Hai-Nam Ha

Hanoi, Vietnam namhaiha0308@gmail.com +84924278568 0x2f0713
0x2f0713

Final-year IT student with 3 + years of practical 4G/5G RAN project and solid C/C++ skills on telecom opensource platforms Open5GS, OAI, and srsRAN. Passionate about telecom systems, embedded software, and high-performance, low-level code, seeking roles that deepen this intersection.

Experience

Viettel Cyber Security, Software Engineer, Part-time — Hanoi, Vietnam
Dec 2021 – Apr 2025

- Acquired foundational knowledge in telecommunications, covering network architecture, wireless signal processing, and data flows.
- Participated in setting up a flexible 5G Standalone (SA) laboratory environment, enabling rapid deployment and testing with commercial devices.
- Explored radio signal jamming techniques for testing interference resilience and security measures.
- Participated in the development and customization of 5G RAN protocol layers, including RLC, RRC, S1AP, and NAS 5G, tailored to specific product requirements.
- Utilized and integrated open-source software platforms such as Open5GS, OpenAirInterface, and srsRAN for protocol implementation, testing, and deployment.
- Conducted in-depth research on 4G/5G network protocols to identify potential vulnerabilities, supporting product development with enhanced network security insights.
- Integrate software with telecom radio hardware to control the configuration of hardware.

Education

BS **VNU University of Engineering and Technology**, Information Technology — Sept 2021 – Dec 2025

- GPA: 3.55/4.0
- **Coursework:** Computer Architecture, Operating System, Computer Network, Wireless Networks

Certifications & Scholarships

TOEIC — Mar 2023
Score: 790

Techcombank Future Gen 2023 — Aug 2023

Projects

Dino Game (C++, SDL2) — github.com/0x2f0713/dino

- Developed a Dino-themed game inspired by Google's Chrome offline game, featuring dynamic gameplay, increasing difficulty, and score tracking.
- Utilized SDL2 for graphics, audio integration, and user interaction; implemented effective memory management using pointers and standard containers like arrays and deque.
- Structured the project using CMake and adopted the Pitchfork layout, enhancing build process efficiency, dependency management, and code modularization.