

Armaan Sengupta

 armaansengupta.ca

 647-877-8942

 senguptaarmaan@gmail.com

 [LinkedIn](#)

Relevant Skills


Software: Python, C, C++, Git, CMake, Bash, VHDL, GDB, Verilog | FreeRTOS, SciKit Learn, NumPy, Pandas
Technologies/Protocols: STM32, NXP, Ambiq, ESP32, AVR | I2C, SPI, CAN, PWM, UART, JTAG, BLE, MIPI DSI
Tools: KiCad, EasyEDA, MATLAB/Simulink | Oscilloscope, Arbitrary Signal Generator, Digital Logic Analyzer

Work Experiences

Embedded Systems & Firmware Engineering Intern

Ixana

 West Lafayette, Indiana


 August 2024 – December 2024

- Designed firmware for a bare metal MCU to stream video from smart glasses to smart watch
 - Developed efficient firmware implementing video compression, transmission, and decompression improving frame rate from 4 fps to 13 fps and reducing power consumption by 8% (C++)
 - Designed amoled display driver using MIPI DSI protocol enabling reliable video rendering (C++)
- Implemented automatic gain control for custom RF chip improving packet reception rates by 45%
 - Developed Visa python library for Rigol Arb signal generator to facilitate HIL testing of AGC
 - Configured chip registers implementing scheduled binary search for ideal gain (C++)
- Debugged inter-IC signal timing and encoding issues using a digital logic analyzer

Data Science & Controls R&D Engineering Intern

Magna

 Brampton, Ontario

 January 2024 – April 2024

- Optimized a robotic gantry's motion profile, increasing production speed by 38%, while reducing wear
- Designed an anomaly detection algorithm using strain gauge data from a hydraulic press to identify cracked parts and automatically filter them from the production line with a 100% accuracy during 150 part burst run (Python)

Robotics and Embedded Systems Intern

Fyelabs

 Hamilton, Ontario


 May 2023 – September 2023

- Implemented checksums in custom RF protocol reducing data corruption by 15%
- 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)

Mobile Robotics R&D Engineering Intern

Studica

 Mississauga, Ontario

 June 2022 – August 2022

- Develop code for AMRs to autonomously navigate complex environment, enabling component testing
- Designed 11 novel parts (e.g.), and reduced the cost of an existing product by ~40%

Engineering Design Teams

Waterloo Rocketry

Embedded firmware and PCB design for active in flight controls

 September 2023– Present

- Developed C code for an STM32 using FreeRTOS to manipulate aerodynamic control surfaces
 - Used Kalman filters to determine rockets current orientation and RK4 methods to predict the rocket's trajectory (C)
 - Implemented a PID controller based on predicted trajectory to command servo motor over PWM, which during flight allowed our Air Brakes to bring the rocket 1500 ft closer to it's target apogee (C)
- 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


Education

University of Waterloo

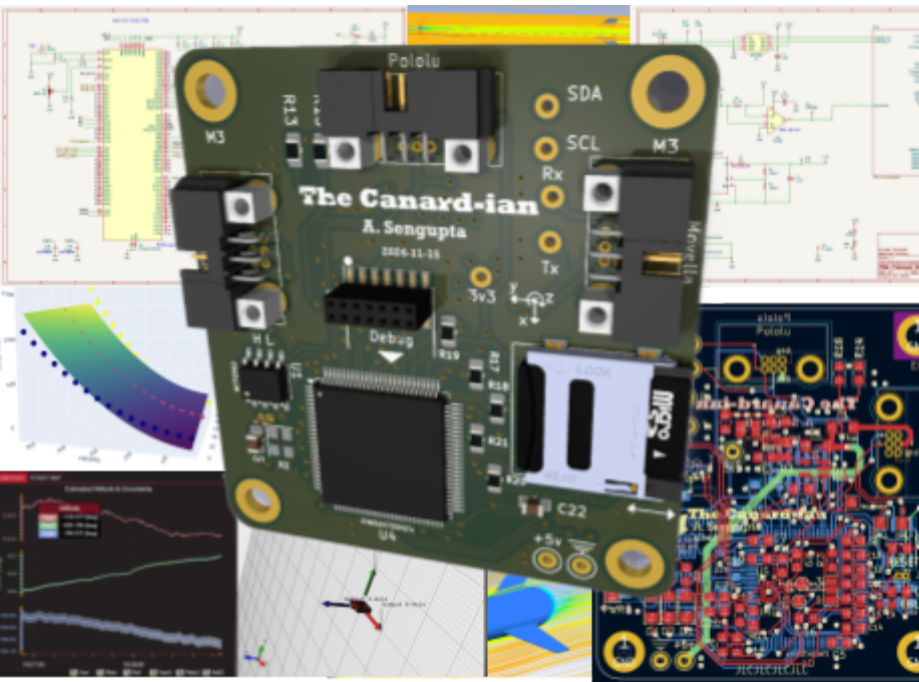
Bachelors of Mechatronics Engineering | GPA: 89.4% (Dean's Honors List)

