

Артём Уваров — портфолио

Artem Uvarov

Артём Уваров — Full-stack software and hardware engineer	1
О себе	1
Что я делаю сейчас	1
Стек	1
Проекты	2
Контакты	2
Умный дом: интеграция с Яндексом и устройства-прослойки	3
Роль и результат	3
Устройства	3
Сопутствующие разработки	4
Технологии	4
Фото	4
Зарядное устройство для литий-ионных АКБ	5
Что сделано	5
Технологии	5
Фото	5
Управление бесщёточными двигателями постоянного тока (BLDC/PMSM)	7
Задача	7
Решение 1. Контроллер на базе китайского драйвера JY01 (99F2)	7
Решение 2. Плата драйвера собственной разработки	7
Прошивка контроллера (общая для обоих решений)	8
Технологии	9
Фото	9
CNC Tools	12
Что сделано	12
Фото	12
Изготовление печатных плат	13
Зачем	13
Фото	13

Артём Уваров — Full-stack software and hardware engineer

IMPORTANT

Портфолио в процессе наполнения

Если вы не нашли нужного примера или хотите узнать подробности о моём опыте, просто [свяжитесь со мной напрямую](#). Я с радостью отвечу на все вопросы и расскажу об интересующих вас деталях!

О себе

Разрабатываю программные и аппаратные решения полного цикла — от печатных плат и ПО для микроконтроллеров, до облачных сервисов и пользовательских интерфейсов.

Основной стек — Java, PostgreSQL и современные веб-технологии. В области embedded-разработки работаю с STM32, ESP32, промышленной автоматизацией, IoT, иногда с силовой электроникой.

Благодаря опыту одновременно в software и hardware согласую все уровни разработки и делаю 'бесшовный' продукт.

Более 15 лет в IT и инженерной разработке. Начинал работать, когда для динамического контента на сайте нужно было создавать отдельное окно. За это время участвовал в создании веб-платформ, промышленных устройств, IoT-систем и специализированного программного обеспечения.

Что я делаю сейчас

Веду продукт по всей вертикали — от схемы и платы до веба и сервисов:

- **Frontend и backend** — сервисы на Java, базы данных, API, серверная инфраструктура.
- **Связанные устройства под ключ** — электроника, прошивка и облачная часть вместе.
- **Мобильные приложения**, не любитель, но с последними фреймворками и наборами готовых библиотек разработка становится приятной прогулкой.

Стек

- Java как web, так и desktop (Swing). Про JEE писать особо смысла нет.
- C/C++ в основном gcc, с web-сервером для клиентской части. Поверхностно знаком с WinApi и QT.
- Web: react, vue (TS) использую как основу front-end. HTML, CSS, JS, jQuery знаю тоже.
- Опыт работы с базами данных PostgreSQL (в основном), MySQL, NoSQL и KV-хранилищами. SQL, PL/pgSQL, любой сложности запросы/процедуры и функции, разработка структуры бд и немного знаком с кластеризацией.

- Умение вести документацию, осуществлять её версионирование.
- Разработка сценариев для автоматического распространения конфигурации серверов.

Могу вести проект один.

Дополнительно: встраиваемые системы и железо

Опыт с железом помогает проектировать backend и API под реальные ограничения устройств — тайминги, обновления, объём телеметрии.

Прошивки

- **ESP32, ARM Cortex-M4/M7 (ST, CH); FreeRTOS.**
- Работа с датчиками и телеметрия в 'реальном' времени, обновление по сети, веб-конфигуратор на устройстве.
- **C/C++** — прошивки и десктоп-инструменты.

Аппаратная часть

- **Altium Designer** — схемотехника, трассировка плат, разработка плагинов.
- Прототипирование SMD-плат (шаг ~127 мкм/мп) для обкатки до заказа производства.
- Подготовка документов, заказ производства.
- **Fusion 360** — корпуса и 3D-моделирование.

