Aarjav Jain

Aarjavjain2736@gmail.com | Website: aarjavjain.netlify.app | 587-664-2736

OBJECTIVE: Join a team of passionate embedded systems or firmware developers. **(8 months)**

TECHNICAL SKILLS

Languages: C, Python, C++, System Verilog (SV), ARM Assembly, Bash.
Hardware and Peripherals: STM32 MCUs, CAN, UART, I2C, I2S, SPI, ADC, GPIO, PWM, DMA.
Frameworks and Applications: FreeRTOS, Embedded C, Linux.
Software/Agile Tools: STM32Cube IDE, GDB, Git/GitHub, Visual Studio Code, Jira.

EDUCATION

University of British Columbia **Expected Graduation: May 2027**
Bachelor of Applied Science - Computer Engineering **CGPA: 4.33**
Related Courses: Computing Systems I & II, Algorithms and Data Structures, and Software Construction

ENGINEERING STUDENT TEAMS

Solar, UBC **September 2023 – Present**
Embedded Systems Team (EMBD) Co-Lead *June 2024 – Present*

- Proactively strategized EMBD’s future with Solar’s executives by planning recruitment, resource management, member onboarding, and EMBD projects by consulting other teams and translating strategy requirements.
- Focused on simplifying our EMBD systems, such as the driver dashboard and motor control system, by collaborating with other electrical sub-teams to reduce unnecessarily complex aspects of the systems.
- Designed a standard testing framework including Hardware-in-Loop testing, automation scripts, and JTAG boundary scan testing for new PCBs.
- Increased EMBD's work efficiency by reducing HW issues and setup time with an improved integration pipeline.
- Performed routine design reviews and pushed documentation of all decisions and work using monday.com.
- Conducted reviews and approvals for GitHub PRs and planned releases for our firmware and Python repositories.
- Our determination resulted in 6th out of 24 SOV class vehicles in the 2024 FSGP race, qualifying Solar for ASC!

Embedded Systems Engineer *September 2023 – June 2024*

- Developed our telemetry and motor control system by configuring FreeRTOS middleware and CAN, I2C, UART, and DMA peripherals for STM32F103RC chips on custom PCBs. Wrote the firmware using VSCode.
- Designed a Python Flask backend to read PCAN, XBee radio, and randomized data using threading and cantool libraries. The backend parses, stores, and visualizes CAN data on InfluxDB and Grafana.
- Efficiently debugged issues using STM32CubeIDE’s OpenOCD + GDB server to use a StackAnalyzer and Expressions which assisted in identifying bugs in our telemetry and main control board firmware.
- Debugged hardware CAN issues with an oscilloscope, multimeter, reworking, and root cause analysis.

WORK EXPERIENCE

Intel – Emulation Engineer Co-op **January 2025 – Present**

- Enabled microcontroller (µC) FW debugging and trace for Emulation on Synopsys ZeBus using C++ and SV.
- Improved µC debugging via access to Core internals using JTAG, ARM DAP, and a custom AXI4 test interface.
- Utilized Verdi to debug and view test program execution on HW via waveforms and SW trace/logs.
- Implemented and documented a standard build and runflow to create our Emulation team's workflow.

NETGEAR – Software Developer Intern **April 2024 – December 2024**

- Developed a Python client-server model to monitor and validate Orbi Topology Optimization against TCP data rates, generating PDF reports on the network topology over time, which speeds up customer support by 90%.
- Took initiative for constructing multiple labs. This involved AutoCAD layout prototyping, numerous test resources, an executive team, and task management to improve our Orbi mesh product’s development environment.
- Utilized Jira and GitHub for tracking project progress, collaborative development, and distributing scripts.