 September 2022 - April 2027

Projects

 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
 - Serial communication with IMU (VN-300)
 - Board to board communication over CAN
 - 4th order Runge-Kutta methods for trajectory prediction
 - PID controller for closed loop control of the fin extension during flight
- Designed schematic and PCB layout for central computing board in KiCad
 - Improved power efficiency of board from 66% to 95% by using buck converter
 - I2C communication to two IMUs and UART to external AHRS sensor

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](#)

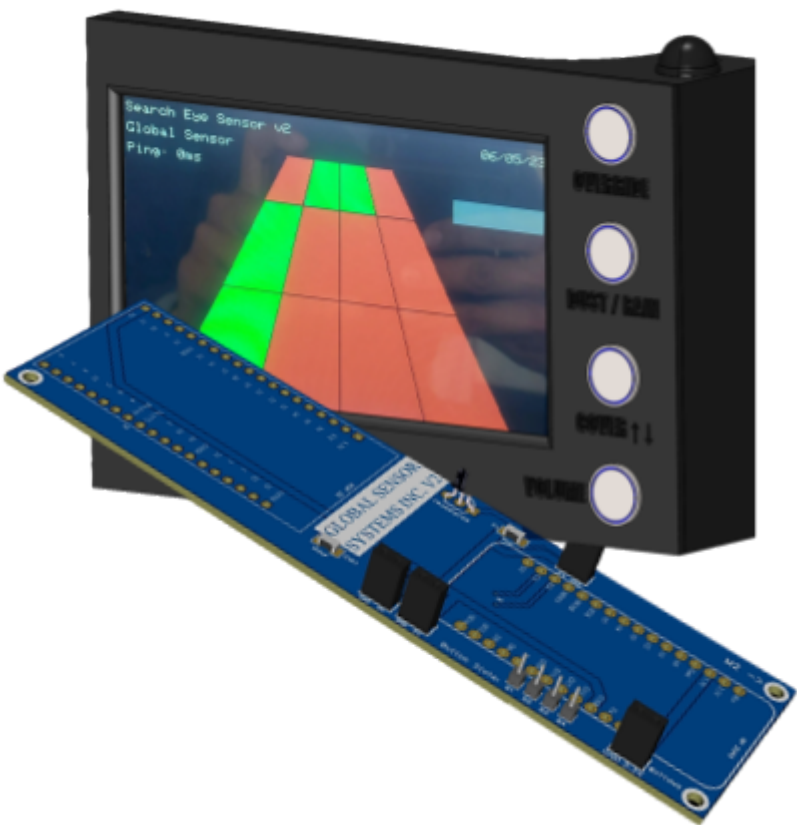


Smart Glass To Smart Watch Video Streaming *Ixana*

- 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

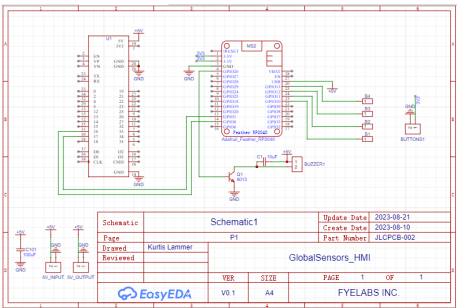


[Last Year's CES Demo](#) or [Some Early Prototypes](#)

