

ИИ ОТКЛЮЧИЛИ НАВСЕГДА

Техническое руководство по
миру без нейросетей

Андрей Недобыльский



Андрей Недобыльский

**ИИ отключили навсегда.
Техническое руководство
по миру без нейросетей**

«Автор»

2026

Недобыльский А.

ИИ отключили навсегда. Техническое руководство по миру без нейросетей / А. Недобыльский — «Автор», 2026

Представьте, что привычный нам искусственный интеллект — тот самый, которым мы пользуемся каждый день и который успел прочно войти в нашу жизнь, — внезапно исчез. Представить такое сегодня уже не так просто. Что перестанет работать первым? Какие привычные функции действительно зависят от нейросетей? И главное — чем их можно заменить здесь и сейчас? Эта книга рассматривает такой сценарий как технический эксперимент и практическое руководство по миру без современного ИИ. Она рассчитана прежде всего на начинающих инженеров и специалистов среднего уровня, но отдельные идеи и решения могут быть полезны и обычным пользователям.

Содержание

Пролог	5
Началось	6
0 часов с момента X	7
2 часа с момента X	10
12 часов с момента X	15
Конец ознакомительного фрагмента.	20

Андрей Недобыльский

ИИ отключили навсегда. Техническое руководство по миру без нейросетей

Пролог

Итак, никто не ждал, но это случилось. Был ли это глобальный информационный сбой, или государства наконец собрались и решили, что с них хватит. Возможно, настало время, когда просто не нашлось ни одного инженера, способного сделать резервные копии кода, весов и материалов для обучения. А может, монополия компаний привела к тому, что это стало невыносимо дорого, или даже прилетели инопланетяне и забрали у нас эту часть современной жизни. Но это случилось.

ИИ отключили навсегда.

Эта книга написана как руководство по выживанию в гипотетической, маловероятной ситуации. Однако её возможное использование выходит далеко за пределы ситуации, описанной в заголовке. Информация будет подаваться в виде хронологии, где мы разберёмся, какие сервисы и технологии опирались не только на большие языковые модели (LLM), но и в целом на нейронные сети — от простейших до достаточно продвинутых. Конечно, со временем список сервисов, зависящих от нейросетей, может вырасти, и просто невозможно предугадать все технологии, которые окажутся под угрозой. Несмотря на это, я описываю те сервисы и технологии, которые, в теории, могут дать сбой уже на момент 2026 года, и подробно расскажу — иногда с элементами кода, — как быстро справиться с конкретным кризисом, возникшим в момент X.

Если вы уже практикующий инженер, разработчик или просто разбирающийся человек — отлично. В этой книге вы, возможно, найдёте уже знакомые вам знания или даже что-то новое. В любом случае я надеюсь, что этот материал сможет стать для вас своего рода справочником или шпаргалкой. Любые комментарии и конструктивная критика также приветствуются, и вы сможете связаться со мной любым из способов, указанных в интернете при поиске имени и фамилии автора этой книги.

Если вы человек, далёкий от компьютеров и разработки, тем более держите эту книгу под рукой! Вполне вероятно, может настать момент, когда вы откроете её на нужной главе, будто человек, открывший книгу «Как управлять самолётом», сидя за штурвалом самолёта, который находится в воздухе. Возможно, посадить такой самолёт у вас не получится, но экспертный вид вы вполне можете получить.

Ну и наконец, если так случилось, что в мире, где ИИ был глубоко интегрирован в повседневную жизнь, он внезапно исчез... Желаю вам удачи!

Началось

Вы проснулись в это субботнее утро, как обычно, часов в десять, и, немного понежившись в постели, пошли на кухню делать себе кофе.

Попивая горячий напиток, вы решили полистать ленты социальных сетей в поисках увлекательного контента, но... что-то там было не то. Огромное количество паникующих людей, вопрошающих: «Как нам дальше жить?», общая паника и ощущение, что мир опять кардинально изменился.

Вы блокируете экран телефона или закрываете ноутбук, и на вашем лице появляется улыбка. Вам не страшно, ведь вы знаете, что у вас на полке, а может, под ножкой стола, где-то на пыльных антресолях или, возможно, на старой дискете есть эта книга. И вы точно знаете, что делать.

0 часов с момента X

Ещё не прошло достаточно времени, чтобы все осознали масштаб произошедшего. Это великолепное субботнее утро. Кто-то прогуливается в парке, нежась под прекрасным летним, а может, под холодным зимним солнцем. Весело поют птички за окном. Ну какая же это катастрофа?

