

TP-Link TL-WDR4300

Dual Band (concurrent) and Gigabit Ethernet. Advertised as 750 Mbps it is Dual-Stream (2x2) on the 2.4 Ghz Band and Triple-Stream (3x3) on the 5 Ghz Band. Same as the TL-WDR4310 Released earlier this year in China. FCC ID = TE7WDR4300.

Related to TL-WDR3600, which has only two instead of three antennas.

Manufacturer product page is here [http://www.tp-link.com/en/products/detail/s/?category_id=2166&model=TL-WDR4300], while the support download page is here [http://www.tp-link.com/en/support/download/?model=TL-WDR4300&version=V1#tbl_1].

WARNING: Security warning: unpatched http/ftp backdoor in **original** firmware: <http://sekurak.pl/tplink-httplibtftp-backdoor/> [<http://sekurak.pl/tplink-httplibtftp-backdoor/>]

Supported Versions

The latest firmware available is the release build of Barrier Breaker 14.07 [https://downloads.openwrt.org/barrier_breaker/14.07/ar71xx/generic/], with working ethernet and dual-band wireless (disabled by default), webUI, and support for all hardware revisions through at least 1.7.

Version/Model	S/N	Board ID	Release Date	OpenWrt Version Supported	Model Specific Notes
v1.0	-	-	2012/02	r32683 [https://dev.openwrt.org/changeset/32683]	TP-Link TL-WDR4310.
v1.1	-	-	2012/06	r32461 [https://dev.openwrt.org/changeset/32461]	Similar to TP-Link TL-WDR4310
v1.2	-	-	2012/11	-	-
v1.3	-	-	2012/11	-	-
v1.4	-	-	2013/04	r36676 [https://dev.openwrt.org/changeset/36676]	NEED to Toggle HW Wifi switch after first flash
v1.5	-	-	2013/07	r37374 [https://dev.openwrt.org/changeset/37374]	NEED to Toggle HW Wifi switch after first flash. Similar to v1.4, released between 2013/04 & 2013/07
v1.6	-	-	2013/08	r37759 [https://dev.openwrt.org/changeset/37759]	-
v1.7	-	2050500272 rev 1.3	2014/01	Barrier Breaker - 14.07 [https://downloads.openwrt.org/barrier_breaker/14.07/ar71xx/generic/openwrt-ar71xx-generic-tl-wdr4300-v1-squashfs-factory.bin] [https://downloads.openwrt.org/barrier_breaker/14.07/ar71xx/generic/], or trunk image [Current: https://downloads.openwrt.org/changeset/42651], took for openwrt-ar71xx-generic-tl-wdr4300-v1-squashfs-factory.bin. The precompiled images does not activate the wireless feature by default (you will have to use ethernet for the initial configuration).	-

NOTE: The ethernet switch in this device, AR8327N, is working fine with the OpenWRT default configuration. But some of the more advanced functions of the switch are not yet fully supported by the driver in 12.09. Most notably, a port cannot be tagged in some VLAN's and untagged in another, see Bug 12181 [<https://dev.openwrt.org/ticket/12181>]. The fix had been merged into trunk with r42652 [<https://dev.openwrt.org/changeset/42652>] and r42653 [<https://dev.openwrt.org/changeset/42653>] but it hadn't been backported to BB yet.

If your wireless cannot be enabled when using wide channel modes, this may be due to the friendly neighbour "feature" that prohibits operation of such a mode and you may have to use the standard modes before wireless can be enabled.

Quick Start Guide

Barrier Breaker 14.07 provides full support for this router and has Luci (webUI) built-in.

