# TalosII firmware update

Eric Grosse

2021-03-14

If you're the owner of a Raptor Engineering TalosII computer you have demonstrated willingness to expend time and money on security and open design. So you may want, as I did, to switch your machine to OpenBSD.

My 2018-era machine was still running factory firmware. Although good enough for installing Ubuntu, with that firmware my OpenBSD install froze at Petitboot kexec. A query to the `openbsd-ppc` mailing list got a prompt reply 2021-03-07 from Theo de Raadt that my firmware was too old, and I set out to update following the `wiki.raptorcs.com` advice. There were enough hurdles that I thought it might help you if I showed my notes.

Recall from your TalosII user manual and schematic that on the mainboard near the PCI connectors is a socketed BMC flash part, for booting the ASPEED 2500 when "standby power" is applied by connecting the chassis power supply to a wall outlet. (No power switch exists—this is always on and always listening on the network, so as will be emphasized below you *must* reset the root password in disconnected mode if you update the firmware.) There is also an adjacent socket for the PNOR BOOT flash part, which holds the multi-component boot sequence for the POWER9 cpus when "main power" is applied by pushing the front panel power button. As recommended by Raptor's wiki, leave the third flash part, for FPGA, untouched.

There are enough timing delays and things that can go wrong that I recommend having a serial console to watch what is happening, and occasionally to issue configuration commands. Here and below, I'm not saying this is the only way to do it, just a way that works and gives you a good safety margin against bricking your system.

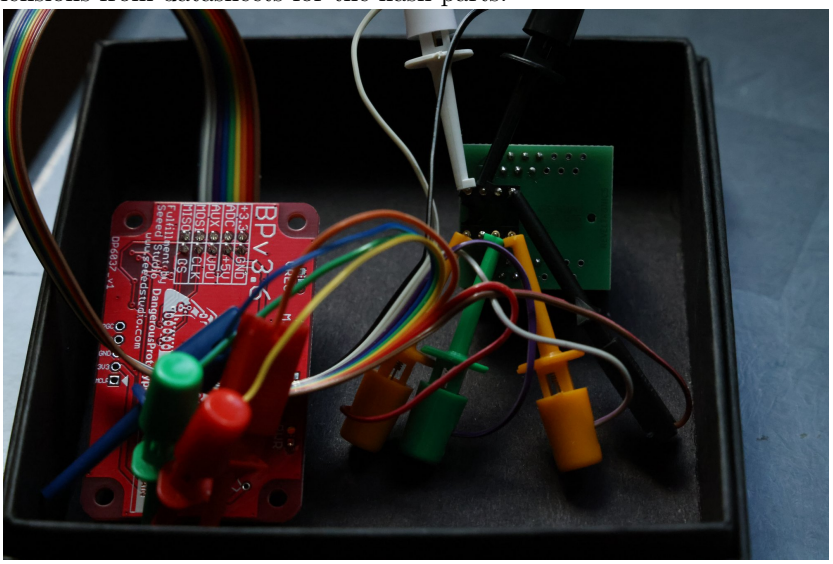
For the BMC console, add a DB9 extension card with ribbon cable connected to internal header J7701, illustrated on page 24 of the user manual. You need a cable from the DB9 to an auxiliary machine; I used a Prolific Technologies USB adapter. Finally, you need software such as `minicom` on Linux or `TeraTerm` on Windows, set to speed 115200. On a working system, before starting any updates, learn what to expect from the BMC boot by watching this console. About two minutes after connecting to wall power, you should see a Linux login prompt from the ASPEED. Type your personally set root password and note your current working network configuration with `fw_printenv` and `ifconfig`. You will need the MAC addresses; if you forget to record them now, you can get them later from a label on the mainboard.

Next move your serial cable over to the DB9 on the back corner of the mainboard, which lets you observe bringing up the Power CPUs and RAM and such. When you turn on main power, you'll see messages from OPAL Skiboot, which loads Skiroot, which loads Petitboot, which loads whatever operating system is installed on the main storage device. Even without updating firmware, you'll see a Petitboot option for the OpenBSD install if you've connected a USB install stick. But that's where you'll get stuck because of an ELF bug in older firmware. We're going to fix that now.

