DANIEL RUSH

Senior Software Engineer

rush.daniel95@gmail.com San Francisco, California daniel-rush95 theDrsh

EXPERIENCE

Lead Senior Firmware Engineer

Carbon, Inc.

June 2017 - Current Redwood City, California

- Led the firmware engineering team at Carbon.
- Delivered the Automatic Operation (AO) Backpack as a solo firmware engineer.
- Served as the main firmware engineer on the M3 and M3 Max project—Carbon's smartest, fastest, and most cost-effective printer to date.
- Built a Brushless DC Motor Controller, critical to all Carbon products, from the ground up.
- Developed Hardware-in-the-loop (HIL) testing to identify and fix bugs before shipping.
- Brought up many new circuit boards and developed an automated HIL system for board bring-up.
- Established Over-The-Air (OTA) updates for motors and other subsystems in the printer.

LANGUAGES

C++	●●●●●
C	●●●●●
Python	●●●●○
Verilog	●●○○○
Bash	●○○○○

PROFICIENCIES

I2C · SPI · UART · USB · MQTT · Protobuf · Docker · System Integration · Board Bring-up · Linux · Bazel · Cmake · Jenkins · HIL Testing · RTOS · YAML · Mako

EDUCATION

B.Sc. Mechatronics Engineering

California State University, Chico

Aug 2013 - May 2018

B.Sc. Computer Engineering

California State University, Chico

Aug 2013 - May 2018

PROJECTS

Automation Operation(AO) Backpack

Carbon

- Worked with Mechanical and Electrical engineers to design behaviors and operate hardware to make printers fully autonomous.
- Led the effort to move from multiple motor vendors to utilizing a single motor controller (also a project of mine) to reduce costs and increase our ownership of code, as well as enable OTA updateability and logging.
- Delivered the product in the shortest timeline ever seen at Carbon—on time and under budget as a team of one.
- Drove firmware development using customer feedback in the cycle and continued to release updates based on that feedback, improving the product over time.

M3 and M3 Max

Carbon

- Collaborated with hardware engineers, product managers, and print development engineers to create firmware requirements.
- Developed all new subsystems in-house to create Carbon's "smartest" printer.
- Wrote firmware with extensibility, ease of debugging, and robustness in mind to create a lasting product.
- Addressed challenges, including supply chain shortages, which necessitated creating new drivers from scratch on a tight timeline.

Brushless DC Motor Controller(BLDC)

Carbon

- Developed a motor controller from concept to production, featuring cascaded control loops for torque, velocity, relative position, and absolute position control.
- Created drivers for USB logging, UART API interface, SPI-controlled FET-driver chip, and I2C EEPROM for hardware version control.
- Designed and implemented hardware fault detection and safety states to prevent damage to mechanics and electronics.
- Architected Hardware-in-the-loop(HIL) testing using a dynamometer for system and firmware validation.

OTHER PROJECTS

Board Testing Rig

Carbon

- Implemented a method for electrical engineers to hand off new circuit board designs by allowing them to describe hardware changes using YAML
- YAML files from Electrical Engineers were able to be used to test any new circuit board at a low-level to test for manufacturing correctness
- Created a bridge so that firmware team could take new YAML file and use it to make changes to code and then test the new code against the file so that there was a single source of truth for the electrical hardware/firmware testing and development.

Microtest

Carbon

- Developed a firmware image which was able to test core interfaces of the microcontroller item Server-side application to interface the microcontroller with gtest on Linux, using **Nanopb**
- Developed Jenkins-based job to automatically run tests upon pull request.
- Created ability write firmware that could be automatically called and tested via Jenkins.
- Recorded statistics like execution time and logs on target processor.

Code Generation Projects

Carbon

- Worked to develop python-generated firmware to automate different parts of the firmware stack:
  - Server <-> embedded system communication.
  - microcontroller <-> microcontroller communication and commands.
  - Hardware faults and alerts generated through entire software stack micro -> server -> cloud.

Board Bootloader and Over-The-Air(OTA) Update Software

Carbon

- Helped build the bootloaders for smart subsystems of printer
- Extended OTA updates to incorporate updating all smart subsystems.
- Helped with designs for updater that manages all the firmware versions across the printer/device to only update applicable subsystems.