- Download [openwrt-ar71xx-generic-tl-wdr4300-v1-squashfs-factory*.bin](https://downloads.openwrt.org/barrier_breaker/14.07/ar71xx/generic/openwrt-ar71xx-generic-tl-wdr4300-v1-squashfs-factory.bin) [https://downloads.openwrt.org/barrier_breaker/14.07/ar71xx/generic/openwrt-ar71xx-generic-tl-wdr4300-v1-squashfs-factory.bin]
- connect your PC to a LAN port of the TP-link via ethernet. Login to the TP-link web administration webpage. Under 'System Tools' select 'Firmware Upgrade'. Browse to the previously downloaded *.bin file. Click Upgrade.
- Connect to <http://192.168.1.1> with your web browser
- Set your password and configure the router through the web UI. Basic Config [<http://wiki.openwrt.org/doc/howto/basic.config>]

Note: Factory default IP address range is 192.168.0.1 while OpenWrt uses 192.168.1.1 by default. If you have trouble accessing your router after initial flash, check that you have a 192.168.x.x IP address on your PC.

Hardware Highlights

CPU	Flash	RAM	Network	WAN	USB	Serial	JTag	VLANs
Atheros AR9344@560MHz	8MB	128MB	4x1 GigE	1x1 GigE WAN	x2 v2.0	Yes	Yes	128

Installation

- It is recommended to update to stock firmware 3.13.33(130617) before installation. This firmware features bootloader with handy TFTP recovery mode – [flashing via tftp, de-brick, or oem.installation.using.the.tftp.recovery](#)
- obtain firmware Download a pre-compiled release image [Attitude Adjustment 12.09](http://downloads.openwrt.org/attitude_adjustment/12.09/ar71xx/generic/) [[https://downloads.openwrt.org/attitude_adjustment/12.09/ar71xx/generic/](http://downloads.openwrt.org/attitude_adjustment/12.09/ar71xx/generic/)], Barrier Breaker 14.07 [https://downloads.openwrt.org/barrier_breaker/14.07/ar71xx/generic/], or trunk image [Current](http://git.caesar.lan/downloads/openwrt.org/changeset/42651) [<https://downloads.openwrt.org/changeset/42651>], took for openwrt-ar71xx-generic-tl-wdr4300-v1-squashfs-factory.bin. The precompiled images does not activate the wireless feature by default (you will have to use ethernet for the initial configuration).

NOTE: Trunk images dont have luci you must install manually [luci essentials](#)

NOTE: Images with "-il-" in the name is specialised for devices sold in Israel, e.g. openwrt-ar71xx-generic-tl-wdr4300-v1-il-squashfs-factory.bin. For devices in Israel, try flashing the original image first, in most cases it will work just fine. Devices that require the Israeli firmware will show a warning on the Firmware Update page. If you see this warning, fallback to the "-il-" image. The Israeli firmware differs only in the Hardware ID, in order to enable flashing from the original firmware interface. There is no difference between the images otherwise. See this thread [<http://www.mail-archive.com/openwrt-devel@lists.openwrt.org/msg17573.html>] for details.

- [generic.flashing](#) Now write this firmware-file onto the flash-chip of your device

(For a brand new router, you could just use the vendor web UI to flash the .bin image)

- To use wifi, you need to activate wifi in the configuration, see [wireless configuration](#).
- You can always build your own image based on Attitude Adjustment/ trunk. Choose Atheros AR71xx/AT7240/AR913x platform and use the "TP-LINK TL-WDR4300 board support" profile.

Images for the TP-Link 3600 are largely compatible with a simple modification to the header of the firmware image. The PCB for both models is identical, or close to identical. The third external antenna on the 4300 is on the PCB of the 3600, but not connected to an external antenna.

WARNING: Do not flash the sysupgrade firmware via the vendor firmware web interface - only the 'factory' images should be flashed from the vendor firmware.

