OpenWrt Wiki

Linksys WRT54G, WRT54GL and WRT54GS

The original WRT54G was first released as a SOHO router in December 2002. The product line supports WiFi and five switched Ethernet ports. (The WAN port is part of the same internal network switch, but on a different VLAN.) The devices have two removable antennas connected through Reverse Polarity TNC connectors. For additional background information, see Linksys WRT54G series [https://en.wikipedia.org/wiki/Linksys WRT54G series].

Supported Versions

The different models within the WRT54G series may all look identical. Please refer to the model information sticker on the underside of the unit in order to determine the precise model number and hardware version of your device.

Note: The supported versions of OpenWrt can be found on the Table of Hardware

For wireless support on Kernel 2.6 it is recommended to use trunk snapshots or releases newer than and including "Backfire" 10.3.

Note: the wireless driver names are different in 2.6 from 2.4. You may need to do:

```
opkg install kmod-b43
wifi detect > /etc/config/wireless
```

in order to get the correct wireless configuration created.

For versions of the OpenWrt "brcm47xx" target prior to "Attitude Adjustment" 12.09-final, you may wish to use Broadcom's proprietary wl driver due to longstanding issues with the b43 driver in Linux kernel versions 2.6 and newer (https://dev.openwrt.org/ticket/7552 [https://dev.openwrt.org /ticket/7552]). After installing the brcm47xx image, you will need to execute the following commands while logged into the router over TELNET or SSH:

```
opkg update
opkg install kmod-brcm-wl wlc nas
rm /etc/modules.d/*b43*
```

After rebooting the router, configure wireless as usual, only using the Broadcom driver instead of the b43 driver.

Notes on specific WRT54G hardware versions

WRT54G

* The Linksys WRT54G 1.1 hardware (4 MB of flash) has trouble with OpenWrt 10.03.1-rc6 and maybe all 10.03 releases as of 2011-12-08. In a test with OpenWrt 10.03.1-rc6, the OS will install but LuCI will be unable to update settings because there isn't enough flash left free.

- References:
  - "Kamikaze, brcm47xx, WRT54G v1.1: jffs2 marker not detected, rootfs_data not mounted", https://dev.openwrt.org/ticket/5071 [https://dev.openwrt.org/ticket/5071]
  - "Linksys WRT54G v1.1 default after reboot", https://forum.openwrt.org/viewtopic.php?id=28223 [https://forum.openwrt.org /viewtopic.php?id=28223]
  - "config changes aren't permanent", https://forum.openwrt.org/viewtopic.php?id=20125 [https://forum.openwrt.org /viewtopic.php?id=20125]
- The solution is to go back to OpenWrt 8.09 r14511 (code name "kamikaze") – the link to the Broadcom Linux 2.4 chipset version is http://downloads.openwrt.org/kamikaze/8.09/brcm-2.4/openwrt-brcm-2.4-squashfs.trx [http://downloads.openwrt.org/kamikaze/8.09/brcm-2.4/openwrt-brcm-2.4-squashfs.trx].

WRT54GL

Testing with the WRT54GL 1.1 (16MB RAM, 4MB flash) showed it can run the following versions:

		7.09	8.09.2	10.03	10.03.1	12.04	12.09	14.07
brcm-2.4		works	works	works	works	n/a	n/a	n/a

		7.09	8.09.2	10.03	10.03.1	12.04	12.09	14.07
brcm47xx	b43/legacy	untested	untested	works (somewhat unstable)	works (somewhat unstable)	very low free ram & jffs	see note below	unviable, not enough RAM to run wifi

* **10.03.1 brcm-2.4** had frequent WiFi drops when in client mode w/ psk2 (did not drop when connecting to same AP w/ encryption disabled).

* **12.04** almost certainly needs to rebuilt with unnecessary packages (e.g. LuCI) and daemons (uhttpd) removed to make enough free ram & jffs to obtain long uptimes.. — *tc424 2013/08/26 17:28*

* **12.09** "Only have 688kb available on filesystem /overlay, pkg kmod-brcm-wl needs 695" Impossible to install proprietary **wl** driver into 12.09 — *jikuja 2013/11/18 12:29*

* **14.07** had slow LuCI web interface, after enabling Wifi, the entire router became inaccessible. A custom cut-down image worked slightly better, but would not let WAN and Wifi work at the same time due to low system RAM. Ref: Forum Thread [https://forum.openwrt.org/viewtopic.php?id=51729]

As the WRT54GL has only 4Mb flash, any image sent to the device must be 3866624 bytes or smaller.

WRT54G-TM

According to the Linksys WRT54G series [https://en.wikipedia.org/wiki/Linksys WRT54G series], the WRT54G-TM is nothing but a renamed WRT54GS v3.0. Because these models have additional RAM and FLASH they do not suffer from the constraints of the WRT54G and WRT54GL series and can run all versions of OpenWRT as of 11/1/2014.

Basic configuration

Please follow the basic.config guide.

