

Sargis Yonan

CONTACT

E-mail: sargis@yonan.org
Website: sargis.yonan.org,
github.com/SargisYonan
Phone: +1 (209) 418-SARG

SKILLS

Software Engineering: C, C++, Python, Java, Real-Time Operating Systems, Embedded Software Design, x86/MIPS/ARM Assembly

Computer Engineering: Computer Architecture, Digital Logic Design

Embedded Systems: ARM, AVR, PIC, CAN, I2C, UART, SPI, sensor integration, protocol debugging hardware

Robotics Engineering: Optimal State Estimation, Sensor Fusion, Feedback Control, Computer Vision, UAVs

Mechanical/Electrical Engineering: Sensor Design, Hardware Filter Design, Electro-Mechanical System Design

Software/Libraries: Linux/UNIX, MATLAB, OpenCV, TensorFlow, SOLIDWORKS, Eagle CAD, PSpice

Computational & Applied Mathematics: Kalman Filters, Control Theory, Linear Dynamical Systems, Machine Learning, Geostatistics, Graph Theory, Frequency Domain & State Space Analysis

EDUCATION

M.S. Computer Engineering

Sep 2017 — Dec 2018

University of California, Santa Cruz

- Emphasis in Robotics Engineering & Control Theory
- Thesis: Aerial Field Prediction and Path Planning Using The Universal Kriging Method
- Autonomous Systems Lab

B.S. Computer Engineering

Sep 2013 — June 2017

University of California, Santa Cruz

- Robotics & Control Theory Concentration
- Project/Research Topics: Autonomous UAVs, Sensor Node Networking, Power Systems
- Graduated with Honors in Major

Minor in Computer Science

Sep 2013 — June 2017

University of California, Santa Cruz

WORK EXPERIENCE

SpaceX

June 2018 — Present

Satellite Development Intern

Flight software engineering intern working on projects involving embedded software and hardware systems.

UC Santa Cruz

Jan 2018 — June 2018

Teaching Assistant for Senior Design Sequence

Teaching assistant for twenty teams of senior-level engineering students taking their capstone sequence. Meetings with each team for one hour a week where project details are discussed. Assisted teams in their system design, architecture, low-level implementations, and engineering approaches. The projects ranged from autonomous robots to large-scale mesh networking systems.

CityBlooms Urban Micro Farms

June 2017 — Oct 2017

Embedded Software and Hardware Engineering Intern

Designed and implemented various sensor and peripheral drivers for shipping systems that acquired data from live farming and micro-agricultural environments. This involved writing various system level modules in bare-metal C/C++ on a microcontroller as well as C and Python on a Linux system. Designed a deployment automation system and wrote system level code for a sensor node network.

Pearl Automation

Jan 2016 — Dec 2016

Firmware Engineering Intern

Developed software components and features on both an ARM based controller with a Real-Time Operating System, and a system with an Embedded Linux Operating System. Developed a dynamic frequency scaling algorithm for a processor that drastically improved the battery life of a device as well as other physically apparent aspects in a shipping product.

PROJECTS

Jay - The custom flight controller software package written for my senior design project, PAVx. This package includes the feedback controller for a quad-copter flight control alongside the drivers and protocols for a 9DOF IMU, IR camera, wireless communication, and GPS. The software was intended to run on a custom made all-embodying printed circuit board which contained an Atmel AVR ATmega 328P, but could also be run on any AVR microcontroller with slight modifications. Jay enables a pod of drones to communicate and scan an unknown area both aurally and autonomously. The aerial pod-aggregated data feeds to a display on a ground station in real time.

github.com/PAVx/jay

snOS - A framework built to assist the creation of low-cost and low-powered IoT systems. With networking capabilities built-in, two or more microcontrollers can run snOS and create a mesh of sensor nodes that can ultimately connect to the internet. A controller in a snOS network could subscribe and publish messages to another snOS device on the network. Messages have the ability to interrupt the operation and thread execution of other controllers in the network, creating a real-time event-driven IoT network.

github.com/SargisYonan/snOS