

Audrey Reinhard

AudreyDReinhard@gmail.com | (951) 203 - 3028 | [LinkedIn](#) | [GitHub](#)

PROFESSIONAL SUMMARY

Embedded systems and robotics engineer with hands-on experience developing firmware, debugging hardware–software interfaces, and building Linux-based robotic systems. Experienced in real-time embedded programming, sensor integration, and system-level debugging using oscilloscopes and protocol analyzers. Passionate about prototype-to-product engineering and solving complex low-level system challenges.

SKILLS

Embedded & Low-Level Systems: C, C++, Embedded Linux, Real-Time Systems, UART, SPI, I2C, memory management, driver development

Robotics and Hardware Integration: Sensor Integration, hardware bring-up, serial protocols, system debugging, packet design

AI/ML: TensorFlow, TinyML, BERT, CNN inference on embedded systems, NPU utilization

Tools & Debugging: Oscilloscope, Logic Analyzer, Multimeter, UART/SPI protocol decoding, performance profiling

Software & Automation: Python, Bash, Git, GitHub Actions, CI/CD pipelines, SQL

EDUCATION

University of California, Riverside

Expected Graduation: July 2026

B.S. in Computer Engineering GPA - 3.8/4.00

Honors: UC Regents Scholar, NSF S-STEM Scholarship

Relevant Coursework: Embedded Systems, Electronic Circuits, Logic Design, Computer Architecture, Operating Systems, Data Structures

PROFESSIONAL EXPERIENCE

Robotics Lab - UCR Library, UC Riverside

September 2024 – Present

Robotics Lab Student Lead

- Develop embedded firmware in C/C++ for multi-board robotic systems, integrating microcontrollers with Linux-based compute modules over UART, SPI, and I2C.
- Perform system bring-up and debugging of hardware–software interfaces using oscilloscopes, logic analyzers, and protocol decoders.
- Designed and implemented a magnetic-stripe device driver and communication protocol, enabling secure equipment access tracking and increasing equipment utilization by 93%.
- Diagnose and resolve timing, packet integrity, and synchronization issues in distributed embedded systems.
- Created technical documentation and automated test workflows improving system reliability and reproducibility.
- Taught a 6-week robotics engineering curriculum covering Linux robotics control, serial communication, and embedded programming.

National Science Foundation, UC Riverside

June 2024 – July 2024

Data Science Fellow

- Built Python automation tools for audio preprocessing pipelines (noise filtering, segmentation), reducing manual processing time 72%.
- Fine-tuned a DistilBERT classifier achieving 83% accuracy, demonstrating foundational ML and model optimization skills.
- Collaborated with cross-disciplinary researchers, integrating model outputs into scientific analysis workflows.

The Three Logistics, Riverside, CA

February 2024 – May 2024

Full Stack Development Intern

- Built backend features using SQL, REST APIs, and server-side processing, improving transaction throughput by 25%.
- Strengthened data reliability and error handling within shipping workflows, contributing to operational scalability.

PROJECTS

Autonomous Rover Navigation System

ESP32 + NavQPlus + Arduino Nano | **Oct 2025 – February 2026**

- Integrated IMU, GPS, and embedded compute hardware into a multi-sensor robotic system.
- Implemented a real-time computer vision lane-segmentation pipeline for autonomous lane centering.
- Developed modular ROS2 nodes for synchronized sensor acquisition, processing, and real-time telemetry streaming.
- Implemented register-level motor driver control, PWM-based closed-loop speed regulation, and precision servo steering.

- Diagnosed system-level faults across hardware interfaces, firmware, and ROS2 middleware layers.
- Validated spatial and temporal synchronization of sensor data during live field testing.
- Improved system reliability and fault tolerance through structured debugging, logging, and iterative integration testing.

Space Invaders - Bare-Metal AVR, UCR

May 2025 – June 2025

- Built a real-time embedded game in C using a synchronous task scheduler on bare-metal AVR.
- Programmed ST7735 LCD graphics over SPI, joystick ADC sampling, and PWM-driven audio output
- Optimized SPI bandwidth and rendering pipelines to reduce display tearing and improve frame timing.
- Implemented modular firmware architecture with hardware abstraction and interrupt-driven timing control.

Chronotype Scheduler, UCR

February 2025 – March 2025

- Developed a Qt-based scheduling application using modular SOLID-compliant architecture.
- Wrote unit tests using GoogleTest and managed development through GitHub Agile sprints.