Flashing via TFTP

Pressing the WPS/Reset button during powerup makes the bootstrap loader enter the TFTP recovery mode. The procedure can be used to transfer a firmware image:

- assign 192.168.0.66 to your local network interface (the router uses 192.168.0.86)
- publish a firmware image via tftp: cp openwrt-ar71xx-generic-tl-wdr4300-v1-squashfs-factory.bin /srv/tftp/wdr4300v1_tp_recovery.bin
- configure your tftp server
- wait for the firmware transfer and subsequent reboot

Upgrading OpenWrt

If OpenWrt is already installed and you wish to upgrade to a newer version, you have two methods available:

- Flash Overwrite
- Generic Sysupgrade

(prior to actual flashing given availability of a serial console it's a nice idea to do dry runs, by ad-hoc RAM booting a factory.bin from a TFTP host server, via uboot "tftp" + "bootm", served via WDR3600's default 6F01A8C0.img filename).

Flash Overwrite

- Login as root via SSH
- Check memory usage with the [free](#) or [top](#) commands. The image can be up to 8MB, so only proceed if you have as much free RAM as the image size plus 6-8MB; this should not be a problem on a device with 128 MB RAM.
- An easy way to free up some RAM is to delete the symlinks to [/etc/modules.d/20-cfg80211](#), [/etc/modules.d/21-mac80211](#), [/etc/modules.d/2*-ath*](#) and [/etc/modules.d/4-9j*-*](#) and reboot. Drop caches can be useful too:

```
echo 3 > /proc/sys/vm/drop_caches
```

- wget or scp the new firmware build to /tmp/
- And finally:

```
cd /tmp
wget http://domain.tld/openwrt-ar71xx-generic-tl-wdr4300-v1-squashfs-factory.bin
mtd -r write /tmp/openwrt-ar71xx-generic-tl-wdr4300-v1-squashfs-factory.bin firmware
```

Generic Sysupgrade

Alternately, you can follow the [generic.sysupgrade](#) procedure. Don't forget to populate your `/etc/sysupgrade.conf` first.

mtd-utils

For systems where OpenWrt mtd is not available, mtd-utils commands need to be used (subsequent commands boldly assume that it's mtd5 which equals the "firmware" mtd partition name - cat /proc/mtd to verify!!):

```
flash_eraseall /dev/mtd5
handwrite /dev/mtd5 /tmp/openwrt-ar71xx-generic-tl-wdr4300-v1-squashfs-factory.bin
```

(write operation will take about 5 minutes to complete)

Note that output of newer mtd-utils flash_eraseall recommends using "flash_erase <dev> 0 0" instead (did not test it).

Flash Layout

Please read the article [Flash Layout](#) for a better understanding. It contains a couple of explanations. Then let's have a quick view at flash layout of this particular device:

TP-Link WDR4300 Flash Layout stock firmware			
Layer0	m25p80 spi [http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus]0:0: s25fl064k 8192KiB		
Layer1	mtd0	mtd1	mtd3
Size in KiB	128KiB	8000KiB	64KiB
Name	<i>u-boot</i>	<i>firmware</i>	<i>art</i>
mountpoint	none	/	none
filesystem	none	SquashFS-LZMA 4.0	none
TP-Link WDR4300 Flash Layout			
Layer0	m25p80 spi [http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus]0:0: s25fl064k 8192KiB		
Layer1	mtd0 <i>u-boot</i> 128KiB	mtd5 <i>firmware</i> 8000KiB	mtd4 <i>art</i> 64KiB
Layer2	mtd1 <i>kernel</i>	mtd2 <i>rootfs</i>	
mountpoint		/	
filesystem		overlayfs	
Layer3		mtd3 <i>rootfs_data</i>	
Size in KiB	128KiB		64KiB
Name	<i>u-boot</i>	<i>kernel</i>	<i>rootfs_data</i>
mountpoint	none	none	/rom
			/overlay
filesystem	none	none	SquashFS
		JFFS2	none

ART = Atheros Radio Test - it contains mac addresses and calibration data for the wifi (EEPROM). If it is missing or corrupt, ath9k won't come up anymore.

Failsafe mode

Power up your router. When the 'SYS' light (asterisk symbol right of the power light) starts to blink, immediately push the WPS/Reset button on the back-left of the router for a short time (>1 sec). The 'SYS' light should now start to blink very fast.

