

Learn about OpenWrt

- Supported devices
- Packages
- Downloads
- Documentation
- Quick start guide
- User guide
- Developer guide
- Security
- FAQ
- Forum

Contributing

- Submitting patches
- Reporting bugs
- Contributing to wiki

Project

- About OpenWrt
- Rules
- Infrastructure
- Donate
- Merchandise
- Website
- Trademark policy
- License
- Contacts

TP - Link Archer VR1200v (v2)



Under Construction!

This page is currently under construction. You can edit the article to help completing it.

The TP - Link Archer VR1200v (v2) is an ADSL2 / VDSL2+ modem with 4x gigabit LAN ports, 1x gigabit WAN port, 2 RJ-11 VoIP ports. It has 2 MediaTek wifi chips, one with 2.4Ghz b/g/n and the other with 5.0 Ghz abgn+ac.

OpenWrt support

Not supported yet. Work in progress.

At current, there is a pull request which builds an image that can boot to a console with working serial port and NAND flash. Wifi (PCIe) is expected to be relatively easy to accomplish, but Ethernet, ADSL, and VoIP DSP support are of unknown complexity.

Supporting activities

- 🔗 [OpenWrt support for Zyxel PMG5617GA, first GPON support !?](#) is the central discussion point for en75 platform support.
- 🔗 [Pull request 17928](#) adds "boots-to-a-console" initial support.
- 🔗 [New Platform: EcoNet EN75xx MIPS \(Mailing List\)](#)

Experimental firmware

You can currently install OpenWRT based on 🔗 [Pull request 17928](#) which will boot to a console but has (almost) no working drivers yet. In order to interact with it, you will need [serial access](#) access (which requires soldering).

- 🔗 [openwrt-en75-en751221-tp link_archer-vr1200v-v2-squashfs-sysupgrade.bin](#) (based on Feb 28 2025 main branch)

Hardware highlights

| CPU | Ram | Flash | Network | WLAN | USB | Serial | JTag |
|-------------------|---------|---------|----------------|---------|-----|--------|------|
| EN7513G @ 900 MHz | 128 MiB | 128 MiB | 5x 10/100/1000 | abgn+ac | No | Yes | No |

Installation

This device uses A/B Partitioning, where there is a flag in flash that tells the bootloader whether to try booting partition A, or partition B. OpenWRT is designed to install in partition A "only", and not to overwrite other parts of the flash, so that the installation is effectively dual-boot between OpenWRT and the factory QS.

TP - Link uses 🔗 [cryptographically signed updates](#) and since we don't have the private key, we cannot make updates that will work in the router's Web UI. At this time, the only way to install OpenWRT is from the bootloader using a serial connection.

Installing OpenWRT from the bootloader

- Download or build the image file `openwrt-en75-en751221-tp link_archer-vr1200v-v2-squashfs-sysupgrade.bin`
- Get the length of that file in hex (you'll need this for step 5)

```
printf '%x\n' "$(stat -c%s openwrt-en75-en751221-tp link_archer-vr1200v-v2-squashfs-sysupgrade.bin)"
```

- Connect to device with xmodem capability, e.g.

```
picocom --send-cmd lsx -vv -b 115200 /dev/ttyUSB0
```

- Switch device on and press a key to stop the bootloader

```
Press any key in 1 secs to enter boot command mode.
.....
```

- Type the xmodem receive command (don't press enter yet)

```
xmdm 80020000 <file length hex>
```

- Copy the filename to your clipboard, you'll need it in a second.

- Press enter and then "quickly" start xmodem and send the file, in picocom that is: `ctrl+a ctrl+s <paste-the-file-name> enter`.

```
bldr> xmdm 80002000 786630
CCCC
*** file: openwrt-en75-en751221-tp link_archer-vr1200v-v2-squashfs-sysupgrade.bin
$ lsx -vv openwrt-en75-en751221-tp link_archer-vr1200v-v2-squashfs-sysupgrade.bin
Sending openwrt-en75-en751221-tp link_archer-vr1200v-v2-squashfs-sysupgrade.bin, 61644 blocks: Give your local XMODEM receive command now.
```

Normally you should see it begin counting the bytes as it transfers. If the transfer fails to start, wait 30 seconds to a minute for the bootloader prompt to return and then try steps 5,6,7 again.