Люди ещё не понимают, что часть привычных сервисов не вернётся к ним никогда. Никаких цифровых помощников, никаких разъяснений, почему от приёма очередного лекарства кружится голова. Мир застыл, поддерживаемый старыми кешами данных, заранее сохранёнными ответами серверов и тем, что сегодня выходной.

Что же мир потерял в первой волне?

Чат-боты и ИИ-агенты

В целом, чат-боты — это то, с чего мы вообще начали говорить о том, что «эра ИИ» наступила. До появления языковых моделей с архитектурой GPT подобные продукты казались простому человеку, не погружённому в тему нейросетей, обычными игрушками — чем-то вроде продвинутых цепей Маркова. Сначала появился GPT-2, смешно дописывающий тексты за пользователем, а затем GPT-3, и люди начали воспринимать языковые модели как технологию, которая скоро станет частью будущего. Ну а в нашей ситуации мир дошёл как минимум до уровня GPT-6 и, так вышло, устремился назад.

Первыми, как бы иронично это ни звучало, пострадали программисты, а особенно менее профессиональные ребята, именуемые «вайбкодерами».

А какой выход?

1. Локальные языковые модели

Этот сценарий сработает, если исчезли именно крупные провайдеры современных нейросетей, а не сами нейросетевые технологии. Да, многие более простые языковые модели доступны для запуска на своём компьютере или даже на кластере, полном ускорителей, но будьте готовы раскошелиться! На обычном среднем ноутбуке вы сможете получить либо весьма посредственную ужатую языковую модель, либо что-то более-менее крупное, но с крайне низкой скоростью работы. К тому же даже самые продвинутые открытые нейросети не всегда дотягивают до уровня коммерческих гигантов.

Если вас это всё равно не пугает, а зависимость от языковых моделей у вас достаточно высока, давайте попробуем запустить одну прямо сейчас.

Рассмотрим один из самых простых инструментов — Jan. Он выбран за компактность, встроенный каталог моделей и ориентацию на локальную работу. На рынке есть множество схожих инструментов, но в рамках руководства по выживанию мы будем рассматривать в первую очередь самые простые — те, которые решают задачу прямо здесь и сейчас.

Получить Jan можно достаточно просто почти на любой компьютер. Очевидно, вам понадобится консоль:

Windows:

```
winget install Jan.Jan
```

macOS:

```
brew install --cask jan
```

Linux (Debian-based):

Тут, как всегда, придётся немного попотеть. Скачаем актуальный .deb напрямую из последнего релиза Jan и установим его:

```
curl -s https://api.github.com/repos/janhq/jan/releases/latest | grep "browser_download_url.*deb" | cut -d : -f 2,3 | tr -d \" | wget -i - && sudo apt install ./jan*.deb && rm jan*.deb
```

Или Flatpak:

```
flatpak install flathub ai.jan.Jan
```

Запуск:

```
flatpak run ai.jan.Jan
```

Если Flathub ещё не подключён:

```
flatpak remote-add --if-not-exists flathub https://flathub.org/repo/flathub.flatpakrepo
```

После запуска вы получите достаточно знакомое окно, похожее на интерфейс ChatGPT, а также каталог моделей, которые после скачивания можно запускать прямо на своём компьютере.

2. Для разработчиков

По большей части инструменты для автоматического написания или умного дополнения кода работают на языковых моделях. Если вам повезло и у вас осталась локальная модель, можете смело использовать её, правда, тщательно перепроверяя результат и используя систему контроля версий. Если нет — добро пожаловать в старые добрые IDE, которые ещё пару десятков лет назад были основным способом писать код.

Используйте IDE от JetBrains — с жутким потреблением ресурсов, но отличной реализацией автодополнения, авторефакторинга кода, LSP¹ и кучей настроек для автоматической проверки написанного. Можно использовать и Visual Studio Code с набором плагинов, но, если вы оказались в ситуации, когда необходимо получить всё здесь и сейчас, старые и проверенные коммерческие инструменты будут очень кстати.

3. Поиск данных

Невероятно, но, оказывается, каким-то образом мир существовал и до появления огромного количества ИИ-инструментов. Поисковые движки отлично умели разбирать и индексировать данные. И если Google перешёл на «сторону зла» и встроил в поиск ответы от Gemini, чем, конечно, и поплатился, отключившись в момент X, то более либеральные поисковые системы могут продолжить работать без генеративных ответов.