Проекты

- Ссылка: [Управление BLDC-двигателями](#) — от собственной платы драйвера и прошивки до сервисных инструментов и мобильного приложения для диагностики.
- Ссылка: [Зарядное устройство для Li-ion АКБ](#) — многоканальная система заряда с дисплеем, диагностикой и выбором тока.
- Ссылка: [Умный дом](#) — интеграция в «Умный дом Яндекса» как производителя устройств: серверная часть (серверы устройств и обновлений по интернету), веб-интерфейсы, мобильное приложение и сами устройства.
- Ссылка: [Утилиты CNC](#) — Разработки для станков.
- Ссылка: [Изготовление плат](#) — прототипирование SMD-плат (шаг ~127 мкм/мп) для обкатки решений на реальном железе до заказа серийного производства.

Контакты

- E-mail: a.uvarov@icloud.com
- Telegram: +79094680218

Умный дом: интеграция с Яндексом и устройства-прослойки

Для одного из заказчиков выполнил интеграцию в «Умный дом Яндекса» как производителя устройств и разработал линейку устройств-«прослоек», которые подключают «не умную» технику к экосистеме Яндекса. Полный цикл: железо, прошивка, интерфейс устройства, сервер устройств и обновлений.

Роль и результат

Интегрировал заказчика (как производителя устройств) в систему «Умный дом Яндекса» — со всеми сопутствующими разработками: интеграция, сервер устройств и сервер обновлений.

Разработал несколько устройств-прослоек между системой умного дома Яндекса и «не умными» устройствами. По каждому устройству вёл полный стек:

- **Аппаратная часть** — схемотехника и плата устройства.
- **Прошивка** — на базе RTOS, платформа ESP32.
- **Графический интерфейс устройства** — веб-интерфейс для настройки (Wi-Fi, режимы настройки и восстановления).
- **Сервер обновлений ПО** — обновление прошивок устройств по интернету по мере публикации новых версий в средах alpha / beta / production.

Связка с Яндексом работает через «Дом с Алисой»: голосовое управление и приложение Яндекса на смартфоне.

Устройства

Модуль для системы контроля протечек воды

Надстройка над сторонней системой контроля протечек: создаёт «прослойку» между ней и «Умным домом Яндекса» — передаёт состояние протечки и управляет кранами. Подключается кабелем в слот расширения, работает по Wi-Fi и не требует отдельного питания.

Модуль управления электроприводом штор

Подключается в разрыв провода питания двигателя штор (или отдельным кабелем) и через разъём brcb непосредственно для управления. Интегрируется с «Умным домом Яндекса» — открытие/закрытие штор голосом через Алису или из приложения.

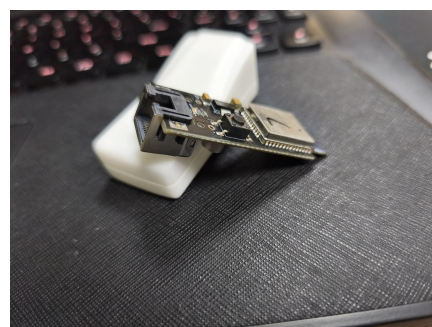
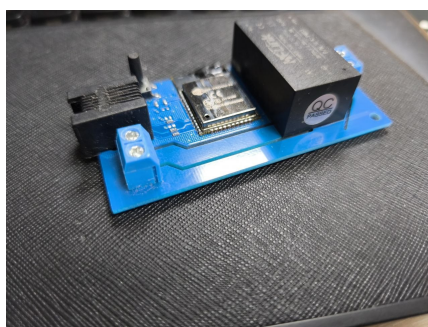
Сопутствующие разработки

- **Сервер устройств** и интеграция производителя в облако Яндекса.
- **Сервер обновлений** с каналами alpha / beta / production для поэтапной выкатки прошивок.
- **Мобильное приложение** (Android) — мастер настройки и пусконаладки устройств.
- Веб-интерфейс настройки на устройстве (режим точки доступа).

Технологии

- RTOS, ESP32, Wi-Fi.
- Обновление прошивок по интернету, разделение сред alpha/beta/production.
- Интеграция с «Умным домом Яндекса».
- Схемотехника и трассировка плат устройств.

Фото



Зарядное устройство для литий-ионных АКБ

Модульное зарядное устройство для литий-ионных аккумуляторов: от силового модуля заряда до многоканальной системы с дисплеем, диагностикой и выбором тока.

