-a1-squashfs-sysupgrade.bin root@192.168.1.1:/tmp/
```

### ssh登陆OpenWrt路由器

```
ssh root@192.168.1.1
cd /tmp
```

### sysupgrade升级固件并取消保留原来配置文件

```
root@OpenWrt:/tmp# sysupgrade -n openwrt-ar71xx-generic-dir-505-a1-squashfs-sysupgrade.bin
```

参数 `-n` 表示升级时不保留原来的配置文件。

等待两分钟等刷新固件并重启完成。

# 登陆并设置 DIR-505 OpenWrt 翻墙固件

ADSL Modem网线连接到路由器的有线接口。路由器的开关拨到刻有 Router/AP 字样的档位，如果你没有给路由器动过手术，就是从上往下数的第一档。本文以router模式为例，如果你的应用场景是ap模式，请自行相应变通。

## 电脑连接路由器

电脑连接到无线 网络 **eastking-dir505**

## ssh 登陆 OpenWrt 翻墙固件

```
ssh root@192.168.1.1
```

输入密码 `fanqiang` 登陆ssh

有时会提示错误：

```
@@@@@@@  
@ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! @  
@@@@@@@  
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!  
Someone could be eavesdropping on you right now (man-in-the-middle attack)!  
It is also possible that a host key has just been changed.  
The fingerprint for the RSA key sent by the remote host is  
cf:c5:12:34:56:0b:4d:1c:56:48:6a:87:04:cf:b8:83.  
Please contact your system administrator.  
Add correct host key in /home/openwrt-fanqiang/.ssh/known_hosts to get rid of this message.  
Offending RSA key in /home/openwrt-fanqiang/.ssh/known_hosts:  
remove with: ssh-keygen -f "/home/openwrt-fanqiang/.ssh/known_hosts" -R 192.168.1.1  
RSA host key for 192.168.1.1 has changed and you have requested strict checking.  
Host key verification failed.
```

解决办法就是复制并运行提示中的清理命令：

```
ssh-keygen -f "/home/openwrt-fanqiang/.ssh/known_hosts" -R 192.168.1.1
```

然后就可以正常登陆了。

## 登陆后用vi修改设置：

```
root@OpenWrt:~# vi /etc/shadowsocks.json  
root@OpenWrt:~# vi /usr/bin/router  
root@OpenWrt:~# vi /usr/bin/ap    #如果是ap模式  
root@OpenWrt:~# vi /usr/bin/shadowsocks-firewall
```

分别修改以下值：

- shadowsocks.json中，server改成你的服务器实际IP
- router/ap中 adsiname 和 adslpassword改成实际值
- shadowsocks-firewall中，107.89.0.12必须改成你的服务器实际IP

如果你还改了其他默认值，请自行修改相应文件。不建议修改其他默认值，以提高一次成功率。

## 执行以下命令使修改生效

