

LinuxとRealtek RTL9210 USB-NVMeブリッジ

概要：

- **症状：** Linuxでの繰り返しUSBリセット、I/Oエラー、またはディスクの消失。
- **影響を受けるもの：** Realtek RTL9210（確認済み）およびRTL9220（可能性あり）。
- **原因：** チェックサム失敗後の内部ROM（**f0.01**）へのフォールバック。
- **影響：** 永続的な不安定性、Linux用のリフラッシュツールは利用不可。
- **解決策：** OEMのWindowsユーティリティのみがファームウェアを復元可能 - Realtekはオープンソースの代替をブロック。

前書き

2025年には、USB経由で接続されたSSDからRaspberry Piを起動することは完全に合理的であるはずです。しかし、Realtekのファームウェアの特異性により、この合理的な目標は冒険に変わりました。数ヶ月にわたる原因不明の不安定性 - ランダムなリセット、ディスクの消失、破損したファイルシステム - の後、著者はすべての一般的な修正を試みました：新しいケーブル、電源付きハブ、カーネルのアップデート、USBの調整、ファームウェアの微調整。突破口は、ChatGPTが深夜の奇妙な質問に答えたときによろしく訪れました：「USB-NVMeブリッジが古いファームウェアに戻った可能性はありますか？」

はじめに

RealtekベースのNVMeエンクロージャが、数週間の完璧な動作後に突然不安定になる場合 - 繰り返しUSBリセット、I/Oエラー、またはディスクの消失 - あなたは一人ではありません。このパターンは、無名のユニットからSabrentやOricoなどの有名なOEMまで、複数のブランドで現れています。共通の分母：**RealtekのRTL9210および可能性としてRTL9220 USB-NVMeブリッジチップ**。

最初はすべて正常に機能します。しかし、明らかな理由もなく、デバイスは負荷がかかったときや長時間の使用中に、特にLinuxやRaspberry Piシステムで切断し始めます。真の原因はSSDでも電源でもありません - それはファームウェアコントローラ自体が、Realtekが内部でまだ**f0.01**として出荷しているROMに埋め込まれたバックアップコードに静かに戻ることです。

隠されたメカニズム - 設計によるファームウェアのフォールバック

Realtekのブリッジチップは、動作ファームウェアと構成データを外部SPIフラッシュに保存します。電源投入時に、コントローラは単純なチェックサムを確認します。そのチェックサムが一致しない場合、外部ファームウェアの読み込みを拒否し、代わりに内部ROMから起動します。

このバックアップファームウェアは古く、欠陥があります。後のリビジョンに存在するいくつかのUSB安定性修正やリンク状態管理の改善が欠けており、すべてのLinuxユーザーが認識する典型的なシーケンスにつながります：

```
usb 3-2: xhci-hcdを使用して高速USBデバイス番号2をリセット
usb 3-2: デバイス記述子の読み取り/64、エラー -71
EXT4-fs警告（デバイスsda2）：inodeへの書き込み中のI/Oエラー ...
```

チェックサムは、構成データが書き換えられるときに無効になる可能性があります - たとえば、ブリッジが電源管理やUAS設定を更新するときに、デバイスが書き込み中に電源を失う場合です。次のブートで破損したチェックサムを検出し、永続的にROMファームウェアにフォールバックします。

この時点で、あなたの「高性能NVMeエンクロージャ」は、最も安価な無名シェルとまったく同じように動作します。なぜなら、内部ではシリコンに焼き付けられた同じ欠陥のあるベースコードを実行しているからです。

問題の検証

Linuxでこの状態を簡単に確認できます：

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

正常なRealtekブリッジは、ファームウェアリビジョン（**bcdDevice**）が1.00以上であると報告します。フォールバックしたものは以下を表示します：

```
bcdDevice f0.01
```

この**f0.01**の署名は、コントローラがROMから起動していることを意味します - そして、どれだけ抜き差ししても、再フォーマットしても、カーネルの調整をしても修正されません。

このフォールバックメカニズムは**RTL9210で確認されています**。**RTL9220**は同じ設計アーキテクチャとファームウェアのレイアウトを共有しているように見えるため、同一の動作を示す可能性があります、これは**可能性があるだけで証明されていません**。

