



- Designed and built a vehicle network emulator to simulate network traffic on real hardware. Developed hardware, firmware, and software for data collection, emulation, and validation.
- > Analyzed 100GB+ of vehicle network data using Python; extracted characteristics into 'streamdb' files that are now used company-wide for data flow analysis and network security investigations.
- > Designed a 4-layer passthrough board to inject errors into 100M/1G automotive Ethernet lines; matched BaseT impedance standards. Uncovered critical issues with missing data and fault detection.
- > Optimized network data analysis speed using the MapReduce paradigm and designed data pipelines for Zoox's compute cluster.
- > Utilized skills in hardware design, firmware development, software engineering (Python, C/C++), and network protocols to create the 'Network Systems Tester' (NST) emulator platform.
- > Designed Test setups and Test Cases to verify the whole Time Synchronization system on Vehicle - including the GrandLeader, Boundary Clocks, and Endpoints. This involved creative test setups, packet analysis, and experimentation with plotting / data analysis.
- > Maintain scripts that verify functionality of Telecommunications and Time Synchronization on the manufacturing line

- > Developed FPGA-based hardware accelerators for Homomorphic Encryption on a Xilinx Zynq ZC702, focusing on key generation, encryption, and large integer arithmetic. Created a novel key-switching scheme for faster homomorphic operations.
- > Designed and optimized Verilog modules, including an implementation of the Miller-Rabin primality test, pseudorandom number generators based on LFSRs, and efficient modular exponentiators.
- > Interfaced FPGA fabric with Zynq PS using AXI IP cores and C code. Integrated ADC modules to process real-world sensor data securely.
- > Extensively debugged and validated designs using Xilinx Vivado's Simulation Suite and Integrated Logic Analyzer (ILA) on both simulated and real hardware.

- Mentored 10 Students in the development of an autonomous robotic arm with computer vision and high level control.
- > Started from basic Python lessons, and got all the way to control using DH-parameters, Finite State Machines, and Computer Vision.
- > Wrote 25+ example scripts and 15+ lessons to help facilitate student understanding of various topics.
- > Wrote a ROS package to integrate computer vision, inverse kinematics, and control into a simple template stack for the students.

- > **3D Modeled** and printed parts for a variety of applications, including product enclosures, accessories, and test rigs. Designed and maintained **25+ 3D printed models**, each with **10+ iterations**.
- > Wrote documentation and drew diagrams for **patents**. Experienced with **technical writing** and research.

Programming	Python	C	C++	Verilog	CUDA	Javascript	Julia	Scala	ROS	Linux/Unix	bash	bazel	OpenCV	PyTorch	Polars	Matplotlib	SciPy	NumPy
E	A	I	I	I	I	I	I	I	I	A	A	A	A	I	I	E	E	E

- Domain Knowledge
 - Networking & OSI
 - TCP/IP Stack
 - Ethernet L1-4
 - FTP/NTF
 - DDS
 - Robotics & Control
 - IK/FK
 - D-H Parameters
 - PID Control
 - RL for Control
 - Computer Vision
 - Calibration
 - CNNs
 - SLAM
 - Data Processing
 - MapReduce
 - Distributed Systems

- Tools & Software
 - Hardware Design
 - Vivado/Vitis
 - Altium
 - Simulation
 - MATLAB/Simulink
 - CTKWave
 - Isaac Lab/Sim
 - CAD/CAM
 - Inventor
 - SolidWorks
 - Onshape
- Practical Skills
 - Fabrication
 - 3D Printing (FDM/SLA)
 - Laser Cutting
 - Wood/Metal shop tools
 - Welding
 - Electronics
 - THS SMT Soldering
 - PCB Assembly
 - Test Equipment

```

/projects
├── FPGA Based Ethernet MAC/PHY
│   ├── description.txt
│   └── Designed and Tested Verilog modules for a Simple 10BaseT MAC and PHY following IEEE 802.3 to transmit fixed Layer 2 Ethernet packets. Debugged and Tested using Verilator and gtkwave. Working on expanding to Layer 3+
├── Chess Robot
│   ├── description.txt
│   └── Using an old 3d printer frame, built a chess robot. Designed motor control circuits, wrote Firmware for low level control, and code for move detection using fiducial markers (AprilTags) and generation. Demo videos on Youtube.
├── Gyro-Boat
│   ├── description.txt
│   └── Built a model boat stabilized by a gyroscope, inspired by Sperry's Gyro-Stabilizer. Demo videos on Youtube.
└── 3 directories, 4 files

```

		Awards & Recognition	
Year	Award	Organization	Details
2024	Hack for Social Impact (1st Place)	UNCCD	Developed "Arboren" to help UN analysts visualize and analyze desertification data. Presented at COP 16 in Riyadh.
2023	SII2024 Conference Paper	IEEE	FPGA-based sensor encryption system paper accepted to Symposium on System Integration in Ha Long Bay, Vietnam.
2022	President's Undergraduate Research Award (PURA)	Georgia Tech	Funded to do research on Homomorphic Encryption with FPGAs. Mentored a student on FPGA development, and presented findings at a research symposium.
2021	NASA SpaceApps Best Use of Technology	NASA	Developed novel space component separation system, used electromagnetic and hydraulic principles to apply a 16kN preload.
2020	MIT COVID-19 Challenge Winner	MIT	Created "COValert" SMS chatbot for rural COVID

	File: US-11287348-B2.pdf
Tire Balance Measurement: Apparatus for measuring imbalance forces of a tire/hub assembly of a vehicle during motion of the vehicle (Issued Mar 29, 2022)	
	File: US-10701266-B2.pdf
Video File Reader: Method for reading out contents of a video file having a predefined video file format (Issued Jun 30, 2020)	
	File: US-10469750-B1.pdf
Motion Data Embedding: Method for embedding motion data of an object into a video file to allow for synchronized visualization of the motion data (Issued Nov 5, 2019)	
	File: US-10284752-B1.pdf
Video-IMU Synchronization: Method for determining a start offset between a video recording device and an inertial measurement unit	

Video-TMU Synchronization: Method for determining a **start offset** between a **video recording device** and an **inertial measurement unit**

Video-TMU Synchronization: Method for determining a **start offset** between a **video recording device** and an **inertial measurement unit**

Video-TMU Synchronization: Method for determining a **start offset** between a **video recording device** and an **inertial measurement unit**

Video-TMU Synchronization: Method for determining a **start offset** between a **video recording device** and an **inertial measurement unit**

Video-TMU Synchronization: Method for determining a **start offset** between a **video recording device** and an **inertial measurement unit**

user@coohas:~\$