- Once the transfer has completed successfully, type the following

```
flash 1c0000 80020000 <file length hex>
```

- Type `re` or simply restart the device to boot into OpenWRT. If it boots the factory firmware, that just means you need to switch the QS, see below.

Debricking

→ [generic.debrick](#)

For recovery, flash backup, or other testing/development, you can use `xmdi` to load the Generic EN751221 initramfs kernel into memory and start it without ever touching flash.

- Download 🔗 [openwrt-en75-en751221-en751221_generic-initramfs-kernel.bin](#)
- Follow steps 1 through 8 of *Installing OpenWRT from the bootloader* (above)
- Instead of flashing, type `jump 80020000` to boot the kernel from memory

Switching OS

You can switch between booting partition A and partition B in the bootloader or in OpenWRT. There is currently no known way to access the factory QS without modifying it.

The switch is in the first 8 bytes of the `reserve` partition (example: `0000000101010003`). Byte 7 is the sum of all other bytes, if either byte 4 or byte 5 are zero, the master QS (A) will be booted. The example above boots QS B.

Switching in the bootloader

If you are unable to get to the factory QS, you can also switch the flag in the bootloader.

- Stop the bootloader and login with the username and password.
- Write the flag to memory that you wish to use:
 - Booting to partition A (OpenWRT)
 - `memwL 80020000 00000001`
 - `memwL 80020004 01000002`
 - Booting to partition B (Factory)
 - `memwL 80020000 00000001`
 - `memwL 80020004 01010003`
- Commit the data to flash `flash 03fe0000 80020000 08`
- Type `re` to reset the device, or power cycle it to boot into the other QS

Switching in OpenWRT

The en75 port of OpenWRT comes with a utility for switching the boot flag, it is called `en75_chboot`.

To read the current state of the flag, use `en75_chboot`

To set the flag to QS B (Factory), use `en75_chboot factory`

To set the flag to QS A (OpenWRT), use `en75_chboot openwrt`

Hardware

Info

| W Architecture | W MIPS |
|----------------|---|
| Vendor | 🔗 EcoNet |
| Bootloader | Modified 🔗 TcBoot |
| System-On-Chip | EcoNet EN7513G - MIPS 34Kc V5.8 |
| CPU/Speed | 2 Thread 1 Core 900 MHz |
| RAM Chip | Unknown 128MB |
| Flash Chip | ESMT F50L1G41LB 128MB SPI NAND |
| Wireless | MT7592(N) 2x2 U-FL 2.4Ghz Wi-Fi 4 (a/b/g/n) + MT7613BE(N) 2x2 5Ghz Wi-Fi 5 (a/b/g/n/ac) |
| Ethernet | 4x GbE LAN (10/100/1000Mbit/s + Link & Act LED) |
| DSL | 1x via EcoNet EN7556(N) driver |
| Switch | Built-in MT7530 to SoC |
| Serial | Yes soldering required |
| GPIO Buttons | 3 (WiFi, WPS, Reset) |
| GPIO LEDs | 11 (Red/Blue) |
| Power | External 12V 1.5A, with a 5.5mm barrel connector |

Flash Layout

The OEM and bootloader use a custom bad block remapping algorithm to make the NAND flash more reliable. There is approximately 118 MiB of usable space, though the exact amount depends on how many bad blocks there are. Supporting the bad block algorithm in OpenWRT is obligatory because it is used in the bootloader.

The firmware and firmware_slave partitions each contain a kernel and rootfs partition, they are prefixed by a 512 byte custom header containing the kernel and rootfs sizes and a CRC32 of the content. The bootloader reads bytes in the "reserve" partition to decide whether it will boot the main kernel or the slave kernel, while booting if checks the CRC32 and if it is not a match, it attempts booting the other kernel.

