https://farid.ps/articles/rtl9210_usb_to_nvme_bridge/sv.html

Linux och Realtek RTL9210 USB-till-NVMe-brygga

Sammanfattning:
• **Symptom:** Upprepade USB-återställningar, I/O-fel eller diskar som försvinner i Linux.
• **Drabbade:** Realtek RTL9210 (bekräftat) och RTL9220 (möjligen).
• **Orsak:** Återgång till intern ROM (**f0.01**) efter misslyckad kontrollsumma.
• **Påverkan:** Permanent instabilitet, inga omprogrammeringsverktyg tillgängliga för Linux.
• **Lösning:** Endast OEM Windows-verktyg kan återställa firmware – Realtek blockerar open source-alternativ.

Förord

År 2025 borde det vara fullt rimligt att starta en Raspberry Pi från en SSD ansluten via USB. Men på grund av Realteks firmware-egenskaper har detta rimliga mål förvandlats till ett äventyr. Efter månader av oförklarlig instabilitet – slumpmässiga återställningar, försvunna diskar, korrupta filsystem – uttömde författaren alla vanliga lösningar: nya kablar, strömförsörjda hubbar, kärnuppdateringar, USB-justeringar och firmware-finjustering. Genombrottet kom först när ChatGPT svarade på en konstig fråga sent på natten: ”Är det möjligt att USB-till-NVMe-bryggan har återgått till en gammal firmware?”

Inledning

Om din Realtek-baserade NVMe-kapsling plötsligt blir instabil efter veckor av felfri drift – upprepade USB-återställningar, I/O-fel eller diskar som försvinner – är du inte ensam. Detta mönster har dykt upp hos flera märken, från namnlösa enheter till välkända OEM-tillverkare som Sabrent och Orico. Den gemensamma nämnaren: **Realteks RTL9210 och möjligen RTL9220** USB-till-NVMe-bryggchip.

I början fungerar allt. Sedan, till synes utan anledning, börjar enheten koppla från under belastning eller vid långvarig användning, särskilt på Linux- eller Raspberry Pi-system. Den verkliga orsaken är varken SSD:n eller strömförsörjningen – det är själva firmware-kontrollern som tyst återgår till sin **inbyggda backup-kod i ROM**, en version som Realtek fortfarande levererar internt som **f0.01**.

Den dolda mekanismen – Firmware-återgång genom design

Realteks bryggchip lagrar sin operativa firmware och konfigurationsdata i en extern SPI-flash. Vid uppstart kontrollerar kontrollern en enkel kontrollsumma. Om denna kontrollsumma inte stämmer, vägrar den att ladda den externa firmwaren och startar istället från sin interna ROM.

Denna backup-firmware är gammal och defekt. Den saknar flera USB-stabilitetsfixar och förbättringar i länkhantering som finns i senare revisioner, vilket leder till den klassiska sekvensen som varje Linux-användare känner igen:

> usb 3-2: återställ höghastighets-USB-enhet nummer 2 med xhci-hcd
> usb 3-2: läsning av enhetsbeskrivning/64, fel -71
> EXT4-fs varning (enhet sda2): I/O-fel vid skrivning till inode ...

Kontrollsumman kan bli ogiltig när konfigurationsdata skrivs om – till exempel när bryggan uppdaterar sina energihantering eller UAS-inställningar – och enheten förlorar ström under skrivningen. Nästa uppstart upptäcker en korrupt kontrollsumma och återgår permanent till ROM-firmwaren.

Vid denna punkt beter sig din ”högpresterande NVMe-kapsling” exakt som den billigaste namnlösa skalet, eftersom den internt nu kör samma defekta baskod som är inbränd i kisel.

Verifiering av problemet

Du kan enkelt bekräfta detta tillstånd i Linux:

> lsusb -v | grep -A2 Realtek

En frisk Realtek-brygga rapporterar en firmware-revision (**bcdDevice**) över 1.00. En som har återgått visar:

> bcdDevice f0.01

Denna **f0.01**-signatur innebär att kontrollern startar från ROM – och ingen mängd avkoppling, omformatering eller kärnjusteringar kommer att fixa det.

Denna återgångsmekanism har **bekräftats för RTL9210**. **RTL9220** verkar dela samma designarkitektur och firmware-layout, så den kan uppvisa identiskt beteende, men detta förblir **sannolikt snarare än bevisat**.

