

# Audion Media Tools

---

[English](#) · [Руководство](#) · [Замеры](#) · [Решения](#)

## Содержание

- [Зачем это сделано](#)
- [Главное решение: новее не значит лучше](#)
  - [Что умеет каждый путь](#)
- [Почему этому можно верить](#)
  - [С чего всё началось](#)
  - [Проверено на том, что снимают камеры](#)
  - [Синхрон измерен, а не предположен](#)
  - [Семнадцать неточностей FFmpeg — компенсированы](#)
- [Принципы](#)
- [Что внутри](#)
- [Дальше](#)
- [Техническая часть](#)
  - [Запуск](#)
  - [Рабочая область](#)
  - [Значения по умолчанию](#)
  - [Отчёты](#)
  - [Что осталось за рамками](#)

Портативная оболочка для настоящей работы с видео и звуком: скачать, проверить, перекодировать, подготовить к монтажу, упаковать для заливки, положить в архив, аккуратно разобрать аудио.

## Зачем это сделано

Всё это умеет FFmpeg — и в этом же проблема. Правильная команда для монтажного кодека, для архивной мастер-копии и для заливки на площадку выглядит по-разному, занимает полторы строки ключей, и половина этих ключей выясняется опытом: что контейнер не держит выбранный поток, что смена канала перевернула порядок байтов в звуке, что метаданные камеры потерялись при перепакровке.

Программа хранит этот опыт в пресетах, а управление отдаёт окну: списки, кнопки, галочки, память путей, терминал операции и видимая команда. Дух старой версии для командной строки сохранён — изменился способ ею пользоваться.

## Главное решение: новее не значит лучше

Каждая сборка FFmpeg собрана под конкретную версию заголовков аппаратного кодирования NVIDIA, и каждая требует своего минимума драйвера. Поставьте самую свежую на драйвер постарше — аппаратное кодирование не ускорится, а перестанет работать.

| сборка FFmpeg | заголовки NVENC | нужен драйвер NVIDIA |
|---------------|-----------------|----------------------|
| 9.0.1         | n13.1.15.0      | <b>610.0</b>         |
| 8.0.1         | n13.0.19.0      | <b>570.0</b>         |
| 7.1.1         | n13.0.19.0      | <b>570.0</b>         |
| 7.1           | n12.2.72.0      | 551.76               |

Третья строка важнее остальных: 7.1.1 собрана теми же заголовками, что и 8.0.1, и требует те же 570.0 — «откатиться на версию назад» на старом драйвере не даёт ничего. Помогает только переход на 7.1.

Поэтому установщик подбирает сборку по вашему драйверу, а не берёт последнюю.

**В поставке 8.0.1** — и это выбор, а не забытое обновление. Машины для монтажа сегодня живут на драйверах примерно с 571 по 609; ветка 610 стоит у считанных единиц. И 8.1, и 9.x требуют именно её: поставить их значит заявить аппаратное ускорение и не дать его большинству тех, кому оно обещано. В 8.0.1 есть всё, что используют эти программы, и она работает на тех драйверах, которые у людей действительно стоят.

Нет видеокарты NVIDIA — всё это неважно: ставится последняя сборка, кодирование идёт на процессоре.

## Что умеет каждый путь

Все три аппаратных пути поддержаны одинаково полно: NVIDIA, Intel и AMD закрывают H.264, HEVC и AV1, каждый со своими родными регуляторами качества и своими пробами декодирования. Различия в таблице ниже — это пределы самих кодировщиков, а не пробелы в программе.

| путь                        | H.264 | HEVC | AV1       | 10 бит     | 4:2:2               |
|-----------------------------|-------|------|-----------|------------|---------------------|
| CPU (x264 / x265 / SVT-AV1) | да    | да   | да        | да         | да                  |
| NVIDIA NVENC                | 8 бит | да   | av1_nvenc | HEVC и AV1 | только новое железо |
| Intel QuickSync             | 8 бит | да   | av1_qsv   | HEVC и AV1 | нет                 |
| AMD AMF                     | 8 бит | да   | av1_amf   | HEVC и AV1 | нет                 |

**H.264 на любом аппаратном пути — только 8 бит.** Нужно 10 — берите HEVC, AV1 или процессорный x264.

Декодирование — отдельный стек: на машине без аппаратного кодирования аппаратный декод может работать прекрасно. Проверяется своими пробами: `CUDA decode`, `QuickSync decode`, `AMD/D3D11VA decode` и `dav1d` для AV1.

Регуляторы качества всегда родные для пути: `CRF` и предустановка SVT-AV1 для процессора, `CQ/QP` и предустановки NVENC, QuickSync или AMF для железа. Список тонкой настройки читают только x264 и x265, поэтому он и показывается лишь при выборе процессора.

