Audion System Manager — руководство пользователя

English

Содержание

- Запуск
- Как устроено окно
- POWERSHELL 7
- TERMINAL
- PYTHON
- VS CODE
- GIT
- AI CLI
- ЗАГРУЗКИ
- Папка портативной сборки
- Где что лежит
- Клавиши
- Сеть и приватность

Audion System Manager — одно окно для рабочего окружения разработчика на Windows: **PowerShell 7**, **Windows Terminal**, **Python 3.12, 3.13 и 3.14**, **VS Code**, **Git** с **GitHub CLI** и агенты в консоли — **Claude Code**, **Codex**, **Grok Build**. Каждую программу можно поставить в систему или развернуть портативной папкой — со всеми опциями, которых не даёт менеджер пакетов, и с настройками Audion. Всё делается кнопками: без чёрного окна консоли и без ключей командной строки.

Запуск

Запускайте **Start.exe**. Программа работает с правами обычного пользователя. Права администратора нужны только для установки на всю машину (PowerShell 7, Python для всех пользователей, VS Code «на машину»): Windows спросит их окном UAC в момент, когда запускается установщик, и только для него. Отказались — ничего не установлено и ничего не изменилось. Скачивание, портативные сборки, настройки терминала и редактора, расширения прав не требуют.

Язык берётся из системы при первом запуске (русский в русской Windows, в остальных английский). Кнопки **RU | EN** справа вверху переключают язык, луна и солнце — тему. Размер шрифта (**a | A | A+**) — в **ABOUT**, карточка **НАСТРОЙКИ**. Выбор запоминается.

Второй запуск из той же папки не открывает вторую копию, а показывает окно уже работающей: два установщика сразу мешали бы друг другу.

Как устроено окно

Верхняя строка — две тёмных плашки: шесть разделов и справа язык и тема. Название и версия программы — в заголовке окна и в ABOUT.

Разделы — POWERSHELL 7, TERMINAL, PYTHON, VS CODE, GIT, AI CLI. У каждого свой цвет; цвет линии под заголовком показывает, какой выбран. Под названием раздела — версия в выбранном на его странице режиме и значок режима (экран — в системе, флешка — портативно). Точка рядом с версией: зелёная — версия последняя, оранжевая — не установлено или вышла новая («1.132.0 → 1.142.0»), серая — источник не ответил. У AI CLI вместо версии — сводка «2 из 3 · вход 2»: сколько агентов стоит и в скольких выполнен вход.

Страница раздела начинается с переключателя режима — двух кнопок в одной рамке слева вверху:

- **В СИСТЕМУ** — программа ставится на машину (или в профиль, где так задумано) со своими опциями;
- **ПОРТАТИВНО** — программа распаковывается в папку и ничего не ставит; папку можно унести на другую машину.

Под ним — карточки выбранного режима в две колонки с линией между ними; обе колонки относятся к режиму, выбранному переключателем. Слева — что есть: состояние (что стоит, где, какая версия последняя, что лежит в загрузках). Справа — что с этим сделать: опции установки, а в портативном режиме папка и **РАЗВЁРТЫВАНИЕ** (как сборка разворачивается из архива в

Download, её кнопки). Общие для обоих режимов карточки — профиль PowerShell, шрифт терминала, настройки и расширения VS Code, GitHub CLI и настройки Git — стоят в тех же колонках. В AI CLI агенты идут по очереди слева и справа, в ABOUT слева сведения о программе, справа настройки, клавиши и лицензия. В узком окне колонки встают одна под другой: сначала левая, потом правая.