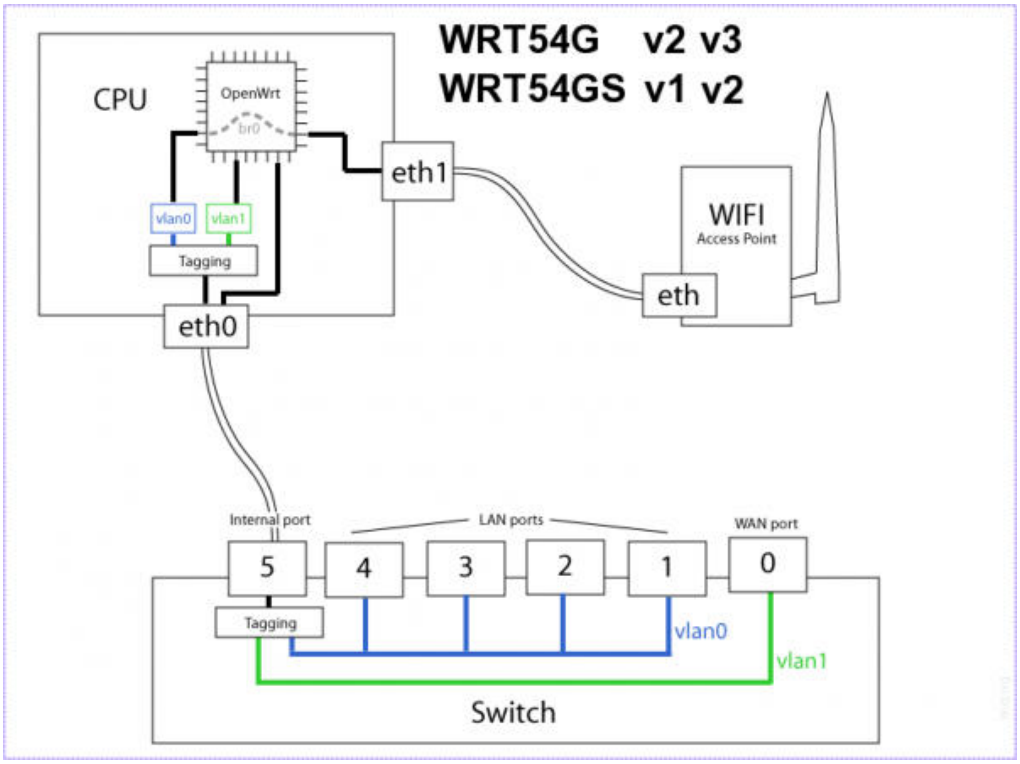
Specific Configuration

Interfaces

The default network configuration is:

Interface Name	Description	Default configuration
br-lan	LAN & WiFi	192.168.1.1/24
vlan0 (eth0.0)	LAN ports (1 to 4)	None
vlan1 (eth0.1)	WAN port	DHCP
wl0	WiFi	Disabled

Internal Architecture - WRT54G (v2, v3) & WRT54GS (v1, v2)



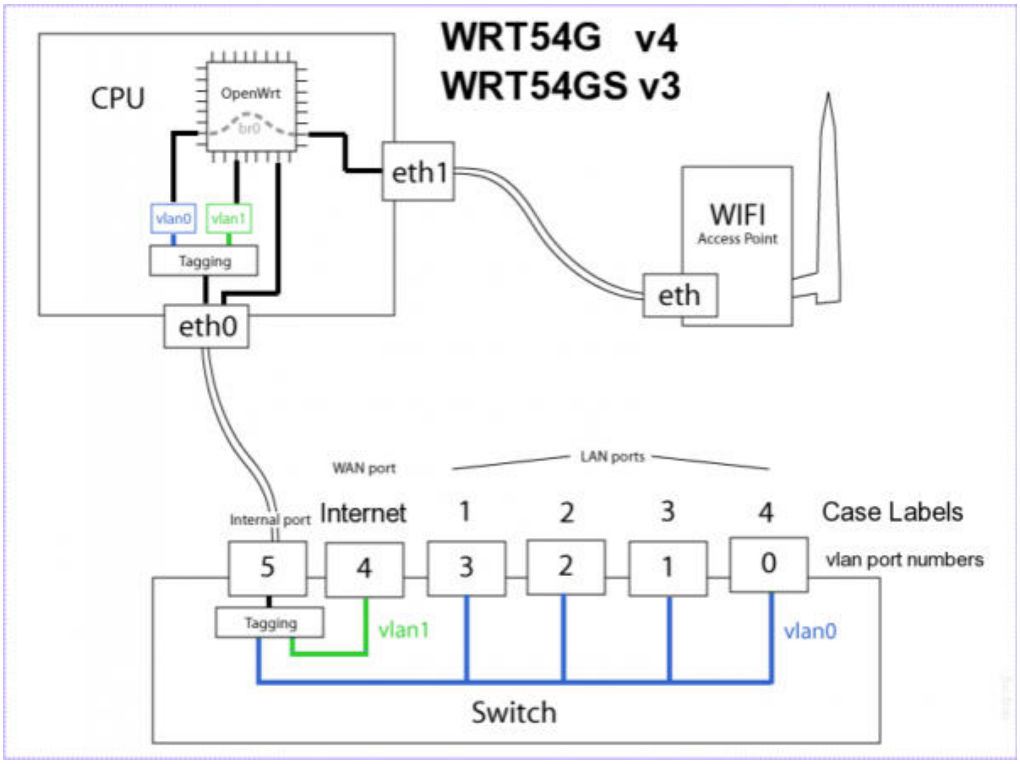
Switch Ports (for VLANs) - WRT54G (v2, v3) & WRT54GS (v1, v2)