**Перед первым запуском на новой машине — «Возможности железа».** Результат ложится в кэш, и профили сверяются с ним. Отсутствие пути — нормальный ответ, а не поломка: значит нет драйвера или самого железа. Кэш привязан к машине: если он чужой или его нет, диагностику прогоняют заново.

## Почему этому можно верить

Точность резки, синхрон звука и пределы FFmpeg здесь не заявлены, а измерены. [Свод замеров](#) держится на одном правиле: **ни одно утверждение не взято из документации** — каждая ячейка таблицы это реальная попытка, каждое число замер. Где измерение оказалось неверным,

это тоже записано: отдельный раздел собирает случаи, когда правдоподобный вывод пришлось отменить.

## С чего всё началось

Shutter Encoder с версии 19.1 читает дробную частоту как целую: 29,97 становится 30. Ошибка  $\times 1,001$  копится — 1,8 секунды рассинхрона на каждые полчаса, четыре секунды на час. На 20.2 дефект на месте, обращение [#402](#) открыто.

Здесь счёт идёт точной дробью:

| частота | дробь      | запрос | правильно    | через <code>round(fps)</code> |
|---------|------------|--------|--------------|-------------------------------|
| 23,976  | 24000/1001 | 50 с   | <b>1199</b>  | 1200                          |
| 29,97   | 30000/1001 | 4 мин  | <b>7193</b>  | 7200                          |
| 59,94   | 60000/1001 | 4 мин  | <b>14386</b> | 14400                         |

Величины, за которые идёт разговор: кадр — 33 мс на 29,97, звуковой пакет — 8 мс, сэмпл — 20 микросекунд. Смещение на один кадр монтажёр потом ищет руками.

## Проверено на том, что снимают камеры

Не на сгенерированных роликах — на материале с настоящих аппаратов, от Sony до ARRI.

Расхождение по кадрам везде ноль:

| источник                | запрос  | кадров | расхождение |
|-------------------------|---------|--------|-------------|
| ProRes 25p              | 3 → 9   | 150    | 0           |
| HEVC long-GOP 29,97     | 12 → 30 | 600    | 0           |
| H.264 all-intra 23,976  | 3 → 9   | 144    | 0           |
| 29,97 drop-frame        | 5 → 15  | 330    | 0           |
| 59,94 drop-frame        | 3 → 9   | 360    | 0           |
| MPEG-TS, старт 1,44 с   | 2 → 8   | 300    | 0           |
| 4 канала PCM            | 2 → 8   | 150    | 0           |
| Canon HEVC 4:2:2 10 бит | 3 → 10  | 166    | 0           |

| источник                   | запрос | кадров | расхождение |
|----------------------------|--------|--------|-------------|
| ARRI ProRes 4444 XQ 24,000 | 2 → 8  | 144    | 0           |

Отдельная находка: Alexa Mini, снимающая честные 24,000, объявляет **24/1** — и читать такой файл как 23,976 значит **создать** ошибку  $\times 1,001$ , а не устранить её. Пять кадров за четыре минуты, ровно то, за что критикуют чужую нарезку. Частота берётся у файла как есть; ту, что не прочиталась, честнее отклонить, чем заменить догадкой.

## Синхрон измерен, а не предположен

Маркер, существующий в обоих потоках в один момент: белый кадр в начале каждой секунды и щелчок 20 мс там же. Отклонение от собственного смещения источника:

| контейнер | расхождение |
|-----------|-------------|
| MP4       | 0...2 мс    |
| MOV       | 0...5 мс    |
| MKV       | 1...3 мс    |
| MXF       | 9...18 мс   |

Сдвигает только MXF — его муксер равняет звук по своим единицам монтажа, это меньше половины кадра и объявлено предупреждением. Прежнее предупреждение про MKV точное измерение опровергло, и оно снято, а не оставлено правдоподобной страшилкой.

## Семнадцать неточностей FFmpeg — компенсированы

Ни одна не очевидна из документации, каждая стоила отдельного разбора и каждая обойдена в коде. Среди них: **-ss** до **-i** сбрасывает отсчёт времени; конец, округлённый «к ближайшему», теряет кадр, стоящий на 29,9966; HEVC с В-пирамидой пишет на 2–4 кадра меньше запрошенного; **avg\_frame\_rate** на переменной частоте превращает 12 запрошенных секунд в 16; **-n** на существующий файл отвечает «уже существует» и **кодом возврата 0**, из-за чего пустой прогон читается как успех; PCM при копировании в MXF получает 384 лишних сэмпла в начале.

Полный перечень с проявлениями — в [замерах](#), раздел 10.

[Список проверок](#) показывает, что именно прогоняется перед выпуском.

## Принципы

**Проверено на настоящем файле, а не на сгенерированном.** Синтетический ролик не несёт ни тегов камер, ни звука с обратным порядком байтов — а именно на них ломались две вещи, которых иначе никто бы не увидел. Матрица проверок построена на файлах с камер.

**Форма показывает, что останется, а не что убирается.** Список того, что будет в результате, читается однозначно; список исключений — нет.