Что сделано

Зарядный модуль

Разработал модуль зарядного устройства с мощностью заряда до **90 Вт – 120 Вт** (требуется радиатор). Теоретически 160 Вт на модуль - штатный, рабочий режим цикла с адекватным охлаждением, но реальных тестов не производил, т.к. запрос по мощности был 2 А, максимум 4 А для увеличенных сборок на 6 и более А·ч. Пакет модулей для 8 АКБ, где всего два посадочных места имеют возможность зарядки до 6 А, а остальные 2 А - в пике загрузки потребляют 600 Вт.

Силовая управляющая плата

Разработал силовую-управляющую плату на **5 таких модулей**, а также плату управления на **две силовые платы**.

Интерфейс и диагностика

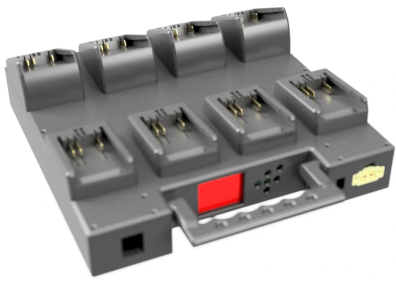
Дисплей отражает **8 посадочных мест** для аккумуляторов с индикацией хода зарядки, возможностью упрощённой диагностики каждого АКБ и выбора тока зарядки для двух посадочных мест — **2 / 4 / 6 А**.

Технологии

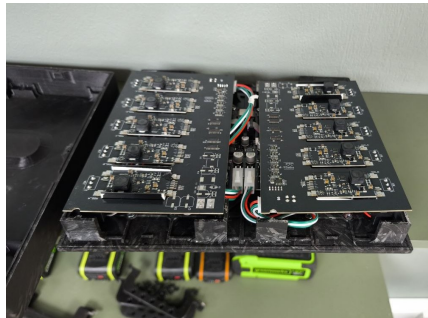
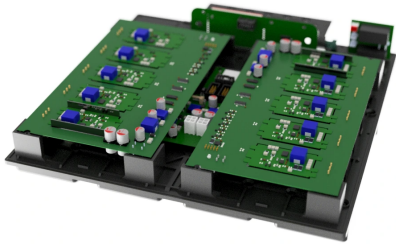
- Силовая схемотехника и трассировка плат (зарядный модуль, силовая и управляющая платы).
- Многоканальное управление зарядом, индикация и диагностика.
- RTOS, ESP32.

Фото

Общий вид



Внутренности



Управление бесщёточными двигателями постоянного тока (BLDC/PMSM)

Контроллеры и прошивки для управления бесщёточными двигателями: от адаптации готового драйвера до собственной платы с полноценным FOC и трёхфазным контролем тока.

Задача

Реализовать надёжное управление мотор-колёсами (в конфигурациях на 2 и 4 колеса): плавный пуск и торможение, контроль тока и температуры, аварийные защиты, удобную механику монтажа и укладки проводки.

Решение 1. Контроллер на базе китайского драйвера JY01 (99F2)

Получил в работу плату драйвера JY01 (99F2): требовалась разработка платы-контроллера и прошивка, чтобы приручить этого китайского 'драйвера'.

Что сделано:

- Разработана плата-контроллер для управления драйверами 99F2 (JY01).
- Реализовано ПО для асинхронного управления двигателями на основе PID-алгоритма.
- Продумана система монтажа и укладки для 2/4 колёс.

Результат. Рабочее, быстрое в сборке решение на доступной элементной базе. Опыт и ограничения этого варианта легли в основу собственной платы драйвера — см. [Решение 2](#).

Решение 2. Плата драйвера собственной разработки

Хотя изначально я был против первого решения, его доработка дала немало опыта. Собственную плату драйвера спроектировал «с нуля», учтя ограничения варианта на готовом драйвере ([Решение 1](#)): переработан токовый контур, заложены трёхфазный контроль тока и совместимость с полноценным **FOC**.