Внизу, постоянно:

- строка состояния: **F1 DOCS** (это руководство), последнее сообщение, ход операции с кнопкой **СТОП** и кнопка **ЖУРНАЛ**;
- инструменты **ЗАГРУЗКИ** и **ABOUT** (ещё раз по кнопке — назад к разделу); в ABOUT, кроме сведений о программе, — карточка **НАСТРОЙКИ**: размер шрифта, проверка версий при запуске, папка настроек Audion;
- справа две главные кнопки для раздела, выбранного вверху, в режиме, выбранном на его странице: **СКАЧАТЬ** (последнюю версию в папку Download; серая — скачана уже последняя) и **УСТАНОВИТЬ** / **РАЗВЕРНУТЬ** / **ОБНОВИТЬ** (поставить или развернуть с опциями страницы; нет скачанного — сначала скачивает). Под названием кнопки — что именно она сделает.

Журнал — внизу над кнопками; полоса между страницей и журналом перетаскивается мышью, и высота журнала запоминается. В нём всё, что делает программа, и вывод установщиков, pip, WinGet, VS Code. Открывается сам, когда начинается операция; **Ctrl+J** или кнопка **ЖУРНАЛ** прячут и показывают его. Файл журнала за день и журналы установщиков (msiexec, python.org, VS Code) лежат в `._runtime\logs`.

Одна долгая операция за раз: пока идёт скачивание или установка, кнопки, которые начали бы другую, серые. **СТОП** останавливает скачивание, распаковку, установку расширений. Установщик, который работает с правами администратора, остановить нельзя: прерванная на полпути установка хуже доведённой.

POWERSHELL 7

В систему. PowerShell 7 ставится из MSI релиза на GitHub — того же пакета, что ставит WinGet, но с опциями, которых WinGet не даёт. Опции — чипы карточки **ОПЦИИ УСТАНОВКИ** (янтарный — включено):

Опция	Что даёт
ОТКРЫТЬ ЗДЕСЬ	«Открыть здесь PowerShell 7» в меню папок

Опция	Что даёт
ВЫПОЛНИТЬ .PS1	«Выполнить с помощью PowerShell 7» в меню файлов .ps1
PWSH В PATH	команда `pwsh` в любой консоли
ЖУРНАЛ СОБЫТИЙ	сообщения PowerShell в «Просмотре событий»
MICROSOFT UPDATE	обновления через Microsoft Update и WSUS — PowerShell обновляется вместе с Windows
УДАЛЁННОЕ ВЗАИМОДЕЙСТВИЕ	`Enable-PSRemoting`: открывает слушателя WinRM, поэтому выключено, пока вы его не включите
БЕЗ ТЕЛЕМЕТРИИ	телеметрия выключена; если её выключали раньше, программа это видит и не включает обратно

Выбор переживает обновления: пакет сам хранит его и подставляет при следующей установке, кто бы её ни делал — Microsoft Update или WinGet. Что включено сейчас, программа читает там, куда пишет сам установщик (`HKLM\SOFTWARE\Microsoft\PowerShellCore\InstallerProperties`); строка под чипами говорит, что изменит **УСТАНОВИТЬ**.

УСТАНОВИТЬ берёт самый свежий MSI из Download (нет — скачивает), сверяет его сумму SHA-256 с релизом и запускает msiexec — Windows спросит права администратора. Та же версия уже стоит — доустанавливает к ней опции; всё уже так — ничего не делает. **ПЕРЕУСТАНОВИТЬ** ставит пакет заново поверх.

Портативно. zip того же релиза распаковывается в папку (по умолчанию `Portable\PowerShell` рядом с программой; поле **Папка** меняет её). Профиль и модули пользователя портативный PowerShell берёт из «Документов», как и установленный. **ЗАПУСТИТЬ** открывает его в отдельной консоли.

Профиль PowerShell — общий для обоих режимов: скрипт `Documents\PowerShell\Microsoft.PowerShell_profile.ps1`, который выполняется в начале каждого окна. Профиль Audion (`Presets\PowerShell`) — кодировка UTF-8, подсказки PSReadLine из истории списком, поиск по истории стрелками. **ПЕРЕНЕСТИ ПРОФИЛЬ** кладёт его (прежний откладывается рядом с датой в имени), **ВЗЯТЬ С МАШИНЫ** делает набором профиль этой машины. С галочкой **Настройки автора: профиль Audion** (нажата по умолчанию) профиль кладётся и при каждой установке и развёртывании; без неё PowerShell 7 ставится чистым, профиль на машине не трогается.