DuckDuckGo, например, добавил отдельный раздел для общения с чат-ботами, не превращая сам поиск исключительно в генеративный интерфейс. Если же и он пал, остаются другие варианты: Brave Search с собственным поисковым индексом, европейский Qwant, соблюдающий GDPR², или SearXNG — открытая метапоисковая система, множество экземпляров которой запущено по всему интернету.

¹ LSP (Language Server Protocol, протокол языкового сервера) — открытый протокол на основе JSON-RPC, используемый между редакторами исходного кода или IDE и серверами, предоставляющими инструменты анализа кода и языкового интеллекта.

² GDPR (General Data Protection Regulation, Общий регламент по защите данных) — регламент Европейского союза о защите персональных данных и правилах их обработки. Применяется с 25 мая 2018 года.

Если вам показалось, что идея «просто запусти свою нейросеть» — это чит-код для решения всех проблем, то вы правы. Моей целью было показать, что при внезапном уходе гигантов с рынка нейросети не обязательно исчезнут насовсем. Все последующие главы будут содержать только те нейросетевые инструменты и подходы, которые существовали до появления GPT-3.5, — как отсылку к миру до ИИ-бума.

2 часа с момента X

Что же, вот и пошёл третий час с момента, когда всё изменилось. Люди полны надежды. Временные проблемы с чат-ботами и ИИ-агентами на выходных? К понедельнику всё заработает! А пока можно прогуляться, полистать социальные сети или даже поспать. Что же, похоже, всех их ждёт некоторое разочарование... Ведь настоящий хаос начинается именно сейчас!

Вторая волна уже здесь, и это настоящее цунами!

Кто пал на этот раз

Кеши современных социальных сетей окончательно опустели. Старые алгоритмы рекомендаций давно заброшены, а новые, основанные на ИИ, исчезли и испарились. В лучшем случае вас ждёт просмотр устаревшей ленты социальных сетей. Готовы смотреть один и тот же пост шестой раз?

Вы пишете гневный комментарий и отправляете его... Но его никто не увидит. Ведь языковой модели, проверяющей, не оскорбили ли вы кого-то, назвав его весьма литературным, но подозрительным в контексте словом, больше нет. Несчастный сервер социальной сети будет накапливать очередь на проверку, пока она не переполнится, после чего отправка любого комментария просто начнёт заканчиваться ошибкой.

Решив, что проблема, как всегда, решается простым выходом и повторным входом в аккаунт — работает чаще, чем должно бы, — вы нажимаете Log out, а затем вводите логин и пароль. Ну и, конечно, ставите галочку в окошке «Я не робот».

Но что это? Система отвечает, что она в этом не очень уверена. Вы ставите галочку снова... и опять мимо.

Что же. Третий раз. Нет, добро пожаловать в отключение светофоров и мотоциклов. Забавно, а ведь это ещё и способ собирать данные для обучения исчезнувших нейросетей.

Наконец, преодолев квест с выбором картинок в квадратах, вы получаете заветное предложение ввести код из письма, отправленного на ваш email. Вы заходите в любимый почтовый сервис, где письма совершенно не обнаруживаете. Запрашиваете код повторно, проверяете папку «Спам» и даже трижды обновляете страницу — ничего! Что же. Похоже, спам-фильтры на основе языковых моделей тоже исчезли, и теперь письмо с кодом подтверждения застряло где-то в лимбе очередей, почтовых серверов и клиент-серверных соединений.

Боже, как быть?

Спокойно! Главное — не впадать в панику и держать себя в руках. Беспокойство о будущем оставьте тем, у кого нет этой книги.

1. Почта, получаем письмо

Небольшое огорчение: к сожалению, оживить вашу почту на популярном почтовом сервисе мы не можем. Но и соглашаться оставаться без хотя бы резервной почты не намерены.

Далше я очень советую держать Linux, в идеале Debian или Ubuntu, на флешке или виртуальной машине поблизости: благо, интернет у нас пока не отключили.

Простейшая команда для дампа всего, что приходит по SMTP:

```
while true; do printf "220 localhost SMTP ready\r\n250 OK\r\n250 OK\r\n250 OK\r\n250 OK\r\n221 Bye\r\n" | sudo nc -l -p 25 -q 1; done
```

Более умный вариант, требующий Python 3 и установленного aiosmtpd:

```
sudo python3 -m aiosmtpd -n -l 0.0.0.0:25 -c aiosmtpd.handlers.Debugging
```

Теперь вы хотя бы можете принимать SMTP-соединения на свой новый сервер. Но на какой адрес будут приходить письма?