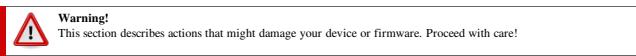
On a TL-WDR4300 Ver 1.6 and Barrier Breaker Bleeding Edge, r39211, the above instructions were not terribly successful. The only way that I was able to get the router into failsafe mode was to quickly and repeatedly press the WPS/Reset button *starting before the front panel "star" LED started flashing*. When that LED finally lit, it appeared to go directly into the rapid-flashing "failsafe" indication. If the WPS LED lights (rightmost, "yin-yang arrows"), it may be that you started clicking the button a little early in the boot sequence.

For what you can do in failsafe, go to the [OpenWrt Failsafe Mode](#) page.

Back to original firmware

 (untested)

DON'T TRY to flash wrd4300 with wrd4310 firmware and vice-versa!



The stock firmware is obtained from the OEM: <http://www.tplink.com/en/support/download/?model=TL-WDR4300> [http://www.tplink.com/en/support/download/?model=TL-WDR4300]. As with the WR1043ND router, there is also a catch with the WDR4300!

- in case the file name of this firmware file does not contain the word "*boot*" in it, you can simply revert back to original firmware. → generic.uninstall
- in case the file name of this firmware file does contain the word "*boot*" in it, you need to cut off parts of the image file *before* flashing it

An example of an image file with the word "boot" in it is `wdr4300v1_en_3_13_17_up_boot(128426).bin`.

Cut the first 0x20200 (that is 131,584 = 257*512) Bytes from original firmware: (1*512 Vendor-info + 256*512 U-Boot)

If you want to find an image that does not contain the word "boot" from the OEM, try downloading smaller zip-files first.

```
wget or scp the stock firmware file to /tmp/
cd /tmp
dd if=orig.bin of=tplink.bin skip=257 bs=512
```

(Note: File size should now be exactly: 8,126,464 Bytes)

Other caveats (from vendor web UI):

- If the firmware path is too short, it will fail with the incorrect error 'firmware path too long'. For instance, flashing c:\openwrt.bin will not work.
- If the firmware path is too long, it will fail with the error 'firmware path too long'.

Now follow → generic.uninstall

de-brick or OEM installation using the TFTP recovery

The stock firmware (3.13.33(130617)) features a TFTP recovery client in bootloader. To activate it press and hold WPS/Reset Button during powering on until WPS LED turns on. Connect computer to LAN1. Using TCPdump, you should see ARP requests from router having address 192.168.0.86 looking for address 192.168.0.66.

```
# tcpdump -ni eth0 arp
ARP, Request who-has 192.168.0.66 tell 192.168.0.86, length 46
```

Set up your computer to address 192.168.0.66, netmask /24 (255.255.255.0).

```
# ip addr add dev eth0 192.168.0.66/24
```

Using TCPdump, you should now see request for new firmware image:

```
# tcpdump -npi eth0 udp
IP 192.168.0.86.2195 > 192.168.0.66.69: 44 RRO "wdr4300v1_tp_recovery.bin" octet timeout 5
```

Rename factory image to given name and put it into TFTP server root. → generic.flashing.tftp

① In case you are flashing back original firmware, make sure original firmware image name does not contain word **boot** → back.to.original.firmware.

```
# cp openwrt-ar71xx-generic-tl-wdr4300-v1-squashfs-factory.bin wdr4300v1_tp_recovery.bin
```

attpd --no-fork --daemon .

After downloading, the flashing starts immediately. After cca. 1 minute, the router reboots automatically.

de-brick or OEM installation using the TFTP and RS232 (serial) method

If you want to de-brick/upgrade your router using TFTP you follow these steps:

Pre-requisites:

- serial RS232 connected from your machine to TL-WDR4300 & terminal program (e.g. minicom, screen) set to **115200 8N1**, no flow control, 3,3V
- copy a working & full OpenWrt firmware image into your tftp server folder (e.g: openwrt-ar71xx-generic-tl-wdr4300-v1-squashfs-factory.bin)

(in case you want to flash the original TP-Link firmware it might needed to delete the first 200 Bytes from this firmware before flashing, plz check Video Flash Steps!)