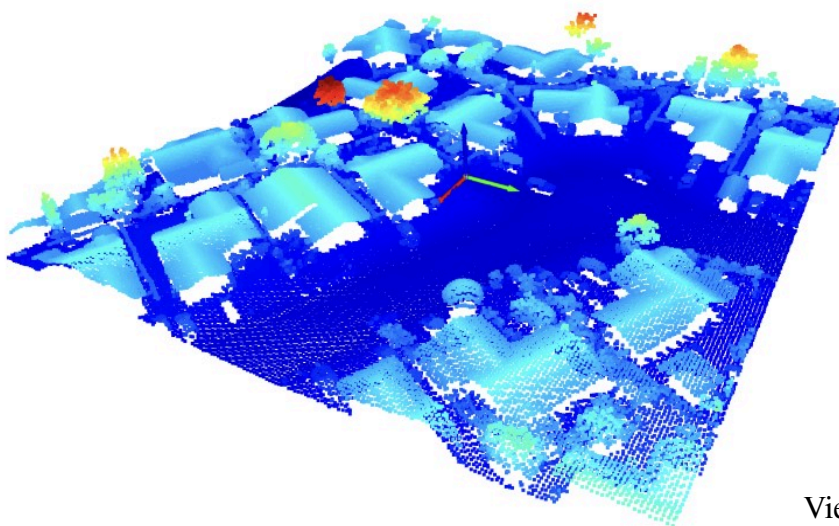


- Created a graphics library in **C++** to display incoming data from the [lidar board](#).
- Designed PCB to support 2 microcontrollers and various other peripherals within the HMI (**EasyEDA**)
- Created a zone mapping algorithm to project a detected object to one of 12 zones so it could be visually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

[View More](#)



LiDAR Data Processing Personal Project

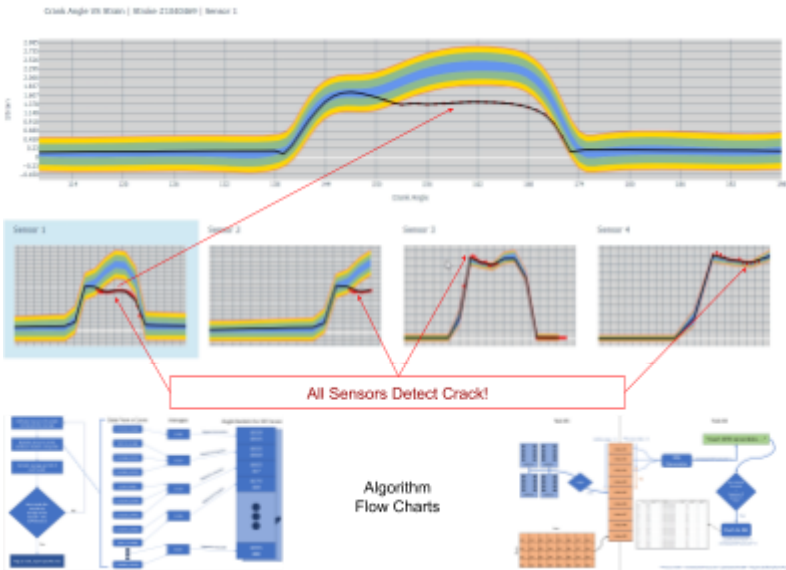


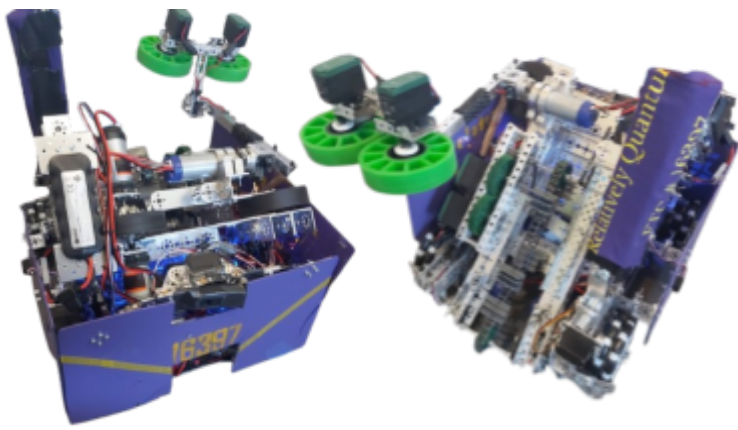
- Used **Python** with laspy, open3D, and numpy
 - laspy: tool for LiDAR data handling and analysis
 - open3D: 3D data processing and visualization
- Manipulation, cleansing, filtering, and analysis of LiDAR point cloud data
- Applied voxel grouping to downsize the data
- Removed artifacts such as dust 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
[View LiDAR implementation on physical hardware](#)

