

Linux og Realtek RTL9210 USB til NVMe-bro

Resumé:

- **Symptomer:** Gentagne USB-nulstillinger, I/O-fejl eller diske, der forsvinder under Linux.
- **Berørte:** Realtek RTL9210 (bekræftet) og RTL9220 (muligvis).
- **Årsag:** Tilbagefald til intern ROM (**f0.01**) efter fejl i tjeksum.
- **Påvirkning:** Permanent ustabilitet, ingen Linux-værktøjer til genindlæsning tilgængelige.
- **Løsning:** Kun OEM Windows-værktøjer kan gendanne firmwaren – Realtek blokerer open source-alternativer.

Forord

I år 2025 burde det være helt rimeligt at starte en Raspberry Pi fra en SSD forbundet via USB. Men takket være finurlighederne i Realteks firmware er dette rimelige mål blevet et eventyr. Efter måneder med uforklarlig ustabilitet – tilfældige nulstillinger, forsvindende diske, korrupte filesystemer – udtømte forfatteren alle almindelige rettelser: nye kabler, strømforsynede hubs, kerneopdateringer, USB-justeringer og firmware-finjustering. Gennembruddet kom først, da ChatGPT svarede på et mærkeligt spørgsmål sent om natten: „Er det muligt, at USB-til-NVMe-broen er vendt tilbage til en gammel firmware?“

Introduktion

Hvis din Realtek-baserede NVMe-kabinet pludselig bliver ustabil efter ugers fejlfri drift – gentagne USB-nulstillinger, I/O-fejl eller forsvindende diske – er du ikke alene. Mønsteret er opstået på tværs af flere mærker, fra ukendte enheder til velkendte OEM'er som Sabrent og Orico. Fællesnævneren: **Realteks RTL9210 og muligvis RTL9220** USB-til-NVMe-bro-chips.

I starten fungerer alt. Men tilsyneladende uden årsag begynder enheden at afbryde under belastning eller ved længerevarende brug, især på Linux- eller Raspberry Pi-systemer. Den sande årsag er ikke SSD'en eller strømforsyningen – det er selve firmwaren, der stille vender tilbage til sin **ROM-indlejrede reservekode**, en version, som Realtek stadig leverer internt som **f0.01**.

Den skjulte mekanisme – Firmware-tilbagefald ved design

Realteks bro-chips gemmer deres driftsfirmware og konfigurationsdata i en ekstern SPI-flash. Ved opstart tjekker controlleren en simpel tjeksum. Hvis tjeksummen ikke stemmer, nægter den at indlæse den eksterne firmware og starter i stedet fra sin interne ROM.

Denne reserve-firmware er gammel og defekt. Den mangler flere USB-stabilitetsrettelser og forbedringer i link-tilstandsstyring, der findes i senere revisioner, hvilket fører til den klassiske sekvens, som enhver Linux-bruger genkender:

```
usb 3-2: nulstil højhastigheds-USB-enhed nummer 2 ved hjælp af xhci-hcd
usb 3-2: enhedsbeskrivelse læs/64, fejl -71
EXT4-fs advarsel (enhed sda2): I/O-fejl ved skrivning til inode ...
```

Tjeksummen kan blive ugyldig, når konfigurationsdata omskrives – for eksempel når broen opdaterer sine strømstyrings- eller UAS-indstillinger – og enheden mister strøm midt i skrivningen. Næste opstart registrerer en korrupt tjeksum og falder permanent tilbage til ROM-firmwaren.

På det tidspunkt opfører dit „højtydende NVMe-kabinet“ sig præcis som det billigste ukendte kabinet, fordi det internt nu kører den samme fejlbehæftede basiskode, der er brændt ind i silicium.

Verificering af problemet

Du kan nemt bekræfte denne tilstand under Linux:

```
lsusb -v | grep -A2 Realtek
```

En sund Realtek-bro rapporterer en firmware-revision (**bcdDevice**) over 1.00. En tilbagefaldet viser:

```
bcdDevice f0.01
```

Denne **f0.01**-signatur betyder, at controlleren starter fra ROM – og ingen mængde af frakobling, omformatering eller kernejustering vil løse det.

Denne tilbagefaldsmekanisme er **bekræftet på RTL9210**. **RTL9220** ser ud til at dele den samme designarkitektur og firmware-layout, så den kan udvise identisk adfærd, men dette er **sandsynligt snarere end bevist**.