Switch port numbers 1-4 are LAN Ports 1-4 as labeled on the unit, number 0 is the Internet (WAN) port, and number 5 is the internal connection to the router's CPU (labeled "CPU" in LuCI). Use these *switch* port numbers when specifying a VLAN's ports via the UCI (i.e., in the `ports` option of a VLAN's `config switch_vlan` section, in `/etc/config/network`).

Port	Switch port
Internet (WAN)	0
LAN 1	1
LAN 2	2
LAN 3	3
LAN 4	4
CPU (internal)	5

See: Network configuration (Switch)

Internal Architecture - WRT54G (v4) & WRT54GS (v3)



Switch Ports (for VLANs) - WRT54G (v4) & WRT54GS (v3)

Switch port numbers 0-3 are LAN Ports 4-1 as labeled on the unit, number 4 is the Internet (WAN) port, and number 5 is the internal connection to the router's CPU (labeled "CPU" in LuCI). Use these *switch* port numbers when specifying a VLAN's ports via the UCI (i.e., in the `ports` option of a VLAN's `config switch_vlan` section, in `/etc/config/network`). Don't Be Fooled: LAN Port 1 on the unit is switch port 3 when configuring VLANs on the switch.

Port	Switch port
Internet (WAN)	4
LAN 1	3
LAN 2	2
LAN 3	1
LAN 4	0
CPU (internal)	5

See: Network configuration (Switch)

Failsafe mode

If you forget your password, have broken one of the startup scripts, firewalled yourself out, or corrupted the JFFS2 partition, you can get back in by using OpenWrt's failsafe mode:

* Unplug the power cord, press and hold the reset button, put in the power cord, when DMZ-LED lits up release the reset button. When done right, both Power-LED and DMZ-LED will start blinking. Now you can ping and telnet into 192.168.1.1

See: generic.failsafe

Buttons

The Linksys WRT54G has two buttons. They are Reset and Secure Easy Setup. The buttons can be used with hotplug events. Please see the WiFi toggle Wiki page.

BUTTON	Event
Reset	reset
Secure Easy Setup	ses

Hardware

Opening the case

To remove the front cover you simply pop the front of the case off after removing the antennas. Please note that this will void the warranty.

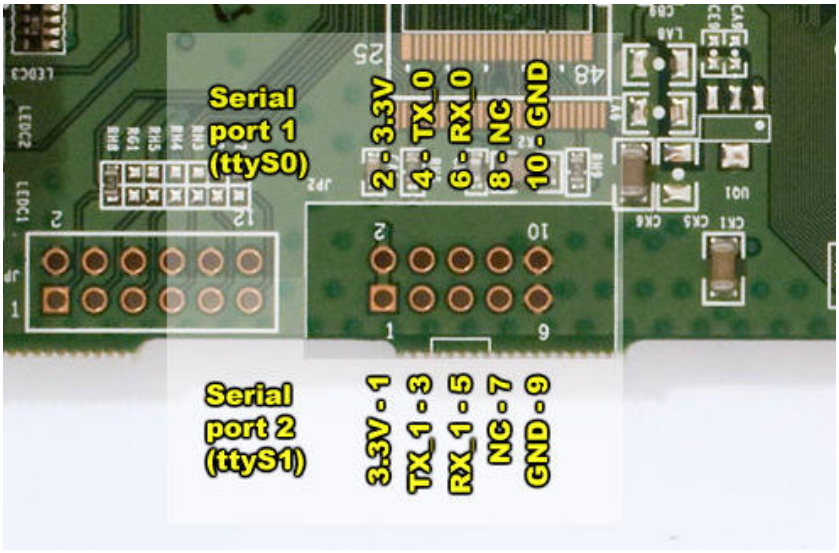
Usually there are two screws holding the PCB to the bottom cover, but on some newer versions (e.g. v2.2) there's only a single screw.

