NGUYEN CONG THANH

Embedded Software Engineer

(+84) 372 788 796 | congthanh.ml.2509@gmail.com | Hanoi
LinkedIn: LinkedIn.com/in/nguyencongthanh25 | GitHub: github.com/thanhnc25 | Portfolio: nguyencongthanh.id.vn

---

EDUCATION

Posts and Telecommunications Institute of Technology (PTIT)
Major: Electrical and Electronics Engineering | 09/2022 – 03/2027 (Expected)

- Relevant coursework: Embedded Systems, Microprocessors, Data Structures & Algorithms, Computer Architecture, Computer Networks.
- English: TOEIC 775/990.

TECHNICAL SKILLS

- Languages: C/C++, Python (Scripting/Tools).
- Communication protocols:
  - Standard: UART, SPI, I2C, TCP/IP, MQTT, HTTP/HTTPS.
  - Automotive: CAN (ISO 11898), UDS (ISO 14229 subnet), CAN-TP (ISO 15765-2).
- Embedded OS & Frameworks: FreeRTOS, ESP-IDF, STM32 HAL/SPL/Register-level, Arduino.

KEY PROJECTS

1. Secure Automotive FOTA System | Independent R&D Project | 12/2025 – 02/2026

Tech Stack: C/C++, STM32, ESP32, FreeRTOS, CAN-UDS, Wi-Fi, Ethernet, Security (SHA-256, ECDSA, AES), SquareLine Studio.

- Designed and implemented a secure bootloader with pre-boot authenticity and integrity checks: Validates ECDSA P-256 signatures and SHA-256 hashes at startup to prevent tampering and unauthorized code execution.
- Implemented the core UDS/CAN-TP stack for both ESP32 (Client) and STM32 (Server), managing packet segmentation and flow control (STmin, Block Size) to ensure reliable firmware transport.
- Implemented an end-to-end "verify-then-decrypt" pipeline (ECDSA P-256, AES-128-CBC), supported by a custom Python desktop app for automated firmware packaging and signing.
- Developed an ESP32 gateway (with LVGL UI) managing dual-interface updates: Remote OTA via Wi-Fi (HTTPS) and local deployment via Ethernet, utilizing UDP broadcasts for dynamic device discovery and TCP for reliable payload transfer.
- Optimized STM32 RAM by implementing stream-based AES decryption and chunked SHA-256 processing, eliminating the need for full-firmware buffering.
- Ensured system reliability with read-back verification and strict 4-byte alignment checks during the dual-slot copy process to prevent flash corruption.

2. Low-Power Digital Clock | Coursework: Embedded Systems | 10/2025

Tech Stack: C++, STM32, FreeRTOS, SPL.

- Applied C++ OOP to wrap hardware drivers (RTC, LCD, Encoder), making the code cleaner and easier to manage.
- Handled low-power modes by programming the system to enter STM32 Standby Mode automatically after 10s of inactivity to save battery.
- Used FreeRTOS to manage display tasks and user inputs efficiently without blocking the CPU.

OTHER PROJECTS

- IoT Weather Station (ESP32, MQTT, FreeRTOS): Built a multi-sensor data logger with cloud telemetry, operating stably for 10 months.
- Smart Home Network (ESP-IDF, C++): Developed customized IoT devices leveraging BLE Mesh networking and ESP Rainmaker framework.
- Laser Engraver Automation (STM32): Controlled stepper motors and emulated USB-HID keyboard events to automate industrial workflows.