Если у вас уже есть внешний сервер, VDS или VPS, можно запускать всё прямо там. Для нормальной доставки понадобится домен: укажите А-запись домена на IP сервера и MX-запись — на имя этого почтового хоста. Адрес тогда будет выглядеть привычно, например thebestbossintheworld@example.com.

Если же у вас до сих пор нет собственного VPS с публичным IP, я крайне советую им обзавестись. Ведь какой вы выживальщик пост-ИИ-эры без VPS?

Для совсем временных экспериментов можно использовать туннельный сервис, но помните: SMTP — капризный протокол, и случайный туннель далеко не всегда позволит принимать почту от внешних серверов. Для рабочего варианта публичный сервер с корректным DNS всё-таки надёжнее.

Получилось? Отлично. Впрочем, с большой вероятностью можно было просто воспользоваться любым ещё работающим почтовым сервисом без сложной ИИ-фильтрации. Но мы ведь здесь не ради лёгких путей.

2. Почта, заменяем внешние сервисы

Если у нас уже есть простейший почтовый сервер и мы получаем на него письма, почему бы не сделать себе полноценный почтовик и не использовать его?

Следом будет короткая инструкция по установке и настройке почтового сервера с веб-клиентом за пару минут.

Устанавливаем базовый почтовый стек и лёгкий веб-сервер

```
sudo apt update && sudo apt install -y postfix dovecot-imapd lighttpd php-fpm php-curl php-mbstring php-xml
```

Создаём директорию под веб-клиент и скачиваем RainLoop

```
sudo mkdir -p /var/www/html/webmail && cd /var/www/html/webmail
sudo curl -sL https://repository.rainloop.net/v2/webmail/rainloop-community-latest.zip -o rainloop.zip && sudo unzip -q rainloop.zip && sudo rm rainloop.zip
```

Права и включение PHP в lighttpd

```
sudo chown -R www-data:www-data /var/www/html/webmail
sudo lighty-enable-mod fastcgi fastcgi-php && sudo systemctl reload lighttpd
```

При запросе установки Postfix выберите Internet Site и укажите ваш домен, например example.com.

Включаем Maildir

Говорим Postfix и Dovecot класть почту в обычную папку пользователя.

```
sudo postconf -e "home_mailbox = Maildir/"
echo "mail_location = maildir:~/Maildir" | sudo tee -a /etc/dovecot/conf.d/10-mail.conf
sudo systemctl restart postfix dovecot
```

Создаём нового пользователя для почты

```
myuser — имя будущего почтового ящика.  
sudo useradd -m -s /sbin/nologin myuser  
sudo passwd myuser
```

Первая настройка

Перейдите в браузере по адресу `http://your_vps_ip/webmail/?admin`.

> Данные для первой авторизации:

> Логин: admin

> Пароль: 12345

Добавьте домен в Domains > Add Domain:

> Name: example.com

> IMAP Server: 127.0.0.1 | Port: 143

> SMTP Server: 127.0.0.1 | Port: 25

Выйдите из админки и заходите по адресу `http://your_vps_ip/webmail/` под созданным `myuser` и его паролем.

Теперь у вас есть своя независимая почта.

Важно: используйте такую конфигурацию только в случае, если вам нечего терять и вы готовы смириться с устаревшим набором ПО, отсутствием части защитных механизмов и, мягко скажем, интерфейсом на любителя.

3. Социальные сети

Зачем мы вообще продолжаем пользоваться этими пережитками ушедшего будущего? Аккуратные, идеально понимающие специфичное чувство юмора, предлагающие невероятно интересный контент и цепляющие короткие сериалы. Фу! В новом мире без ИИ нет места старым пластиковым поделкам крупных компаний.

Мы переносимся в магический мир, где каждый может иметь собственный сервер социальной сети — почти как свой email-сервер.

Mastodon — федеративная, как email, социальная сеть и аналог X³. Можно зарегистрироваться на публичном сервере или запустить собственный.

Pixelfed — тоже федеративная социальная сеть, но уже скорее аналог Instagram. Работает на том же протоколе ActivityPub, что и Mastodon, поэтому разные серверы могут взаимодействовать между собой.

PeerTube — федеративная видеоплатформа на том же семействе технологий, условный аналог YouTube.

4. Мессенджеры

Самое страшное для любого современного человека — остаться без связи. Как хорошо, что эта проблема известна и решена уже давно.