TERMINAL

В систему. Windows Terminal из Магазина Windows. На Windows 11 он уже есть и обновляется Магазином; если его нет или он старый, кнопка рядом с версией ставит или обновляет его через WinGet (тот же пакет, для пользователя, без прав администратора).

Карточка **НАСТРОЙКИ AUDION** показывает, что в файле настроек (`Presets\Terminal\settings.json`: схема Audion с оранжевым курсором #FF9807, русский интерфейс, три сочетания клавиш, PowerShell по умолчанию, шрифт Cascadia Mono), и куда он ляжет: терминал держит настройки в разных местах в зависимости от того, откуда поставлен (Магазин, Preview, распакованный). **ПЕРЕНЕСТИ НАСТРОЙКИ** кладёт файл во все найденные; прежние настройки откладываются рядом с датой в имени — вернуть их можно обычным копированием. Файл сначала проверяется на правильность JSON. Если терминал открыт, лучше закрыть его окна: при выходе он может записать поверх своё состояние. **ВЗЯТЬ С МАШИНЫ** делает набором настройки терминала этой машины — так набор не отстаёт от того, что поменяли в самом терминале.

Терминал по умолчанию (Windows 11) — в каком окне открываются консольные программы: **WINDOWS TERMINAL**, **КОНСОЛЬ WINDOWS** или **ПУСТЬ РЕШАЕТ WINDOWS**. Хранится в реестре пользователя, прав не нужно. Портативный Windows Terminal терминалом по умолчанию стать не может.

Шрифт — Cascadia Code и Cascadia Mono с вариантами NF (Nerd Font: значки в строке приглашения и списках файлов) и PL с GitHub, для пользователя, без прав. Начертание, которое уже есть в Windows (в Windows 11 — Cascadia Mono), не трогается; те же файлы повторно не ставятся.

Портативно. zip релиза с GitHub распаковывается в папку (по умолчанию `Portable\Terminal`), рядом с WindowsTerminal.exe кладётся файл `.portable` — и терминал держит настройки в `settings\` рядом с собой. С галочкой **Настройки автора** туда же кладутся настройки Audion; та же галочка в карточке **НАСТРОЙКИ AUDION** решает, класть ли их в установленный терминал после установки. Нажата по умолчанию; без неё терминал ставится чистым, со своими настройками. При обновлении `settings\` и `.portable` остаются.

PYTHON

Под заголовком страницы — **Версия**: 3.12, 3.13 или 3.14. Версии стоят рядом, каждая в своей папке, со своим установщиком, архивом и своей последней версией на python.org; `py -3.12`, `py -3.14` выбирают нужную, `py` без номера — самую новую из установленных. По

умолчанию 3.12: на ней встроенные рантаймы проектов Audion (тег колёс cp312). Выбор один на оба режима и хранится в настройках программы.

В систему. Установщик python.org со своими ключами — для всех пользователей, в свою папку (по умолчанию `C:\Python312`, `C:\Python313` или `C:\Python314` — у каждой версии своя: в Program Files каждый `pip install` упирается в запрос прав, в своей папке — нет). Ключи — чипы карточки **УСТАНОВКА**: путь в начало системного PATH (заглушка Магазина больше не перехватывает `python`), лаунчер `py`, tcl/tk; ассоциации .py, ярлыки в «Пуске», документация и тесты по умолчанию выключены. После установщика программа убирает заглушки Магазина (файлы python.exe и python3.exe нулевого размера в профиле) и кладёт рядом с python.exe его копию python3.exe, проверяет, какие python и python3 отвечают в новой консоли, и обновляет pip. Заглушки — псевдонимы App Installer, и Windows возвращает их с каждым его обновлением; но python3 тогда отвечает настоящий Python — его папка стоит в PATH раньше, чем папка заглушек. Строка **Заглушки Магазина** говорит, перекрыты ли они.

