/* REVERSE ENGINEERING */

Инженер по Reverse Engineering

Веретененко Антон Николаевич

РЕЗЮМЕ

Инженер по Reverse Engineering, специализируюсь на встраиваемых системах и недокументированном железе. Если datasheet нет — сначала строю инструмент: модели машин в QEMU, дамперы прошивок, анализ в Ghidra + MCP, доказательства с логического анализатора и осциллографа. Более 18 лет в embedded, прошивках и проектировании плат; восстановил ~3 900 функций из вендорского mask-ROM монолита и сделал целую стоковую прошивку наблюдаемой в эмуляции: я управляю AI-агентом, который работает с Ghidra, и проверяю каждое утверждение.

ОПЫТ

Общий инженерный стаж — более 18 лет.

Инженер по Reverse Engineering (AI-first) 2026 – по настоящее время

NDA

- Цель: SoC Smartlink SL6801 (класс ARM Cortex-M4F, Thumb-2/VFP) в MP3-плеерах Jointbees / SMTLINK / YP3 — экран 128×160 RGB565 по SPI/DBI (класс GC9106/ST7735S), тач CST726/CST816D по I²C, microSD через SDIO, встроенный кодек KADC/DAC, 5-кнопочная ADC-лестница плюс PMU-кнопка питания, Bluetooth на кристалле. Нет публичного datasheet, SDK и карты регистров: единственный достоверный код — mask ROM ~0x5C000.
- Сначала построил инструмент, потом анализ: модель машины в QEMU 8.2 (emulator/qemu/smartlink_sl6801.c) — окна CPU/SRAM, mask ROM и 2 МиБ SPI NOR с XIP (класс Puya P25Q16), LCD-контроллер с DCS-парсером (CASET, RASET, RAMWR), тач, SD/MMC с FatFs, аудио-DMA/кодек с PCM-тапом в хост, key ADC, PMU/CPLD, RTC и подсветка; плюс Python-GUI с живым framebuffer, инъекцией тача/кнопок, переключателями железа, записью звука и скриншотами.
- Восстановил контракт загрузки: mask ROM → загрузчик HLKJ (RAM 0x00804C00, вход 0x00820001) → образ FIRM с заголовком, таблицей разделов, CRC16 и блоком RAM 4 КиБ → XIP-приложение. Стоковый образ — единый монолит, где UI, медиа и Bluetooth перемешаны, отдельного SDK-раздела нет; на диске он не меняется — только volatile-патчи в RAM (wakeup ViewTask / context-2, заглушки записи в DSP) позволяют однопоточной модели его выполнить.
- Решил сложные места: wakeup context-2 / CPU1, которых однопоточная модель не даёт; аудиотракт (дескрипторы DMA, регистры кодека, заглушки DSP), без которого воспроизведение встаёт на 75 % или даёт тишину; ADC-лестницу кнопок плюс PMU-кнопку питания, без «space available» wakeup которых менеджер страниц уходит в дедлок; и вендорский Bluetooth-стек, общающийся с контроллером на кристалле через проприетарное окно регистров.
- AI-first цикл анализа: я управляю LLM-агентом, который работает с Ghidra 12.2 через форк HTTP-сервера OGhidraMCP — /disassemble, /decompile_function, /create_function, /create_data, /undefine, /create_label, /variables, /data_types, /set_comment, /basic_blocks, /call_graph, /search_bytes, /bookmark, /batch, /undo, /redo, /save_program (добавлено 18 эндпоинтов). Деструктивные операции требуют confirm=true, записанной причины и конкретных доказательств; каждое утверждение проверяется по дампу, эмулятору или устройству.
- Починил инструменты там, где они отказывали: форк smartlink_flash (таймауты/повторы USB, dump_chunks.py с чанками 64 КиБ, продолжение и --compare, flash_ranges.py, verify_flash.py), потому что загрузчик перестаёт отвечать примерно после 30 команд, а на 2 МиБ нужно 512+; и форк OGhidraMCP для создания функций/данных, undo, пакетных правок и save_program.

- Результат: стоковая прошивка полностью загружается и наблюдаема в эмуляции (UI, медиа, Bluetooth), в базе Ghidra размечено ~3 900 функций, задокументированы 22 фазы step-* и стадии эмулятора M0–M7, опубликованы два форка инструментов, десктопная сборка — AppImage в один клик (Electron + PyInstaller). Стек: QEMU, Ghidra, MCP/LLM, ARM Cortex-M4F, Embedded C, Python, FreeRTOS/LVGL/FatFs (наблюдаемые), Docker, GNU Make.

Embedded-разработчик (Фриланс) Янв 2014 – Дек 2019 (6 лет)

NDA

- Reverse Engineering устройств и прошивок, восстановление протоколов взаимодействия
- Эмуляторы/симуляторы узлов на основе оригинала; различные конвертеры шин
- Разработка электроники на заказ; разработка печатных плат
- Программирование микроконтроллеров, в основном MSP430, AVR, Cortex-M
- Датчики с беспроводным интерфейсом Bluetooth 4 / LE / BLE / Smart / 5; фреймворк MBED

Ведущий инженер Янв 2008 – Янв 2014 (6 лет 1 месяц)

NDA

- Автоматизированная система тестирования торговых роботов, скрипты на Windows и Linux; визуализация результатов и генерация отчётов
- Плагин для MT4 на C-подобном языке
- Бэкенд на PHP/Symfony и фронтенд HTML5/Canvas/JavaScript

НАВЫКИ

Reverse Engineering

Ghidra, Декомпиляция, Дизассемблирование, Xrefs, Граф вызовов, Восстановление функций, Типы данных, Анализ патчей, Статический анализ, Динамический анализ, Дамп прошивки, Анализ загрузчика, Восстановление протоколов

Эмуляция и инструменты

QEMU, Собственная модель машины, GDB, Снятие framebuffer, PCM-тап звука, SDIO/MMC, I²C, SPI, DMA, Логический анализатор, Осциллограф

Железо и прошивки

ARM Cortex-M4F, Thumb-2/VFP, Mask ROM, SPI NOR XIP, USB Boot ROM, FreeRTOS, LVGL, FatFs, Embedded C, Python, GNU Make, Docker

AI-assisted процесс

MCP, LLM-агенты, Промпт-инжиниринг, Форки и расширения инструментов, Проверка на доказательствах, Пакетная автоматизация, Git

СЕРТИФИКАТЫ

2012

MITx 6.002x: Схемы и электроника

MIT (edX)

Проверить ↗ (https://verify.edx.org/cert/317037ee95ed49ebb6142ef20f527706)

PDF ↗ (https://veretenenko.ru/anton/mitx-6002x-certificate.pdf)

2014

MITx 6.00.1x: Введение в информатику и программирование на Python

MIT (edX)

Проверить ↗ (https://verify.edx.org/cert/07e1b2377c044bb8aec8b55c19418eb2)

2014

MITx 6.01x: Введение в EECS I — «Shape the World»

MIT (edX)

Проверить

ЯЗЫКИ

Русский — Родной, Английский — B2 (Выше среднего)

Контакты

ПОЧТА
anton@veretenenko.ru

ТЕЛЕГРАМ
@egaistek

GITHUB
github.com/anton-veretenenko