Serial port

The WRT54G/S/L has a 10 pin connection slot on the board called JP1 (JP2 on some v1.1 boards). This slot provides two TTL serial ports at 3.3V. Neither of the ports use hardware flow control, you need to use software flow control instead. Other routers may have similar connections. These two TTL serial ports on the WRT54GL router can be used as standard Serial Ports similar to the serial ports you may have on your PC. In order to do this though you need a line driver chip that can raise the signal levels to RS-232 levels. You can not directly connect a serial port header to the board and expect it to work. That method will only work with devices that can connect to TTL serial ports at 3.3V. Connecting two which have 3.3V directly will work (TX - RX, RX - TX, GND - GND). Standard RS-232 devices cannot be directly connected which accounts for nearly all serial PC devices.

Once the modification is made you can have at most two serial ports to use for connecting devices etc. By default, OpenWrt uses the first serial port to access the built-in serial console on the router. You can connect to it at 115200,8,N,1 using a terminal program like Putty, SecureCRT or minicom for example. This is helpful because if you have problems communicating with your router this method will allow you easy access connecting over a serial console. By default this leaves you with one serial port left, however, there is a method to turn the console off giving you access to both ports if you really need them. It isn't recommended but it can be done.

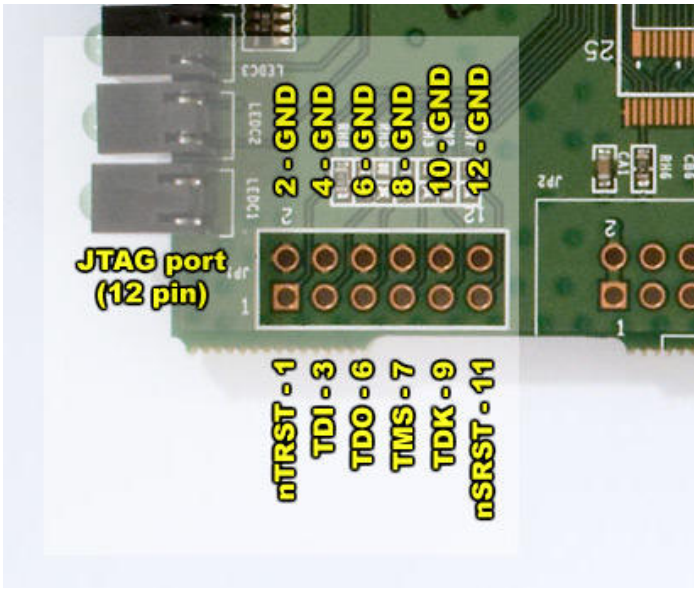
Pin 2	**3.3V**	**Pin 4**	**TX_0**	**Pin 6**	**RX_0**	**Pin 8**	**Not connected**	**Pin 10**	**GND**
Pin 1	**3.3V**	**Pin 3**	**TX_1**	**Pin 5**	**RX_1**	**Pin 7**	**Not connected**	**Pin 9**	**GND**



JTAG

The JTAG port is a unpopulated 12-pin header and is located next to the serial port header. A simple unbuffered should work fine.

Pin 2	**GND**	**Pin 4**	**GND**	**Pin 6**	**GND**	**Pin 8**	**GND**	**Pin 10**	**GND**	**Pin 12**	**GND**
Pin 1	**nTRST***	**Pin 3**	**TDI**	**Pin 5**	**TDO**	**Pin 7**	**TMS**	**Pin 9**	**TCK**	**Pin 11**	**nSRST***



See here for more JTAG details.

Photos

WRT54GL v1.1 - Serial number: CL7B

Front:

Back:

Hardware Mods

Adding an MMC/SD card

The GPIO (General Purpose Input/Output) lines can be used to add a SD card in SPI mode. Please see the GPIO page in the oldwiki.

To add an SD card with backfire and kernel 2.6, You need to mask GPIO's from b43 module. To achieve it, edit:

```
vi /etc/modules.d/30-b43
```

… putting masking just after b43, so content of above file looks like:

```
b43 gpiomask=0
```

What's presented above, is a failsafe examaple, that masks ***all*** GPIO diodes from b43. in reality, You're using only 4 GPIO's, so after ensuring that card work properly, You may tweak gpiomask to mask *only* used GPIO's. For example, value **'0x1'** disables all diodes except WiFi and power (+ 4 port switch, wchich isn't connected to GPIO at all). After doing that, **reboot**, and install following packages (install one by one, as for reasons unknown, device does like to crash when provided with many packages to install at once):

```
opkg install kmod-mmc
opkg install kmod-mmc-over-gpio
opkg install kmod-mmc-spi
opkg install kmod-spi-bitbang
opkg install kmod-spi-gpio-old
```

Then, carefully edit:

```
vi /etc/config/mmc_over_gpio
```

…setting 'enable' to "1", and providing GPIO numbers for SD's clock, data-in, data-out and select-chip. Actually, You may use other GPIO's, presented here are just example. This setup, render amber/white SES and DMZ diodes unusable, with the latter, hoever, working properly during boot. You could as good "sacrifice" connection to WiFi and power diodes, in any combination - it's up to You.