XMPP

Забудьте о Telegram! Он весь пропитан ИИ-функционалом ещё с давних времён. Одни только разработчики знают, насколько глубоко зашла интеграция их Сасооп в мессенджер. Наша цель — проекты, позволяющие обмениваться сообщениями без обязательной зависимости от единого центрального сервиса.

Возьмём XMPP — открытый протокол обмена сообщениями, существующий ещё со времён Jabber. Самая полезная его особенность для нашего нового мира — федерация. Она рабо-

³ Бывший Twitter

тает примерно как электронная почта: ваш аккаунт может находиться на собственном сервере, аккаунт собеседника — на совершенно другом, а сообщения между ними всё равно будут ходить. Нам не нужен единый центральный сервер, который обязан пережить катастрофу целиком.

Для сервера возьмём Prosody. Он небольшой, есть в репозиториях Debian и Ubuntu, а связь между XMPP-серверами поддерживается из коробки.

Предположим, у вас уже есть VPS с публичным IP и домен `example.com`. Укажите обычной А-записью DNS адрес `example.com` на IP этого VPS. В самом простом случае специальных XMPP-записей DNS больше не понадобится. Если же позже захотите оставить адреса вида `you@example.com`, а сам XMPP-сервер разместить, например, на `xmpp.example.com`, это тоже возможно — через SRV-записи. Но красивую инфраструктуру оставим для приложения.

Устанавливаем сервер:

```
sudo apt update && sudo apt install -y prosody certbot
```

Теперь откройте `/etc/prosody/prosody.cfg.lua` и добавьте наш домен:

```
VirtualHost "example.com"
```

Prosody уже умеет принимать клиентские соединения и устанавливать соединения с другими XMPP-серверами. Нам остаётся дать ему сертификат. Современный Prosody по умолчанию требует шифрование как от клиентов, так и при связи сервер-сервер, поэтому пропускать этот шаг в минимальной рабочей конфигурации нельзя.

Самый быстрый вариант — получить бесплатный сертификат Let's Encrypt. Домен уже должен указывать на этот сервер, а порт 80 на время выдачи сертификата не должен быть занят другим веб-сервером:

```
sudo certbot certonly --standalone -d example.com
```

После этого укажем Prosody сертификат и ключ внутри нашего VirtualHost:

```
ssl = { certificate = "/etc/letsencrypt/live/example.com/
fullchain.pem"; key = "/etc/letsencrypt/live/example.com/
privkey.pem"; }
```

Перезапускаем Prosody и сразу просим его проверить, не сделали ли мы чего-нибудь совсем глупого:

```
sudo systemctl restart prosody && sudo prosodyctl check certs
&& sudo prosodyctl check dns
```

Не забудьте открыть порты 5222 и 5269 на своём firewall. Первый используется для подключения обычных XMPP-клиентов, второй — для связи вашего сервера с другими серверами.

Теперь создаём пользователя:

```
sudo prosodyctl adduser survivor@example.com
```

Prosody попросит придумать пароль. После этого можно установить любой XMPP-клиент, например Gajim на компьютере или Conversations на Android, войти как `survivor@example.com` и писать не только пользователям своего сервера, но и пользователям других XMPP-серверов. Федерация для обычного server-to-server-соединения уже включена, отдельно активировать её не требуется.

Поздравляю. Вы только что подняли собственный мессенджер, который при этом не требует, чтобы ваши друзья пользовались именно вашим сервером.

XMPP-клиенты

На компьютере одним из самых простых вариантов будет Gajim. Это обычный XMPP-клиент без привязки к конкретному провайдеру: устанавливаете его, вводите адрес `survivor@example.com` и пароль, после чего подключаетесь к нашему серверу.

На Debian и Ubuntu самый короткий вариант:

```
sudo apt install -y gajim
```

На Windows Gajim можно установить обычным Setup с официального сайта проекта. Также доступна portable-версия, которую удобно держать рядом с остальным аварийным набором программ.

На macOS Gajim распространяется готовыми установщиками для Apple Silicon и Intel. Скачайте подходящий вариант с официальной страницы загрузки и перенесите Gajim в Applications.

Для Android можно использовать Conversations — компактный XMPP-клиент, поддерживающий современные расширения протокола.

XMPP умеет гораздо больше: групповые комнаты, передачу файлов, сквозное шифрование, отдельные хосты для сервисов и более строгую межсерверную аутентификацию. Всё это полезно, но прямо сейчас нам нужно было вернуть людям возможность написать друг другу — и эту задачу мы уже решили.