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

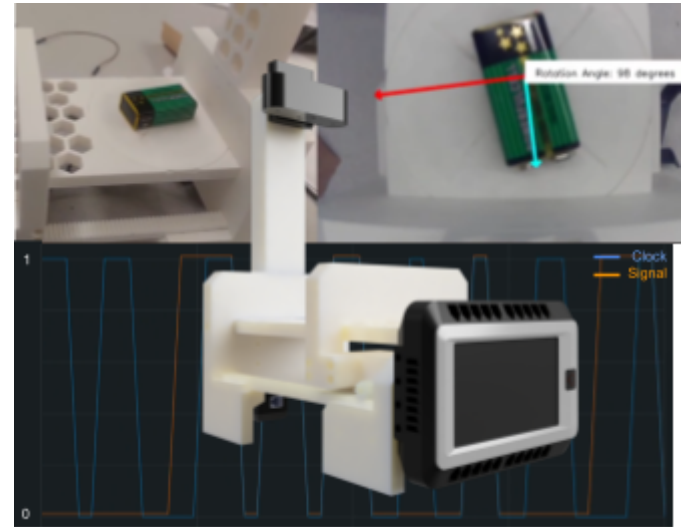
View [demo video](#), [2 page technical report](#), or [top match](#)

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](#)

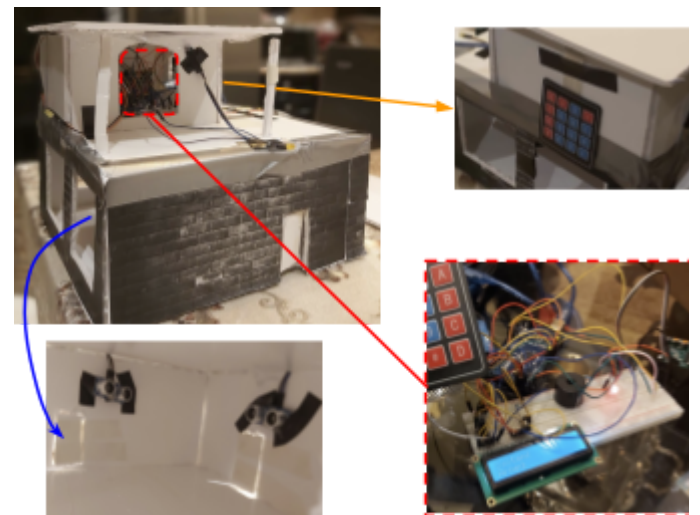


Home intrusion detection system

Personal Project

- Programmed in **C++** using Arduino as the MCU
- Components include lights, buzzers, motion sensors, distance sensors, a key pad and an LCD screen
- Loitering detection with automatic warning lights achieve through Passive Infrared Sensor (PIR) sensor
- Intrusion detection performed using a matrix of distance sensors to determine where the intruder is in the house
 - Matrix of ultrasound sensors detect change in reading from calibrated baseline
 - System needs to be disarmed by entering the password into the keypad following instructions on the LCD

[View Demo Video](#) or view [full presentation](#)



Autonomous egg collecting + sorting robot

Team Project (4 people)

- Wrote code to sweep an area to collect and sort endangered and invasive species of bird eggs (**C++**)
- Used proportional feedback control system (PIDF)
 - Uses a gyro sensor to accurately turn
 - Correct for drift in angle when driving straight
- Used a touch and color sensor to detect and categorize the type of egg collected
- Encoder-based intelligent anti-jam code
- Developed an algorithm based on the percentage of time a color was detected to detect eggs with a 100% accuracy (over 40 trials).

[View Demo Video](#) or view [full engineering design report](#)



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