- start a tftpd server on your local machine on LAN address 192.168.1.100/24 and connect your LAN-port to one of the routers LAN ports

Video Flash Procedure: How to debrick TL-WDR4300 [http://www.youtube.com/watch?v=Ng5chUjgkE&feature=youtu.be]

Written Flash Procedure:

1. router should be unplugged & your serial line connected & terminal open & ftp server installed not yet running
2. copy your desired openwrt image for the tp-link-4300 into your tftp server folder and rename it into openwrt.bin (to save some typing within the flash procedure)
3. first goal is to get the command prompt from the u-boot bootloader on your router
4. you should only plug in the serial into the router's serial port AFTER it initialises for a split second after powering on BUT BEFORE Autobooting starts otherwise it might hang at the initialisation process
5. login in your router and be ready to type tnt & hit ENTER after you see the line Autoboot in 1 seconds:

- Official repository on GitHub: U-Boot 1.1.4 modification for routers [https://github.com/pepe2k/u-boot_mod]
- Discussion about this project on OpenWrt forum [<https://forum.openwrt.org/viewtopic.php?id=43237>]
- An article (in Polish) about one of the first version of this project on www.tech-blog.pl [<http://www.tech-blog.pl/2013/03/29/zmodyfikowany-u-boot-dla-routerow-tp-link-z-atheros-ar9331-z-trybem-aktualizacji-oprogramowania-przez-www-i-konsola-sieciowa-netconsole/>]

Original bootloader settings

(for 1.7, at least)

```
dh12x> printenv
bootargs=console=ttyS0,115200 root=31:02 rootfstype=squashfs init=/sbin/init mtdparts=ath-nor0:256k(u-boot),64k(u-boot-env),6336k(rootfs),1408k(uImage),64k(mib0),64k(ART)
bootcmd=boot 0x9f020000
bootdelay=1
baudrate=115200
ethaddr=0xXX:0xXX:0xXX:0xXX:0xXX:0xXX
ipaddr=192.168.1.111
serverip=192.168.1.100
dir=
lufttftp 0x80060000 ${dir}u-boot.bin&&erase 0x9f000000 +$filesize;cp.b $fileaddr 0x9f000000 $filesize
lf=tftp 0x80060000 ${dir}db12x${bc}-jffs2&&erase 0x9f050000 +0x630000;cp.b $fileaddr 0x9f050000 $filesize
lk=tftp 0x80060000 ${dir}vmlinux${bc}.lzma.uImage&&erase 0x9f680000 +$filesize;cp.b $fileaddr 0x9f680000 $filesize
stuartserial
stdns=serial
stdns=serial
ethact=eth0
```

Environment size: 686/65532 bytes

db12x>

Changing variables through 'setenv' doesn't seem to make the changes stick, unfortunately.

Hardware

Info

Instruction set [http://en.wikipedia.org/wiki/Instruction_set :	MIPS [http://en.wikipedia.org/wiki/MIPS_architecture] MIPS 74Kc
Vendor:	Qualcomm Atheros [http://en.wikipedia.org/wiki/Qualcomm_Atheros]
Bootloader:	U-Boot
System-On-Chip:	AR9344 (MIPS)
CPU/Speed	560 MHz
Flash-Chip:	Spansion FL064KIF does [http://www.spansion.com/Products/memory/Serial-Flash/Pages/Spansion%20FL.aspx]
Flash size:	8192 KiB
RAM:	128 MiB
Wireless No1:	SoC-integrated: Atheros AR9340 2x2 MIMO for 2.4GHz 802.11b/g/n
Wireless No2:	separate Chip: Atheros AR9580 3x3 MIMO for 5GHz 802.11a/n
Switch:	Atheros AR8327N
USB:	Yes 2 x 2.0 (GL850G chip - 4 port's capable)
Serial:	Yes
JTAG:	Yes