Varför du inte kan fixa det själv

I princip är lösningen trivial: omprogrammera rätt firmware till SPI. I praktiken gör Realtek detta omöjligt.

Företaget tillhandahåller en sluten källa-uppdaterare för Windows till OEM och integratörer. Linux-användare erbjuds ingenting. Gemenskapsutvecklare omvände kompatibla flash-verktyg (**rtsupdater**, **rtl9210fw**, **rtsupdater-cli**) som möjliggjorde fullständig firm-

ware-återställning från Linux-system – tills Realtek utfärdade **DMCA-borttagningsmeddelanden** för att undertrycka dem.

Det finns ingen trovärdig immateriell rättighetsmotivering för att blockera sådana verktyg: de exponerar inte mikrokod, utan orkestrerar bara uppdateringssekvensen via USB. Realteks borttagningar handlade inte om skydd. De var ideologiska.

Kostnaden för en ideologi

Detta handlar inte om open source-idealism. Det handlar om **en hårdvarutillverkares ideologiska fientlighet mot öppna system** som förstör enheter som marknadsförs som *Linux-kompatibla*.

Realteks motstånd mot dokumentation och öppna verktyg har pågått i två decennier, omfattande Wi-Fi, Ethernet, ljud och nu lagringskontrollers. Denna isolering kan gå obemärkt förbi i en värld med enbart Windows, men blir giftig när samma chip integreras i multiplattformsprodukter som **Sabrent EC-SNVE**, som öppet visar Linux-logotypen på sin förpackning.

Genom att förbjuda Linux-flash-verktyg och blockera gemenskapsunderhåll har Realtek effektivt **kriminaliserat självreparation**. Konsekvenserna sprider sig utåt:

- Linux-användare ser ”stött” hårdvara försämras till instabilitet.
- OEM-tillverkare som Sabrent och Orico står inför onödiga RMA- och garantikostnader.
- Realteks långvariga rykte för dålig Linux-kompatibilitet förstärks återigen.

I slutändan är det inte open source som förstör Realteks enheter – det är **Realteks fientlighet mot open source** som förstör dem.

En rationell väg framåt

Lösningen kräver ingen ideologisk förändring, bara pragmatism. Realtek skulle kunna:

1. Släppa ett leverantörsignerat kommandorads-uppdateringsverktyg för Linux (ingen källkodspublicering krävs).
2. Publicera kontrollsummealgoritmen så att integratörer kan verifiera flash-bilder säkert.
3. Anta ett DFU-liknande läge som accepterar uppdateringar via USB-masslagring, oberoende av operativsystem.

Vart och ett av dessa skulle förhindra garantikostnader, skydda OEM-relationer och återställa förtroendet för Realteks bryggchip bland professionella Linux-användare – från arbetsstationsbyggare till Raspberry Pi-utvecklare.

Vad du kan göra

Om du misstänker att din kapsling har återgått till ROM-firmware:

- Kontrollera med **lsusb -v | grep bcdDevice**.
- Om det visar **f0.01**, rapportera problemet till din OEM.
- Inkludera utdraget från **dmesg** och peka på denna dokumenterade återgångsmekanism.
- Be din leverantör att eskalera problemet till Realtek, med hänvisning till behovet av ett Linux-kompatibelt uppdateringsverktyg.

Realteks firmware-policy stör inte bara entusiaster; den skapar konkreta ekonomiska förluster för deras eget ekosystem. Ju tidigare denna verklighet erkänns inom företaget, desto snabbare kan Linux-användare och OEM-partners sluta slösa tid på undvikbara RMA-cykler.

Svar från tillverkare

Både Realtek och Sabrent har bjudits in att ge uttalanden om det firmware-återgångsproblem som beskrivs ovan. Deras svar – om de mottas – kommer att läggas till här.

Bilaga – Identifiering av drabbade enheter

Kontroller	Leverantör-ID	Produkt-ID	Anmärkningar	Status
RTL9210	0x0bda	0x9210	USB 3.1 Gen 2 10 Gb/s brygga	**Bekräftat** återgångsbeteende
RTL9220	0x0bda	0x9220	USB 3.2 Gen 2×2 20 Gb/s brygga	**Möjligt**, liknande arkitektur

Firmware-återgångssignatur: **bcdDevice f0.01**
Kända stabila revisioner: **1.23** – **1.31**