Then, just run:

```
/etc/init.d/mmc_over_gpio start
```

, and You should see message like one presented below:

```
root@OpenWrt:~# dmesg|tail
<snip>
gpio-mmc: MMC-Card "default" attached to GPIO pins di=2, do=4, clk=3, cs=7
mmc_spi spi32766.0: can't change chip-select polarity
mmc0: host does not support reading read-only switch. assuming write-enable.
mmc0: new SD card on SPI
mmcblk0: mmc0:0000 00000 1.88 GiB
 mmcblk0: p1
```

You can mount it with

```
mount /dev/mmcblk0p1 /mnt/
```

…or, mount it anywhere else, add it to be automatically enabled at startup, do some extroot, etc… There are, literally, gigabytes of possibilities :)

Adding USB ports.

Yes it is possible to add USB ports on WRT54 GSv3/GL, only downside are that it is only USB v 1.1.

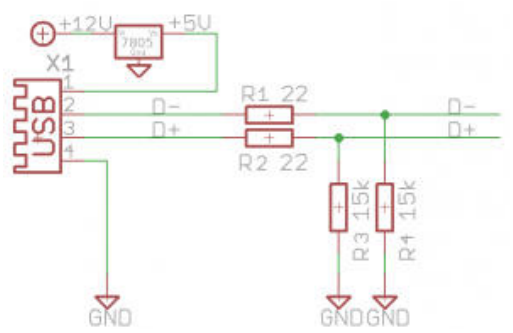
So how then? You will need this.

- Breadboard
- 4x 15 kohm resistors
- 4x 22 ohm resistors
- 2x female usb connectors

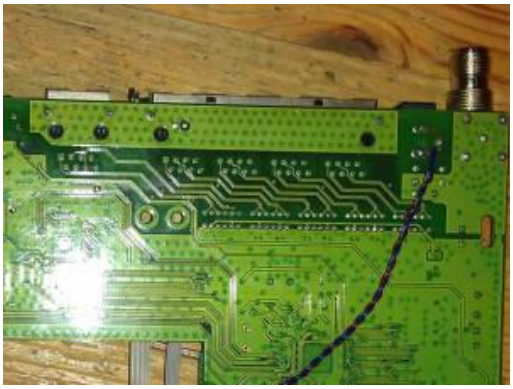
You should also be confident with a soldering iron and basic knowledge with a multimeter will always help.

This circuit also needs 5 volt output which is standard for USB, 5 Volt Regulator [http://lmgtfy.com/?q=7805+circuit+for+usb]. It's recommended to add heatsink to 7805 chip.

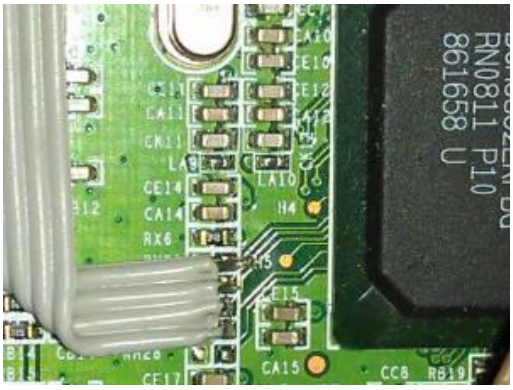
Schematics:

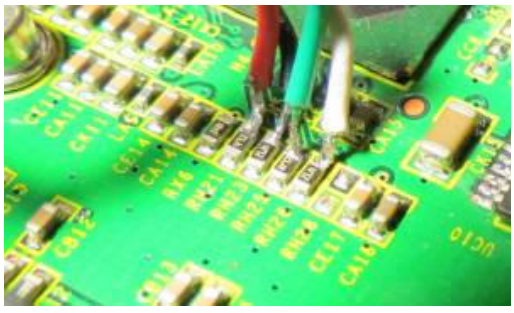


12 volt source:



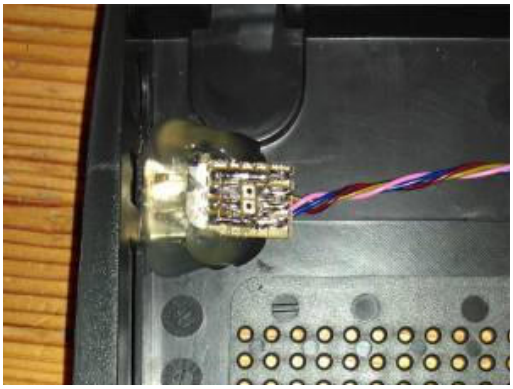
USB source soldered:





RH21 = USB1 Data + RH23 = USB1 Data - RH25 = USB2 Data + RH26 = USB2 Data -

USB Port:



When the circuits are done and everything is soldered onto the pcb of the router, it's time to install the software.

```
opkg update
opkg install kmod-usb-ohci kmod-usb-storage kmod-usb-core
```