Почему 3.12: на этой версии работают рантаймы внутри проектов Audion, и метка колёс (cp312) должна совпадать. python.org выпускает 3.12 после 3.12.10 только исходниками, поэтому последняя версия здесь — последняя, у которой есть установщик для Windows.

Карточка **В СИСТЕМЕ** показывает то, что обычно и ломается: все Python, что знает реестр (и те, что не в PATH), лаунчер py, какой python отвечает в консоли и заглушки Магазина. **ОБНОВИТЬ PIP У ВСЕХ** обновляет pip у всех Python, которые знает лаунчер py (где писать без прав нельзя, Windows спросит их).

Портативно. Полный архив python.org (`python-3.N.M-amd64.zip`, тот, что берёт Python Install Manager) распаковывается в папку своей версии (по умолчанию `Portable\Python` для 3.12, `Portable\Python313` и `Portable\Python314` для остальных): весь Python с pip, ничего не ставится и не пишется в реестр. Папка Doc не распаковывается, если не включён чип **ДОКУМЕНТАЦИЯ**. При обновлении ваши пакеты (`Lib\site-packages` и `Scripts`) переходят в новую сборку. Лаунчер py портативный Python не видит — запуск полным путём к python.exe или кнопкой **ЗАПУСТИТЬ**.

VS CODE

В систему. Установщик Microsoft — **НА МАШИНУ** (для всех пользователей, нужны права администратора) или **В ПРОФИЛЬ** (без прав, только для вас) — со всеми галочками мастера: меню файлов и папок, ассоциации, `code` в PATH, ярлык на рабочем столе. Затем настройки Audion и расширения. Та же или более новая версия уже стоит — установщик не запускается,

настройки и расширения делаются. Установщик заменяет программу, поэтому VS Code нужно закрыть.

Портативно. Архив распаковывается в папку (по умолчанию `Portable\VSCode`), рядом с Code.exe — папка `data`, где редактор держит настройки и расширения и ничего не пишет в профиль. Сам себя портативный VS Code не обновляет — обновляет эта программа: свежий архив, программа заменяется, `data` не трогается.

Настройки и расширения — общие для обоих режимов. Настройки — файл `Presets\VSCode\settings.json`; в нём выключено доверие к папкам (`security.workspace.trust.enabled`), поэтому редактор не спрашивает про каждую папку. Класть ли их и **ripgrep для Todo Tree** вместе с ними, выбирается в окне выбора (ниже): Todo Tree ищет пометки через ripgrep, путь к которому внутри VS Code меняется с каждым обновлением, поэтому берётся отдельно установленный (нет — ставится через WinGet) и его путь пишется в настройки. Та же галочка **Настройки автора** стоит и прямо на странице — в карточке установки каждого режима.

ВЗЯТЬ С МАШИНЫ делает набором настройки, клавиши (`keybindings.json`) и сниппеты VS Code этой машины — установленного или портативного, по режиму страницы; машинный путь ripgrep при этом не берётся. Клавиши и сниппеты из набора кладутся вместе с настройками. **ОБНОВИТЬ РАСШИРЕНИЯ** обновляет все расширения сейчас самим VS Code (`code --update-extensions`); обычно они обновляются и сами — в настройках Audion включено `extensions.autoUpdate`.

Расширения — из трёх списков в `Presets\VSCode`: **ЯДРО** (`extensions.txt`, ставится всегда), **ИИ-АГЕНТЫ** (`extensions-ai.txt`: Claude, Codex и Kilo Code по умолчанию; Grok — по желанию, это панель над отдельным Grok Build CLI) и **ПО ЖЕЛАНИЮ** (`extensions-optional.txt`). Kilo Code берёт модели разных поставщиков по ключу API или через шлюз Kilo, подписки Codex и Grok к нему подключаются официально; подписку Claude к нему не подключайте — Anthropic не разрешает работать по ней через сторонних агентов и блокирует за это аккаунт (Claude в Kilo Code — только по ключу API).

