

# Linux и мост USB-NVMe Realtek RTL9210

## Краткое содержание:

- **Симптомы:** Повторяющиеся сбросы USB, ошибки ввода-вывода или исчезновение дисков в Linux.
- **Затронуты:** Realtek RTL9210 (подтверждено) и RTL9220 (возможно).
- **Причина:** Откат к внутренней ROM (**f0.01**) после сбоя проверки контрольной суммы.
- **Влияние:** Постоянная нестабильность, инструменты для перепрошивки под Linux отсутствуют.
- **Решение:** Только утилиты OEM для Windows могут восстановить прошивку – Realtek блокирует альтернативы с открытым исходным кодом.

## Предисловие

В 2025 году запуск Raspberry Pi с SSD, подключенного через USB, должен быть совершенно разумной задачей. Однако из-за особенностей прошивки Realtek эта разумная цель превратилась в приключение. После месяцев необъяснимой нестабильности – случайных сбросов, исчезновения дисков, поврежденных файловых систем – автор исчерпал все стандартные решения: новые кабели, питание через хабы, обновления ядра, настройки USB и оптимизация прошивки. Прорыв наступил только тогда, когда ChatGPT ответил на странный вопрос, заданный поздно ночью: «Возможно ли, что мост USB-NVMe откатился к старой прошивке?»

## Введение

Если ваш корпус NVMe на базе Realtek внезапно становится нестабильным после недель безупречной работы – повторяющиеся сбросы USB, ошибки ввода-вывода или исчезновение дисков – вы не одиноки. Этот шаблон проявился у нескольких брендов, от безымянных устройств до известных OEM, таких как Sabrent и Orico. Общий знаменатель: **чипы моста USB-NVMe Realtek RTL9210 и, возможно, RTL9220.**

Сначала всё работает. Затем, казалось бы, без причины, устройство начинает отключаться при нагрузке или во время длительного использования, особенно в системах Linux или Raspberry Pi. Настоящая причина – не SSD и не источник питания, а сам контроллер прошивки, который незаметно возвращается к своему **встроенному в ROM резервному коду**, версии, которую Realtek всё ещё поставляет внутренне как **f0.01**.

## Скрытый механизм – Откат прошивки по дизайну

Чипы моста Realtek хранят свою рабочую прошивку и данные конфигурации во внешней флэш-памяти SPI. При включении контроллер проверяет простую контрольную

сумму. Если эта сумма не совпадает, он отказывается загружать внешнюю прошивку и вместо этого загружается из внутренней ROM.

Эта резервная прошивка устарела и содержит ошибки. В ней отсутствуют несколько исправлений стабильности USB и улучшения управления состоянием канала, присутствующие в более поздних версиях, что приводит к классической последовательности, которую узнаёт каждый пользователь Linux:

```
usb 3-2: сброс высокоскоростного USB-устройства номер 2 с использованием xhci-hcd
usb 3-2: чтение дескриптора устройства/64, ошибка -71
Предупреждение EXT4-fs (устройство sda2): Ошибка ввода-вывода при записи в inode ...
```

Контрольная сумма может стать недействительной, когда данные конфигурации перезаписываются – например, когда мост обновляет настройки управления питанием или UAS – и устройство теряет питание во время записи. Следующая загрузка обнаруживает поврежденную контрольную сумму и навсегда возвращается к прошивке ROM.

На этом этапе ваш «высокопроизводительный корпус NVMe» ведёт себя точно так же, как самый дешёвый корпус без названия, потому что внутренне он теперь работает на том же дефектном базовом коде, записанном в кремний.

## Проверка проблемы

Вы можете легко подтвердить это состояние в Linux:

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

Здоровый мост Realtek сообщает о версии прошивки (**bcdDevice**) выше 1.00. Откатанный показывает:

```
| bcdDevice f0.01
```

Эта подпись **f0.01** означает, что контроллер загружается из ROM – и никакое количество отключений, переформатирования или настройки ядра этого не исправит.

Этот механизм отката **подтверждён для RTL9210. RTL9220**, похоже, имеет ту же архитектуру дизайна и компоновку прошивки, поэтому может демонстрировать идентичное поведение, но это остаётся **вероятным, а не доказанным**.