自分で修正できない理由

原則として、解決策は簡単です：正しいファームウェアをSPIにリフラッシュするだけです。実際には、Realtekがこれを不可能にしています。

同社はOEMやインテグレーターにクローズドソースのWindowsアップデーターを提供しています。Linuxユーザーには何も提供されていません。コミュニティの開発者は、Linuxシステムから完全なファームウェア復元を可能にする互換性のあるフラッシュユーティリティ（**rtsupdater**、**rtl9210fw**、**rtsupdater-cli**）をリバースエンジニアリングしました - Realtekが**DMCA削除通知**を発行してそれらを抑制するまで。

そのようなユーティリティをブロックする合理的な知的財産の理由はありません：それらはマイクロコードを公開せず、USBを介した更新シーケンスを調整するだけです。Realtekの削除は保護のためではありませんでした。それはイデオロギー的でした。

イデオロギーのコスト

これはオープンソースの理想主義についてではありません。ハードウェアベンダーの**オープンシステムに対するイデオロギー的な敵対心**が、**Linux互換**として市場に出されているデバイスを壊しているのです。

Realtekのドキュメントやオープンなツールに対する抵抗は、Wi-Fi、Ethernet、オーディオ、そして今ではストレージコントローラに至るまで、20年にわたって続いています。この孤立主義はWindowsのみの世界では気づかれなくてもかもしれませんが、**Sabrent EC-SNVE**のようなマルチプラットフォーム製品に同じチップが統合されると、それが有毒になります。この製品はパッケージにLinuxロゴを堂々と表示しています。

Linuxフラッシュユーティリティを禁止し、コミュニティのメンテナンスをブロックすることで、Realtekは効果的に**自己修復を犯罪化**しました。結果は外に広がります：

- Linuxユーザーは「サポートされた」ハードウェアが不安定に劣化するのを見ます。
- SabrentやOricoのようなOEMは、不要なRMAや保証コストに直面します。
- RealtekのLinuxとの互換性が悪いという長年の評判がさらに強化されます。

最終的に、Realtekのデバイスを壊しているのはオープンソースではありません - **Realtekのオープンソースに対する敵対心**がそれらを壊しています。

合理的な前進の道

解決策にはイデオロギー的な転換は必要なく、ただ現実主義が必要です。Realtekは以下を行うことができます：

1. Linux用のベンダー署名付きコマンドラインアップデーターをリリースする（ソースコードの公開は不要）。
2. インテグレーターがフラッシュイメージを安全に検証できるようにチェックサムアルゴリズムを公開する。
3. OSに依存しないUSBマスストレージを介してアップデートを受け入れるDFUスタイルのモードを採用する。

これらの各ステップは保証コストを防ぎ、OEMとの関係を保護し、ワークステーションビルダーからRaspberry Pi開発者まで、プロフェッショナルなLinuxユーザーの間でRealtekのブリッジチップへの信頼を回復します。

あなたができること

エンクロージャがROMファームウェアに戻ったと疑う場合：

- `lsusb -v | grep bcdDevice`で確認してください。
- `f0.01`が表示された場合、OEMに問題を報告してください。
- `dmesg`の抜粋を含め、このドキュメント化されたロールバックメカニズムを指摘してください。
- ベンダーにRealtekに問題をエスカレーションするよう依頼し、Linux互換のアップデートの必要性を引用してください。

Realtekのファームウェアポリシーは愛好者を悩ませるだけでなく、彼ら自身のエコシステムに具体的な経済的損失を生み出します。この現実が社内で認識されるのが早ければ早いほど、LinuxユーザーとOEMパートナーは回避可能なRMAサイクルで時間を無駄にするのをやめることができます。

製造者の反応

RealtekとSabrentの両方が、上記のファームウェアルールバック問題に関する声明を提供するよう招待されました。受け取った場合の回答はここに追加されます。

付録 - 影響を受けるデバイスの特定

コントローラ	ベンダーID	製品ID	備考	ステータス
RTL9210	0x0bda	0x9210	USB 3.1 Gen 2 10 Gb/s ブリッジ	確認済み ロールバック動作
RTL9220	0x0bda	0x9220	USB 3.2 Gen 2×2 20 Gb/s ブリッジ	可能性あり 、類似アーキテクチャ

ファームウェアルールバックの署名：**bcdDevice f0.01**
既知の安定リビジョン：**1.23 – 1.31**