Tucker J. Polomik

Email: tuckerpo@buffalo.edu
Phone: (845) 381-8007

GitHub: https://github.com/tuckerpo
LinkedIn: https://linkedin.com/in/tuckerpo
Personal Site: http://buffalo.edu/∼tuckerpo

Education

- **University at Buffalo, School of Engineering** — Buffalo, NY
  Bachelor of Science, Computer Engineering — *Sept 2018*

Experience

- **INFICON** — Syracuse, NY
  R&D Software Engineer — *October 2018 - Present*
  - **R&D**: Software Engineer working as part of a multidisciplinary, cross-functional team doing R&D work in the areas of gas chromatography, mass spectrometry, thin film deposition and vacuum leak detection.
  - **Project Aruba**: One of three engineers driving a major software update for our gas chromatography instrument. Responsible for writing the post-processing events that allowed instrument users to clean up chromatography errors due to electrical noise and external artifacts. Done in Python, JavaScript, and Bash.
  - **Web Interface**: Built a web interface on top of our in-house chromatography synthesis engine. Allowed chemists to remotely run chromatogram scenarios with varying noise profiles to determine problem areas in our instruments. Python, JavaScript, HTML5.
  - **GC/MS**: Work with a team developing a new combined gas chromatography/mass spectrometry instrument, NextGen HAPSITE, a modern replacement for INFICON's highest revenue stream product. Maintained 90%+ code coverage with GoogleTest. C++11, Bash.
- **KGB AVIATION SOLUTIONS, LLC.** — West Seneca, NY
  Capstone Project - Student Engineer — *January 2018 - May 2018*
  - **FDR Interfacing Tool**: Worked with the company CEO and several other interns to build an interfacing system for flight data recorders (FDRs).
  - **Reverse Engineering**: Determined the proprietary handshaking protocol & RS-422 signal timing patterns of FDRs.
  - **Analog Tooling**: Used a Tek oscilloscope and Digilent bus analyzer to determine character encoding, start-stop, parity and baud rate of serial streams.
  - **Shipped Product**: Managed to take an idea from the planning stages and progress to a tangible, secure, shippable embedded system in four months time.

Personal Projects

- **InMoov Robotics Simulation**: Worked on a team of 6 to create a bipedal motion simulator for an open-source CAD tool called "Choreonoid". Wrote C++ code that models bipedal kinematics and Python scripts that automate animations. Followed the agile software development pattern with bi-weekly sprints.
- **Carduino (Autonomous RC vehicle)**: Write C code on an ATmega328P chip that controls servos and motors on an RC vehicle. Concurrently parse data from an ultrasonic sensor to calculate the vehicle's distance to things around it, changing the angle of the servos to avoid collisions accordingly.
- **RTCV**: Platform-independent real-time computer vision project. Accesses a system's webcam and allows the user to apply various image filters and segmentations in real-time with minimal FPS loss. Written in Python.
- **MicroSpaceInvaders**: ASCII rendition of the 1978 arcade game written in ARM7 assembly language and C. Developed on bare-metal NXP LPC213x series of ARM microprocessor boards over JTAG.

Programming Skills

- **Languages**: C++, C, Python, Java, Bash, asm (ARM, MIPS)
- **Tools & Frameworks**: GNU Toolchain, GoogleTest, PyTest, JIRA, LaTeX, git
- **OS/Platforms**: Linux, Windows