

● Mijn ervaring met de Raspberry Pi 5 ●

Johan Swenker

De eerste testjes

NVMe-SSD

Na de fysieke installatie van de Raspberry Pi in de behuizing van Argon ben ik ermee gaan stoeien. Als eerste de NVMe-SSD: doet die het en hoe snel is die? Dat bleek gelukkig mee te vallen. Op internet zijn lijsten te vinden van ondersteunde SSD's, meestal getest door de leverancier van de PCIe-naar NVMe-converter. Ik heb een kleine, maar verder willekeurige SSD gekocht. Het geheel werkt, zonder een bijzondere NVMe- of PCIe-optie in de config-file op te nemen. Het PCIe-interface van de Raspberry Pi 5 voldoet formeel aan de normen voor de tweede generatie PCIe. Met een enkele lane geeft dat een theoretische snelheid van 500 Mbyte/sec. Eenvoudige metingen ondersteunen dit: schrijven 400 Mbyte/sec; lezen 450 Mbyte/sec. Maar het interface kan ook PCIe-Gen 3.0 aan, maar dat moet wel heel specifiek aangezet worden met de optie `dtparam=pciex1_gen=3` in de config-file. Dat zou tweemaal zo snel moeten zijn. Een simpel testje dat veel data leest en schrijft, bevestigt dat de snelheid bijna verdubbeld is.

Temperatuur

Het pakket *smartmontools* heb ik geïnstalleerd om met het programma *smartctl* de temperatuur van de NVMe-SSD op te vragen. Tijdens het schrijven van dit artikel geeft een van de temperatuursensoren een temperatuur van 47 graden aan. De onderkant van de behuizing, daar waar de SSD zit, is aangenaam warm. Je kunt je handen eraan opwarmen, als aan een warme kop chocolademelk. In mijn laptop zit een ouderwetse (?) Sata-SSD. Deze NVMe-SSD in de Raspberry is voor mij voor het eerst dat ik met een NVMe-SSD kan experimenteren. Er is een commando *nvme*, waarmee je informatie over de NVMe-SSD kunt opvragen; wel eerst *nvme-cli* installeren. Het meest interessant vond ik de subcommando's *list*, *show-topology* en *id-ns*. Nee, ik heb niet gedurfd om een namespace te verwijderen, en twee nieuwe te creëren. Dat laatste had ook niet gewerkt, NVMe-controllers voor consumenten kunnen slechts één namespace aan. Ook de processor van de Raspberry Pi kent een temperatuursensor. Die is uit te lezen met `vcgencmd measure_temp`, en geeft op dit moment `temp=49.4'C` aan. Met `vcgencmd` zijn nog meer interessante parameters van de Raspberry Pi op te vragen.

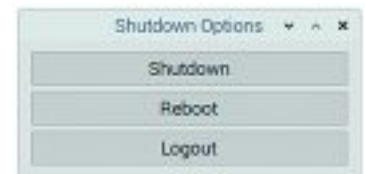
- `vcgencmd commands` geeft alle commando's weer;
- `vcgencmd bootloader_config` geeft de eeprom-configuratie weer. Mijn Raspberry geeft als uitvoer:

```
johan@rp... -nvme: ~ ▾ ▹ ▸ ×
File Edit Tabs Help
johan@RPi5-nvme:~ $ vcgencmd boot
loader_config
[all]
BOOT_UART=1
POWER_OFF_ON_HALT=1
BOOT_ORDER=0xf164
PCIE_PROBE=1
johan@RPi5-nvme:~ $
```

Boot-parameters

Als je shutdown geeft op een Raspberry Pi 4 of ouder, dan blijft die tamelijk veel stroom gebruiken. Bij de Raspberry Pi 5 kun je kiezen voor de oude situatie: de 3,3 Volt blijft actief, de processor blijft actief, en een eventuele HAT blijft gewoon doorwerken. Je kunt er ook voor kiezen dat de processor helemaal spanningsloos wordt. N.B.: Oudere HAT's kunnen in de war raken als ze wel 5 Volt krijgen, maar geen 3,3 Volt. Het verschil in stroomverbruik is groot: 0,3 A (dus 1,5 Watt) in de oude situatie, tegen minder dan 0,01 A met `'POWER_OFF_ON_HALT=1'`.

Met een simpele druk op de aan/uit-schakelaar, die nieuw is op de Raspberry Pi 5, kun je de Raspberry Pi weer tot leven wekken. Als je deze schakelaar indrukt, terwijl de Raspberry draait, dan krijg je het standaard menuutje met de shutdown opties. En inderdaad, als je de schakelaar te lang ingedrukt houdt, dan gaat de stroom uit, alsof je de stekker uit het stopcontact gehaald hebt. Dit moet je niet te vaak doen. Het file-systeem vindt het niet leuk, de SSD vindt het niet leuk, en de SD-kaartjes vinden het al helemaal niet leuk. Er gaat vast niets kapot, tot je het op een verkeerd moment doet.



