

Manish Soni

Senior Embedded Linux & Firmware Engineer, Gurugram, India

soni.97man@gmail.com

+91 85270 25409

0mani.github.io

github.com/0mani

I write the firmware between the silicon and the product. About 7 years of production work across medical training devices, carrier optical networking, power-grid signal processing and telephony, from bare-metal MCUs through FreeRTOS to embedded Linux and UEFI.

Experience
~7 years

Worked with
Ciena, Cisco, Micatu

Industries
**Medical, telecom,
energy**

Availability
Immediate start

Core skills

Languages	C, C++ (modern C++, STL, design patterns), Python, Bash
MCUs and SoCs	ESP32 / ESP32-S3, STM32 (Cortex-M), AVR ATmega, NXP KE-02Z, ARM Cortex-A
RTOS and OS	FreeRTOS, ESP-IDF v5.x, Zephyr (exposure), embedded Linux, Wind River real-time Linux
Connectivity	BLE (GAP, GATT, pairing), Wi-Fi, MQTT, TCP/UDP, NFC
Peripherals	I2C, SPI, UART, GPIO, ADC, DMA, RMT, timers, interrupts
Linux platform	Yocto / BitBake recipes, bbappends and layers; UEFI (AMI Aptio V), BIOS, iPXE, secure boot; eBPF / libbpf; OpenWrt; IPC and Pthreads
Control	Field-oriented motor control, SVPWM, PI tuning; MATLAB / Simulink
Debug and test	GDB, JTAG/SWD, QEMU, oscilloscope, logic analyser, CMocka, pytest, Python BLE test tools
Quality and process	MISRA C, Coverity, Jenkins and GitHub Actions CI, Git, JIRA, Scrum

Projects

Open source

TP1

ebpf-drvlat: driver latency tracer for embedded Linux

2026, github.com/0mani/ebpf-drvlat, eBPF, libbpf CO-RE, C

- Log2 latency histograms for hard IRQs, softirqs, threaded-IRQ wake-up, I2C transfers and SPI messages.
- Tracepoints only, so it runs on production kernels without kprobes; per-CPU and per-thread start maps keep overhead low.
- Slow-event log through a ring buffer, JSON Lines output, and CI that builds it and loads it into a live kernel on every push.

M2

foc-motor-control: portable FOC library with closed-loop simulator

2026, github.com/0mani/foc-motor-control, C99, Python, Cortex-M4F

- Speed loop, dq current loops, decoupling feed-forward, min/max-injection SVPWM and anti-windup; about 2 KB of flash on Cortex-M4F.
- Model-based auto-tuning in firmware that rejects unsafe bandwidths.
- Verified against a Python PMSM model with encoder quantisation, sensor noise and a one-cycle delay: start-up to 2000 rpm in 17.6 ms with 1.9 % overshoot; decoupling cuts the flux-current disturbance from 0.20 A to 0.07 A.

U1

CPR training manikin

ESP32-S3, FreeRTOS, ESP-IDF, BLE, ToF

Time-of-flight sensor measures compression depth and rate; an airflow sensor on the artificial lungs measures ventilation. Multi-task FreeRTOS firmware streams live data to the training app over BLE and handles its control commands in real time.

L1

Auscultation trainer

ESP32-S3, FreeRTOS, NFC, audio IC, BLE

NFC tags identify where the stethoscope is placed on the chest; the firmware plays the matching heart or lung sound through an audio IC. Scenarios are selected and controlled from the app over BLE.

D1

Anatomical-zone RGB LED controller

Dual-core ESP32, ESP-IDF v5.2, FreeRTOS, RMT, BLE GATT, ADC

- Drives 33 WS2812 LEDs across heart, anterior lung, posterior lung and bowel zones from the app.
 - Replaced a multi-packet text protocol with a binary format (4-byte header with CRC8, 4 bytes per LED): a full frame is 136 bytes and fits one GATT write at MTU 185.
 - Core-pinned task design (BLE and system tasks on core 0, LED flush and OTA on core 1) with double-buffered RMT output driven by semaphores.
 - Found and fixed ADC2 readings corrupted by the BLE radio, a watchdog reset from a missing yield, and polled BLE state; built Python BLE test tools.
-

U4

Shared firmware platform and release support

ESP32, DMA, OTA / bootloader, QA

One hardware-abstracted ESP32 codebase for several training products. DMA offload and a dual-core task split for real-time throughput; fixed a bootloader overflow by moving the partition-table offset in ESP-IDF v5.2; regression-tested the companion app across BP, auscultation, CPR and ECG modules.

M1

BLDC FOC speed-control proof of concept

MATLAB, Simulink, delivered to client, Oct 2026

Simulation model of field-oriented speed control for a BLDC motor, with an agentic workflow that automates controller parameterisation and tuning. Completed and shared with the client for their project decision.

J1

RLS-OSRP Layer 1 control plane

Linux, C/C++, MISRA C, Coverity

- Developed and maintained L1 control-plane components for carrier optical transport: channel provisioning and software-upgrade paths.
- Built an RPC-based test-injection mechanism that bypasses full network configuration, sharply cutting bug-reproduction time.
- Tested protocols on the PINS RLS simulator without hardware; wrote upgrade sanity tests; cleared Coverity findings before every monthly release.

U2

Cisco NCS1004 / NCS1K platform firmware

Embedded Linux, Yocto, IOS-XR, UEFI, OpenWrt

- Integrated AMI Aptio V UEFI updates, applied Intel-recommended vulnerability fixes, and validated GPIO, memory clock and DMA on hardware over JTAG.
- BIOS and iPXE network boot, secure boot and firmware encryption; Yocto image, build and boot-time optimisation.
- Wrote and modified Yocto recipes and bbappends and created layers to integrate applications; multi-process services with IPC and Pthreads; FPGA/PCIe access through Linux system and socket programming.
- CMocka unit tests that raised coverage in Jenkins CI before merge.

Chetu India, Software Engineer (C/C++), client Micatu (USA), Nov 2021 – Feb 2023

U3

M410B grid data processor

ARM Cortex-A, Wind River real-time Linux, embedded C, DSP

Ported Python/NumPy harmonic and noise filters, including temperature compensation, to optimised C for an optical-sensor power-grid processor. Validated first on QEMU and then on hardware. Also wrote ARM Cortex-M I2C, SPI and UART drivers.

Earlier roles, 2020 – 2021

U5

Telephone exchange product line and 32-zone alarm panel

Copper Connections, Oct 2020 – Nov 2021, AVR, STM32F103, NXP KE-02Z, RTOS

Firmware for a 4-junction, 32-extension exchange, a PAX, a digital PBX and key telephone sets, with SPI, UART and I2C drivers; plus a commercial burglar alarm with 32 wired zones.

U6

Pulse monitor and touch-free dispenser

LarkAI Healthcare, Jan – Oct 2020, STM32F103, AVR ATmega

SPI and I2C sensor drivers, interrupt routines and data processing for a pulse monitor; IR and ultrasonic sensing firmware for an automatic hand dispenser.

Personal

J2

Raspberry Pi 3 kernel and custom Linux (in progress)

Cortex-A53, Linux kernel, Yocto, QEMU

Kernel and device-driver development on the board, and a custom Linux image built with Yocto, tested on QEMU and hardware.

Working together

Engagement

Fixed-scope projects, monthly retainer or hourly contract. Remote worldwide, or on site in Delhi NCR.

Availability

Immediate start.

Education

B.Tech ECE, Bharati Vidyapeeth's College of Engineering (GGSIPU), 2019. Diploma ECE, Guru Nanak Dev Polytechnic, 2016.