

Rajveer Chaudhari

Firmware & Low-Level Systems | B.Tech CSE, IIIT Vadodara

+91-6355395811 • rajveerchaudhari.com • rajveer@rajveerchaudhari.com • github.com/Rajveer1496 • Linux kernel patches

EDUCATION

IIIT Vadodara – International Campus Diu

B.Tech, Computer Science & Engineering CPI: 7.07/10

Aug 2024 – May 2028

Diu, India

Relevant Coursework: Operating Systems, Computer Organization, Digital Logic & Electronics, Computer Networks

PROJECTS

MetalBird – FPV Drone Flight Controller ↗ | STM32F411, ARM Thumb-2 asm, C, KiCad

Feb 2026 – Present

Custom flight-controller PCB and bare-metal firmware – no dev board, no HAL, no RTOS

- **Firmware:** hand-written Cortex-M4 assembly with a custom vector table, startup code and linker script; HSE → PLL clock-tree setup, SysTick timing, a USART debug console with register-dump helpers, and a register-level I2C master driver (ACK-failure handling, multi-byte reads) that brings up the MPU-6050 IMU.
- **Hardware:** designed the 2-layer board in KiCad: MP2331H buck (7.4V→5V) plus AMS1117 3.3V LDO, PMOS reverse-polarity protection, MPU-6050 and BMP280 on I2C, ELRS receiver, four ESC outputs, camera/VTX and SWD headers. Exported Gerbers and had the v1 board fabricated; a v2 redesign around a 4-in-1 ESC is in progress.
- **Bring-up & debug:** set up wireless flashing and GDB debugging over SWD through an ESP32 running CMSIS-DAP with OpenOCD. Wrote separate C diagnostic firmware that streams the IMU FIFO at 1 kHz over a CRC-framed 1 Mbaud UART link to a Python/3D host viewer, and used it to trace bad sensor readings to a faulty IMU.

PenguOS – x86 Operating System from Scratch ↗ | C, x86 asm (NASM), QEMU

Dec 2025 – Present

Bootable 32-bit OS with no external libraries

- **Boot & CPU:** two-stage bootloader (real→protected mode, GDT, kernel load from disk); IDT with ISRs for all 32 CPU exceptions and remapped PIC hardware IRQs.
- **Memory:** bitmap physical page allocator, two-level paging and `sa1loc`, a custom kernel heap allocator.
- **Drivers:** PS/2 keyboard (Shift/Caps/Ctrl) and mouse (3-byte packets), ATA PIO disk with a sector-bitmap file store, PIT, x87 FPU and UART serial debug logging.
- **Scheduling & graphics:** PIT-driven preemptive context switching; VGA mode 13h and VBE linear-framebuffer graphics with double buffering; a software 3D engine that parses `.obj` models and renders them with perspective projection and rotation.

OPEN SOURCE

Linux Kernel – IIO & Staging Subsystems ↗ | C, git send-email, checkpatch

Feb 2026 – Mar 2026

Contributor – 4 commits in mainline Linux | patch archive

Mainline

- Converted manual `mutex_lock()/mutex_unlock()` pairs to scope-based `guard(mutex)` in IIO accelerometer drivers, removing goto-based unlock paths so every error path releases the lock by construction. `adx1313` and `adx1372` were applied by the IIO maintainer and are in mainline.
- Took a follow-up series for `sca3000`, `bmc150`, `mma8452` and `mma9551` from v1 to v4 on `linux-iio`, reworking error paths to return directly and splitting header re-ordering into separate patches (not merged).
- Removed unused `#includes` in staging drivers: `most/dim2` and `most/net` were applied by Greg Kroah-Hartman and are in mainline; `media/imx` and `axis-fifo` were sent.

TECHNICAL SKILLS

Languages: C, ARM Thumb-2 assembly (GAS), x86 assembly (NASM), Python, Bash

Embedded: STM32F4 / Cortex-M4 at register level: RCC/clock tree, GPIO, USART, I2C, SysTick; startup code & linker scripts

Hardware: KiCad schematic & PCB layout, power regulation (buck + LDO), datasheet & reference-manual driven bring-up

Debug & Tools: GDB, OpenOCD, SWD / CMSIS-DAP, QEMU, arm-none-eabi, GCC, Make, Git

Kernel: x86 protected mode, paging, interrupts; Linux IIO subsystem; mailing-list patch workflow