```
root@OpenWrt:~# /etc/init.d/shadowsocks stop
root@OpenWrt:~# /etc/init.d/shadowsocks start
root@OpenWrt:~# /etc/init.d/dnsmasq restart
root@OpenWrt:~# /etc/init.d/network restart
```

等待约两分钟，就可以测试是否可以在网上畅行无阻了。

# 附录

---

翻墙常用资源及如何贡献本项目

# 翻墙教程资源汇总

---

## 翻墙软件

---

- [Shadowsocks Download](#)
- [shadowsocks libev](#)
- [gohop - VPN in GO lang](#)
- [obfuscated openssh](#)

## 翻墙方案

---

- 打造OpenWrt智能自动透明翻墙路由器
- [FreeRouter\\_V2](#)
- [Autowpn for OpenWrt](#)
- [Proxy for GFW](#)
- [fqrouter, Android as router](#)
- [openwrt GFW](#)
- [dnsforwarder](#)
- [Autddvpn beta](#)

## 翻墙辅助

---

- [dnsmasq China List](#)

## 翻墙教程

---

- [FreeRouter V2完全手册PDF下载](#)
- [在华为HG255D OpenWrt上安装和配置Shadowsocks并实现智能流量转发](#)
- [用PDNSD + Google DNS 获得高速正确的dns解析](#)
- [基于OpenWRT的自动翻墙路由器](#)
- [Tunlr-style DNS unblocking](#)
- [DNS unblocking using Dnsmasq and HAProxy](#)

## OpenWrt教程

---

- [跟hoowa学做智能路由](#)
- [跟 UMU 一起玩 OpenWRT](#)

# 本地阅读本教程的方法

## git clone项目

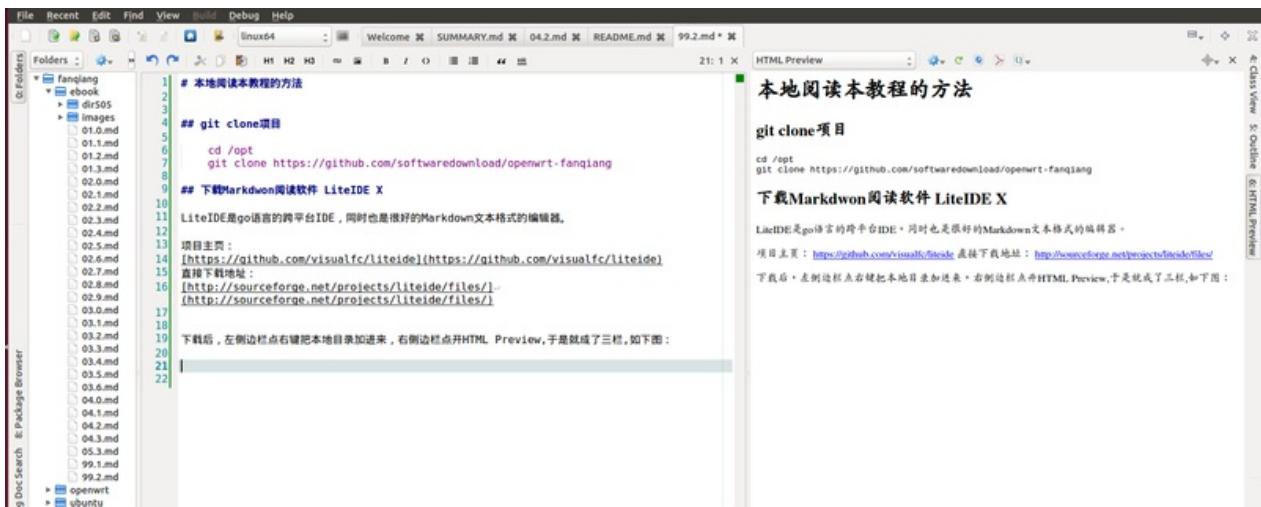
```
cd /opt  
git clone https://github.com/softwaredownload/openwrt-fanqiang
```

## 下载Markdwon阅读软件 LiteIDE X

LiteIDE是go语言的跨平台IDE，同时也是很好的Markdown文本格式的编辑器。

- 项目主页：<https://github.com/visualfc/liteide>
- 直接下载地址：<http://sourceforge.net/projects/liteide/files/>

下载后，左侧边栏点右键把本地目录加进来，右侧边栏点开HTML Preview,于是就成了三栏,Ubuntu下如下图：



## LiteIDE 使用方法

- 阅读时把中间栏宽度拉小,左边栏是导航栏，右边栏是HTML Preview
- 把中间文档栏宽度拉大，阅读Markdown源文件，贡献本项目

你是个有爱心的人，阅读了本教程，想要回馈这个开源项目，在阅读时顺便修改一些错字，加进补充内容，增加一章你的路由器应用本教程翻墙的过程等等，然后提交 pull request.

# 知识若不分享，实在没有意义

---

## 这个世界为什么圣人这么少？

---

人类历史上存在过无数人，他们都不见了，他们都到哪里去了，他们曾有过什么样的故事，可曾有人在想起他们的笑容？通过历史书，我们知道了一些历史上存在过的人物的名字，其中少数人，为人类的发展作出了特别的贡献，我们可以称他们为圣人，这样的人，一只手就数得过来。

历史上存在过的人这么多，为什么圣人却这么少？

我认为，这是因为，普通的一生，主要是在思考怎么得到更多，而较少想到去付出。得到越多越好，付出越少越好，这就是普通人。

圣人是怎么样的，是不是只想着付出，不计收获？不是的，我认为圣人是付出得到比较均衡的人。只付出而不得到，自己就很快会陷入困境，就没有能力去帮助更多人。圣人得到什么，就会想着怎么样去回馈外界，回馈社会，在回馈过程中自己得到快速成长，从而有更大的能力去回馈更多，圣人于是逐渐长成。

我这么说，并不是希望谁成为圣人。圣人并不知道自己是圣人，也不会去想这个事情。有一个信念，就要去实行，生命的意义就在于点滴的行动，能做多少就做多少，当生命之花最终凋落时，我们得到的都将失去，我们付出的也许还会存在于这个世界很长的时间。

## 我为什么写这个教程

---

生在天朝，上网各种不方便，很是苦恼，什么OpenWrt，没有听说过，不知道哇。上网查相关论坛，非注册用户附件下载隐藏，图片隐藏，各种限制。也有一些教程散布在网上，需要自己整合。终于，花了N个白天，给家里的路由器翻墙了。我是个习惯于换位思考的人，想想自己花了很多时间查各种资料，何不花时间整合各种资源并加上自己的心得，写成系列教程，公布在网上？

于是，又是N个白天（ $N > 10$ ），学习Git, GitHub, GitBook, Ubuntu, Markdown, OpenWrt，各种调试、编译。经常一天的绝大部分时间在写这个教程。钱可以少赚些，当下够用就行，这个教程还得认真写，没有想过要得到什么，只是觉得白发已生，人生不能虚度，给这个世界留下一些自己的印记也总是好的。虽然不对别人说，但也未尝不可在人少时偷偷笑一声，并对自己说：我这样的人，在这个世界上可是不多呢，哈哈。

## 为什么以开源方式发布在GitHub

---

为什么不写在博客上呢？如果写在博客上，就要自己维护博客，一直维护下去总是个麻烦事。GitHub总比自己维护的博客稳定，或者说能存在更长时间。即使GitHub倒闭，也就一个git命令就可以托管到其他网站，何况GitHub至少现在看来是来日方长呢。

开源方式发布，更是希望阅读本教程翻墙成功的朋友，如果你的路由器型号不被本教程覆盖，就写下自己的翻墙实践过程，提交到本项目中，以帮助相关朋友。我在教程中以D-Link DIR-505为范例，演示了如何参与到本项目中来，将在下一节详述。

# 如何贡献本项目

虽然说原理是通用的，本教程内容可以应用到绝大多数路由器中去。然而，高手毕竟少数，多数有翻墙需求的人可能都没有用过Linux系统，没有听说过OpenWrt，针对他们，最好是一种路由器类型（型号）一个教程。并且最好提供预编译的固件，刷上这个预编译的固件后，修改极少的参数就可以自动翻墙。

在你应用本教程原理翻墙的过程中，把详细应用过程一步步写下来，并贡献到本项目中，以帮助更多的人。

假如你的路由器是 netgear wndr3800

## github fork 本项目，然后

