

[https://farid.ps/articles/rtl9210\\_usb\\_to\\_nvme\\_bridge/id.html](https://farid.ps/articles/rtl9210_usb_to_nvme_bridge/id.html)

# Linux dan Jembatan USB ke NVMe Realtek RTL9210

## Ringkasan:

- **Gejala:** Reset USB berulang, kesalahan I/O, atau drive menghilang di bawah Linux.
- **Terkena Dampak:** Realtek RTL9210 (dikonfirmasi) dan RTL9220 (kemungkinan).
- **Penyebab:** Kembali ke ROM internal (**f0.01**) setelah kegagalan checksum.
- **Dampak:** Ketidakstabilan permanen, tidak ada alat reflash untuk Linux yang tersedia.
- **Perbaikan:** Hanya utilitas Windows OEM yang dapat memulihkan firmware - Realtek memblokir alternatif open-source.

## Pendahuluan

Pada tahun 2025, seharusnya sangat wajar untuk mem-boot Raspberry Pi dari SSD yang terhubung melalui USB. Namun, karena keanehan firmware Realtek, tujuan yang wajar ini telah menjadi sebuah petualangan. Setelah berbulan-bulan ketidakstabilan yang tidak dapat dijelaskan - reset acak, drive yang menghilang, sistem file yang rusak - penulis telah mencoba semua perbaikan biasa: kabel baru, hub bertenaga, pembaruan kernel, penyesuaian USB, dan penyetelan firmware. Terobosan hanya datang ketika ChatGPT menjawab pertanyaan aneh di tengah malam: "Apakah mungkin jembatan USB ke NVMe kembali ke firmware lama?"

## Pengantar

Jika enclosure NVMe berbasis Realtek Anda tiba-tiba menjadi tidak stabil setelah berminggu-minggu beroperasi tanpa cela - reset USB berulang, kesalahan I/O, atau drive yang menghilang - Anda tidak sendirian. Pola ini telah muncul di beberapa merek, dari unit tanpa nama hingga OEM terkenal seperti Sabrent dan Orico. Penyebut umum: **chip jembatan USB ke NVMe Realtek RTL9210 dan mungkin RTL9220.**

Pada awalnya, semuanya berfungsi. Kemudian, seolah tanpa sebab, perangkat mulai terputus saat berada di bawah beban atau selama penggunaan yang lama, terutama pada sistem Linux atau Raspberry Pi. Penyebab sebenarnya bukan SSD atau catu daya - melainkan pengontrol firmware itu sendiri yang diam-diam kembali ke **kode cadangan yang tertanam di ROM**, sebuah versi yang masih dikirimkan Realtek secara internal sebagai **f0.01**.

# Mekanisme Tersembunyi - Rollback Firmware Secara Desain

Chip jembatan Realtek menyimpan firmware operasional dan data konfigurasi mereka di flash SPI eksternal. Saat dinyalakan, pengontrol memeriksa checksum sederhana. Jika checksum itu tidak cocok, ia menolak untuk memuat firmware eksternal dan malah mem-boot dari ROM internalnya.

Firmware cadangan ini sudah usang dan cacat. Ia kekurangan beberapa perbaikan stabilitas USB dan peningkatan manajemen status tautan yang ada di revisi berikutnya, yang mengarah ke urutan klasik yang dikenali setiap pengguna Linux:

```
| usb 3-2: reset perangkat USB berkecepatan tinggi nomor 2 menggunakan  
| xhci-hcd  
| usb 3-2: pembacaan deskriptor perangkat/64, kesalahan -71  
| Peringatan EXT4-fs (perangkat sda2): Kesalahan I/O saat menulis ke inode ...
```

Checksum dapat menjadi tidak valid ketika data konfigurasi ditulis ulang - misalnya, ketika jembatan memperbarui pengaturan manajemen daya atau UAS - dan perangkat kehilangan daya di tengah penulisan. Boot berikutnya mendeteksi checksum yang rusak dan kembali secara permanen ke firmware ROM.

Pada titik ini, "enclosure NVMe berkinerja tinggi" Anda berperilaku persis seperti casing tanpa nama termurah, karena secara internal sekarang menjalankan kode dasar cacat yang sama yang dibakar ke dalam silikon.

## Verifikasi Masalah

Anda dapat dengan mudah mengkonfirmasi status ini di bawah Linux:

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

Jembatan Realtek yang sehat melaporkan revisi firmware (**bcdDevice**) di atas 1.00. Yang kembali menunjukkan:

```
| bcdDevice f0.01
```

Tanda tangan **f0.01** ini berarti pengontrol mem-boot dari ROM - dan tidak ada jumlah pencabutan, pemformatan ulang, atau penyesuaian kernel yang akan memperbaikinya.

Mekanisme rollback ini telah **dikonfirmasi pada RTL9210. RTL9220** tampaknya memiliki arsitektur desain dan tata letak firmware yang sama, sehingga mungkin menunjukkan perilaku yang identik, tetapi ini tetap **kemungkinan, bukan terbukti**.

