# Armaan Sengupta

armaansengupta.ca GitHub LinkedIn senguptaarmaan@gmail.com

## Skills

**Software:** Python, C, C++, Rust, Git, CMake, Bash, GBD, Protobuf | ROS2, FreeRTOS, Pytest, NumPy
**Embedded Systems**: STM32, NXP, Ambiq, ESP32, AVR | I2C, SPI, CAN, gRPC, UART, JTAG, BLE, MIPI DSI
**Tools:** KiCad, Altium, MATLAB, Simulink/Simscape, SimulationX, SolidWorks, Inventor, Fusion 360
**Robotics:** Controller Design, Forward/Inverse Kinematics, Trajectory Planning, State Estimation

## Work Experience

**Implant Firmware & Embedded Systems Intern** — California, USA
*Neuralink* — Jan 2026 – August 2026

- Led firmware development for 1st gen neural stimulation implant, implementing interrupt driven, DMA fed, high throughput code coordinating waveforms and sampling for 3k+ electrodes (Rust & C)
- Architected the full app to ASIC stack for Blindsight, our human visual prosthesis product
- Shipped code reducing registration time from 8m to 15s per trial enabling delivery of ~40M pulses per electrode in 13 weeks; making the safety study feasible and hitting FDA submission deadline (Rust)
- Brought up impedance & calibration, enabling self test of chips, unblocking production line

**Vehicle Firmware Body Controls Intern** — California, USA
*Tesla* — May 2025 – August 2025

- Brought up ~20 exterior lights & related ADCs on Semitruck from hardware to app layer (Altium & C)
- Fixed Cybertruck bug causing left/right headlamp stepper motors to calibrate out of sync (C)
- Developed current-sense self-tests for Cybercab lights, preventing faulty units passing production (C)
- Wrote SIL tests verifying lighting functionality across models preventing regressions (Python: Pytest)

**Embedded Systems & Firmware Engineering Intern** — Indiana, USA
*Ixana* — August 2024 – December 2024

- Designed firmware to implement video transmission from smart glasses to smart watch improving frame rate from 4 fps to 13 fps and reducing power consumption by 8% (C++)
- Developed library for Rigol Arb signal generator to facilitate HIL testing of custom silicon (Python)

**Robotics and Embedded Systems Intern** — Ontario, Canada
*Fyelabs* — May 2023 – September 2023

- Designed a LiDAR dust filtering algorithm, enhancing object localization in harsh environments (C++)
- Developed PCB board for a custom HMI panel enabling driver-system communication (EasyEDA)
- Developed UART driver for LiDAR module to enable data logging and live HIL debugging (Python) & (C)

## Design Team, Research & Project

**Stewart Platform (3RRS Parallel Manipulator) Project** — September 2025 – December 2025

- Derived inverse kinematics of platform: platform plane → 3 servo's rotations (Python)
- Built a multi-body dynamics model of platform–ball interaction (Simscape/Simulink)
- Implemented a force-torque ball contact localization system; benchmarked against OpenCV (Python)
- Evaluating different controller designs and trajectory based feedforward control (Simulink)

**Research Assistant @ Robotic Interaction, Perception and Learning Lab** — September 2025 – Jan 2026

- Enabled torque-based contact localization research by building a MuJoCo - ROS2 simulation bridge enabling external force injection to evaluate torque-based contact localization algorithms (Python)

**Active Control Systems @ Waterloo Rocketry** — September 2023– December 2024

- Developed C code for an STM32 using FreeRTOS to actuate aerodynamic control surfaces via a PID controller and RK 4 trajectory prediction bringing the rocket 800 ft closer to its target apogee
- Designed 4 layer central processing board PCB in KiCad improving power efficiency from 66% to 95% and incorporated an onboard IMU over I2C, as well as an external AHRS through UART (KiCad)