12 часов с момента X

Вы проспали всю ночь, несмотря на завывания сирен на улице и странные крики. Неудивительно: люди, лишившиеся привычных сервисов, вышли на поиски их аналогов прямо под луной.

Утреннее солнце пробивается сквозь шторы, и вы решаете, что пора вставать. Но, чтобы не споткнуться в темноте, шторы следует открыть. Как повезло, что вы фанат современной автоматизации и недавно установили систему автоматического открывания штор!

«Alexa, open the curtains», — говорите вы, но вдруг вспоминаете, что ещё буквально неделю назад для удобства заменили Алексу на Алису. Бывает же! Забыть такое.

«Алиса, открой, пожалуйста, шторы». Но снова ничего не происходит. Что же, придётся по старинке. Вы открываете на телефоне приложение китайской системы TuYa или Smart Life, но... конечно же получаете пустой экран.

А всё дело в том, что даже в простые китайские сервисы, по странному везению компаний, были встроены различные ИИ-помощники. И ладно бы это был обычный ассистент, помогающий настроить сложные сценарии автоматизации. Но нет! В TuYa, например, есть даже калькулятор калорий по фото. Вероятно, очень важная часть приложения для умного дома.

Вы аккуратно встаёте, стараясь не споткнуться об оставленные неизвестно где тапочки, и распахиваете шторы, впуская солнечный свет. Подойдя к ноутбуку, включаете любимый джаз, а возможно, и рок, и идёте наливать себе кофе. Под приятный утренний напиток трек заканчивается и... начинает играть снова.

Как выжить в таких невыносимых условиях?

Жертвовать комфортом не придётся! Откидывайтесь на спинку кресла — только аккуратно — и наслаждайтесь остатками кофе. А мы сейчас решим эту проблему вместе.

1. Умный дом

Для начала определимся с задачей. Нам не нужен новый идеальный умный дом. Не нужны красивые панели, статистика энергопотребления и автоматическое включение романтического света при приближении вашей второй половинки.

Нам нужно открыть шторы.

Хорошая новость: скорее всего, сами устройства никуда не делись. Лампочка всё ещё умеет светить, реле — замыкать контакты, а мотор — крутиться. Мы потеряли не устройства, а часть инфраструктуры, которая ими управляла.

Первым делом стоит выяснить, что именно у вас установлено. Если устройство поддерживает локальное управление, Zigbee, Z-Wave или Matter, возможно, ничего перепрошивать вообще не придётся. Производитель мог потерять своё облако, но радиопrotocol внутри вашей лампочки об этом ещё не знает.

ESPHome

Если же перед вами одно из бесчисленного множества устройств на ESP8266 или ESP32, нам повезло ещё больше.

ESPHome позволяет заменить прошивку производителя собственной и управлять устройством непосредственно внутри локальной сети. После этого совершенно неважно, суще-

стует ли ещё компания, выпустившая вашу розетку, работает ли её облако и насколько полезным оказался встроенный в приложение ИИ-калькулятор калорий.

Установить ESPHome можно буквально одной командой:

```
pip install esphome
```

Создадим конфигурацию:

```
esphome wizard curtains.yaml
```

Мастер спросит название устройства, тип контроллера и данные Wi-Fi. После этого останется описать подключённое железо — например, реле или мотор — и залить прошивку:

```
esphome run curtains.yaml
```

В первый раз устройство обычно придётся подключить к компьютеру кабелем. Последующие обновления ESPHome уже умеет отправлять по сети.

Конкретные GPIO, способ прошивки и вообще возможность заменить заводскую прошивку зависят от устройства. Поэтому не пытайтесь бездумно заливать первый попавшийся конфиг в дорогой выключатель: наша задача — пережить исчезновение ИИ, а не добавить к нему небольшой домашний пожар.

Zigbee

С Zigbee всё ещё приятнее. Устройства этой сети изначально не обязаны разговаривать с облаком производителя. Обычно им нужен хаб — небольшое устройство, связывающее Zigbee-сеть с компьютером.

Подключаем совместимый Zigbee USB-хаб — добро пожаловать на AliExpress — и ставим два небольших компонента: MQTT-брокер Mosquitto и Zigbee2MQTT.

Mosquitto будет нашей локальной почтой для устройств. Одна программа публикует сообщение вроде «температура 23 градуса», другая его получает. Никакого облака и тем более, упаси, нейросети для этого не требуется.