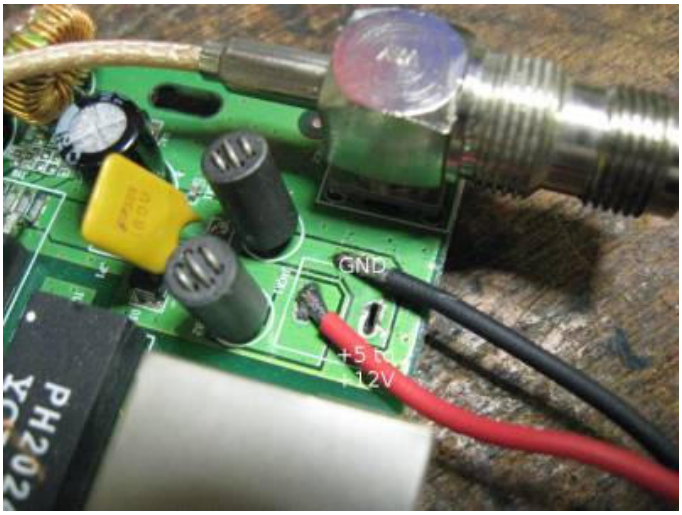
Do not forget to install the filesystem you wish to use. This page has all the extra information needed about USB on this device. usb.storage

External resources and more indepth information. Void Main's WRT54GSv3 USB Mod [http://voidmain.is-a-geek.net/wrt/wrt_usb_mod.html] Linksys WRT54GL ajout de 2 ports USB [http://knight-galaad.no-ip.org/wordpress/?p=1622] (french) > translated [http://translate.google.com/translate?hl=en&sl=auto&tl=en&u=http%3A%2F%2Fknight-galaad.no-ip.org%2Fwordpress%2F%3Fp%3D1622]

Power Supply Mods

(Disclaimer: this has only been tested on WRT54G v1.1 and a WRT54GL v2, it should be the same for other models but I can't be certain. If in doubt check voltages with a multimeter.)

If you've lost your power brick or want to power the WRT54G from an alternate source its possible to solder power cables directly to the power jack connectors. The WRT54G seems to run on anything from 5 to 12 (maybe more) volts. At 5 volts it needs about 800 milliamps, I had thought it might be possible to run it off USB but USB only (officially) supplies 500 milliamps. However, some USB ports will supply 800 milliamps and a lot of USB mains adaptors (e.g. the one for the Amazon Kindle or iPhone) supply 1 amp.

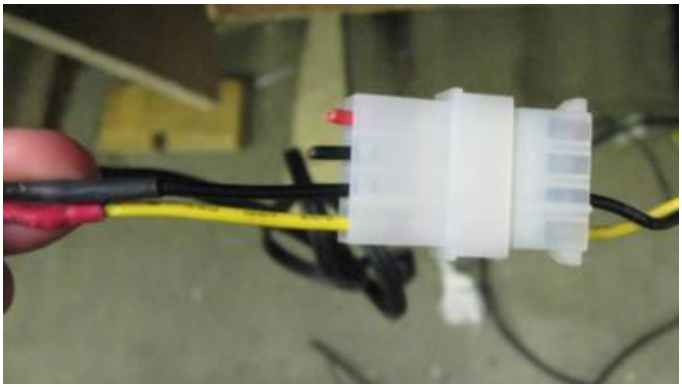


To connect up an alternative power supply, open up the case and locate the small black connector where the power input goes.

You can (as shown in the picture above) desolder the power connector (this took quite a lot of effort and I broke the connector in the process). If you want to keep it just solder to the underside of the board instead, you might need to file away a bit of plastic from the outer casing to make room for the wires.

There are 3 legs to the power connector, each just under 1cm long. The one closest towards the front of the router (the LED side) and running across the router is the positive (red wire in the picture). The one in the middle, running from back to front is the ground (black wire in the picture). Just solder a wire to each of these and connect to your power supply of choice.

If you want to run the WRT54G from USB, cut up a USB cable and solder the black wire to negative and the red wire to the positive. Or you can run it from a PC power supply by getting a male 4 pin molex hard disk power connector (as found on some PC fans or molex splitter/extension cables). Connect the yellow wire (12 volts) to the positive side and one (or both) of the black wires to the ground. Now connect this to a spare hard disk power connector on a PC power supply and your WRT54G will power up.



You can also use a 12 volt lead acid battery (e.g. a car battery) to run the WRT54G. These can peak at around 14 volts when fully charged, but this doesn't seem to cause any problems for the router.

Installing OpenWrt

Using the Linksys web GUI

It is possible to install OpenWrt directly with the Linksys web GUI. If you are initially installing OpenWrt use the Linksys web GUI, this is the easiest way.