## Education

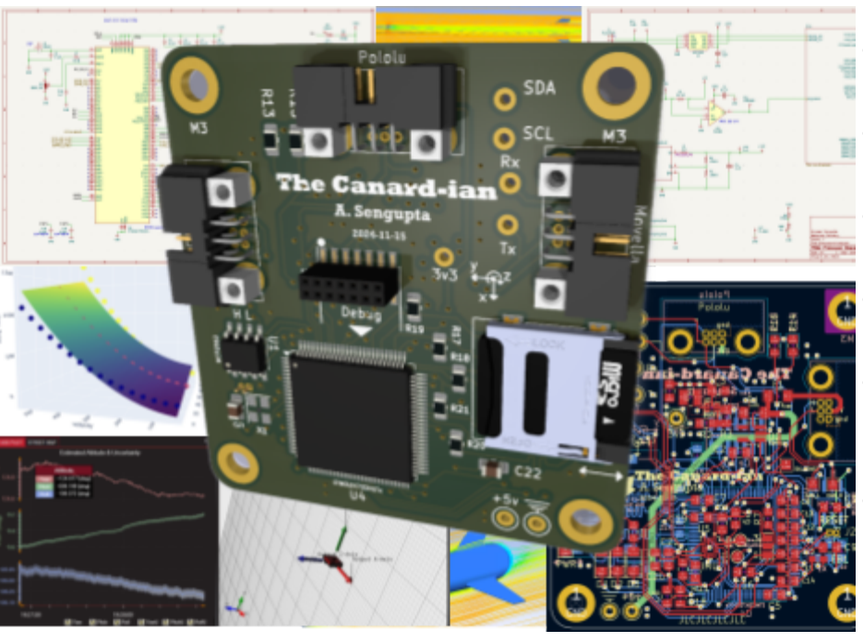
**University of Waterloo**
*Bachelors of Mechatronics Engineering | GPA: 90% (Dean's Honors Role)* — September 2022 - April 2027

# Portfolio

*More projects on my website:* https://www.armaansengupta.ca

### Rocket Air Brakes & Canards

*Waterloo Rocketry (Student Design Team)*




- Developed embedded C code on an STM32 microcontroller within a FreeRTOS environment
- Evaluated extended kalman filter vs madgwick filters for state estimation
- Applied 4th order Runge-Kutta methods for rocket trajectory prediction
- PID controller used for closed loop control of the fin extension during flight to modify trajectory
- Designed schematic and PCB layout for central computing board improving power efficiency from 66% to 95% (KiCad)
- Independently designed the Airbrakes mechanical subsystem to convert a servo's rotational motion to linear motion, simultaneously extending 3 fins from the rocket body to increase CD (Solidworks)

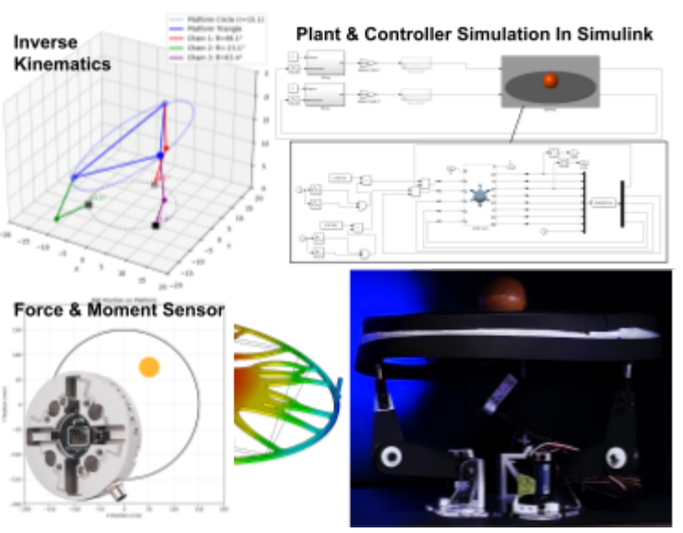
More details on what I did OR the full rocket