Looking at TalosII schematic, page 75, you see that we want a 32MB Macronix MX25L25635FMI flash part for the "BMC Firmware ROM", and a 64MB Micron MT25QL512 flash part for the "Boot Flash ROM". I suggest getting a few from Digikey or wherever. You can in principle overwrite the existing working parts directly on the running TalosII, but I prefer to write externally onto new parts and hold the old ones in reserve. (Similarly, I choose to install OpenBSD on a fresh, larger SSD and save the old Ubuntu one as a backup. But if you're up to writing flash parts, you're already an old pro at operating system install and can make up your own mind.)

Since this was my first time for a flash programmer at home, I also purchased a Bus Pirate v3.6 and cable from Adafruit. There are plenty of other choices and plenty of tutorial videos and bewildering comments online about whether the Bus Pirate firmware should or should *not* be updated. I left mine at the delivered Bus Pirate firmware 6.1, accepting the limitation to 2MHz SPI clock.

Get a socket for holding the new flash part during programming. Mine is OTS-28-1.27-04 SOIC16 but lost the order paperwork. I recall that there are many different sizes, so be sure what you order matches the physical pin dimensions from datasheets for the flash parts.




Here is a photo of my setup, with GND brown wire / black clip, +3.3 red / orange, CLK purple / green, MOSI gray / orange, MISO black / black, CS

white / white.

Source for the software used to talk to this is available from `flashrom.org` and after compiling, test with

```
# flashrom -p buspirate_spi:dev=/dev/ttyUSB0
```

to probe the chip and confirm it is the model you expect, such as

```
Found Macronix chip "MX25L25635F/MX25L25645G" (32768 kB, SPI)
```

I ran into a problem for the PNOR part:

```
Found Micron/Numonyx/ST flash chip "N25Q512..3G" (65536 kB, SPI)
Found Micron flash chip "MT25QL512" (65536 kB, SPI)
Multiple definitions match chip(s): "N25Q512..3G", "MT25QL512"
specify which definition to use with the -c <chipname> option.
```

but flashrom with the chipname option refused read or write commands. I worked around this problem by commenting out the section for N25Q512..3G in `flashrom-v1.2/flashchips.c` and recompiling.

As a sanity check and possible recovery path, I read the contents of the flash parts from the factory with command

```
flashrom -p buspirate_spi:dev=/dev/ttyUSB0 --read factory.pnor
```

which took 25 minutes at the limited Bus Pirate clock speed.

From `wiki.raptorcs.com/wiki/Talos_II/Firmware` fetch the new firmware. There is a pointer there to instructions for installing in place, but we push onward using the `flashrom write` method. I did fetch the source; OPAL built easily but OpenBMC had issues I confess I did not take time to work past. (This was before reading the forum post "holy brittle build toolchain, batman!" with a bitbake fix.) It is good to have the source but, together with all the dependencies, there is way more than I'm going to review in detail myself right now. So I accepted the pre-compiled binaries from Raptor. After extracting from the tar, run

```
flashrom -p buspirate_spi:dev=/dev/ttyUSB0 --write talos-ii-v2.00-image-bmc
```

wait patiently until the verification completes, then swap chips and

```
flashrom -p buspirate_spi:dev=/dev/ttyUSB0 --write talos-ii-v2.00-image-pnor
```

Unclip the factory chips from their sockets on the mainboard with help from a jewler's screwdriver, instal the new chips, restore serial console to the BMC, *make sure the network cable is unplugged,* and finally reconnect to wall power. There may be various error messages along the way, including automatic fixing of flash problems; don't panic, just let it proceed. Eventually you should see a BMC login prompt. All customization, including root password and system MAC addresses has been lost, so login with the well-known static factory password given in the user manual and reset to your own secret `# passwd`. Restore the MAC addresses with commands like

```
fw_setenv ethaddr 2c:09:4d:xx:xx:xx
fw_setenv eth1addr 2c:09:4d:xx:xx:xx
```

and reboot. At this point you should be able to safely reconnect the cable to the network port in the back, next to the mainboard DB9. Log back into the BMC, and `ifconfig` to restore network configuration as needed. Note that this is for the BMC ssh console, which (as you can read about in wikipedia) uses NC-SI to share the network cable with the main operating system.

Now move your serial console over to the mainboard DB9. Insert the OpenBSD install USB stick containing miniroot, and push the main TalosII power button a couple minutes after standby power has been on so that the BMC had time to finish startup. You should see the usual skiboot messages and eventually the Petitboot menu. Let it boot "OpenBSD install" and proceed as with any other OpenBSD install.

Congratulations, you're done!