

# Dylan Cabahug-Almonte

✉ +1 484 356 6487 | @ dcabahug@andrew.cmu.edu | linkedin.com/in/dcalmonte 📍 Pittsburgh, PA

## Education

### Carnegie Mellon University

B.S. in Electrical and Computer Engineering | Minor in ML

August 2022 – December 2026

Pittsburgh, PA

**Relevant coursework:** Distributed Systems, Embedded Systems, Deep Learning Systems, Generative AI

**Current coursework:** Database Systems, Deep Reinforcement Learning

## Skills

**Programming:** C/C++, C#, Python, Go, Java, TypeScript, Bash, SystemVerilog, ARM/x86 Assembly

**Frameworks/Libraries:** PyTorch, NumPy, Pandas, CUDA

**Technologies:** Git, Cassandra, Java Spring Boot, Linux, FPGA, AWS

## Work Experience

### Captial One

June 2025 – August 2025

Software Engineer Intern | EP Tech

Mclean, VA

- Engineered enhancements to an internal performance testing platform for Customer Relationships APIs (10K+ TPS, 5ms latency), optimizing tooling used to benchmark the reliability and efficiency of high-volume, mission-critical services.
- Integrated the testing system with 15+ microservices powering customer data for 185M+ users and 411M+ accounts, enabling seamless API validation across all services.
- Automated test execution and report generation pipelines using Python, Jenkins, and AWS (Lambda, S3), reducing manual overhead and accelerating developer feedback cycles.

### Ansys Inc.

Jan 2025 – April 2025

Software Developer Engineer Intern | Mechanical Team

Canonsburg, PA

- Led the migration of Visual Studio solutions to CMake, optimizing build processes, improving cross-platform compatibility, resulting in increased development efficiency and maintainability.
- Developed and implemented comprehensive unit tests to ensure code reliability, improve test coverage, and facilitate continuous integration, reducing bugs and enhancing software maintainability.

## Projects

### Fast Fourier Transform (FFT) | Python, CUDA

December 2025

- Implemented the Fast Fourier Transform (FFT) from first principles using the recursive Cooley–Tukey algorithm, reducing computational complexity from  $O(N^2)$  to  $O(N \log N)$ .
- Validated correctness and numerical accuracy by comparing outputs against NumPy's FFT implementation across multiple input sizes and signal types.
- Analyzed and benchmarked performance differences between naive DFT and FFT implementations to demonstrate scalability improvements for large signals.
- Applied FFT-based spectral analysis to time-domain signals, enabling frequency-domain interpretation and verification through visualization.

### Real Time Operating System | C, ARM Assembly

November 2024

- Developed a real-time kernel with functionalities for context switching, task scheduling, and mutex synchronization using ARM Cortex-M architecture.
- Implemented fixed-priority rate-monotonic scheduling (RMS) with schedulability checks using the Utilization Bound test to ensure real-time performance guarantees.
- Designed a thread control block (TCB) system to manage multi-threading, including priority-based scheduling.

### Distributed Bitcoin Miner and Live Sequence Protocol (LSP) | Go

September 2024

- Accomplished a reliable communication protocol over UDP to manage packet loss, duplication, and out-of-order delivery. Features included sliding window protocol, exponential backoff for retransmissions, and checksum-based data integrity.
- Devised a client-server architecture where server distributed mining tasks to multiple miners. Server handled load balancing, fault tolerance, and task redistribution to optimize performance.
- Ensured robust fault tolerance with server-side task reassignment in case of miner failures and accomplished concurrency using Go's goroutines and channels to enable scalable, efficient shared computing.