Что сделано:

- Переработал токовый контур, выбрал оптимальные компоненты под задачу.
- Заложил трёхфазный контроль тока.
- Совместимость и с «лёгким управлением» (для быстрой адаптации прошивки от Решения 1, чтобы не терять наработки), и с полноценным FOC-решением.
- Датчики контроля температуры и error-флаг: при перегрузе или неисправности колёс —

мягкое завершение работы, чтобы не повредить электронику.

Прошивка контроллера (общая для обоих решений)

Прошивка контроллера — на текущий момент **общая для обоих решений**: управление колёсами с обратной связью, телеметрия, обновление по сети и защиты. Вокруг неё — сервисные инструменты и мобильное приложение.

Возможности:

- Плавный пуск, торможение и регулирование скорости по каждому колесу с обратной связью по датчикам.
- Координация колёс в поворотах (конфигурации на 2 и 4 колеса).
- Телеметрия в реальном времени для сервисных инструментов и мобильного приложения.
- Обновление прошивки по сети.
- Самодиагностика и защиты: обработка аварийных ситуаций, безопасное завершение работы при перегрузе или неисправности.
- Точка доступа Wi-Fi и веб-настройка устройства.

NOTE

Железо рассчитано и под FOC, и под простое управление по ШИМ. Текущая прошивка использует простое управление по ШИМ и токи не контролирует.

Сервис и диагностика устройств

Два инструмента для обслуживания и диагностики: онлайн-станция для обновления и снятия телеметрии в поле и офлайн-анализатор для разбора логов.

Сервисная станция:

- Обнаружение устройств и подключение к ним.
- Массовое обновление прошивки множества устройств за один прогон с прогрессом по каждому.
- Снятие телеметрии и доступ к настройкам устройства.
- Поставляется одним исполняемым файлом, без установки.

Анализатор телеметрии:

- Загрузка логов телеметрии и наглядные графики.
- Наложение и сравнение нескольких прогонов, базовая статистика.
- Рабочее место для разбора поведения контроллера.

Мобильное приложение

Приложение-компаньон для устройства: пусконаладка, телеметрия в реальном времени, диагностика и самообновление.

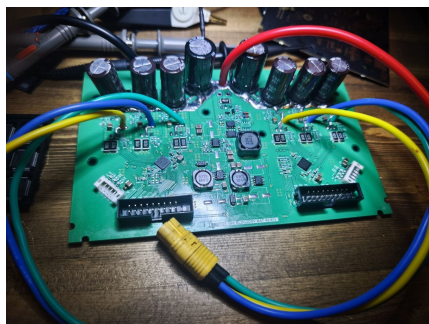
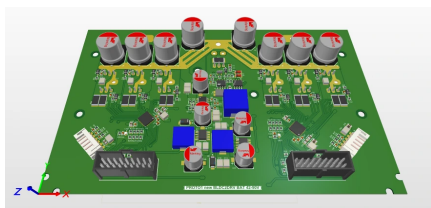
- Пусконаладка и настройка устройства со смартфона.
- Телеметрия в реальном времени: дашборд по колёсам с подсветкой аномального колеса.
- Логирование сессий и выгрузка для анализа.
- Самообновление приложения.

Технологии

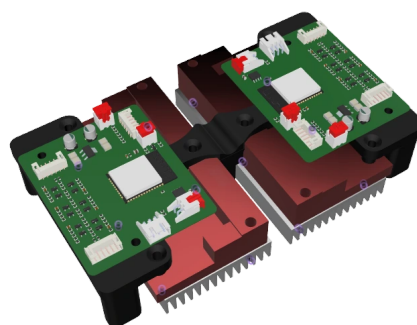
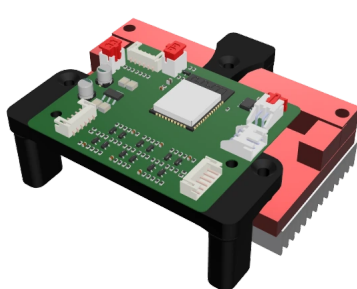
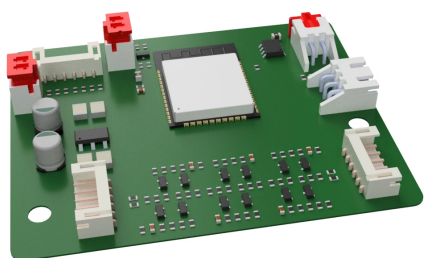
- Схемотехника и трассировка платы контроллера / драйвера.
- Токовый контур, трёхфазный контроль тока.
- Алгоритмы управления: PID, 1 | 6-step, FOC.
- Защиты: контроль температуры, error-флаг, мягкое завершение при перегрузке.