The layout of the flash is as follows:

| Layer0 | raw flash | | | | |
|--------|--|--------|------------|------------------|--------------|
| Layer1 | user servicable area (bad block table) | | | | reserved |
| Layer2 | "bootloader" | "misc" | "firmware" | "firmware_slave" | "reserve" |
| Layer3 | | kernel | rootfs | kernel_slave | rootfs_slave |

A detailed representation is below, with unpartitioned and bad-block-table reserved space shown in brackets.

| Start Address | End Address | Partition | Size | Notes |
|---------------|-------------|---------------------------|------------|---|
| 0x00000000 | 0x00080000 | "bootloader" | 512 KiB | The tcboot.bin bootloader binary |
| 0x00080000 | 0x001c0000 | "misc" | 1.25 MiB | Appears to be encrypted or compressed |
| 0x001c0000 | 0x009c0000 | "kernel" | 8 MiB | A 512 byte custom header followed by an LZMA compressed kernel |
| 0x009c0000 | 0x02000000 | "rootfs" | 22.25 MiB | A squashfs rootfs |
| 0x001c0000 | 0x02000000 | "firmware" | 30.25 MiB | Kernel and rootfs concatenated, this is also the update file format |
| 0x02000000 | 0x02800000 | "kernel_slave" | 8 MiB | Same as above |
| 0x02800000 | 0x03e40000 | "rootfs_slave" | 22.25 MiB | Same as above |
| 0x02000000 | 0x03e40000 | "firmware_slave" | 30.25 MiB | Same as above |
| 0x03e40000 | 0x03fe0000 | <unpartitioned space> | 1.625 MiB | All ffff |
| 0x03fe0000 | 0x04000000 | "reserve" | 128 KiB | Only first 8 bytes used, rest is ffff |
| 0x04000000 | 0x075e0000 | <unpartitioned space> | 53.875 MiB | All ffff |
| 0x075e0000 | 0x08000000 | <reserved> (inaccessible) | 10.125 MiB | Bad block table and remapped blocks |
| TOTAL | | | 128 MiB | |

Photos

Opening the case

To open the case, first take out the 4 screws on the back, then go around the crack between the faceplate and the back shell using a small flathead screwdriver to pry apart the snap hooks that hold the two together. Once open, the board comes out without disconnecting any wires.

Serial

→ [port.serial](#) general information about the serial port, serial port cable, etc.

Serial pins are present on the Archer VR1200v (v2), but you need to solder to them. The order of pins (starting from the side closest to the ports) is TX, RX, GND, VCC. You can connect with normal 115200 8n1 3.3v uart.

Bootlogs

OEM bootlog

| |
|--|
| <pre>BGA IC Xtal:1 DDR3 init. DRAMC init done. Calculate size. DRAM size=128MB Set new TRFC. ddr-1066 7512DRAMC V1.2.2 (0) Set SPI Flash Clock to 25 Mhz EN751221 at Thu Feb 9 13:01:21 UTC 2023 version 1.1 free bootbase Memory size 128MB Set SPI Flash Clock to 25 Mhz spi_nand_probe: mfr_id=0xc8, dev_id=0x1 Decoded SPI NAND Flash - >SPI_NAND_DEVICE_ID_F50L1G41LB, Flash Size=0x8000000 bmt pool size: 81 BMT & BBT Init Success tclinux partition size = 0x7f00000, real size = 0x81e192c tclinux partition size < its real size</pre> |
|--|

OpenWrt bootlog

| |
|--|
| <pre>BGA IC Xtal:1 DDR3 init. DRAMC init done. Calculate size. DRAM size=128MB Set new TRFC. ddr-1066 7512DRAMC V1.2.2 (0) Set SPI Flash Clock to 25 Mhz EN751221 at Thu Feb 9 13:01:21 UTC 2023 version 1.1 free bootbase Memory size 128MB Set SPI Flash Clock to 25 Mhz spi_nand_probe: mfr_id=0xc8, dev_id=0x1 Decoded SPI NAND Flash - >SPI_NAND_DEVICE_ID_F50L1G41LB, Flash Size=0x8000000 bmt pool size: 81 BMT & BBT Init Success tclinux partition size = 0x7f00000, real size = 0x81e192c tclinux partition size < its real size</pre> |
|--|

Notes

Tags

How to add tags

- EXAMPLETAG
- 🚫 Unsupported devices

Last modified: 2025/03/03 11:10 by cjd

Self-registration in the wiki has been disabled.
If you want to contribute to the OpenWrt wiki, please post 🔗 [HERE](#) in the forum or ask on IRC for access.

