Daniel Ye

dzyye@uwaterloo.ca | 408-385-6805 | danielye.me | in/danielzyy | github.com/danielzyy

EDUCATION

University of Waterloo – Ontario, Canada 09/2020 – 04/2025 (Expected)

- Candidate for BASc in Mechatronics Engineering (Cumulative Average: 95.1%, 4.0 GPA, Dean's List Recipient)
- Courses: Control Systems, Algorithms and Data Structures, Microprocessors and Digital Logic, Real-Time Systems

SKILLS

Languages/Frameworks: C++, C, Python, Java, RTOS, CUDA, ROS, Bazel, Docker, MATLAB, HTML, CSS
Applications/Tools: GDB, Git, Arduino, Logic Analyzers, Oscilloscope, KiCad, SolidWorks, AutoCAD

EXPERIENCE

Firmware Engineering Intern – Tesla Motors Chassis Control Team (Steer-by-wire) 01/2024 – 08/2024

- Implemented motor force derating while vehicle is on jump-post, **preventing power loss** due to excessive current draw
- Developed monitoring strategies in **C** for interpolated **autopilot steering command**, improving system robustness
- Implemented **TX buffer** clearing when CAN bus open/off, preventing stale messages from causing false monitor trips
- Enabled phase-disconnect IC sleep over SPI, reducing current draw of the steering system by **25%** while in sleep

Embedded Software Engineering Intern – Skydio Inc. 05/2023 – 08/2023

- Developed firmware updater **state machine** in **C++** to enable new firmware images received over-the-air to be packetized and automatically written to MCU memory through a daisy-chained serial interface with redundancy
- Implemented A/B **bootloader** for STM32 MCUs in **C** with **linker scripts** built using **Bazel**, to improve firmware update robustness and prevent memory corruption from halting program execution by having two bootable image partitions
- Improved update time by **60%** using message batching, and implemented protocol migration functionality

Firmware Engineering Intern – Tesla Motors Chassis Control Team (Steer-by-wire) 09/2022 – 12/2022

- Architected a **state-follower** bootup sequence between **multi-core** ECUs to synchronize initialization states over **CAN**, check **NVM** calibrations, and verify motor shutoff to ensure a deterministic startup and prevent **3+** safety-critical alerts
- Developed **RTOS** module in **C** to monitor gate driver V_{DS} thresholds set on startup over **SPI** using a **finite state machine**
- Implemented angle sensor calibration routine over **CAN** and verified ADC readings to determine steering motor position
- Automated the generation of motor parameters and ADC pin configurations using **Embedded Ruby** and **Python** scripts

System Software Intern – Nvidia DriveIX 01/2022 – 04/2022

- Implemented **Temporal Noise Reduction** (TNR) algorithm in C++ onto stitched camera output to improve vehicle safety
- Developed feature to generate fixed virtual camera paths in **OpenGL** to provide repeatable scenarios for validating TNR
- Unified **CUDA** streams and kernel invocations for camera stitcher to optimize GPU usage and improve latency by **80%**

Test Automation Intern – Ford Motor Company 05/2021 – 08/2021

- Automated and configured **10+ test cases** for the Advanced Driver-Assistance System (ADAS) with **Python** using **Slash**

Radar Team Lead – WATonomous (Autonomous Vehicle Design Team) 05/2021 – 08/2022

- Led a team of 5 to research and implement **DBSCAN clustering algorithm** for radar object detections using **ROS**

PROJECTS

Automated Sea Asparagus Harvester – Capstone Team Salico 9/2024 – Present

- Implemented **closed-loop velocity control** with torque limiting in **C** for BLDC motors to optimize harvesting process
- Designed state machine and **fault monitoring system** using **RTOS** for deterministic operation of solenoids and motors

Haptic Smart Knob – C, STM32, RTOS, KiCad, SolidWorks

- Simulated haptic feedback with a BLDC motor using closed-loop **field-oriented-control** and tuned **PID** values
- Developed **RTOS** threads to manage motor, force sensor, LEDs, display, and USB using queues and mutexes
- Designed 4-layer form factor PCB with a STM32F4 MCU using **KiCad** and 3D printed housing in **Solidworks**