De bootvolgorde 0xf164 is een beetje cryptisch. Je moet dit van achteren naar voren lezen: boot eerst volgens methode 4; als dat niet lukt, boot dan volgens methode 6; als dat niet lukt, probeer dan methode 1; en als dat ook niet lukt, begin dan opnieuw.

Nummer	Device
1	Boot van SD-kaart
4	Boot van USB disk
6	Boot van NVMe
f	Opnieuw van achteraf aan

https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#BOOT_ORDER geeft informatie over de andere nummers.

Het aanpassen van de bootloader-configuratie gaat met `sudo rpi-eeprom-config --edit`

Ik heb er voor gekozen om altijd eerst te proberen van een USB-disk te booten. Als er iets mis is gegaan met de SSD of de SD, dan kun je de Raspberry zonder schroeven-draaiër weer aan de praat krijgen. Je moet dan wel een USB-disk klaar hebben liggen met een werkend Raspberry Pi OS. Bij mijn eerste experimenten stond op mijn USB-SSD een oud Raspberry Pi OS, dat wel geschikt was voor de Pi4, maar niet voor de Pi5. Ik heb de NVMe-SSD even los moeten maken toen ik een niet-werkende kapotte configuratie had.

Config-file

De eerste partitie van de bootdisk werd altijd gemount als `/boot`. De nieuwe versie van Raspberry Pi OS mount de eerste partitie als `/boot/firmware`. Als je het eenmaal weet, dan is er niets aan de hand, je past `/boot/firmware/config.txt`

aan en alles werkt zoals vroeger. Als je het niet weet, dan heb je een probleem: de handleidingen die je op internet kunt vinden kloppen geen van alle meer.

Bij een van mijn experimenten stopte de SSD met werken, ik kan het nu uiteraard niet meer reproduceren. Ik weet ook niet waarom het nu wel werkt. De beschrijvingen op internet geven aan dat `pci_probe=1` in de eeprom moet staan, en dat `dtparam=nvme` in de config-file moet staan. Ik heb ze net verwijderd en na de reboot werkt alles nog steeds.

Ik heb twee aanpassingen gemaakt aan de config-file. Zoals al eerder beschreven heb ik de regel `dtparam=pciex1_gen=3` toegevoegd. Hierdoor wordt de SSD sneller. Zie ook: <https://community.volumio.com/t/prepare-raspberry-pi-for-boot-from-usb-nvme/65700> (step 5 Use NVMe)

De Raspberry Pi 5 heeft een Real Time Clock. Als de Raspberry Pi uit staat, dan krijgt deze RTC stroom van een kleine accu, die je apart moet kopen. Maar dan moet de accu wel af en toe opgeladen worden en dat moet je expliciet aanzetten in de config-file. De documentatie bij raspberrypi.com <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#enable-battery-charging> zegt dat `dtparam=rtc_bbat_vchg=3000000` nodig is, maar ze vertellen niet waarom.

De RTC doet z'n werk. Dat is het makkelijkst te controleren door de kernel-log te bekijken met het `dmesg`-commando:

```
john@pi5-nvme ~
File Edit Tabs Help
1.394887 mousedev: PS/2 mouse device common for all mice
1.403881 rpi-rtc soc:rpi-rtc trickle charging enabled at 3000000v
1.415358 rpi-rtc soc:rpi-rtc registered as rtc0
1.421781 rpi-rtc soc:rpi-rtc setting system clock to 2024-04-17T18:53:53 UTC (171338933)
```