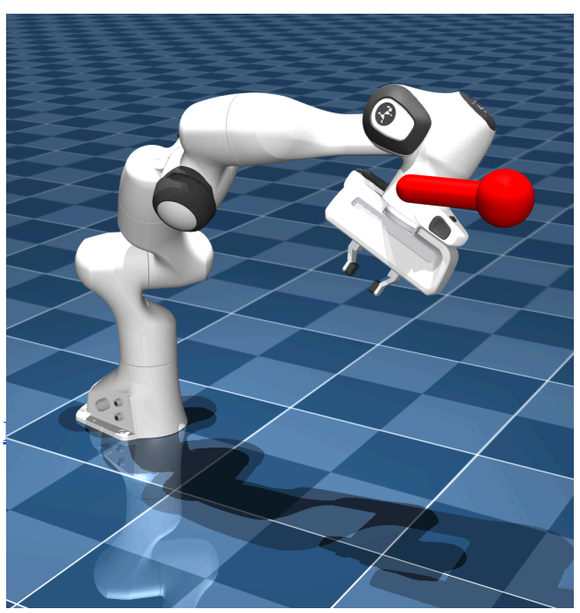
### Stewart Platform (3RRS Parallel Manipulator)

*Personal Project*

- Derived inverse kinematics mapping representative platform plane to 3 servo rotation angles and validated on real platform (Python)
- Built a multi-body Simscape/Simulink model with custom interaction script written in C to simulate platform + ball dynamics and controller behavior
- Currently testing trajectory planned velocity profiles, to dictate feedforward control and comparing system response with traditional PID
- Integrated a 3-axis force-torque sensor for ball contact localization and compared performance to a OpenCV vision based model.

Video demos



### MuJoCo Simulator Support For ROS2 Based Missions

*Robotic Interaction, Perception and Learning Lab*

- Developed a MuJoCo-based ROS2 simulation bridge for the Franka Panda, enabling our existing robot missions to run in a high-fidelity physics environment and allowing realistic external force injection with live torque sensing feedback
- Contributed to the lab's broader contact localization research, using joint torque signals to identify external interaction points on the robot, supporting safer and more intelligent manipulation behaviors for humanoid robots

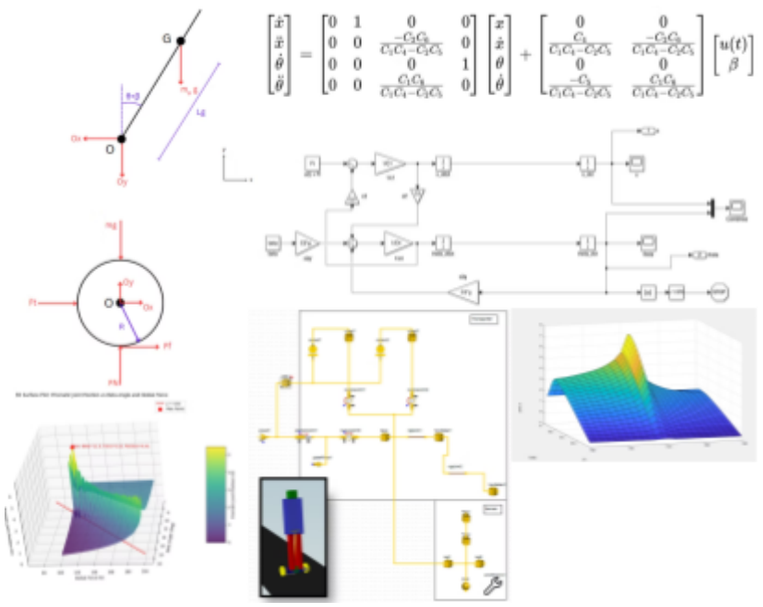
View Demo Video

**Autonomous Truck Back Up Safety System**
*FYELABS*

- Communicated between 6 microcontrollers using wireless and serial communication (C++)
    - Defined custom serial communication byte standard including checksums, and error codes
- Developed LiDAR array subsystem firmware (C++)
    - Custom dust/noise filters based on experimental data
    - Zone mapping to compute what grid an object is in
    - Autonomous alley mode detection feature
- Used Python to visualize live LiDAR data over UART, representing it as a 2D top view of the environment

