

Nolan Chang, EIT

nolanchang8@gmail.com | (626) — 202 — 9167
LinkedIn: linkedin.com/in/nolan-chang
Github: github.com/nychang1/Resume-and-Projects

EDUCATION

California State Polytechnic University, Pomona

Bachelor in Computer Engineering; GPA: 3.41/4.0

Pomona, CA

Sep 2014 - May 2019

SKILLS

- **Programming Languages:** C++, C#, Verilog, Python
- **Technologies:** PSpice, Vivado, Github, PyCharm, Visual Studio, OpenCV, Matlab, Jupyter, Photoshop, Solidworks, AutoCAD

PROJECTS

- **ZYNQ S-Curve Motion Controller with Configurable Kinematics:** Python, Verilog, C
 - Created an open-source, multi-feature motion controller with user-definable kinematics seeking to improve the flaws of industrial and open-source motion controllers
 - Highly optimized S-Curve motion profiler, trajectory planner and PWM pulse generator using a combination of Verilog, Python, and C
 - Implemented homing logic and a safety supervisor
 - Functional prototype implemented on the PYNQ FPGA board
 - Chosen as the sole representative of the ECE Department at the Cal Poly Pomona 2019 College of Engineering Showcase
- **Scantron Scanning and Grading Application using Image Detection:** Python
 - Reads in an answer key and scantron submissions and outputs questions marked incorrectly using Python 3 with OpenCV
 - Uses image detecting parameters to read an image preferably in grayscale
 - Adaptive thresholding applied on a pixel-to-pixel basis to accurately display outlines
 - First implementation uses AND to superimpose both scantrons onto a single image for comparison
 - Second implementation uses XNOR to detect differences in the image, additional adaptive thresholding for enhancement, and a Gaussian blur to remove noise for blob detection
- **Errors and Problems in Software:** C++, Python
 - Explored and tested computation errors in different applications, primarily, floating point operations in Visual Studio C++, floor() in MATLAB, and computations with leading zeroes in Python 2.7.2
- **Postfix Notation using Stacks:** C++
 - Implements a Stack data structure to convert mathematical expressions from standard to postfix notation where operands come before their operators
 - Functions for checking whether the stack is empty, data entry, data retrieval, and data display
- **Grades on a Standard Bell Curve:** C++
 - Generates IDs and randomized scores for n amount of students specified by the user stored in an array
 - Calculates the mean and standard deviation of all the scores
 - Assigns a grade value to the students on a standard bell curve
- **Markov Chain:** C++
 - Prompts user for size of matrix and values to fill the matrix
 - Checks whether the matrix generated by the user is a Markov Chain
- **BST Tree with Inorder, Postorder, and Preorder:** C++
 - Generates a standard Binary Search Tree using nodes
 - Traverses each node and displays the tree with inorder, postorder and preorder
- **Exp() as a Taylor Series Expansion:** C#
 - Contains a method that implements exp(x) using Taylor Series
 - Compares the values generated by this program to Math.exp(x) from the Math library in C#

CERTIFICATIONS & SKILLS

- Certified Engineer-in-Training for Electrical and Computer Engineering
- Certified Solidworks Associate (ID: C-JGF9Y4MEA5)
- Spoken Languages: English, Mandarin Chinese