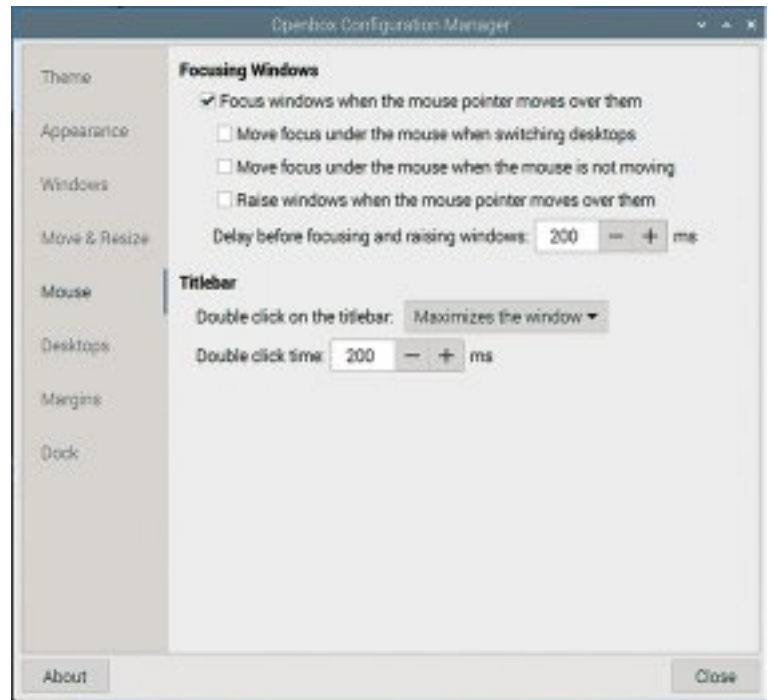
SD kopiëren naar SSD

De installatie op het SD-kaartje beviel me wel. Die wilde ik dus graag naar de NVMe-SSD kopiëren. Er zijn wel wat beschrijvingen te vinden hoe dat zou moeten. Het lukte mij niet. Dan maar doen wat er staat: `apt update && apt upgrade`. Dat ging op zich snel genoeg. De Raspberry is snel, en ook het SD-kaartje is voldoende snel. Het ging pas mis toen het pakket `wolfram-engine` ge-upgradet werd. Uitgepakt is dit pakket 4 GB. Zoveel ruimte was er niet meer vrij op het SD-kaartje. Nu is dat niet zo erg, gewoon dat pakket de-installeren, maar leuk is anders. Maar ik had nog steeds geen werkende Raspberry Pi OS op m'n NVMe-SSD staan. Ook allerhande andere trucs werkten niet.

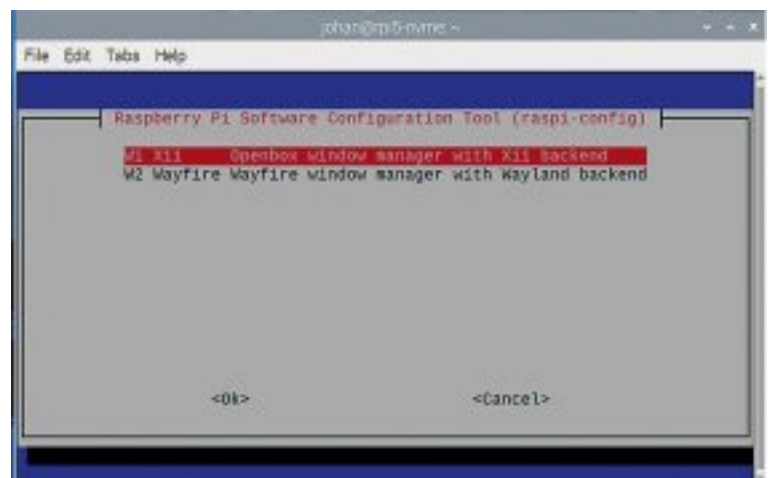
Zoals in het andere artikel staat, heb ik de imager gebruikt om het Raspberry Pi OS op de SSD te zetten. De aanbevolen versie is de 64-bit OS, van 1,1 GB groot. Die beviel me niet: ik had geen werkend geluid, en mijn favoriete performance-monitor `xosview` deed het ook niet. Bij de Full-versie van 2.7 GB had ik die problemen niet.

focus follows mouse

Onder Linux gebruik ik al sinds jaar en dag 'focus follows mouse'. Als ik de muis naar een ander window verplaatst, dan wordt dat window meteen het actieve window. Ik kan er meteen in gaan typen, zelfs als het deels bedekt is door andere windows. Precies zoals alle losse papiertjes op mijn echte bureau werken. Veel mensen zijn gewend aan *click to focus*. Dan moet je eerst in het nieuwe window op de muis klikken, voordat je daarin kunt werken. Bovendien komt dat nieuwe window dan meteen naar voren.



Focus follows mouse kun je met het commando `obconf` instellen, ..., behalve dan dat het niet werkte. Logisch, ik gebruikte Openbox niet als windowmanager, maar wat dan wel? De windowmanager van Wayland is Wayfire. Ik heb niet gevonden hoe je bij Wayfire *focus follows mouse* kunt instellen. Ik heb wel gevonden hoe je het moderne Wayland kunt vervangen door het oudewetse X11 ([raspi-config](#) → [advanced](#) → [wayland](#) → [X11](#)).



Bovendien zijn er nu ook twee andere probleempjes opgelost: ik heb weer een visuele feedback (een zandloper) als ik bijvoorbeeld Firefox van de menubalk opstart; en als ik twee of meer terminalwindows open komen ze naast elkaar en niet bovenop elkaar. Blijkbaar ben ik nog niet klaar voor Wayland.