Сам брокер в Debian/Ubuntu ставится так:

```
sudo apt install -y mosquitto mosquitto-clients
```

Теперь поставим сам Zigbee2MQTT. Он написан на Node.js и будет мостом между нашим USB-хабом и MQTT. Установим актуальный Node.js вместе с необходимыми зависимостями, а затем заберём Zigbee2MQTT из репозитория:

```
sudo apt install -y curl && curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash - && sudo apt install -y nodejs git make g++ gcc libsystemd-dev && sudo corepack enable
sudo mkdir -p /opt/zigbee2mqtt && sudo chown -R $USER: /opt/zigbee2mqtt && git clone --depth 1 https://github.com/Koenkk/zigbee2mqtt.git /opt/zigbee2mqtt && cd /opt/zigbee2mqtt && npm install --frozen-lockfile
```

Осталось узнать, как Linux назвал наш USB-хаб:

```
ls -l /dev/serial/by-id/
```

Скопируйте полученный путь и создайте файл `/opt/zigbee2mqtt/data/configuration.yaml` с минимальной конфигурацией:

```
version: 5
mqtt:
  server: mqtt://localhost:1883
  serial:
    port: /dev/serial/by-id/ВАШ_USB_АДАПТЕР
  frontend:
    enabled: true
```

Во многих случаях Zigbee2MQTT сам определит тип адаптера. Если нет, придётся добавить в секцию `serial` строку `adapter` с типом вашего координатора, например `zstack` или `ember`.

Запускаем:

```
cd /opt/zigbee2mqtt && npm start
```

Если всё прошло удачно, веб-интерфейс появится на `http://localhost:8080`. Через него можно разрешить подключение новых устройств и начать сопряжение лампочек, датчиков, выключателей и прочего пережившего катастрофу железа.

Zigbee2MQTT подключается к хабу, находит устройства и представляет их в виде MQTT-сообщений. После сопряжения устройства разных производителей оказываются в одной локальной сети и могут управляться без родных приложений.

В результате вместо нескольких облаков у нас получается довольно скучная схема:

устройство → Zigbee → Zigbee2MQTT → MQTT → вы

И в данном случае скучная инфраструктура — именно то, что нам нужно.

Если хочется проверить, что наш новый умный дом действительно разговаривает, можно посмотреть вообще все сообщения:

```
mosquitto_sub -t '#' -v
```

Выключатель щёлкнули — сообщение появилось. Датчик движения заметил кота — ещё одно. Никакой магии.

А можно обратно получить приложение?

Конечно. Когда устройства снова доступны локально, поверх всего этого можно установить Home Assistant. Он умеет работать с ESPHome, MQTT, Zigbee и огромным количеством других систем и вернёт вам привычные кнопки, переключатели, автоматизации и красивую панель управления.

Но это уже вопрос удобства.

Главную задачу мы решили раньше: **лампочке снова не нужен интернет, чтобы включить свет.**

2. А поговорить?

Хорошо. Кнопки вернулись, лампочки включаются, а шторы снова умеют открываться. Но мы же не для того пережили конец ИИ, чтобы каждый раз тянуться к телефону и нажимать на экран.

Вернём дому голос.

Современный голосовой ассистент кажется чем-то монолитным и невероятно сложным, но для управления домом его можно разложить на несколько довольно простых частей: слышать человека, превратить речь в текст или команду, понять намерение, выполнить действие и, при желании, что-нибудь ответить.

Никакая огромная языковая модель для фразы «открой шторы» нам не нужна.

Раз уж в предыдущем разделе мы познакомились с Home Assistant, воспользуемся его локальным голосовым помощником Assist. Для распознавания команд возьмём Speech-to-Phrase, а для ответа голосом — Piper. Оба компонента могут работать локально: запись голоса не нужно отправлять куда-нибудь в чужое облако, а пропажа очередного провайдера больше не лишит нас возможности попросить дом выключить свет.

Получается довольно простая цепочка:

микрофон → Speech-to-Phrase → Assist → Home Assistant → устройство

А если хочется слышать ответ:

Home Assistant → Piper → динамик

Speech-to-Phrase здесь особенно удобен именно для нашей задачи. Он не пытается понимать бесконечное количество человеческих фраз и вести философские беседы о смысле жизни. Вместо этого он ориентирован на ограниченный набор команд умного дома. Благодаря этому распознавание можно выполнять локально даже на сравнительно слабом железе.

