

Marco Acea

About



aceamarco@gmail.com



(786) 470-9891



marco-acea



aceamarco

Programming Languages

Python Golang C

HTML/JS/CSS

System Verilog

Software/Frameworks

Selenium Django

Kubernetes

Google Firestore

Argo CD

Education

Carnegie Mellon University

08/2018 - 12/2022

Bachelor's of Science in Electrical and Computer Engineering

- Structure and Design of Digital Systems
- Signals and Systems
- Web App Development
- Intro to Computer Systems
- Intro to Embedded Systems
- Fundamentals of Embedded Control

Experience

Intuit

05/2020 - Present

Software Engineering Intern

- Developed a new feature project, the ArgoCD AppSource Controller, for the Argo Kubernetes Organization

23andMe

07/2020 - 09/2020

Software Engineering Intern

- Developed user interface using Flask API endpoints to display information about hosted applications (statuses, owners, authentication tokens, etc.)

Projects

PD Temperature and FSM Rollercoaster Controller

01/2022 - 05/2022

Fundamentals of Embedded Control

- Developed a Proportional-Derivative controller that approaches and holds a desired temperature within a closed chamber
- Developed a finite state machine controller that manages model rollercoaster carts through the track
- Designed electrical circuit and built testbenches for both controllers

ArgoCD AppSource Controller

05/2021 - 08/2021

- Kubernetes Custom Resource Definitions that gives under-privileged users permission from admins to create ArgoCD applications automatically
- Supports multiple ArgoCD Project "profiles" that limit what actions sub-users can make with their application.
- Currently going through adoption process for the argo-proj-labs collection of vetted community projects

UART/I2C Controller and Real Time Kernel

01/2021 - 05/2021

Intro to Embedded Systems

- Used Memory Mapped I/O (MMIO) to build UART and I2C peripheral drivers
- Built an acoustic "clap" detector that runs on the STM32 Nucleo Board using I2C for acoustic data and UART to print sensor data to the console
- Designed multi-threaded Real Time Operating System (RTOS) with context switching, mutexes, and enforced fixed priority scheduling.

Malloc and Unix Tiny Shell

08/2020 - 12/2020

Intro to Computer Systems

- Implemented dynamic memory allocation library in C using both explicit and segregated free lists
- Implemented a Unix shell program that supports job control using process control and signalling.