## Почему вы не можете исправить это самостоятельно

В принципе, решение тривиально: перепрошить правильную прошивку на SPI. На практике Realtek делает это невозможным.

Компания предоставляет закрытую утилиту обновления для Windows OEM и интеграторам. Пользователям Linux ничего не предлагается. Разработчики сообщества провели реверс-инжиниринг совместимых утилит для прошивки (**rtsupdater**, **rtl9210fw**, **rtsupdater-cli**), которые позволяли полностью восстанавливать прошивку из систем Linux – пока Realtek не выпустила **уведомления о снятии по DMCA**, чтобы подавить их.

Нет правдоподобного обоснования интеллектуальной собственности для блокировки таких утилит: они не раскрывают микрокод, а лишь организуют последовательность обновления через USB. Удаления Realtek не были связаны с защитой. Они были идеологическими.

## Цена идеологии

Это не про идеализм открытого исходного кода. Это про **идеологическую враждебность поставщика оборудования к открытым системам**, которая ломает устройства, продаваемые как *совместимые с Linux*.

Соппротивление Realtek документации и открытым инструментам длится два десятилетия, охватывая Wi-Fi, Ethernet, аудио и теперь контроллеры хранения. Эта изоляция может остаться незамеченной в мире, где используется только Windows, но становится токсичной, когда те же чипы интегрируются в мультиплатформенные продукты, такие как **Sabrent EC-SNVE**, который открыто демонстрирует логотип Linux на своей упаковке.

Запрещая утилиты прошивки для Linux и блокируя поддержку сообщества, Realtek фактически **криминализовала самостоятельный ремонт**. Последствия распространяются вовне:

- Пользователи Linux видят, как «поддерживаемое» оборудование деградирует до нестабильности.
- OEM, такие как Sabrent и Orico, сталкиваются с ненужными расходами на RMA и гарантию.
- Долгосрочная репутация Realtek за плохую совместимость с Linux укрепляется снова.

В конечном итоге, не открытый исходный код ломает устройства Realtek – это **враждебность Realtek к открытому исходному коду** ломает их.

## Рациональный путь вперед

Решение не требует идеологического сдвига, только прагматизма. Realtek могла бы:

1. Выпустить подписанную поставщиком утилиту обновления командной строки для Linux (без необходимости раскрытия исходного кода).
2. Опубликовать алгоритм контрольной суммы, чтобы интеграторы могли безопасно проверять образы флэш-памяти.

3. Принять режим в стиле DFU, который принимает обновления через USB-накопитель, независимо от операционной системы.

Каждый из этих шагов предотвратит затраты на гарантию, защитит отношения с OEM и восстановит доверие к чипам моста Realtek среди профессиональных пользователей Linux – от строителей рабочих станций до разработчиков Raspberry Pi.

## Что вы можете сделать

Если вы подозреваете, что ваш корпус вернулся к прошивке ROM:

- Проверьте с помощью **lsusb -v | grep bcdDevice**.
- Если отображается **f0.01**, сообщите о проблеме своему OEM.
- Включите отрывок из **dmesg** и укажите на этот задокументированный механизм отката.
- Попросите своего поставщика передать проблему в Realtek, ссылаясь на необходимость утилиты обновления, совместимой с Linux.

Политика прошивки Realtek не просто раздражает энтузиастов; она создаёт ощутимые финансовые потери для их собственной экосистемы. Чем раньше эта реальность будет признана в компании, тем скорее пользователи Linux и партнёры OEM смогут прекратить тратить время на неизбежные циклы RMA.

## Ответы производителей

Realtek и Sabrent были приглашены предоставить заявления относительно проблемы отката прошивки, описанной выше. Их ответы – если они будут получены – будут добавлены здесь.

## Приложение – Идентификация затронутых устройств

Кон- троллер	ID Постав- щика	ID Про- дукта	Примечания	Статус
RTL9210	0x0bda	0x9210	Мост USB 3.1 Gen 2 10 Гбит/с	<b>Подтверждено</b> поведение отката
RTL9220	0x0bda	0x9220	Мост USB 3.2 Gen 2×2 20 Гбит/с	<b>Возможно</b> , схожая архитектура

Подпись отката прошивки: **bcdDevice f0.01**

Известные стабильные версии: **1.23 – 1.31**