Power

PSU (power supply)

The TL-WDR4300 DE (v1.1) comes bundled with the following PSU:



Specifications:

Brand/Model	Leader Electronics Inc / LEI F7
Input	100-240V~ (50/60Hz, 0.6A)
Output	12.0V 1.5A
Measured output	12.15V
The plug (on the router side) has the following specifications:	
Outer diameter	5.5mm
Inner diameter	2.1mm
Length of the shaft	9.5mm

GPIO

→ port.GPIO The AR933x platform provides 30 GPIOs. Some of them are used by the router for status LEDs, buttons and other stuff. The table below shows the results of some investigation:

GPIO	Common Name	PCB Name	Voltage level at GPIO in output-mode		gpioX/value in input-mode when GPIO is:		
			gpioX/value=1	gpioX/value=0	Floating	Pulled to GND	Pulled to Vcc
0	JP1-9						
1	JP1-3						
2	JP1-5						
3	JP1-7						
4							
5							
6							
7							
8							
9							
10							
11	LED USB1	DS8,R313					
12	LED USB2	DS8,R314					
13	LED WLAN2G	DS6					
14	LED System	DS4					
15	LED QSS	DS5					
16	WPS Button						
17	WiFi Switch						
18	External LNA0						
19	External LNA1						
20							
21	USB2 Power						
22	USB1 Power						

To make the GPIOs available via sysfs, the required ones have to be exported to userspace, as it is explained on a page of the Squidge-Project [<http://squidge.sourceforge.net/gpio/>]. Kernel modules occupying that resource need to be removed before (e.g. "leds-gpio" and "gpio-buttons"). In output-mode, voltage levels of the GPIOs were measured against GND, after the value 1 or 0 had been written to /sys/class/gpio/gpioX/value. In input-mode, the value of the file /sys/class/gpio/gpioX/value was read when the GPIO was floating (initial state), pulled to GND or pulled to Vcc.

The 5GHz LED seems not to be controlled via GPIO.

XXXXXXXXXXXXXX

HARDWARE MODIFICATIONS

USB Modification

The task was to make ext-root without using the default ports.

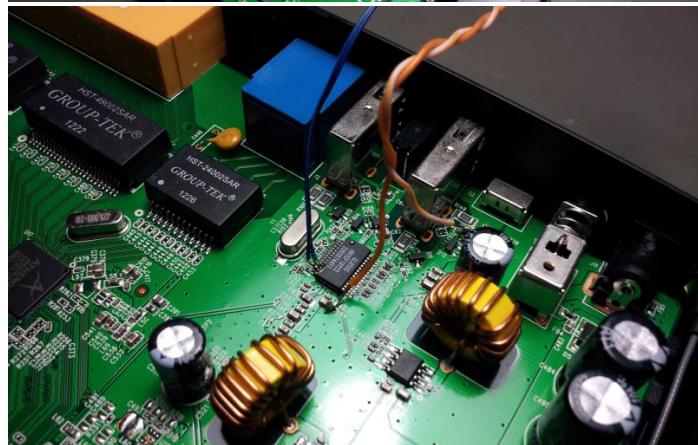
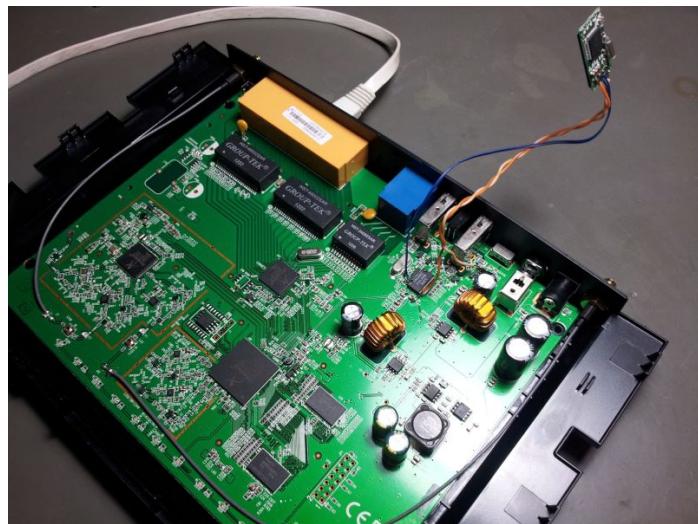
It turns out that the GL850G chipset used by the TP-Link in WDR3600/4300/4900 models can handle up to four ports.