View project's Electronics or Software



**Segway System Analysis & Modelling**
*Class Project*

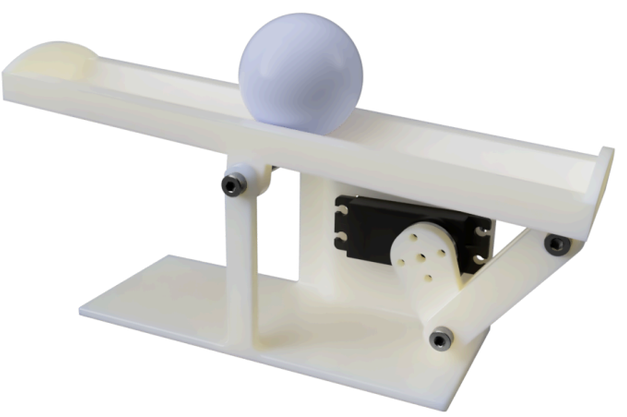
- Derived the nonlinear kinematic and dynamic equations of motion for a human transporter system, then applied small-angle linearization to obtain a simpler state-space model.
- Built a Simulink simulation to sweep lean angle and applied force inputs, identifying parameter combinations that maximize travel distance without closed loop controls
- Validated the hand-calculations / Simulink results by modelling the system in SimulationX which uses a more accurate acausual system

View detailed write up and video

**Beam Balancer**
*Personal Project*

- Designed optimized link lengths using 2D constraints then modelled compact and smooth 3D design (SolidWorks)
- Designed and tuned control system to stabilize ball at center position with minimal %overshoot and oscillations (Python)
- Wrote embedded firmware for servo actuation (C)
- Used OpenCV for ball position tracking and feedback

View Demo Of Ball Balancing





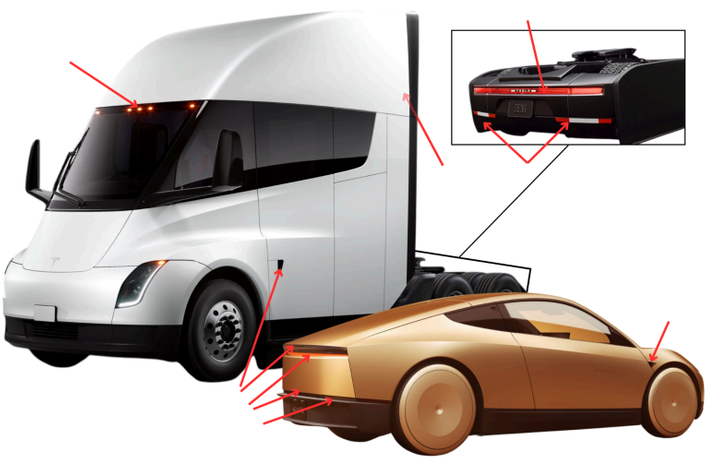
Last Year's CES Demo or Early Prototypes