Что ставить, выбирается в отдельном окне галочками: наверху — **Настройки автора** и **ripgrep для Todo Tree**, ниже — каждое расширение трёх списков по разделам, кнопки **ВСЕ**, **ПО СПИСКАМ** (как задумано списками) и **СНЯТЬ ВСЁ** и счётчик отмеченного и уже стоящего. Окно открывается перед **УСТАНОВИТЬ**, **ПЕРЕУСТАНОВИТЬ** и **РАЗВЕРНУТЬ** (его кнопка внизу и запускает установку), перед **ТОЛЬКО РАСШИРЕНИЯ** (без настроек) и кнопкой **ВЫБРАТЬ РАСШИРЕНИЯ** (кнопка **СОХРАНИТЬ**) — она стоит в карточке установки рядом с числом отмеченного, под галочкой **Настройки автора**. Отметки запоминаются в настройках программы,

когда окно закрыто своей кнопкой; **ОТМЕНА** оставляет всё как было. В подсказке названия раздела — почему в нём именно эти, в подсказке галочки — полный идентификатор и стоит ли расширение. Списки можно править (кнопка **СПИСКИ**): одна строка — один идентификатор, заголовок `# --- Имя ---` открывает раздел.

GIT

В систему. Git for Windows из релиза на GitHub для всей машины, с галочками мастера его установщика: Git в PATH, редактор VS Code, ветка main, перевод строк CRLF, менеджер паролей Git Credential Manager, сертификаты Windows, кэш файловой системы, символьные ссылки (выключены), меню Проводника, Git LFS, ассоциации, профиль Git Bash в Windows Terminal, ярлык на столе (выключен). Каждая галочка передаётся установщику его собственным ключом. Последняя версия уже стоит — установщик не запускается; **ПЕРЕУСТАНОВИТЬ** меняет галочки у стоящего Git. Нужны права администратора.

Портативно. PortableGit того же релиза распаковывается сам рядом с папкой (по умолчанию `Portable\Git`) и встаёт на её место: Git, Git Bash и всё нужное в одной папке.

GitHub CLI (`gh`) ставится в режиме страницы: MSI релиза в систему или zip в `Portable\GitHub CLI`, со сверкой сумм релиза. **ВОЙТИ** открывает `gh auth login --web` в отдельном окне консоли: gh показывает одноразовый код, в браузере код вводится на github.com — после этого и `gh`, и Git ходят на GitHub с этим входом. **ВЫЙТИ** — `gh auth logout`.

Настройки Git — набор `Presets\Git\gitconfig` (перевод строк, длинные пути, кириллица в именах файлов, редактор, ветка main, UTF-8). **ПЕРЕНЕСТИ НАСТРОЙКИ** ставит ключи в глобальные настройки по одному (`git config --global`): имя, почта и вход в `~\.gitconfig` остаются. Галочка **Настройки автора: gitconfig Audion** (нажата по умолчанию) делает это и после каждой установки и развёртывания; без неё Git ставится чистым. Рядом с каждым ключом видно, что действует у вас сейчас. **ВЗЯТЬ С МАШИНЫ** делает набором глобальные настройки этой машины — без личного и машинного (user, credential, url, http и всё, что похоже на адрес с логином или токен).

AI CLI

Агенты в консоли: **Claude Code**, **Codex**, **Grok Build** и последним — **Antigravity CLI** (`agy`, агент Google на смену Gemini CLI). У раздела нет режимов: агенты живут в профиле пользователя и ставятся без прав администратора.