Hvorfor du ikke kan reparere det selv

I princippet er løsningen enkel: genindlæs den korrekte firmware til SPI. I praksis gør Realtek dette umuligt.

Virksomheden leverer en lukket kildekode-opdatering til Windows til OEM'er og integratorer. Linux-brugere tilbydes intet. Fællesskabsudviklere reverse-engineerede kompatible flash-værktøjer (**rtsupdater**, **rtl9210fw**, **rtsupdater-cli**), som tillod fuld firmware-gendan-

nelse fra Linux-systemer – indtil Realtek udstedte **DMCA-nedtagelsesmeddelelser** for at undertrykke dem.

Der er ingen plausibel begrundelse for intellektuel ejendom til at blokere sådanne værktøjer: de afslører ikke mikrokode, kun orkestrerer opdateringssekvensen over USB. Realteks nedtagelser handlede ikke om beskyttelse. De var ideologiske.

Prisen for en ideologi

Dette handler ikke om open source-idealisme. Det handler om en hardwareleverandørs **ideologiske fjendtlighed over for åbne systemer**, der ødelægger enheder, der markedsføres som *Linux-kompatible*.

Realteks modstand mod dokumentation og åbne værktøjer har været i to årtier, spændende over Wi-Fi, Ethernet, lyd og nu lagringscontrollere. Denne isolationisme kan gå ubemærket hen i en verden kun med Windows, men bliver giftig, når de samme chips integreres i produkter til flere platforme som **Sabrent EC-SNVE**, der åbent viser Linux-logoet på sin emballage.

Ved at forbyde Linux-flashværktøjer og blokere fællesskabsvedligeholdelse har Realtek effektivt **kriminaliseret selv-reparation**. Konsekvenserne spreder sig udad:

- Linux-brugere ser „understøttet“ hardware forfalde til ustabilitet.
- OEM'er som Sabrent og Orico står over for unødvendige RMA- og garantinomkostninger.
- Realteks mangeårige ry for dårlig Linux-kompatibilitet forstærkes endnu en gang.

I sidste ende er det ikke open source, der ødelægger Realteks enheder – det er **Realteks fjendtlighed over for open source**, der ødelægger dem.

En rationel vej frem

Løsningen kræver ingen ideologisk ændring, kun pragmatisme. Realtek kunne:

1. Udgive en leverandørunderskrevet kommandolinje-opdatering til Linux (ingen kildekode-åbenhed nødvendig).
2. Offentliggøre tjeksum-algoritmen, så integratorer sikkert kan validere flash-billeder.
3. Udrulle en DFU-lignende tilstand, der accepterer opdateringer over USB Mass Storage, uafhængigt af OS.

Hver af disse ville forhindre garantinomkostninger, beskytte OEM-relationer og genoprette tilliden til Realteks bro-chips blandt professionelle Linux-brugere – fra arbejdsstationsbyggere til Raspberry Pi-udviklere.

Hvad du kan gøre

Hvis du mistænker, at dit kabinet er vendt tilbage til ROM-firmware:

- Tjek med **lsusb -v | grep bcdDevice**.
- Hvis det viser **f0.01**, rapporter problemet til din OEM.
- Inkludér uddraget fra **dmesg** og peg på denne dokumenterede tilbagefaldsmekanisme.
- Bed din leverandør om at eskalere problemet til Realtek og nævn behovet for en Linux-kompatibel opdatering.

Realteks firmware-politik generer ikke kun entusiaster; den skaber håndgribelige økonomiske tab for deres eget økosystem. Jo hurtigere denne virkelighed anerkendes i virksomheden, jo hurtigere kan både Linux-brugere og OEM-partnere stoppe med at spille tid på unødvendige RMA-cykluser.

Producenternes svar

Både Realtek og Sabrent blev inviteret til at give udtalelser om tilbagefaldsproblemet med firmwaren beskrevet ovenfor. Deres svar – hvis de modtages – vil blive tilføjet her.

Tillæg – Identifikation af berørte enheder

Control-ler	Leveran-dør-ID	Produkt-ID	Noter	Status
RTL9210	0x0bda	0x9210	USB 3.1 Gen 2 10 Gb/s bro	Bekræftet tilbagefaldsadfærd
RTL9220	0x0bda	0x9220	USB 3.2 Gen 2×2 20 Gb/s bro	Mulig , lignende arkitektur

Firmware-tilbagefaldssignatur: **bcdDevice f0.01**

Kendte stabile revisioner: **1.23 – 1.31**