

Ronny N. Ismael

Email: ronnyismael@gmail.com | M: (209) 614-9525 | Website: ronnyismael1.github.io | LinkedIn: linkedin.com/in/ronnyismael

EDUCATION

Bachelor of Science in Computer Engineering

Graduation May 2024

San Jose State University | San Jose, CA

Certification: Cybersecurity

Coursework: Algorithms & Data Structure, Circuit Analysis, Digital Design I-II, Microprocessor Design, Computer Networks I, Compiler Design, Computer Architecture and Design, Operating Systems, IoT Platforms, RT Embedded CoDes, Database Systems I

Associate in Science in Mathematics & Physics

Graduated May 2021

Modesto Junior College | Modesto, CA

Coursework: Problem Solving/Programming, Calculus I-III, Linear Algebra, Differential Equations

TECHNICAL SKILLS

Hardware Description Languages: Verilog, SystemVerilog

Programming Languages: C, C++, Python

Software Development: Quartus Prime, Vivado, ModelSim, Embedded Protocols (UART, SPI, I2C, CAN)

System Tools: Linux (SCP, SSH), LogicWorks, Git, Jira, Confluence

Lab Equipment: Oscilloscope, Function generator, Multi-meter, Signal generator, Power supply, Logic/Protocol Analyzers

WORK EXPERIENCE

Engineering Technician | Nextest division of Teradyne

July 2022 - Present

- Engaged in system-level debugging, troubleshooting, and repairs of semiconductor testing systems.
- Managed record retention, international logistics, and data integrity for Magnum systems, employing SQL for database management and creating data analysis and automation scripts.
- Developed a React Native application using JavaScript and Expo Camera for QR code scanning, streamlining board repair and inventory management, integrated with Firebase Firestore for real-time data persistence.

Robot Technician, Site Lead | Starship Technologies

Sep 2020 - June 2022

- Served as the primary point of contact for Mountain View and Intuit operations, managing app support and leading teams through expansion projects, marketing events, and menu configurations.
- Utilized Python in Linux environments, employing SCP and SSH for secure data transfer and remote system management, to ensure optimal performance of the autonomous robotic fleet through rigorous debugging and hardware diagnostics.
- Specialized in complex PCB repairs and operations, including Raspberry Pi, ELMOs Controller PCBs, and TEGRA PCBs, integrating IoT principles for robotic functionality and using Jira, Confluence, Excel, and MRPeasy for project tracking and inventory management.
- Developed technical documentation on AMD and TEGRA components, employed sensor calibration techniques for precision in robotics, and managed staffing and scheduling for the team.

PERSONAL PROJECTS

github.com/ronnyismael1

RT Face Recognition Security System on Embedded Platform | Verilog, Python, FPGA, Raspberry Pi, OpenCV

- Awarded \$5,000 in the David Student Scholarship for outstanding innovation.
- Developed a smart system utilizing facial recognition for secure access, optimized for Linux but adaptable to other environments.
- Employed Python and OpenCV for real-time face detection and identification, integrating hardware control with RasPi and FPGA.
- Designed a facial recognition system that extracts and compares facial encodings from sample images, enabling individual identification through real-time processing of video feeds with advanced algorithms and OpenCV.
- Optimized for performance and reliability by refining video processing and multithreading techniques.
- Raspberry Pi deployment with insights on SSH usage and graphical interface management through X11 forwarding.

Advanced RAM Module Design | Verilog, Quartus Prime, ModelSim

- Engineered and simulated single, dual, and true dual-port RAM modules in Verilog.
- Utilized Quartus Prime for design and ModelSim for simulation
- Achieved successful data storage and retrieval operations.
- Developed comprehensive testbenches for each module type.

Concurrent Word Frequency Counter | C, pthreads, GCC, Valgrind, VS Code, Git, GDB

- Developed a multithreaded application in C, utilizing the POSIX threads library (pthreads) for concurrent processing of large text files.
- Implemented hash table for efficient word count tracking
- Utilized mutexes and condition variables to ensure thread-safe operations on shared data structures
- Employed dynamic memory management in C to handle data storage.