- Download the openwrt-wrt54g-squashfs.bin firmware image from the brcm-2.4 folder to your PC.
- You can find that image at: http://downloads.openwrt.org/backfire/10.03.1/brcm-2.4/ [http://downloads.openwrt.org/backfire/10.03.1/brcm-2.4/]
- Actually the safest version to use would be **8.09**. *Hmmm, it looks like 10.03.1 is safe but you must use the **brcm-2.4** variant like in the link above.* I am not risking bricking my router for the initial flash, so I am going with **8.09**.
- Open http://192.168.1.1/Upgrade.asp [http://192.168.1.1/Upgrade.asp] in your browser or manually go to http://192.168.1.1 [http://192.168.1.1] → Administration → Firmware Upgrade
- Upload openwrt-wrt54g-squashfs.bin
- Wait 2 minutes. The router will reboot itself automatically after the upgrade is complete.
- Your router should now be telnettable at 192.168.1.1. The web interface `luci` is also available at http://192.168.1.1 [http://192.168.1.1] . Telnet is disabled and ssh is enabled once a password has been set.
- Type these commands out in telnet/ssh. This is to ensure that tftp is available, in case your router gets bricked.

```
nvram set boot_wait=on
nvram set boot_time=10
nvram set wait_time=10 #important for some models
```

```
nvram commit && reboot
```

- **You're done!** At this point, you are free to continue using brcm-2.4. However, if you wish to use brcm47xx, proceed below.
- Download the image openwrt-brcm47xx-squashfs.trx here: http://downloads.openwrt.org/backfire/10.03.1/brcm47xx/ [http://downloads.openwrt.org/backfire/10.03.1/brcm47xx/]
- Upload using luci at System > Flash Firmware
- **OR**, you can simply ssh into the router,

```
cd /tmp
wget http://downloads.openwrt.org/backfire/10.03.1/brcm47xx/openwrt-brcm47xx-squashfs.trx
sysupgrade /tmp/openwrt-brcm47xx-squashfs.trx
```

- Wait for the router to reboot and login! In case you somehow found out that you can't telnet, then ssh should be possible since your previous settings before flashing remained (unless you used `sysupgrade -n`)

Using the TFTP method

Right after flashing at your first login to OpenWrt set the following NVRAM parameters to enable tftpd at bootup:

```
nvram set boot_wait=on
nvram set boot_time=10
nvram set wait_time=10
nvram commit && reboot
```

NOTE: Do not touch any other NVRAM parameters. NVRAM is only used as environment for the bootloader. OpenWrt ignores NVRAM parameters.

NOTE: On WRT54GL (at least), you should probably use 'wait_time' instead of 'boot_time'. *bg300: Added*.

NOTE: On WRT54GS v1.1 too you have to use 'wait_time' instead of 'boot_time', anyway if in doubt add both parameters.

Once you have set the NVRAM parameters above it is possible to use a TFTP client to flash OpenWrt. The TFTP method is also **the recommended way to restore the original Linksys firmware or switch to other third-party firmwares.**

First download a firmware image file ending in ".bin", e.g. openwrt-wrt54g-squashfs.bin.

Then follow the Generic TFTP flashing instructions.

Upgrading OpenWrt

Using mtd OR sysupgrade

If you have already installed OpenWrt and like to reflash for e.g. upgrading to a new OpenWrt version. It is important that you put the firmware image into the ramdisk (/tmp) before you start flashing.

```
cd /tmp/
wget http://downloads.openwrt.org/backfire/10.03.1/brcm47xx/openwrt-brcm47xx-squashfs.trx # The file must be the trx file.
mtd write /tmp/openwrt-brcm47xx-squashfs.trx linux && reboot
```

OR

```
cd /tmp/
wget http://downloads.openwrt.org/backfire/10.03.1/brcm47xx/openwrt-brcm47xx-squashfs.trx # The file must be the trx file.
sysupgrade /tmp/openwrt-brcm47xx-squashfs.trx
```

Feel free to explore the rest of sysupgrade options by typing 'sysupgrade' in the terminal.

Using LuCI

1. Select Administration in the top right corner
2. Choose the System > Flash Firmware tab
3. Click Browse and navigate to the firmware (it must be a *.trx* file) you wish to flash to
4. Click 'Upload image'
5. Follow the instructions

Upgrading from Tomato

The instructions from the Tomato upgrade instructions [http://www.polarcloud.com/tomatofaq#installing_another_firmware_or] are confirmed to work with Backfire (10.03.1-rc4, r24045)

Other Info

Flashing via JTAG

Unverified Information!
This page or section contains unverified information. Remove this notice if you can ensure its correctness.

Kamikaze/White Russian Recovery

After successfully running a late model version of Kamikaze. I had some stability issues decided to go with white russian, I downgraded with the web gui to the latest default version of openwrt-brcm-2.4-squashfs.trx and got bricked. These instructions are linux specific, if you are running any other OS your mileage may vary.

Enter Failsafe Mode:

I was able to get into failsafe mode consistently by hitting the reset button for 2 seconds after the DMZ led lights up.

1. Power on the WRT.
2. **Wait for the DMZ light** first (failing to wait for DMZ on before button press would end up completely Linksys-destroying the nvram contents, i.e. **boot wait off** (ouch), 192.168.1.1, SSID linksys).
3. **Then** press and hold the reset button on the back until the DMZ light blinks.