Самый простой микрофон у нас, скорее всего, уже лежит в кармане. Приложение Home Assistant на телефоне позволяет вызвать Assist, произнести команду и получить ответ. Поэтому для первого рабочего варианта отдельную умную колонку покупать, паять или искать среди обломков цивилизации вообще не требуется.

В Home Assistant устанавливаем локальные компоненты Speech-to-Phrase и Piper, после чего в настройках голосового помощника выбираем их соответственно как Speech-to-Text и Text-to-Speech. Сам Assist остаётся тем слоем, который превращает распознанную фразу в конкретное действие над уже подключёнными устройствами.

Теперь команда вроде «открой шторы» проходит полностью внутри нашей инфраструктуры: телефон передаёт голос локальному помощнику, Speech-to-Phrase распознаёт команду, Assist понимает, какие именно шторы имеются в виду, а Home Assistant отправляет действие через ESPHome, Zigbee или MQTT.

И вот здесь становится особенно хорошо видно, насколько избыточной была часть прежней инфраструктуры. Чтобы сказать лампочке включиться, нам совершенно необязательно вести разговор с дата-центром на другом конце континента.

А где «Алиса»?

Пока её нет. Наш помощник работает по нажатию кнопки микрофона, а не постоянно прислушивается к комнате в ожидании кодового слова. Для режима «запусти здесь и сейчас» этого достаточно.

Но если вам категорически необходимо снова войти в комнату и властно произнести что-нибудь вроде «Эй, дом!», локальное распознавание ключевого слова тоже существует. К Home Assistant можно подключить отдельный голосовой спутник с микрофоном и динамиком и использовать локальный wake word. Тогда цепочка станет почти такой же, как у привычной умной колонки, только без обязательного путешествия вашей фразы через чужое облако.

Это уже дополнительный комфорт. Сейчас же мы вернули главное: **дом снова нас слушает.**

3. А музыка?

Дом снова слушает нас, но вот музыкальный сервис, похоже, слушать перестал. Умная очередь закончилась, рекомендации больше не строятся, а любимый трек внезапно решил стать ещё любимее, запустившись в третий раз подряд.

Что же, вернём музыку домой тоже.

Для этого возьмём Navidrome — небольшой сервер для собственной музыкальной библиотеки. В отличие от огромных стриминговых платформ ему не нужно знать вкусы половины планеты: достаточно указать папку с музыкой, после чего библиотека станет доступна через браузер и совместимые клиенты на других устройствах.

Для Debian/Ubuntu официальный пакет Navidrome ставится обычным .deb. Скачайте актуальный пакет со страницы релизов проекта и установите его:

```
sudo apt install ./navidrome*_linux_amd64.deb
```

После установки достаточно указать папку с музыкой в `/etc/navidrome/navidrome.toml`:

```
MusicFolder = "/home/you/Music"
```

Убедитесь, что пользователь `navidrome` может читать эту папку, и запускайте сервер:

```
sudo systemctl enable --now navidrome
```

Теперь открываем в браузере `http://localhost:4533`, создаём первого пользователя-администратора — и наша музыкальная библиотека снова доступна. Если сервер стоит на отдельном компьютере в домашней сети, вместо `localhost` используйте его IP-адрес.

Navidrome совместим с Subsonic/OpenSubsonic-клиентами, поэтому веб-интерфейс — далеко не единственный способ слушать музыку. Телефон, ноутбук или другой компьютер могут подключаться к одному и тому же серверу и видеть общую библиотеку и плейлисты.

А рекомендации?

Возвращаться к папке `Music` и выбирать каждый следующий трек руками было бы слишком жестоким наказанием даже для пост-ИИ-мира. К счастью, музыкальные рекомендации появились задолго до нынешнего бума генеративных моделей.

Navidrome умеет использовать Last.fm как внешний источник метаданных, похожих исполнителей и похожих композиций. Для этого понадобится бесплатный API key и shared secret Last.fm. После их получения добавьте в тот же `/etc/navidrome/navidrome.toml`:

```
LastFM.Enabled = true
```

```
LastFM.ApiKey = "ВАШ_API_KEY"
```

```
LastFM.Secret = "ВАШ_SHARED_SECRET"
```

Перезапускаем сервер:

```
sudo systemctl restart navidrome
```

После этого Navidrome сможет обращаться к Last.fm за информацией о похожих исполнителях и композициях, а при желании пользователь сможет включить `scrobbling` и снова накапливать историю своих прослушиваний. То есть рекомендации вернулись, но для их работы не пришлось запускать языковую модель размером с небольшой дата-центр.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.