**Что контейнер не удержит — отвергается до запуска, а не в середине.** Несовместимое сочетание кодека и контейнера блокируется в окне и проверяется службой ещё раз.

**Остановленный прогон не оставляет того, что нельзя открыть.** Незавершённый файл не притворяется готовым.

**Отказ перезаписать — это пропуск, а не успех.** В отчёте он так и называется.

**Результат кладётся в папку операции, а не в имя файла.** `Source\Day1\clip.mov` и `Source\Day2\clip.mov` не сваливаются в одну плоскую папку и не спорят за имя: структура подпапок повторяется в результате.

## Что внутри

| страница                            | о чём                                                                                     |
|-------------------------------------|-------------------------------------------------------------------------------------------|
| Скачивание                          | ссылка или список, форматные профили, разрешение, субтитры, потоки                        |
| Аудиопотоки                         | извлечение, ресемплинг, нормализация, упаковка; звук в видео без перекодирования картинки |
| Обрезка                             | покадровая, по ключевым кадрам, с памятью меток по каждому файлу                          |
| Перепаковка, частота, таблицы цвета | смена контейнера, работа с частотой кадров, наложение таблиц                              |
| Архив                               | долговременные мастер-копии: FFV1, x264 без потерь                                        |
| Монтажные кодеки                    | ProRes и DNxHR в MOV или MXF                                                              |
| Хранение                            | компактное и качественное: x264, x265, SVT-AV1                                            |
| Доставка                            | просмотр, отправка, заливка                                                               |

| страница    | о чём                                    |
|-------------|------------------------------------------|
| Диагностика | что за железо, что умеет, чем кодировать |

## Дальше

- [Руководство](#) — работа по страницам, рабочая область, отчёты.
- [Замеры](#) — цифры, на которых всё держится.
- [Проверки](#) — что прогоняется перед выпуском.
- [Решения](#) — почему сделано так, с датами и числами.

## Техническая часть

### Запуск

```
launcher_gui.cmd
```

Окружение не собрано:

```
builder_main.cmd
```

В сборщике есть отдельные пункты установки FFmpeg (из двух источников), yt-dlp и 7-Zip в портативную папку `Tools\`.

Окно открывается на `http://127.0.0.1:8080/`.

### Рабочая область

Верхняя строка задаёт пути для всех страниц:

| кнопка           | что делает                     |
|------------------|--------------------------------|
| Источник         | выбрать папку с исходниками    |
| Добавить файл... | выбрать один файл как источник |
| Назначение       | выбрать папку результата       |

| кнопка   | что делает                                                                           |
|----------|--------------------------------------------------------------------------------------|
| Сбросить | вернуть проектные <code>Source\</code> и <code>Transcoded\</code> , файлов не трогая |
| Удалить  | очистить текущие источник и назначение после подтверждения                           |
| Список   | вывести в терминал имена файлов текущего источника                                   |

Плюс `Probe Source` — один запуск разбора с краткой строкой параметров.

Единый словарь для всех проектов Audion. По-английски: **Source**, **Add file...**, **Target**, **Reset**, **Delete**, **List**. Слова «Цель», «Очистить», `Destination` и `Clear` для этих кнопок не используются.

Источник обходится рекурсивно, структура подпапок сохраняется в результате. Результат пишется в папку операции и профиля:

```
Transcoded\Remux\remux_mkv_to_mp4_video_audio\Day1\clip.mp4
```

Выбранная вручную папка назначения просто заменяет `Transcoded`, внутренняя структура остаётся прежней.

## Значения по умолчанию

| что               | значение                                                   |
|-------------------|------------------------------------------------------------|
| качество (CRF/CQ) | 14                                                         |
| звук              | 384 kbps; для M4A есть профиль 256                         |
| ресемплер         | SoX/libsoxr, полное качество ( <code>precision=33</code> ) |
| архив скачиваний  | выключен                                                   |

Пиксельный формат на странице монтажных кодеков не выбирается — его задаёт сама цель: ProRes всегда 10 бит 4:2:2, DNxHR HQ — 8 бит, DNxHR HQX — 10 бит. Чтобы поменять разрядность, выбирают другую цель, а не отдельное поле.

Для MXF доступны PCM 16 или 24 бита либо вывод без звука; несовместимые варианты блокируются в окне и проверяются службой. FLAC недоступен для MOV — этот контейнер его не держит, нужен MP4 или MKV.

## Отчёты

После аудио-операций пишется `audio_report.md` и `.json`: исходный поток, параметры входа, режим обработки, громкость, путь результата.

## Что осталось за рамками

WebM пока входной, скачиваемый и перепаковываемый формат, но не отдельная цель массового кодирования. Понадобится доставка под VP9 или AV1 в WebM — её лучше добавить отдельным профилем с внятными параметрами качества, контейнера и звука, а не пристёгивать к существующим.