## Mengapa Anda Tidak Bisa Memperbaikinya Sendiri

Secara prinsip, perbaikan itu sepele: reflash firmware yang benar ke SPI. Dalam praktiknya, Realtek membuat ini tidak mungkin.

Perusahaan menyediakan pembaruan sumber tertutup untuk Windows kepada OEM dan integrator. Pengguna Linux tidak ditawarkan apa pun. Pengembang komunitas merekayasa balik utilitas flash yang kompatibel (**rtsupdate**, **rtl9210fw**, **rtsupdate-cli**) yang memungkinkan pemulihan firmware penuh dari sistem Linux - sampai Realtek mengeluarkan **pemberitahuan penghapusan DMCA** untuk menekan mereka.

Tidak ada alasan properti intelektual yang masuk akal untuk memblokir utilitas tersebut: mereka tidak mengekspos mikrokode, hanya mengatur urutan pembaruan melalui USB. Penghapusan Realtek bukan tentang perlindungan. Itu bersifat ideologis.

## Biaya Ideologi

Ini bukan tentang idealisme open-source. Ini tentang **permusuhan ideologis vendor perangkat keras terhadap sistem terbuka** yang merusak perangkat yang dipasarkan sebagai *kompatibel dengan Linux*.

Perlawaan Realtek terhadap dokumentasi dan alat terbuka telah berlangsung selama dua dekade, mencakup Wi-Fi, Ethernet, audio, dan sekarang pengontrol penyimpanan. Isolasi ini mungkin tidak diperhatikan di dunia yang hanya menggunakan Windows, tetapi menjadi beracun ketika chip yang sama diintegrasikan ke dalam produk multi-platform seperti **Sabrent EC-SNVE**, yang secara terbuka menampilkan logo Linux pada kemasannya.

Dengan melarang utilitas flash Linux dan memblokir pemeliharaan komunitas, Realtek telah secara efektif **mengkriminalisasi perbaikan sendiri**. Konsekuensinya menyebar ke luar:

- Pengguna Linux melihat perangkat keras “didukung” menurun menjadi ketidakstabilan.
- OEM seperti Sabrent dan Orico menghadapi biaya RMA dan garansi yang tidak perlu.
- Reputasi Realtek yang sudah lama untuk kompatibilitas Linux yang buruk diperkuat lagi.

Pada akhirnya, bukan open-source yang merusak perangkat Realtek - melainkan **permusuhan Realtek terhadap open-source** yang merusaknya.

## Jalan Rasional ke Depan

Solusi tidak memerlukan perubahan ideologis, hanya pragmatisme. Realtek bisa:

1. Merilis pembaruan baris perintah yang ditandatangani vendor untuk Linux (tidak perlu pengungkapan kode sumber).
2. Menerbitkan algoritma checksum sehingga integrator dapat memvalidasi gambar flash dengan aman.

3. Mengadopsi mode gaya DFU yang menerima pembaruan melalui Penyimpanan Massal USB, tanpa bergantung pada OS.

Masing-masing langkah ini akan mencegah biaya garansi, melindungi hubungan OEM, dan mengembalikan kepercayaan pada chip jembatan Realtek di antara pengguna Linux profesional - dari pembuat stasiun kerja hingga pengembang Raspberry Pi.

## Apa yang Bisa Anda Lakukan

Jika Anda menduga enclosure Anda telah kembali ke firmware ROM:

- Periksa dengan **lsusb -v | grep bcdDevice**.
- Jika menunjukkan **f0.01**, laporkan masalah ke OEM Anda.
- Sertakan cuplikan **dmesg** dan tunjukkan mekanisme rollback yang didokumentasikan ini.
- Mintalah vendor Anda untuk mengeskalasi masalah ke Realtek, dengan menyebutkan kebutuhan akan pembaruan yang kompatibel dengan Linux.

Kebijakan firmware Realtek tidak hanya mengganggu penggemar; ini menciptakan kerugian finansial nyata bagi ekosistem mereka sendiri. Semakin cepat realitas ini diakui di dalam perusahaan, semakin cepat pengguna Linux dan mitra OEM dapat berhenti membuang waktu pada siklus RMA yang dapat dihindari.

## Tanggapan Produsen

Baik Realtek maupun Sabrent diundang untuk memberikan pernyataan mengenai masalah rollback firmware yang dijelaskan di atas. Tanggapan mereka - jika diterima - akan ditambahkan di sini.

## Lampiran - Mengidentifikasi Perangkat yang Terkena Dampak

Pengontrol	ID Vendor	ID Produk	Catatan	Status
RTL9210	0x0bda	0x9210	Jembatan USB 3.1 Gen 2 10 Gb/s	<b>Dikonfirmasi</b> perilaku rollback
RTL9220	0x0bda	0x9220	Jembatan USB 3.2 Gen 2×2 20 Gb/s	<b>Kemungkinan</b> , arsitektur serupa

Tanda tangan rollback firmware: **bcdDevice f0.01**

Revisi stabil yang diketahui: **1.23 – 1.31**