Neil Chibber

neilchibberr@gmail.com | in/nchibber | Washington, DC | Open to relocation

Education

University of Maryland, College Park

B.S. Electrical Engineering, Minor in Statistics — *Expected Graduation: May 2029*

- **Relevant Coursework**: Digital Logic Design, Computer Architecture, Microelectronics, Circuit Theory, Signals & Systems, Digital Signal Processing, Control Theory, Communication Network Theory, Power Systems

Technical Skills

Languages & Software: C/C++, Python, MATLAB/Simulink, Verilog/SystemVerilog, R/RStudio, Linux, Git
Hardware Design: PCB Design (Altium, KiCad), SPICE Simulation, Circuit Analysis, Antenna Design
Test & Verification: Hardware-in-the-Loop (HIL) Testing, Logic Analyzer, Oscilloscope, Network Analyzer, Spectrum Analyzer, Signal Generator, DMM, Thermal Camera, Micro-Soldering

Experience

UMD LOOP: Mars Rover Team — College Park, MD

Lead Embedded Engineer — *Sep 2025 – Present*

- Lead a team of **10+** engineers building embedded hardware and firmware on a Mars Rover for the **University Rover Challenge**, integrating **robotic arm**, **life detection**, and **autonomous navigation** subsystems.
- Designed and implemented a lightweight **CAN bus protocol** with message arbitration and acknowledgment, linking **10+** sensors and actuators across the rover's sample-collection subsystem for command and telemetry traffic.
- Brought up two custom **STM32** PCBs driving stepper and servo motors, writing **C** firmware on **Zephyr RTOS** with **Devicetree**-based hardware configuration to coordinate sample collection across multiple actuators.
- Validated the life detection subsystem through **hardware-in-the-loop (HIL)** testing, debugging with logic analyzers and oscilloscopes to reach **99.9%** data integrity across **10,000+ CAN frames** in high-noise conditions.

Alchemity — College Park, MD

Systems Integration Intern — *Jun 2026 – Aug 2026*

- Designed a reactor membrane test skid to **TRL 5**, specifying its instrumentation and balance-of-plant sensors.
- Authored a **DFMEA** covering **300+** failure modes, identifying and mitigating design risks before fabrication.
- Reverse-engineered a third-party **SOFC** test rig, documenting its architecture and **P&ID** to verify compliance.

Electrical Engineering Intern — *Jun 2025 – Jan 2026*

- Designed and tuned a closed-loop **PID** stepper motor controller for membrane tube extrusion, holding speed within **1%** of target RPM across multi-hour fabrication runs to maintain uniform wall thickness along each tube.
- Built a **Python** data acquisition pipeline logging thermocouple and motor telemetry over **serial**, automating test report generation and enabling data-driven tuning of the membrane tube extrusion process parameters.
- Built an **ESP32**-based relay controller with a custom enclosure, **automating** a multi-day ball mill operation.
- Debugged thermal overshoot in a furnace **PID** controller, isolating faulty relay outputs with a **DMM**, redesigning the relay control scheme, and reprogramming alarm thresholds to eliminate the fault and prevent recurrence.

Projects

3-DOF Robotic Wrist Controller PCB | ***STM32, KiCad, C*** — Feb 2026 – Mar 2026

- Designed and fabricated a custom **KiCad** PCB controlling a 3-DOF robotic wrist (pitch, yaw, roll) and gripper, integrating motor drive circuitry for **4 DC motors** and **4 magnetic encoders** on a single compact board.
- Wrote **C** firmware for an **STM32** implementing **PID** position control with real-time encoder feedback.
- Ran hardware/software integration testing, validating closed-loop tracking accuracy within **±2°** on all three axes.

SOFC Thickness Optimization Tool | ***Python, MATLAB/Simulink, Git*** — Aug 2025

- Built a **Python** GUI tool that determines optimal material orientation, optimizing layer thickness to **±10 µm**.
- Modeled cell performance in **MATLAB/Simulink** (Nernst voltage, ohmic and activation losses) to quantify how electrolyte and electrode thickness trade off against cell power output and guide fabrication targets.
- Automated **MATLAB** parsing of experimental datasets, feeding the model and eliminating manual calculation.