Engineering Portfolio

Ziang Tan

Electrical & Computer Engineering • Applied Mathematics: Data Science
University of Washington

This portfolio highlights selected projects in embedded systems, robotics, sensing, testing, and electro-mechanical prototyping. The work emphasizes system integration, debugging, and practical engineering execution across firmware, hardware, and modeling workflows.

Embedded Systems
ESP32-S3 firmware, interrupts, timers, buffering, and motor-control logic for autonomous operation.

Electro-Mechanical Prototyping
Line-tracking robot assembly, custom PCB integration, sensor calibration, and iterative testing.

Robotics & Modeling
6-DOF arm kinematics, trajectory generation, and motion validation in Python.

System Integration
Sensor selection, environmental measurement, and subsystem planning for a Mars-style rover.

Contact: ziangt2@uw.edu • Seattle, WA

Autonomous Embedded Cat Care System

Embedded sensing and control prototype • ESP32-S3 • C

Designed a real-time embedded system that combined sensing, timed logic, and actuator control for autonomous cat-care functions under practical timing and reliability constraints.

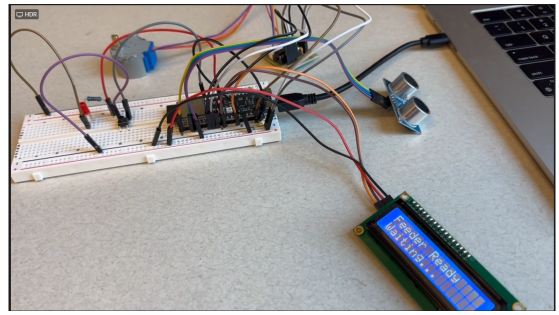
Key technical work

- Implemented interrupt-driven ultrasonic sensor handling on ESP32-S3 to detect events and trigger control logic.
- Configured hardware timers for deterministic sampling intervals and periodic task execution.
- Developed state-machine-based motor control to coordinate automated feeding and cleaning cycles.
- Improved reliability by using buffering, volatile variables, and critical sections to reduce race-condition risk.

Result: Produced a stable prototype capable of autonomous sensing and actuation behavior with improved firmware robustness.

Tools used: ESP32-S3, C, timer interrupts, ultrasonic sensing, motor control, embedded debugging

Engineering note: This project is especially relevant to hardware systems integration because it required both firmware structure and practical coordination between sensors, timing, and actuators.

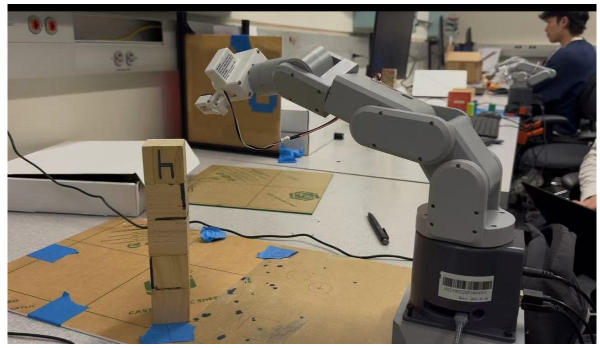


Prototype hardware setup with ESP32, ultrasonic sensor, and LCD interface

Robotic Arm Kinematics and Trajectory Control

Modeling and motion-planning workflow • Python • SciPy

Built a computational workflow for modeling a 6-DOF robotic manipulator and generating feasible pick-and-place trajectories using numerical methods.



Block stacking task using the robotic arm

Object sorting task using the robotic arm

Key technical work

- Modeled a 6-DOF robotic arm using Denavit–Hartenberg parameters to describe link geometry and joint relationships.
- Derived forward kinematics to compute end-effector position and orientation from joint values.
- Implemented inverse kinematics numerically in Python using SciPy optimization methods.
- Generated and validated joint-space trajectories while analyzing kinematic constraints across configurations.