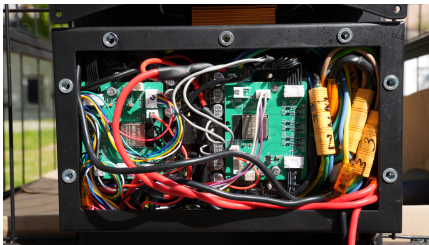
Фото

Решение 2 — плата драйвера собственной разработки



Решение 1 — контроллер на драйвере JY01 (99F2)





Сервис и диагностика — анализ телеметрии



Мобильное приложение — калибровка / детектор отклонений

18:07

7.94 50% 4G+

← Настройки устройства

Калибровка колес [← к настройкам](#)

PWM ML (DRIVE A + BRAKE C)

A: частота, Гц: 20000

A: duty, 0..1000: 333

C (brake) duty: 0

PWM MR (DRIVE B + BRAKE D)

B: частота, Гц: 20000

B: duty, 0..1000: 338

D (brake) duty: 0

КОЛЕСО ML

FAULT (active-LOW) OK

DIR (фактич./конф.): LOW wrd:0

Hall raw (U/V/W → state)

U/V/W уровни: U:0 V:1 W:1

Частота, Гц: 54

Частота ×100: 5413

Рубер всего: 33029

Глитчи: 197

Направление: -1 (назад)

State (компл.): 2

Ветви переходов

vfwid / vrev: 329 / 32700

graymm / slinv: 4 / 193

ssame / deb: 671723 / 657

КОЛЕСО MR

FAULT (active-LOW) OK

DIR (фактич./конф.): LOW wrd:1

Hall raw (U/V/W → state)

U/V/W уровни: U:0 V:0 W:1

Частота, Гц: 54

Частота ×100: 5432

Рубер всего: 32602

Глитчи: 238

Направление: +1 (вперед)

State (компл.): 5

Ветви переходов

vfwid / vrev: 32453 / 149

graymm / slinv: 8 / 230

ssame / deb: 1653845 / 260

Мобильное приложение — настройки

Настройки устройства

-

+

750

Breaks and Waiting settings:

Brake On Zero Throttle
(тормозим при отпущенном курке):
Waiting to Stop Engine (ждем остановку двигателя):

... 'Power Up' (старт системы):

☐ On/Off

... 'Rotation Dir Changed'
(смена направления):

☒ On/Off

Waiting Zero Throttle (ждем отпущенный курок):

... 'Power Up' (старт системы):

☐ On/Off

... 'Rotation Dir Changed' (смена направления):

☐ On/Off

R F | Загрузить Сохранить Готово

Параметры колес

Для каждого колеса задайте диаметр, число зубьев, тормозное усилие и направление.

#	Место	Диаметр	Кол.зубов	Торм.усилие	(-)Коррекция
Unit ID: ty01333TVPR45OKUC2QI9E6K38R (Local)					
1	1	250	27	1000	0
2	2	250	27	1000	0
Unit ID: ty014YVKKFFNWC7GGNQ61TQE0HSL (Remote)					
3	3	0	0	0	0
4	4	0	0	0	0

R F | Загрузить Сохранить Готово

CNC Tools

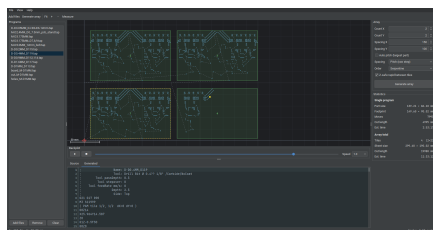
Что сделано

- CNC Tool: PAM (Program Array Manager):

Дублирование программы G-кода по рабочей площади

Фото

Общий вид



Изготовление печатных плат

Прототипирование печатных плат собственными силами: изготавливаю и обкатываю SMD-платы (шаг ~127 мкм), чтобы проверить решение на реальном железе до заказа серийного производства.

