Steak Electronics

Contents

1 Edimax WAP1750

```
From:
https://openwrt.org/toh/devolo/devolo_wifi_pro_1750e
```

[1] The devolo WiFi pro 1750e is a wall mountable access point with:

Dual band 2.4 and 5 GHz WiFi (3 by 3 IEEE 802.11n 450Mbps, 3 by3 IEEE 802.11ac 1299Mpbs)

- 2x 1Gbit ethernet ports (with Power over Ethernet support)

[1] Always type out full URL in links. Don't put hyperlinks. This is done, so that when you print out a webpage, the URLs are still readable.

- 1x USB 2.0 port
- External serial console port (Currently unsupported but internal 4-pin internal Dupont connector is functional)
- 1x GPIO attached piezo beeper

The access point is manufactured by OEM Acelink Technologies Co., Ltd. as a EW-7679WAC and was originally sold by Edimax as a WAP1750. The Edimax unit was reviewed by SmallNetBuilder.

1.1 Flashing Openwrt on to the WAP1750

At the moment, this page:
https://openwrt.org/toh/devolo/devolo_wifi_pro_1750e discusses flashing new firmware via SSH. However, the WAP1750 has a custom console, and while it's running GPLv2 GNU Linux, somehow it managed to lock down the software, so that simply shelling into it, is no longer possible. I'm sure that is a sin. GPLv2 does not mean people can take it and then abuse it for commercial use. It's a community good.

None the less, it can be turned into an upgradeable router via the following means:

- Teardown enclosure
- Remove flash chip by desoldering with hot air
- Read flash with appropriate tool (Flashrom, TL866A).
- Patch existing image sysupgrade for Devolo 1700 series.
- Write flash with new image
- Resolder chip to board
- Access board via FTDI / USB-Serial chip

Simple enough, right? It is simple. But it's not simple. At least 5 years of hobbyist electronics and I'm able to do this, but for the layman... Only with a guide. And let's also talk about the issues that arose during the above steps. It's never as easy as it would be in a normal world. Earth is not normal. Earth is tilted.

1.2 Teardown Enclosure

The first time, I went careful with a plastic spudger. This device is actually well designed as far as teardowns go. The second time, I got angry and snapped off the front plate with a screwdriver. In fact, the screwdriver approach (here) was superior. Much faster. Although scratches are left. Scratches only hurt those who favor form over function. Fools.

1.3 Remove Flash chip by desoldering with hot air

Once you teardown into the board, you'll examine the ICs. The flash is nowhere to be seen. Is it onboard the CPU? No, it's on the bottom of the board. So it must be lifted out of the enclosure. With hot air desoldering, you must also use lead solder first on the pins. This guide is not a tutorial on hot air soldering. I assume you know this kind of stuff. I won't go into it, unless notable.

Not all tweezers can grab these large chips easy. I have one pair that fits them, and about 3 that don't. Can't have enough tweezers, it seems.

1.4 Read flash with appropriate tool

I have a new rule. Every EE should have a toolbox of flash reading / writing equipment. It's necessary.

My weapon of choice for this project was the Beaglebone black with flashrom. A TL866A could also work. Flashrom is compatible with lots of hardware. An RPI can be used. Doesn't matter, but you must set it up. And when you have the chip, you will need:

- Breakout boards for the IC, as those clips are useless [2]

- Soldering Iron

[2] I have two of those flash-in-place clips. Pomona used to sell them for $40 each, then China came in and sells them for less than 12 dollars each now. Some for 5. They can work, but the one I had barely reached to all the pins. It wasn't big enough. Not worth it. Get breakout pcbs unless you do this a lot.

- Hot Air station
- Lead solder
- Tweezers
- Experience to not damage pins during the operation
- Short wires between breakout and BBB
- Proper connecting of wires to BBB

I got a tote/toolbox of a few pages of documentation on flashing via BBB, and also the pinouts of the P8 and P9 headers of the BBB. Printouts help.

A guide on the BBB flashrom setup is here: (insertlibrebootlink)

BBB SRM manual has the P8 and P9 headers.

Since I've used flashrom in the past to flash Libreboot, this was not new territory for me. Experience with this helps. [3]

1.5 Patch existing image sysupgrade for Devolo 1700 series.

This was easy enough (what I tried worked the first time). Based on this link on the forum for tp link devices https://forum.openwrt.org/t/debricking tl-wr1043nd-v4-hard-way-by-external-flashing-solved/7675/6 So the command was similar.

```
sudo dd if=lede-17.01.4-ar71xx-generic-tl-wr1043nd-v4-squashfs-sys
```

Except I knew the proper directory was at 0x70000 on my chip, thanks to the partition map at the devolo page, the bootlog, and also the partitions page here which explains all this:
https://openwrt.org/docs/techref/flash.layout

Another interesting, but not required reading:
https://openwrt.org/docs/techref/flash

[3] Such as, keep wires short, make multiple reads, and compare with md5sum afterwards, don't fry chip.

1.5.1 The Common IT Company Rant

Basically, any IT admin who sells / maintains a flash based device, and doesn't have a plan to replace the flash when it fails (which it very likely will at some point) is a hack. It's the number one cause of built in obsolescence. Even Tesla ran into trouble with their cars[4].

