

Nilesh Warke

✉ warkenilesh3172@gmail.com ☎ 8010804458 🌐 LinkedIn

👤 PROFILE

Enthusiastic engineer with hands-on experience in embedded firmware development for smart electric panels and storage systems using C and C++. Skilled in RTOS, communication protocols (I2C, SPI, UART, CAN), and Modern CMake. Experienced in designing device features, debugging complex system interactions, and improving firmware efficiency in RTOS and Linux environments. Highly motivated, quick to learn new technologies, and passionate about building reliable, scalable embedded solutions.

📁 PROFESSIONAL EXPERIENCE

Software Engineer - II

Jun 2024 – Present

RSL/Ness [🔗](#)

- Partnered with the client to design and develop robust firmware **solutions for a smart electric panel**, enabling advanced remote energy management and detailed consumption analytics.
- Refactored and modernized a **legacy C codebase into a scalable C++ architecture**, streamlining the build system and **achieving 4–5x faster build times**.
- Collaborated with QA and cross-functional teams (mobile, backend, customer support) to identify and resolve complex defects, improving overall product stability.
- Stabilized intermittent sensor **data sampling, processing, and publishing**, significantly enhancing system reliability and performance. During this effort, introduced **stack and thread monitoring** features.
- Designed and deployed new device features, including LED and button modules, expanding functionality and enhancing user interaction.
- Resolved **flaky OTA (Over-the-Air) test cases**, gaining deep expertise in OTA processes and test automation.
- Currently focusing on **OS-level optimizations** to maximize firmware efficiency and performance.

Firmware Intern

Jan 2024 – Jun 2024

Seagate Technology [🔗](#)

- Gained a strong understanding of the **SAN-based product architecture** and product lineup, effectively addressing diverse client needs.
- Built a Python-based tool to improve **data visualization and create a single platform for older tools**. Enhanced the tool's functionality by fixing bugs and resolving legacy issues.

🎓 EDUCATION

Walchand College Of Engineering, Sangli

2020 – 2024 | CGPA : 8.44

B.TECH, Electronics

Vyankatrao High School & Jr. College, Ichalkaranji

2018 – 2020 | 82.92 %

Higher Secondary Certificate

Borawade Vidyalay, Borawade

2017 – 2018 | 95.80 %

Secondary School Certificate

PROJECTS

C++ Migration & Build System Modernization

C, C++, CMake, Python, Makefiles, Shell

- Migrated core embedded application from **C to C++ (OOP)**, refactoring ~600 lines of code from a single file into 10 modular .hpp/.cpp files, with the C++ version now in use and the legacy C version deprecated.
- Despite lack of hardware access, verified code flow through rigorous static analysis and delivered a critical ship-blocker on time.
- Replaced legacy CMake with Modern CMake, enabling **incremental builds and 4–5x faster build times**, introduced *CMakePresets.json* to **centralize toolchain configuration and build customization**, eliminating reliance on Python, shell scripts, and Makefiles.

LED Modes & Button Feature Implementation

C++, SPI

- Designed and implemented multiple **LED operation modes and button features** for an SPI-connected module with 4 LEDs and 1 button.
- Developed advanced button interaction logic supporting **long-press** (3s, 5s) and **multi-press** (3×, 5×) patterns to trigger predefined device functions.
- Implemented **priority-based LED** modes and resolved **button debounce issues** to ensure reliable input detection.

Data visualization tool

Python

- Developed a debugging tool to parse and **decode communication across hardware and Linux services** (TCP/UDP via tcpdump, CAN bus, and system logs via systemctl/journalctl).
- Visualized captured data on a web tool with graphs showing state changes across hardware modules, protocols, and key–value interactions, enabling quick detection of data corruption and bugs.

SKILLS

Programming Languages: C, C++, Python, CMake, Make

Communication Protocols: I2C, SPI, UART, CAN

Operating Environment: Windows, Linux, macOS, RTOS

Version Control / Bug Tracking: Git, GitHub, JIRA

COURSE & CERTIFICATION

- | | |
|---|--|
| • Data Structures and Algorithms | • Object Oriented Programming |
| • Operating System | • SDLC |
| • Problem Solving  | • Embedded C  |