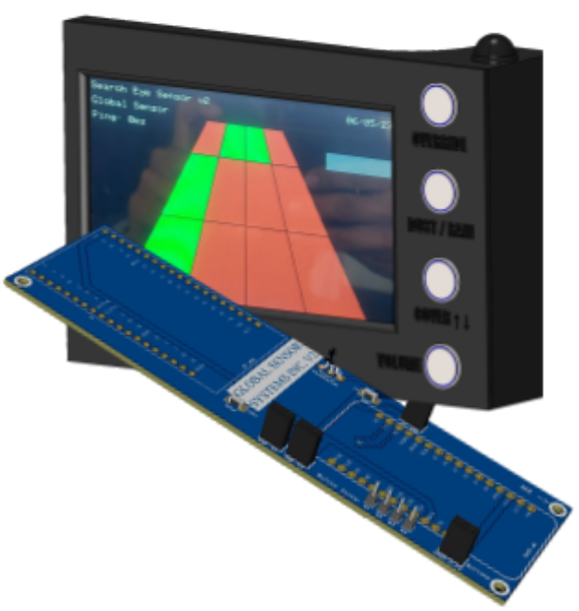
**Smart Glass To Smart Watch Video Streaming**
*Ixana*

- Designed power efficient low level core OS on bare metal chips using C++ to facilitate wireless video transmission from smart glasses to smart watch
- Developed Verilog code for FPGA to interface over MIPI DSI to Amoled display
- Orchestrated data flow between camera, to encoding FPGA, to transmitter microcontroller, then wirelessly transmitted between 2 "Wi-R" chips, to receiver microcontroller, to decoder FPGA, and finally displayed on AMOLED screen
- Debugged extensive data encoding, timing, and signal integrity issues in all parts of system

**Exterior Lighting New Vehicle Platform Bringup**
*Tesla*

- Brought up ~20 exterior auxiliary lighting channels and associated ADC current-sense self tests on Semitruck, spanning hardware initialization, driver development, and app-layer integration (Altium & C)
- Implemented diagnostics and self-test infrastructure on Cybercab, ensuring lights are validated in production and during runtime (C)
- Improved ADC sampling time for Cybercab lighting endpoints from 10ms to 1ms while identifying hardware RC time constant issue causing the sample and hold circuit to incorrectly hold charge causing false passes (Altium & C)





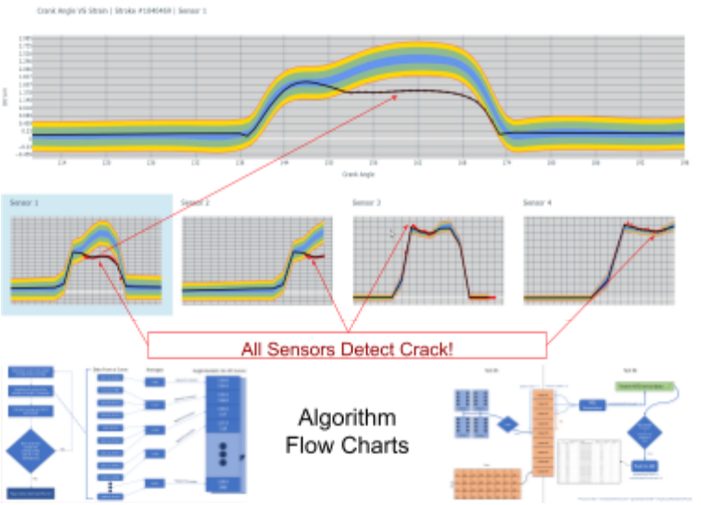
**Human Machine Interface (HMI)**
*FYELABS*

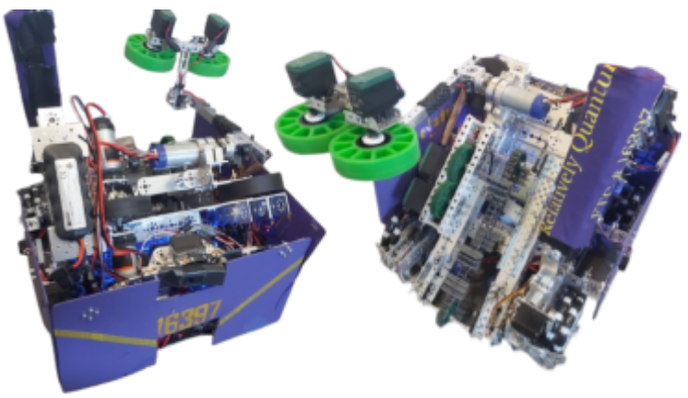
- Created a graphics library in C++ to display incoming data from the lidar board.
- Designed PCB to support 2 microcontrollers and various other peripherals within the HMI (EasyEDA)
- Created a zone mapping algorithm to project a detected object to one of 12 zones so it could be visually displayed on the HMI (C++)
- Outcome of the project is driver can successfully interpret current system behavior and change system settings wirelessly

