

# Jaycee Dylan Alipio

818-310-0274 | jaycee.alipio@gmail.com | [linkedin.com/in/jayceedylanalipio/](https://www.linkedin.com/in/jayceedylanalipio/) | [github.com/nevaRiaD](https://github.com/nevaRiaD)

## EDUCATION

---

### University of California - Los Angeles

Bachelor of Science

Major in Electrical and Computer Engineering

Relevant Coursework: Algorithms, Analog Circuits, Computer Architecture, Digital Signal Processing, Feedback Control

Los Angeles, CA

Expected Dec 2026

## EXPERIENCE

---

### UCLA Bruin Supermileage (Formula SAE Team)

Embedded Engineer Club Member

Los Angeles, CA

May 2025 – Present

- Implemented CAN Bus communication in C on STM32F4 microcontrollers using HAL drivers, CubeMX configuration, Tx/Rx mailboxes, and hardware filters
- Designed and implemented the brake-pressure transducer firmware, including ADC configuration, calibration, and CAN integration, achieving 100 Hz sampling rate and accurate readings up to 500 psi
- Developed fault-tolerant message handling and priority arbitration for real-time subsystem data exchange
- Tested and debugged STM32 CAN firmware using STM32CubeIDE, JTAG/SWD interface, and logic analyzer (PulseView) to verify frame IDs, bit-timing, and bus error recovery

### Southern California Edison

Software Engineer Intern

Alhambra, CA

May 2024 – Present

- Developed a C# .NET service to collect and compress 1 M+ data records daily from 12 field devices, improving storage efficiency by 50%
- Built a C# GUI tool capable of decompressing and visualizing 5 M+ entries via interactive tables and comparison graphs
- Refactored legacy C code to improve readability and maintainability, reducing lines of code by 60%
- Automated data processing workflows using Bash scripts and Python utilities in Linux, modifying 10,000+ items saving 2 hrs
- Created a Splunk Classic dashboard to track six field devices and flag deviations, reducing manual checks by 30+ mins daily
- Captured and decoded network packets with Wireshark, ensuring data collection reliability and accurate data analysis

## PROJECTS

---

### STM32 Environmental Sensor Firmware with Wi-Fi Transmission

Fall 2025

- Programmed STM32 Nucleo board in C to interface with a BME680 environmental sensor over SPI using Bosch API drivers
- Designed modular firmware with FreeRTOS tasks, queues, and semaphores for real-time sensor sampling and communication
- Implemented UART link from STM32 board to ESP32 Wi-Fi module to transmit sensor data wirelessly via TCP

### RISC-V Single-Cycle CPU Simulator

Fall 2025

- Implemented a 32-bit single cycle RISC-V CPU in C++ with modules for ALU, Control Unit, Immediate Gen, Register File, and Memory
- Modeled instruction flow with explicit signal and MUX logic; verified 13 instructions (ADDI-JALR) via binary traces
- Simulated full instruction flow and memory hierarchy to validate ISA compliance via custom binary testbench

### ESP32 Bedroom Security Sensor

Spring 2024

- Programmed ESP32 in C++ to interface with an HC-SR04 ultrasonic sensor using pulseIn() timing for motion detection
- Calibrated ultrasonic sensor readings and validated motion-detection accuracy using serial monitoring and threshold tuning

## SKILLS

---

**Programming Languages:** C, C++, C#, Python, Embedded C

**Embedded Systems:** STM32 (F4), RTOS, ESP32, CAN, SPI, UART, I2C, ADC/DAC, DMA, GPIO, Timers, Interrupts

**Tools:** STM32CubeIDE, CubeMX, ST-Link, Logic Analyzer, Oscilloscope, Wireshark, Git, Linux, Bash, Windows .NET