Figure 3.2 – GL850G 28 Pin SSOP Pinout Diagram

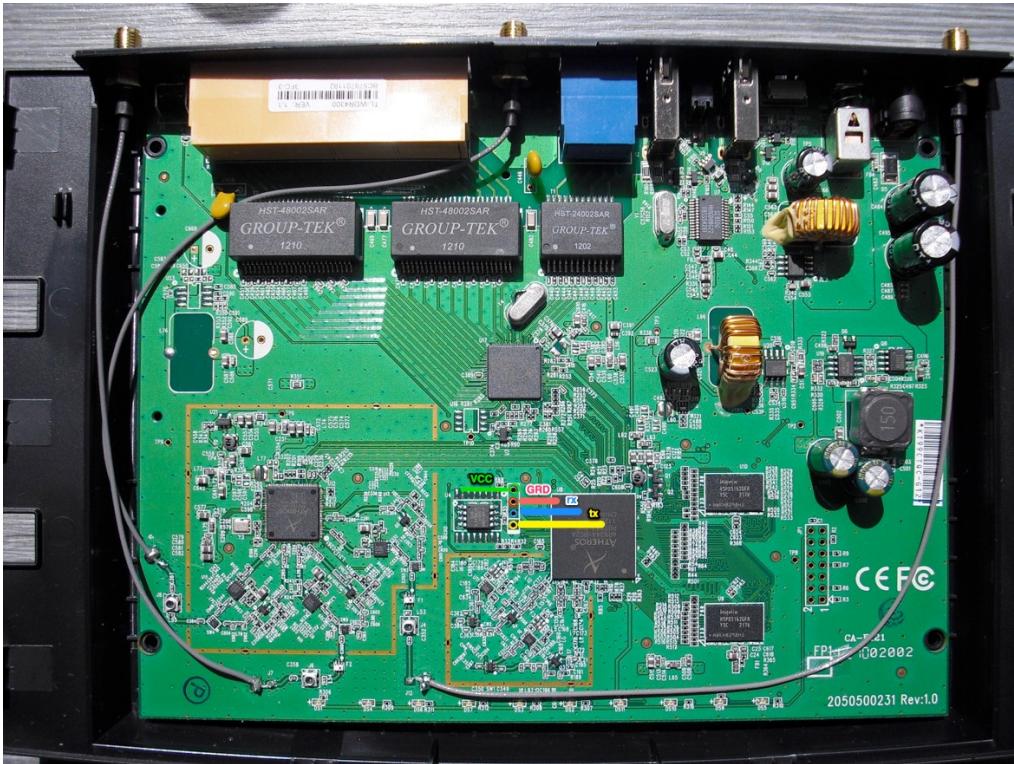
Analysing the router's PCB it appears that pins 8(D-), 9(D+), 11(D-) and 12(D+) are unused. Additionally each factory USB port has separate power section.

GND is at the TP7 pin point. +5 V was taken directly from the MOSFET.





Photos



Opening The Case (V 1.1)

Remove the 4 screws on the bottom of the case.

The top is clipped to the bottom of the case at 9 attachment points: 3 on each side of the case, 1 on the back, and 2 on the front. Each attachment point consists of two pins which fit into holes in tabs which protrude from the other half of the case. All of the tabs are on the bottom of the case, with the exception of the case back, where the single tab is in the center of the top of the case.