View More

**In Die Crack Detection + Die Condition Monitoring**
*Magna*

- Read strain gauge data from press (MQTT), re-factored it, stored it in PostgreSQL database, and developed anomaly detection algorithm to detect when a part cracked based on this data (Python)
    - Algorithm correctly detected cracked parts 100% of the time during a production run (350 strokes)
- Created web-app to visualize/interact with the data (Python)
- Developed data aggregation algorithm for a PLC using multithreading with custom semaphores and data queues to circumvent limited hardware performance
    - Data collection rate improved from 142 Hz to 500 Hz



**Autonomous Mobile Robots (AMRs)**
*Competitive Robotics Team (FTC 16397)*

- Independently wrote all software (Java)
- Implemented PIDF control structure to ensure motors held their position (arm) or velocity (flywheel) under load
- Used TensorFlow to implement computer vision that detected objects to influence robot behavior
- Utilized distance sensors and gyros for localization
- Won the control award in the Asia Pacific Open Championship for the best software reliability

View demo video, 2 page technical report, or top match

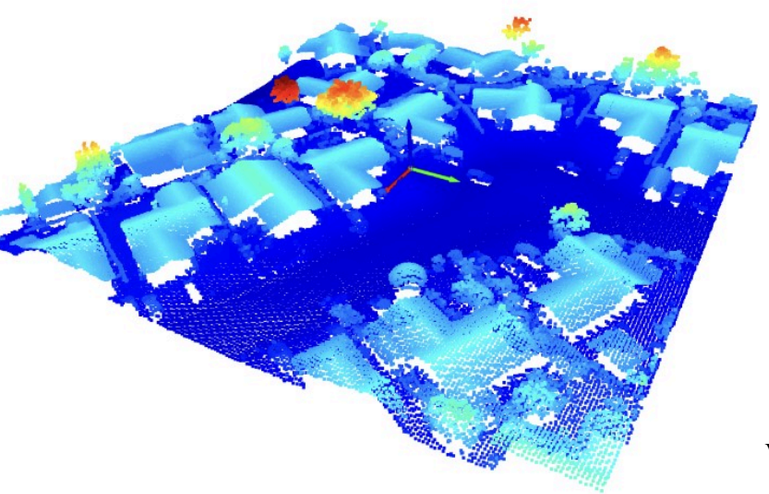
**Automated Multi-Dimensional Tolerance Verification System**
*Team Project (3 People)*

- Wrote C code to control both motors (clamp + turntable), derive object dimensions from encoder values, and user GUI
- Designed and implemented a custom 2 wire communication protocol (C++ & C) (similar to I2C)
- Determined object orientation in OpenCV (Python)
- Accurately rotated objects 100% of the time during testing as well as measured results within 0.3mm of true value

View demo video or technical report



**LiDAR Data Processing**
*Personal Project*



- Used Python with laspy, open3D, and numpy
    - laspy: tool for LiDAR data handling and analysis
    - open3D: 3D data processing and visualization
- Manipulation, cleansing, filtering, and analysis of LiDAR point cloud data
- Applied voxel grouping to downsize the data
- Removed artifacts such as dust using a nearest neighbour outlier algorithm
- Found the furthest point within a region to any other point in the data set using a custom algorithm to vastly decrease time complexity (essential for 1M+ points)

View project details or view the 3D point cloud online

*More projects on my website:* https://www.armaansengupta.ca