У каждого агента — версия против последней, вход и кнопки:

- **УСТАНОВИТЬ / ОБНОВИТЬ** — установщиком его авторов (PowerShell, `irm … | iex`), как он указан в их документации; стоящий обновляется его собственной командой (`claude update`, `grok update`, `agy update`; Codex — тем же установщиком). Папка программы добавляется в PATH пользователя, если установщик этого не сделал.
- **ВОЙТИ** — OAuth в браузере по вашей подписке: в отдельном окне консоли запускается команда входа самого CLI (`claude auth login`, `codex login`, `grok login --oauth`). Antigravity CLI отдельной команды входа не имеет: запускается сам `agy`, и без сохранённого входа он открывает страницу входа Google.
- **ВЫЙТИ** — выход из аккаунта (после вопроса). У Antigravity CLI такой кнопки нет: выход — командой `/logout` в самом `agy`.
- **ЗАПУСТИТЬ** — агент в отдельном окне консоли.

Есть ли вход, программа видит только по тому, что файл входа агента есть; сам файл не читается. Antigravity CLI хранит вход в диспетчере учётных данных Windows — его вход программа не проверяет. Кнопка **УСТАНОВИТЬ** внизу ставит и обновляет всех отмеченных чипом **КНОПКОЙ ВНИЗУ**; Antigravity CLI по умолчанию не отмечен и в сводке раздела наверху не считается, пока не установлен.

ЗАГРУЗКИ

Что скачано в папку `Download` рядом с программой: установщики и портативные архивы, их размер и дата. Разделы берут из неё самый свежий файл своего вида и убирают прежний того же вида, когда скачан новый. Файл из папки сверяется перед установкой: с опубликованной суммой SHA-256, где источник её даёт, иначе с суммой, записанной при скачивании; не сошлось — файл убирается и скачивается заново. Папку можно унести на другую машину вместе с программой — ставить можно будет без сети; изменившийся после скачивания файл без сети не берётся. **В КОРЗИНУ** и **ВСЁ В КОРЗИНУ** перемещают файлы в Корзину.

Папка портативной сборки

Поле **Папка** в портативном режиме принимает путь по Enter или когда вы уходите из поля; **ВЫБРАТЬ...** открывает выбор папки, **ПО УМОЛЧАНИЮ** возвращает `Portable\<программа>`. Подходит пустая папка или папка, где эта программа уже развёрнута: при обновлении программа заменяет её файлы, поэтому папку с чужими вещами она не тронет. Корень диска не годится. Новая сборка распаковывается рядом и встаёт на место папки целиком; если что-то не

удалось (например, открыт файл из папки), остаётся прежняя сборка со всеми данными. **В КОРЗИНУ** перемещает портативную сборку в Корзину (с её настройками), запущенную — нет.

Где что лежит

Путь	Что там
`Start.exe`	запуск программы
`App\`	сама программа (своя среда .NET, ничего ставить не нужно)
`config\settings.json`	ваш выбор: язык, тема, режимы, папки, опции, отметки расширений
`Presets\`	настройки Windows Terminal, VS Code (с клавишами и сниппетами), Git и профиль PowerShell, списки расширений — то, что переносит программа
`Download\`	скачанные установщики и архивы
`Portable\`	портативные сборки (если не указана другая папка)
`._runtime\logs\`	журнал по дням и журналы установщиков

Клавиши

Клавиша	Что делает
F1	это руководство
Ctrl+1 … Ctrl+6	разделы POWERSHELL 7, TERMINAL, PYTHON, VS CODE, GIT, AI CLI
Ctrl+7, Ctrl+8	ЗАГРУЗКИ, ABOUT
Ctrl+J	журнал
Esc	назад к разделу

Сеть и приватность

Программа обращается только к источникам самих программ: GitHub (PowerShell, Windows Terminal, Git, GitHub CLI, Cascadia), python.org, сервису обновлений Microsoft (VS Code), Visual Studio Marketplace (расширения — через сам VS Code), WinGet и установщикам агентов (claude.ai,

chatgpt.com, x.ai). Ответ о версиях хранится шесть часов; чип **ПРОВЕРЯТЬ ВЕРСИИ ПРИ ЗАПУСКЕ** в ABOUT выключает проверку при запуске. Ничего не собирается и никуда не отправляется.