

# Colton Acosta

404.430.1346 • cacost12@asu.edu • US Citizen • coltonacosta.com • linkedin.com/in/colton-acosta/

## EDUCATION

---

**B.S.E, Electrical Engineering**  
Arizona State University, Tempe, AZ

May 2023  
4.00 GPA

## TECHNICAL SKILLS

---

**Design/Modeling:** Altium, LTspice, KiCAD, MATLAB/Simulink, Cadence, NX, SolidWorks  
**Hardware:** Microcontrollers, Soldering (SMD), Multimeters, Oscilloscopes, Verilog, FPGA, Function Generators  
**Software:** C, C++, C#, Python, Assembly, Linux, Git, Make, ARM, Visual Studio

## EXPERIENCE

---

**SpaceX: Hardware Development Engineer, Starship Electronics** October 2023–Present

- Responsible Engineer for a gyroscope that provides inputs to bending compensation algorithms on Starship Booster
- Responsible Engineer for an engineering camera used to stream video of key events in flight to inform design
- Conducted environmental testing including thermal cycling, vibration, shock, leak, humidity, and radiation
- Developed and released PCBs and avionics designs using Altium for schematic capture and layout
- Reduced cost per vehicle stack by more than 150k by developing new technologies for rocket usage such as MEMs gyroscopes, video encoder ASICs, and cutting edge image sensors
- Doubled camera yield by root causing production test failures such as boot failures, loss of streams, and corrupted video packets
- Brought up new PCB designs by testing power trees, serial interfaces, and sensors using oscilloscopes and multimeters
- Coordinated product development by communicating with production, supply chain, and engineering

**Raytheon Missiles and Defense: Software Engineer** July 2023–October 2023

- Implemented new features for a .NET based WebUI used for automating software (SIL), computer (CIL), and hardware (HIL) in the loop missile tests using C# and Javascript
- Wrote a C# method to automatically locate library dll files to eliminate the need for manually setting the path
- Added a new page to the WebUI for monitoring background test status by querying the API using HTTP requests
- Added Javascript keyboard event listeners user interface to implement quit and save keyboard shortcuts

**Undergraduate Research Assistant: Advanced CMOS** January 2022–April 2023

- Assisted in the data collection of a 90nm CMOS process in adverse environments for use in satellite imaging systems
- Designed over ten PCBs for mounting test devices and interfacing with a semiconductor parameter analyser
- Constructed a CMOS measurement setup rated for cryogenic temperatures to emulate temperature conditions in space
- Designed and simulated a transimpedance amplifier to amplify CMOS drain currents and filter high frequency noise

**Garmin Aviation: Embedded Software Engineering Intern** May 2022–August 2022

- Developed certification software for a new Vulkan graphics driver to be used in safety-critical avionics systems
- Wrote unit tests with randomized test vectors in C to test the GPU driver source code with maximal coverage
- Debugged compiler errors of ARM and Windows builds using Visual Studio and MSBuild XML schemas
- Resolved runtime errors caused by randomized test vectors by analyzing the source code functions and manually setting up data structures, pointers, arrays, and buffers

**Sun Devil Rocketry: President and Avionics Team Founder** August 2021–May 2022

- Oversaw all activities of a technical student organization with three rocket propulsion teams, two amateur rocketry teams, a K-12 outreach program, and over 50 members
- Facilitated all project development by holding meetings and design reviews, writing budget proposals, organizing launch logistics, mentoring, and maintaining industry/university relations
- Founded a new avionics team to design the club's first flight computer and promote the development of electrical and software engineering skills among students interested in the aerospace industry

**Pyramid Technologies, Inc, Mesa, AZ: Electrical Engineering Intern** May 2021–August 2021

- Evaluated bill validation errors of a bill acceptor's firmware using an in-circuit debugger and assembly source code

