

# Daniel Seabra de Andrade

---

<b>Address</b>	444 Saratoga Avenue APT 30-J, Santa Clara, CA, 95050	<b>Mobile Phone</b>	+1 (626) 531 5120
<b>Date of Birth</b>	29 <sup>th</sup> July 1994	<b>Email</b>	hoverstandis@gmail.com
<b>Nationality</b>	American		

## Personal Profile

I'm looking for challenging opportunities in radio frequency, control system, and/or high-speed digital design. I am not looking for a software development job at the moment.

## Education

**2012-2016** BS in Electrical Engineering - California Institute of Technology (Caltech) [GPA: 3.7]

## Recent Employment History

**2016 - Present** Mimosa Networks  
*Staff Hardware Engineer (2018-Present), Hardware Engineer (2016-2018)*

I am an integral part of the hardware team at Mimosa. My core responsibilities are in design and testing; these do not just involve a proficiency in schematic capture and board layout tools and a familiarity with general digital, analog, and RF concepts, but also a working knowledge of manufacturing processes, RF calibration software development, and certification. At Mimosa I've earned "Most Valuable Employee of the Quarter" for Q4 2017 and have been trusted to travel to our factory in Shenzhen, China for factory product bringup and hardware debug during critical quarter-end periods. Major accomplishments include:

- Design and full DVT of a 2x2 5GHz 802.11ac client radio
- Full (including RF) characterization and verification of a 4x4 24GHz 802.11ac backhaul radio, including any necessary tuning and optimization. I was also responsible for the full FCC/ETSI certification (intentional and unintentional emissions, safety, etc.) of this product.
- Design of an 8x8 5GHz 802.11ac access point radio
- Bringup of three different products at the contract manufacturer during the last three weeks of the quarter
- DVT and tuning of three different (client, access point, and backhaul) radios and one 11GHz backhaul radio including addressing ESD and EMI concerns as well as RF performance issues
- Participated in every schematic and layout design review meeting for Mimosa when not in China
- Mentored two interns

**Summer 2015** Jet Propulsion Laboratory (NASA), 4800 Oak Grove Drive, Pasadena, CA, United States  
*Intern*  
Helped develop a multi-agent framework for executing behaviors on autonomous ships.

## Relevant Knowledge

### ■ Hardware Topics

*General:* schematic capture, PCB layout, and BOM tools

*Components:* Microcontrollers, Ethernet PHYs, DDR3, baseband processors, power supplies & feedback networks, etc.

*RF:* RF transceivers, FEMs/PAs/LNAs, filters (discrete as well as on-board), mixers, baluns, couplers, splitters/combiners

*Interfaces:* WiFi, PCIe, Ethernet, RGMII, MDI/MDIO, SGMII, UART, SPI, I<sup>2</sup>C

*Equipment:* Oscilloscopes (differential or single-ended), spectrum analyzers, power meters, VNAs, signal generators

### ■ Programming Languages

*Fair warning:* I haven't programmed in a while, but I could get back to it if necessary.

*Python, Java, VHDL, C, C++, Bash*

*Assembly Language:* various flavors [x86, NIOS, ARM]

Less experience with: *PHP, JavaScript, HTML, CSS, Haskell*

### ■ Coursework

*EE 51, EE 52, EE 53* - embedded system design and implementation

*EE 44, EE 45, EE 40* - circuits and systems, semiconductor physics

*EE 113, CDS 110, CDS 112* - advanced feedback circuits and control system design

*EE 111, EE 112, EE 160* - signal processing and some digital filter design, communications

*EE 135, EE 125* - power systems, digital circuit design (VHDL)

*ACM 95ab, ACM 116* - Complex analysis, ODEs, Fourier & Laplace transforms, and stochastics

*CS 1, CS 2, U. of I. Courses* - basic programming, intro. to algorithms and data structures

## Projects

- **Embedded Systems**

*Guitar pedals* - I designed analog distortion and compression pedals for fun.

*FSAE Car* - Hardware team lead for Caltech Racing's electric vehicle for the 2016 FSAE competition. In addition to leading the team I was responsible for the core board hardware which processes all the control code as well as reading data from an IMU, a temperature sensor, and writing to a color pixel display.

*Bubbly* - 20 MHz bubble measurement device running on an Atmel AT91RM9200 processor. This device detects the edges of an incoming (noisy, variable amplitude) square wave and outputs the void fraction (essentially the duty cycle) and the average high and low voltages to a character display. There were plans to output the incoming signal to a computer through USB but that fell through because embedding USB is nontrivial (actually quite difficult in ARM assembly). All the hardware, software, and PCB design was my own. Finished August 2015.

*Automatic Dog Bowl* - Dog bowl that opens automatically whenever the correct dog is nearby. The identification is done via IR and detection of the presence of a large object is done through an ultrasonic sensor. Since the project had no processor (all done through hardware), there was no code. Finished June 2015.

*DS-94 Digital Oscilloscope* - 5 MHz oscilloscope assembled on a custom PCB. I did all the hardware, PCB, and FPGA design. I also wrote the code for interfacing with the hardware, but not the user interface or top-level code. The project was based around an Altera Cyclone III FPGA. Finished June 2014.

*RoboTrike* - Small robot with holonomic motion. I wrote all the code for this project in x86 assembly. This includes the keypad (4x4 grid of buttons) interface, the serial chip interface, the serial parser, the LED display, and the PWM output to the motors. Finished December 2013.

- **Other**

*Auto Terrain Generator* - Set of Java scripts that tried to generate random but realistic-looking terrain.

*Chat Server/Client* - Small Python application that implements a chat server and client using sockets.

*Game Development* - I developed (I did all the programming and art) quite a few computer games through secondary school, some of which I spent at least two years on.

## Other Interests

- **Guitar (acoustic and electric)**
- **Running, cycling, hiking, and camping**
- **Cooking, grilling, and coffee**
- **Reading**