Result: Created a reusable kinematics and trajectory-planning workflow for motion analysis and robotics control learning.

Tools used: Python, SciPy, numerical optimization, robot kinematics, trajectory planning

Engineering note: This work demonstrates system modeling and validation skills that transfer directly to robotics, automation, and electromechanical product development.

Line-Tracking Robot

Course robot build and tuning project • Arduino • custom PCB

Designed and assembled a line-tracking robot from the ground up, integrating custom sensing hardware, control logic, and repeated test-based tuning to improve performance.

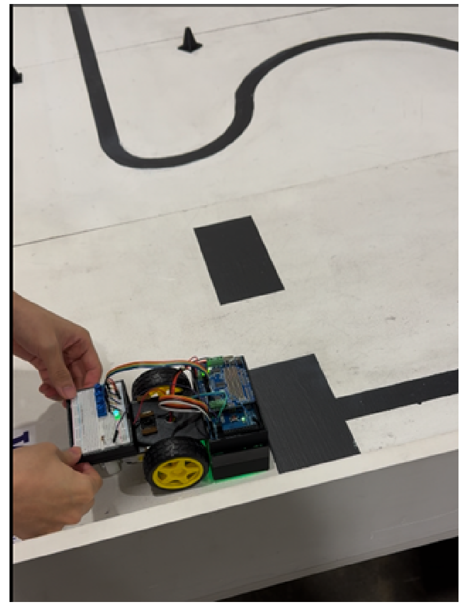
Key technical work

- Built the robot with an Arduino board, motor shield, motors, batteries, and a fully 3D-printed chassis.
- Integrated a custom PCB with LEDs and photoresistors for line sensing and signal detection.
- Contributed to sensor calibration, control tuning, and hardware/software integration during iterative track testing.
- Used repeated optimization and validation to improve line-following consistency and overall speed.

Measured outcome: After testing and optimization, the robot completed the course track in 48 seconds.

Tools used: Arduino, custom PCB, LEDs, photoresistors, motors, batteries, 3D printing, robot integration

Engineering note: This is one of the strongest examples of hands-on electromechanical integration in my project work.



Project visual / system view

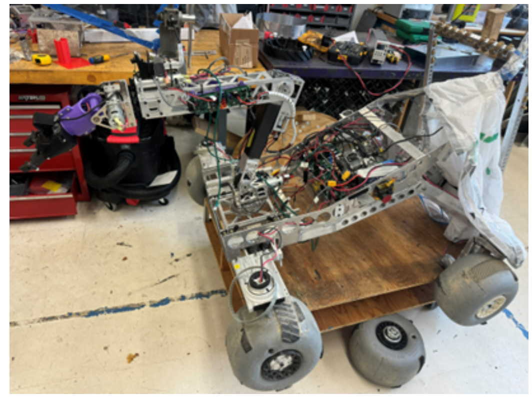
Mars Rover Sensor Subsystem – Husky Robotic

Environmental sensing for a Mars-style rover • subsystem research and integration

Supported sensor subsystem planning for a rover intended to explore environmental conditions and assess life-detection potential under Mars-like constraints.

Key technical work

- Researched and evaluated candidate sensors for CO_2, CH_4, CO, O_2, temperature, humidity, and possible subsurface measurements.
- Compared sensor specifications, operating ranges, and integration constraints for Mars-style environmental use cases.
- Helped align sensor choices with rover science objectives and life-detection experiments.
- Contributed to hands-on subsystem development through sensor integration and enclosure-oriented robotics work.



Mars-style rover platform used for subsystem development and integration

Result: Built practical experience in sensor-driven robotics design and environmental measurement planning for a complex mobile platform.

Tools used: Sensor research, subsystem planning, environmental measurement, robotics integration, 3D-printed enclosures

Engineering note: This project strengthened my ability to reason about sensing choices not only by specification, but also by integration feasibility and field constraints.