```
cd /opt  
git clone https://github.com/yourname/openwrt-fanqiang
```

## 创建目录结构

```
cd openwrt-fanqiang/ebook  
mkdir wndr3800  
mkdir wndr3800/images  
mkdir wndr3800/openwrt
```

在ebook目录下创建以路由器型号为名的目录，你的教程所有用到的文件都在此目录下。images是图片文件目录，openwrt是openwrt配置文件目录。可以参考 dir505目录学习文件命名相关规则。

## 教程结构

推荐的结构如下：

- 0.xxx.md 路由器的硬件信息，购买信息，作者信息等
- 1.xxx.md 怎样进入恢复模式
- 2.xxx.md 刷openwrt固件图文教程
- x.xxx.md 编译翻墙固件的过程
- x.xxx.md
- x.xxx.md 使用预编译固件的方法

如果你的更好的教程结构，则不受此限

推荐在本机上保存两份配置文件，在你的home目录下保存的是你个人专用的配置文件，包含了登陆帐号，编译进固件后，刷好固件就是零设置翻墙。

保存在 /opt/wndr3800/openwrt 目录下的是通用配置文件，不包含真实帐户信息，编译出来的固件，使用者刷好固件后，登陆路由器进行简单设置就可以翻墙。

在你的教程中最好提供预编译固件的稳定下载地址。如果你没有稳定的下载空间，可以提交一个issue，附上临时下载地址，我会上传到稳定下载地址，然后你可以修改教程加上稳定下载地址。

你可以用LiteIDE写教程。

## 教程完成后修改目录文件

修改目录文件， openwrt/SUMMARY.md，把你的教程作为新的一章，放在最后一章之前。

## **pull request**提交你的贡献

---

### 如果你的路由器型号与教程中的相同

---

如果你的路由器型号与教程中的相同或类似，你也可以参与到本项目中来，你可以修正错误，补充不详细的地方，文字润色，提出建议等。