- Revised a switching power supply and serial opto-isolator PCB to be usable with multiple bill acceptors
- Qualified new optocouplers by measuring logic levels and slew rate for ambient temperatures ranging from 0 to 60°C
- Designed a new PCB to protect test fixture pins from overvoltage and overcurrent conditions using schottky diodes and a PTC resettable fuse
- Collected and analyzed phototransistor data on over 150 LEDs to find a viable bill validation LED that would work at scale production without firmware modifications
- Added serial indication LEDs, signal buffering, inrush current protection, and short circuit protection to a USB to MDB serial interface PCB
- Designed a revised bill acceptor software development board by adding an electronic fuse to alleviate faulty supply/loading conditions and provide power supply fault indication
- Performed DC load testing on a new 120V AC power supply to measure power trace voltage drops at full load
- Conducted electrical tests and wrote qualification documents for replacement PCB parts to resolve procurement issues
- Tracked project progress and managed feedback on PCB designs and layouts with git and bitbucket
- Resolved electrical issues with dysfunctional test fixtures and equipment used by engineers and production staff
- Wrote Python scripts to calculate external component design values from input specifications and datasheet guidelines

## PROJECTS

---

### Sun Devil Rocketry: Flight Computer

January 2021–Present

- Developed a flight computer to log flight data and implement parachute recovery in amateur and high power rockets
- Equipped the computer with an ARM Cortex-M4 microcontroller, a barometric pressure sensor, external flash, and a USB interface
- Created three PCB layouts to separately optimize for cost, size, and test by using different design rules (DRC)
- Wrote APIs in C for low level control of external hardware using UART, I2C, SPI, and GPIO
- Wrote a data-logger application to collect flight data, and successfully recovered flight data from several launches
- Implemented the recovery software using a state machine architecture to transition between programming, idle, in-flight, and post-flight states
- Implemented apogee detection, main parachute altitude detection, and landing detection algorithms using a FIFO buffer, Z-transforms, and tuned thresholds
- Tested the recovery software using a vacuum chamber to simulate the pressure loss that occurs during ascent

### Sun Devil Rocketry: Liquid Rocket Engine Controller

August 2019–May 2023

- Architected an avionics system which successfully supported ASU's first static hotfire attempt of a liquid rocket engine
- Designed and built an engine controller for to manage engine hardware and communicate with the ground station
- Designed the PCB using an ARM Cortex-M7 microcontroller, a switching power supply, external flash, an SD card, ignition terminals, sensor peripherals, a USB interface, and an RS485 command and control interface
- Developed ignition and data-logging APIs in C to abstract low-level hardware control functionality
- Programmed a Python interface for real-time visualization of engine state and sensor measurements
- Amplified pressure transducer differential outputs to measurable ranges using a programmable amplifier circuit in order to save upwards of 10% of club funding in new sensor costs
- Wrote sequencing code in C to open/close engine valves in a timed order to automate a static hotfire test
- Wrote an interrupt driven command processor to communicate with the remote ground station over ethernet

### Sun Devil Rocketry: Valve Controller

Spring 2022

- Designed, built, and tested a controller to actuate rocket engine valves using an ARM Cortex-M7 microcontroller, solid state relays, a pulse interface, and motor sensors.
- Calibrated valve shaft initial positions using an optoelectronic photogate sensor with customized form factor
- Designed an optically-isolated voltage monitoring circuit to alert the controller when solenoid power is lost
- Programmed the controller in C to process valve actuation commands from the main engine controller
- Developed a valve control API in C to implement basic valve actuation functions for solenoids and servo valves

### Flow Control Valve Actuator Control System

Fall 2020

- Designed and built a closed loop control system for a valve actuator for use in flow throttling applications
- Examined the relationship between Pulse Width Modulation duty cycle and steady state shaft speed to derive a controller output signal with a linear transfer function from controller output to shaft position
- Characterized the plant transfer function with a series of step response experiments
- Implemented a saturated PI controller with integrator clamping in C++, and simulated the performance using Simulink to meet specifications of zero steady state error of step inputs and complete rejection of step disturbances
- Built the actuator using a brushed DC motor, coupling shaft, Arduino, and quadrature rotary encoder for feedback