Max Freeman

max_freeman@berkeley.edu | (607) 327-7804 | linkedin.com/in/maxfreeman | Mountain View, CA

Education

University of California, Berkeley **August 2024 - May 2025**
Master of Engineering in Mechanical Engineering, *Concentration in Controls & Robotics* GPA: 3.98

Cornell University **September 2020 - May 2024**
Bachelor of Science in Mechanical Engineering GPA: 3.77 | *Magna Cum Laude*

Skills

Electrical & Mechatronics: Sensors & Actuators | Embedded Systesm | Harness Fabrication (Crimping & Soldering)
Networking & Diagnostics: CAN, I2C, UART, Ethernet | Oscilloscopes | Multimeters | PCAN-View | Wireshark
Programming & Software: Python | ROS2 | MATLAB | Simulink | C++ | Git | Linux | Bash | Jira
Prototyping & Mechanical Design: CAD (SolidWorks) | 3D Printing & Rapid Prototyping | Manual Mill & Lathe

Highlighted Experience and Projects

System Integration Intern, *Toyota Research Institute* **July 2025 - Present**

- Support cross-functional integration of mechanical, electrical, and software systems to enable rapid prototyping and testing of next-generation research vehicles.
- Interate a steering wheel encoder on a prototype vehicle, taking ownership of CAN communication setup, wire harnessing and connector fabrication, and the design of a custom 3D-printed mount using SolidWorks.
- Develop a ROS2 node in Python to convert raw steering angle data into vehicle yaw rate estimates using a kinematic bicycle model, publishing the signal to the localization stack to improve pose estimation.
- Integrate an NVIDIA Jetson for on-vehicle monitoring and diagnostics, defining interface requirements with researchers, designing custom sheet metal mounting brackets, and packaging peripherals—including an LCD touchscreen, GPIO-connected button, and harnessing—into a highly confined vehicle space.
- Program the vehicle's PDM in MATLAB/Simulink to implement a controlled shutdown sequence, improving reliability of log file capture and ensuring safe power-down during testing.
- Reverse-engineer undocumented CAN signals by operating vehicle components while scanning logs in PCAN-View, isolating message patterns to identify key control signals.

Lead Robotics Engineer, *Multimodal Autonomous Platform Design (MEng Capstone)* **September 2024 - May 2025**

- Led a team of 4 engineers in developing a hybrid ground-aerial robot, owning hardware bring-up, control system implementation, and testing strategy across both driving and flight modes.
- Built and deployed a ROS2-based flight controller in Python using a cascaded PID architecture with integrated safety checks; validated stability and responsiveness through iterative flight tests.
- Designed and executed a staged flight validation plan, leading 10+ tethered flight tests to verify subsystem functionality, capture debug logs, and identify failure modes while minimizing hardware risk.
- Developed custom Python-based logging and visualization tools to support hardware testing of both control modes, enabling rapid debugging and performance validation.
- Analyzed test logs to isolate root causes of instability and control issues, iteratively refining controller behavior to improve system robustness and tracking accuracy.

Hardware Test & Integration Intern, *Lit Motors* **June 2024 - July 2024**

- Contributed to test infrastructure development and subsystem design refinement at a fast-paced startup building a two-wheeled self-balancing vehicle actuated by Control Moment Gyroscopes (CMGs).
- Led development of a dual-plane dynamic balancing test rig for CMG evaluation, integrating structural hardware, sensors, and a belt-drive system to identify weight imbalances and reduce vibrations.
- Designed the rig in SolidWorks using a mix of 3D-printed and machined parts, enabling rapid prototyping and flexible reconfiguration to support varying shaft lengths and hardware geometries.

Embedded Autonomous Vehicle Project, *Cornell University* **January 2024 - May 2024**

- Managed end-to-end development of a small-scale autonomous ground vehicle, integrating sensors and motor drivers on an Arduino Nano and implementing control logic in C++ and Python.
- Interfaced with Time-of-Flight and IMU sensors over I2C, developing sensor fusion algorithms and software-based filters in C++ to minimize sensor output noise by over 50%, enhancing sensor accuracy.
- Implemented and tuned distance-based PID controllers to improve tracking, reducing settling time by 40%.
- Diagnosed hardware and sensing issues using oscilloscope traces and telemetry data, resolving integration faults, and improving system performance.