I had an argument at work with a 'normal' IT company that advocated that 'Hardware firewalls' were more secure than 'software firewalls'. There's of course, no such thing as a hardware firewall. There are no dedicated ICs for firewalls. There are ARM chips tailored for firewalls (or perhaps just networking equipment), but these are just ARM chips, for the most part. One argument from this IT company was that my x86 PC was less secure than a commercial MIPS or ARM product off the shelf, e.g. Juniper, or Cisco[5]. They considered the Juniper to be a 'hardware firewall'. Bollocks. Those are also software firewalls, just as much as the x86 is. There's a whole page on stack exchange on this. It's not even an argument. It appeals to the uneducated. It's a nonsensical argument. It's like saying a hardware desktop is better than a software desktop. It doesn't make sense.

Ref: https://serverfault.com/questions/268542/hardware-firewall-vs-software-firewall-ip-tables-rhel

The second argument they had was that Open Source is less secure than commercial products because "Any hacker can look at the code". I could rebut this with the identical hyperbole of "Any hacker can buy your commercial firewall off of ebay and hack it", but that's childish. Let's not. Here's what matters: No one with a whit of integrity would say Free and Open source software is worse than Proprietary software. You can't. In 2020, it's 20 years too late for

[4] Some early electric cars by Tesla in the 2020's had their onboard flash fail. Flash is a problem, and a big one for the long game. Never trust flash.

[5] I will ignore the backdoors these companies load into their firewalls for the NSA, although it may be noted here. Complete security is a myth. There is no safe space in life. "There be no shelter here" - Popular song.

that. The success of Google was proof that FOSS works.[6]

I read somewhere the following, and it seems right:"Humans write good software, and they write bad software. Most of it is not good. Whether it's proprietary or foss has zero relation on the quality of the software. "

And if anything, FOSS benefits from the 'many eyes' approach, that will ensure that successful software at least has a lot of people reviewing it. Whether it's good or insane complexity is another matter.

Anyways, my point is: IT companies can be insidious profit chasing vermin. I'm sure they aren't all bad, but the common IT company is about as good as the common hamburger (i.e. fast food). They will sell you everything Windows, and in 5-10 years, they will sell you it again, and again. The idea of updating a server without replacing hardware is almost unheard of. All firewalls are sold, and expected to be landfilled in 5-7 years. It's not sustainable business. Humans need to have a minimal footprint on planet Earth, and they need to reuse as much as possible. It's a sin.

I know this planet appears a joke, but that's not going to stop me from doing what should be done. And a spade will be called a spade.

1.6 Write flash with new image and Resolder chip to board

After dd'ing the new image bin, you write it back to the flash chip. This is not dramatic, just don't forget to do so. If you have a solid flashrom and pcb setup, it will just work.

Resoldering the chip to the board is also trivial, assuming you have hot air and lead solder. Not all the pins have to be connected in a 16 pin EEPROM. This board happens to have two footprints.

[6] Additionally, Sophos bases its XG firewalls off of GNU Linux. So much for that argument.

Underneath the 16 pin IC, is... surprise: a 8 pin footprint (same ic). So you can swap in the smaller package if you want.[7]

1.7 Access board via FTDI / USB-Serial chip

After that, you will again need to have serial access. FTDI is my goto as I've learned over the years that my cheapo PL2302 usb serial adapters are not reliable.[8]

Finally, you can use the board.

1.8 Dumb AP is Ideal AP

I just learned the value of a Dumb AP with this project. What is a dumb ap? It's essentially a Wifi AP with zero configuration. Maybe it can't be logged into, it has no subnet, it doesn't do anything outsde of connect wireless clients to the lan. It's like a wireless switch. It's beautiful.

I know that it isn't 100% future proof[9], but the idea that Wireless APs could be as future proof as a network switch is wonderful. No more management console, no busywork interfaces... Just configure the SSID / Password, and plug it in to ethernet. That's it. It's such an obvious solution, I am surprised I didn't understand it before. It's also quite secure.

Dumb AP is the ideal Wireless AP.[10]

[7] Good design again.

[8] Another thing that must go right.

[9] WPA issues, or side channel attacks probably. However, unlikely if it's not accessible via LAN

[10] Technically, you probably still want to update them every so often, given that RF and Encryption (e.g. WPA) adds complexity to the AP, but a Dumb AP is the way Wireless Access Points should be done, if possible. Avoid configuration overhead, and have something that does what its supposed to, and nothing else. Minimize the attach surface.

1.9 What is required for this to go right

A lot is required for everything to go smoothly. I've touched on I hope most of the points, but experience flashing, soldering, soldering with hot air, proto board usage, tearing down enclosures, configuring Openwrt, using USB to Serial adapters, installing the WAP on the wall with a screw driver, and some drywall screws[11], and finally understanding this WAP was worth the trouble by reviewing the screenshots and having enough electronics experience to recognize quality when you see it.

Not least of all, you have to decipher all the docs as I did to come to this conclusion that flashing would be viable. It is here. It may not be next time. However, with this tool under my belt I'm better prepared for next time.[12]

[11] This WAP also comes with separate magnetic bars that screw into the wall. The AP attaches via magnets to the bars, and is easily removable for service. Great design.

[12] A note about the current support of this device in Openwrt. I've decided to not make a wiki page, as for whatever reason no one else did, and "When in Rome, do as the Romans do". I don't think there are a lot of devices out there, so will avoid promoting it. As they'd follow the fate of other devices which have had internet popularity make them unobtainable (In video game world, the classic example is the SNES game Earthbound). However, I felt an article on my unknown git/book wouldn't hurt. When you browse here, you get such articles.