Зачем

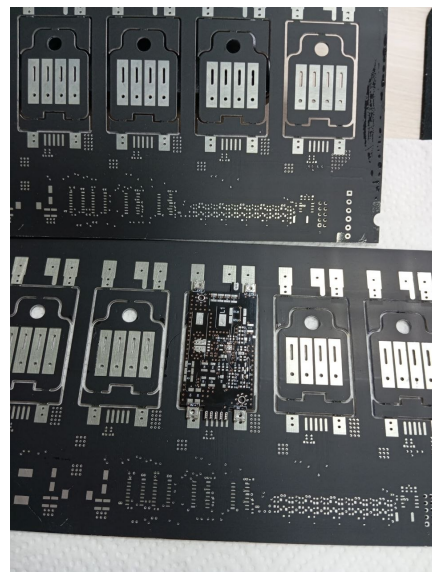
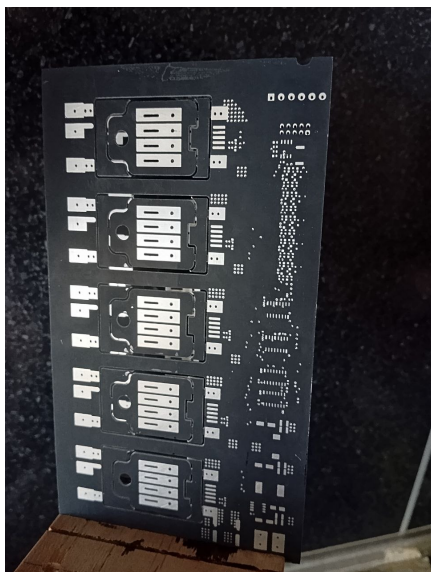
- Быстрая итерация: от схемы до рабочего прототипа без ожидания внешнего производства.
- Обкатка платы и монтажа до заказа серии — ловлю ошибки на прототипе, а не в партии.
- Контроль качества на каждом этапе: от подготовки и травления до пайки SMD и финальной проверки.

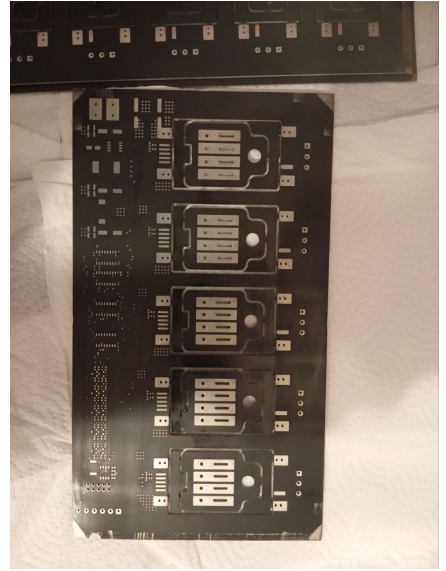
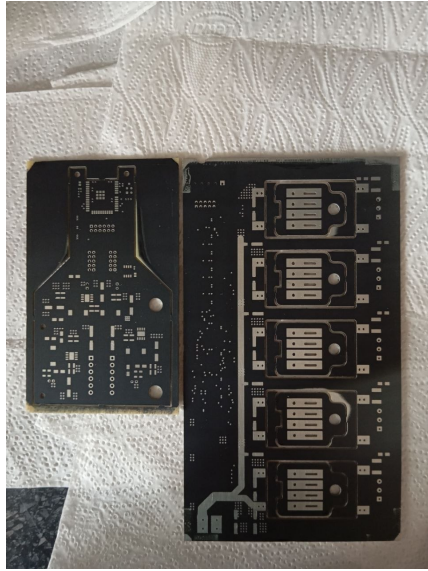
NOTE

Бывают и неудачи — это часть процесса обкатки: поймать проблему на прототипе дешевле, чем в серии.

Фото

Изготовление и монтаж





Бывают и неудачи

