

Resume Part B

This document presents details of work that I've done professionally and otherwise as related to:

- Embedded Systems and Electronics
- Data Science and Machine Learning
- DSP and Audio Programming

Embedded Systems

Some of this isn't "embedded" *per se*, but I think it represents a similar type of work involving bit twiddling, protocol interfacing, and careful management of IO and memory where speed and efficiency at a very low level is second only to correctness.

Professionally:

- Netscape (1999): Implemented low-level bridge between two proprietary network protocols in C and C++
- Océ (2011): Built tools, in C# and C++ to generate all of the C++ UI code for printer drivers so that a family of printers would have a base driver with the UI generated based on the UI/capabilities of a specific model
- Familiar with analyzing network or other "over the wire" signals/traffic to diagnose communication issues

Projects:

- Started playing with Arduinos around 2006 and have prototyped or played with:
 - Driving different stepper motors, built a *simple* 2d "cnc" table out of screws, nuts, and some brackets.
 - Reading and writing to SD cards and USB drives
 - Driving an AM/FM radio chip via I2C (A car stereo was planned)
 - Built a guitar pedal shield kit and programmed simple distortion, delay, and other weird stuff, and here, got started into DSP...
- Wrote a UI for a prototype irrigation control unit that ran Windows CE
- Worked with a friend to retrofit a 1970's Bridgeport milling machine to use modern CNC technology. Lots of details, please ask about it.
- I have a synthesizer module that I can upload PureData or C++ code to. I've done hello-world and am getting ready to try out some filter and delay code that I've written for VCV Rack (see DSP section)

Books:

- "Make: Getting Started with Arduino" by Massimo Banzi and Michael Shiloh
- "Timer, Op Amp, & Optoelectronic Circuits and Projects" by Forrest M. Mims III
- "Robotics, Mechatronics, & AI" by Newton C. Braga
- "Programming Arduino: Next Steps" by Simon Monk
- "Practical Electronics for Inventors" by Paul Scherz and Simon Monk

Courses at Arizona State University: (1993-1998)

- Digital Design: From gates to adders, memories, (de)muxes, ALUs, etc to a simple 8-bit CPU
- Physics for Engineers II: Electricity and Magnetism
- Assembly Language : 68HC11
- Computer Organization and Architecture
- Senior Elective: Embedded Systems

Data Science

Professionally:

- I've worked with many SQL databases since 1997 and have DBA-like proficiency with MySQL and SQLServer
- Simply Hired (2005): Helped to build and run the "largest database of jobs on earth"
- Simply Hired (2008): Ran the web-analytics for one of the top 50 most trafficked sites on the internet
- Simply Hired: I handled regular and ad-hoc reporting and analysis needs for the sales, biz-dev, and executive teams.
- Green Light (2016): Designed and built several neural network systems to help predict the outcomes of medical bill negotiations based on previous negotiation history. We also did some experiments around identifying patient event-histories based on medical claims and predicting patient use of healthcare based on their history. Ultimately a lack of data sidelined these activities and we turned our focus to collecting more data
- Green Light (2018): Designed and implemented a "client dashboard" product which combined useful charts and visualizations with slice/dice/drill capabilities so that our clients could view their healthcare spending from nearly any angle and at any depth, helping to identify areas of potential savings

Courses:

- 2016: "Machine Learning 101" by Stanford via Coursera (mostly about neural networks)
- 2021: "IBM Data Science Professional Certificate" set of courses at Coursera. I am on track to finish 9/2021

DSP / Audio Programming

Projects:

- As I worked with audio on the Arduino and tried to run FFTs in real-time I realized that I needed to learn DSP
- Implemented many of the algorithms in the books below and others as C++ plugins for VCV Rack
 - Oscillators
 - Filters (FIR and IIR)
 - Delays
 - Envelope generation
 - Signal combinators: boolean-logic, min/max, other...

Books (2018 - Present):

- Designing Audio Effect Plugins in C++ by Will Pirkle
- Designing Software Synthesizer Plugins in C++ by Will Pirkle
- BasicSynth by Daniel R. Mitchell

Coursera Course (2016): "Audio Signal Processing for Music Applications":

- Fourier theorems, transform, and FFT
- Sinusoidal, harmonic, and stochastic models of synthesis
- Audio transformations: time and pitch scaling and morphing

Analog and Digital Electronics

Projects:

- At 15 or so, I completely rewired my '69 VW Bug to accommodate a host of modifications including: power windows, remote entry, remote trunk opening, an alarm system, and a "killer" stereo
- Around 2005 I digested a bunch of books (and websites) about how to "hot rod" your guitar and just had to include the kitchen sink on my poor Squire Strat. It didn't turn out well, so I bought a new guitar and an Arduino...
- Using the Arduino forces one to learn at least a minimum of analog electronics to connect things to it
- About 8 years ago, much to my own amazement, I fixed our broken TV by replacing a couple of blown caps
- As I went further down the music/audio path I became intrigued by analog synthesizer circuits and have studied, recreated (on a breadboard and/or simulated), and experimented with analog circuits for oscillators, filters, sequencers, and "signal combiners" of many types as well as building blocks like reverse-polarity power protection, attenuated/attenuverted inputs, mixing, linear to exponential converters, and interfacing with IC chips for op-amps, OTAs, counters, shift registers, PLL chips, digital logic, and more...

See Books and Courses in Embedded (above)