Unit is now available at 192.168.1.1. If, for any reason (like interface configuration being totally messed up), the WRT doesn't show up on that IP in failsafe mode, see section "Last Resort Recovery" below.

Get Your Kit in Order:

Get the proper code into the tmp directory of your computer, for me this was:

```
cd /tmp
wget http://downloads.openwrt.org/whiterussian/newest/default/openwrt-brcm-2.4-squashfs.trx
wget http://downloads.openwrt.org/people/nbd/nvram-clean.sh
```

Update the OpenWrt Image from Failsafe:

Get the code into the tmp directory of the WRT from the failsafe telnet session and then overwrite the firmware with the new image using mtd.

```
cd tmp
scp root@192.168.1.2:/tmp/openwrt-brcm-2.4-squashfs.trx ./
mtd -r write openwrt-brcm-2.4-squashfs.trx linux
```

I found that the machine would not reboot itself nor would it reboot using /sbin/reboot. I had to unplug to make it reload. Make sure the commands you have previously typed have finished completely before you pull the juice.

According to Internet sources your WRT may be fixed at this point. Mine still would not come up, it seems that I had some nvram variables mucked up, so back to failsafe again.

Backing up your nvram variables:

It is a good idea to save these just in case you need to refer to anything previously set.

```
nvram show | sort
```

I copied the output to my laptop using

```
cat > nvram-broke
```

and pasting the output of the "nvram show" command to my terminal.

make sure you got the code by doing:

```
cat nvram-broke
```

Running the Clean Up Script:

The cleanup script gets your nvram variables back to a sane state and gets rid of anything not used by OpenWrt.

```
cd /tmp/
scp root@192.168.1.2:/tmp/nvram-clean.sh ./
chmod a+x nvram-clean.sh
./nvram-clean.sh
nvram commit
```

Throw the bones on the floor and reboot. Your WRT should be up and running.

Troubleshooting

Wireless broken by default because of wrong macaddr:

There's something wrong with mac address settings in the default /etc/config/wireless Make sure the mac address line reads the mac you find in /sys/class/ieee80211/phy0/macaddress Ref: https://dev.openwrt.org/ticket/7102 [https://dev.openwrt.org/ticket/7102]

Read only filesystem

If the file system stays in read only mode type

```
mtd unlock rootfs_data
```

Put this command in `/etc/init.d/nvram` for example, so that it is called on every start up.

(Source and details: https://forum.openwrt.org/viewtopic.php?id=25063 [https://forum.openwrt.org/viewtopic.php?id=25063])

Aircrack

The `aircrack` package doesn't fit on the router with the OpenWRT **10.03 Backfire** image.

Using the OpenWRT **8.09.2 Kamikaze** `aircrack` fits, but the `wl` package needed as well doesn't. By moving `libcrypto` from the aircrack dependency package `libopenssl` into the ramdisk (and backlinking into original directory) you can install wl as well, but `airodump` freezes on start and causes the router to reset.

So this guide is **using Kamikaze 7.09** as latest known OpenWRT version that supports the aircrack suite on WRT54G.

Install Kamikaze 7.09

If you already have a (newer) version of OpenWRT installed, the easiest way is using `sysupgrade` as described in generic.sysupgrade

Be aware that 7.09 still uses the ipkg package manager, not opkg.

IPKG backports source

You need the aircrack from whiterussian backports, because there are no aircrack packages available for 7.09 (anymore?). Edit your `/etc/ipkg.conf` to look like this:

```
src release http://downloads.openwrt.org/kamikaze/7.09/brcm-2.4/packages
dest root /
dest ram /tmp
src whiterussian http://downloads.openwrt.org/backports/0.9
```

Install aircrack and wl

```
ipkg update
ipkg install aircrack-ng wl
```

Start airodump

`ifconfig` should tell you the wifi interface is down (use `ifconfig -a` to show down interfaces as well). We use the `wl` tool for configuring the wifi to use monitor mode needed for `airodump-ng`, not `airmon-ng`.

```
wl up
wl monitor 1
```

Now `ifconfig -a` should show a `prism0` interface.

```
ifconfig prism0 up
airodump-ng prism0
```

And that's it - airodump should show you networks in range.

Capture IVs

For your wardriving purposes, you need to start airodump in background and save IVs to files. Make sure you're not in /tmp (i use / as location) and run airodump like this:

```
airodump-ng --ivs --write wep prism0 &
```

Instead of `wep` you can use any prefix you like, it is used as a prefix to the files with IVs like `wep-01.ivs`.

The `&` at the end makes airodump run in the background (it still spams your console with information). You can just close your terminal and abort the telnet session, airodump will still run as long as the router is powered or until you manually stop/kill it.

view IV list

Use `aircrack-ng` to list the networks and number of captured IVs for each:

```
aircrack-ng *.ivs
```

Tags

bcm53xx, bcm5352, MIPS