One method known to work, once, is to start at one of the rear corners. The corner by the ethernet ports seems to work best. Gently flex the case and slightly separate the top from the bottom at the corner by lifting on, or inserting a fingernail or other thin object into, the crack above the antenna. While doing this insert the tip of a knife blade (upward, given the geometry as the unit normally sits) into the crack between the two halves along the side of the case toward the rear. This will force the pins in the top of the case outward, flex the tab protruding from the bottom of the case inward, and free the pins from the tab. If necessary the knife tip may be levered slightly toward the case interior after insertion. Due to the force separating the top of the case from the bottom near the antenna, the pins should pop out of the tab located on the case side near the rear, lift slightly upward, and remain free.

Continue to free the other tabs, first working from the rear corner toward the front of the case, then across the front of the case, and finally from the front of the case toward the rear along the opposite side. The two halves of the case will then separate without having to work at freeing the last attachment point at the rear of the case.

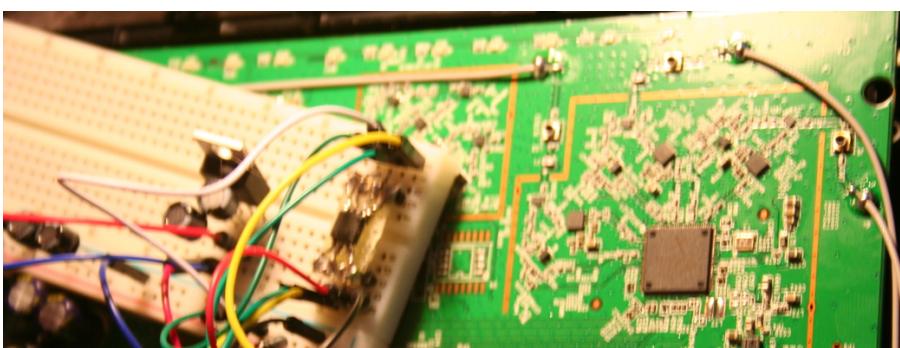
With care, this method leaves no marks on the case.

de-brick using in-system-programming

 **WARNING:** risk of frying your hardware. only do this when you understand basic electric engineering.

When the bootloader was trashed as well, and none of the above recovery methods work, you can de-brick the thing using flashrom, see <http://flashrom.org/ISP> [<http://flashrom.org/ISP>].

If you don't have one of those fancy SOIC clips, desolder the flash chip (google for SOIC desoldering for your favorite method)





Serial console

Serial console is available on the J1 (1.7) connector, 3.3v signals.

1 = TX out
2 = RX in
3 = GND
4 = VCC 3.3V
DO NOT CONNECT VCC. Use only TX/RX/GND.

Baud Rate: 115200

Data Bits: 8
Parity: No
Stop Bits: 1

To break bootstrap sequence, type 'tp1' during the 1-second boot delay.

Factory firmware login credentials are not known at this time (it's not root/up as with other tp-link models).

TP-Link TL-WDR4310 Version 1.0 forum thread

<https://forum.openwrt.org/viewtopic.php?id=36534> [<https://forum.openwrt.org/viewtopic.php?id=36534>]

Add-ons of the Router TL-WDR4300

- Installing packages into the USB storage [<http://nixorids.blogspot.com/2013/03/installing-packages-into-usb-on-tl.html>]
- Adding a USB wireless adapter to the router [<http://nixorids.blogspot.com/2013/03/wireless-adapter-tl-wn7200nd-with.html>]

Performance test with trunk/r35995

Tested with [http over nginx](#) ← wan-tldr4300 ← lan-[Client]

mbit	mode
~880	switched
~400	routed
~300	nat

Tags

GigabitEthernet, 2NIC, 2WNIC, 5Port, USB, 2USB, JTAG, 802.11bgn, ath9k, AR9344, AR9580, 802.11abgn simultan, 3Ant, DetachableAntenna, 128RAM, 8Flash, MIPS, MIPS32, 74Kc, DB120

tob/tp-link/tl-wdr4300.txt · Last modified: 2015/03/02 10:48 by debrah