

Paul R. Potts

Abridged Résumé • Updated March 2025

Ypsilanti, Michigan • paul@thepottshouse.org

*You can find an HTML version of this résumé, with projects links, at
https://thepottshouse.org/paul/portfolio/Abridged_Resume.html*

Employment Goal

I am seeking fully remote work in low-level embedded or related software development, ideally working with a team in Mountain or Pacific time zones. I would prefer to be a full-time direct hire, but will also consider hourly contract work.

Top Skills

ARM Cortex-M	Agile	Bootloaders	C
C++	Device Drivers	Embedded Systems	FreeRTOS
GNU/Linux	Microcontrollers	Microsoft Windows	Python
QEMU	Tech Writing	Test Automation	Testing

Recent Work Highlights

Virtualization Engineer • Randstad USA (Contract with Boeing) • December 2022-October 2024 • Ypsilanti, Michigan (Fully Remote)

At this contract assignment with Boeing, I was fully remote, and worked with a team developing “high fidelity digital twin” QEMU-based virtual machines. These allow simulating, testing, and debugging firmware for custom aerospace hardware on multiple commodity platforms. I worked with a small scrum team of employees and contractors debugging and improving emulated peripherals, some of which are “soft” peripherals implemented in FPGA fabric on Xilinx multicore SOCs. I also tuned VM performance for different guest code workloads.

For a project that emulates a multi-CPU system on a rack of 16-core NXP ARM servers, I wrote ARM instruction emulation code to extend QEMU, adding support for several types of instructions that execute correctly in TCG mode, but in KVM mode produce NISV (Non-Instruction Syndrome Valid) faults. On this team I also worked on other QEMU-based virtual machines deployed on

x86 Windows and Linux platforms, made improvements to the CI/CD pipeline, acted as temporary scrum team leader and product owner, gave regular demos to my scrum team along with managers and architects, presented “lunch-and-learn” topics such as C memory management, and mentored a junior engineer.

Senior Software Engineer • Argo AI • June 2021-July 2022 • Dearborn, Michigan (Hybrid)

At Argo, I first completed the testing project I began as a contractor, continuing to work primarily remotely, but completing some tests that required access to hardware on-site in Allen Park, Michigan. I then joined a small team of four engineers that worked on firmware and tools to support testing: serving as a resource to other teams, writing design documents, presenting them to the team, developing and testing code changes, landing code changes as GitHub pull requests, and giving guidance to the teams that test firmware on the vehicles. I used embedded C and C++, Python, VSCode, Git, GitHub, Jenkins, Jira, Confluence, and Argo’s custom continuous integration system.

Senior Software Engineer • OSI Engineering Contract with Argo AI • April-May 2022 • Dearborn, Michigan (Hybrid)

I worked briefly as a contractor for Argo AI, and then was offered direct employment. While a contractor, I worked on low-level testing of autonomous vehicle hardware. To facilitate testing I began developing a suite of automated and semi-automated tests in Python, using regular expressions to parse the output of command-line tools running on a number of different embedded microcontrollers on the board; I eventually expanded this test framework to about 17,000 lines of code, covering hundreds of test cases. When necessary, I made changes to testing-specific firmware for Infineon Aurix microcontrollers.

Senior Software Engineer • Thorlabs Ultrafast Optoelectronics • June 2015-January 2021 • Ann Arbor, Michigan (Hybrid, then On-site)

As the sole software engineer in the Ultrafast Optoelectronics business unit in Ann Arbor, I worked closely with the product designer to develop high-speed optical transmitter and receiver product lines. Products included the MX10B and a number of related transmitters, phase modulators, modulator drivers, and lasers running common code, and the RXM10 receivers.

The MX10B and related products include the SAM4E16E microcontroller (a single-core ARM Cortex-M4 device, running FreeRTOS), with multiple DACs and ADCs (monitoring over 100 ADC channels), and the CP2102N UART-to-USB bridge chip from Silicon Devices. These product lines integrate tunable and fixed-frequency lasers, variable optical attenuators, Mach-Zehnder modulators, and multistage RF amplifiers. Modulators and attenuators are managed by PID control loops. These products incorporate touch screens from Amulet Technologies, programmed in GEMScript, and support remote control via USB

and RS-232 via SCPI-compatible commands. The RXM10 and related receiver products incorporate the ATtiny461A, running “bare metal” with no operating system at a very low clock rate to save power, monitoring battery level and input level via ADC, controlling the scaling of the LCD display.

Later, I developed a new hardware/software platform based on the SAM4S2A microcontroller, again incorporating FreeRTOS. I hand-built early prototype boards, then developed several revisions of prototype PCBs using Autodesk EAGLE. I implemented all peripheral drivers (flash, UART, SPI, I2C, ADCs, DACs, and PWM) from scratch, and an all-new bootloader, which fits in under 16KiB of flash memory, as well all the application firmware. I also worked through Microsoft’s process to develop a signed USB device driver for this device. The first product to incorporate this platform was the TLX3 tunable laser source.

I also developed unreleased prototype products incorporating smaller microcontrollers such as the ATtiny104. On the ATtiny104 microcontroller, with only 32 bytes of RAM, I implemented a VOA controller with support for encoder knobs, using GCC’s fixed-point math support library.

In addition to embedded code, I wrote several Windows applications in C++, using the Qt framework, which communicated with the device bootloaders to install firmware updates and calibration data during manufacturing. I worked to improve LabVIEW application code to test receivers during manufacturing. I also wrote test plans and customer documentation for multiple products, and sample code for distribution, demonstrating how customers can control these products with Python and LabVIEW.

Tools used: FreeRTOS, Keil MDK 5 (armclang and armcc), Atmel Studio (gcc), LabVIEW, Amulet GEMStudio, Autodesk EAGLE, GraphViz, and GitHub.

Senior Software Engineer • Lectronix, Inc. • October 2005-March 2013 • Ann Arbor, Michigan and Saginaw, Michigan (On-site, then Hybrid)

At Lectronix, I developed embedded client/server, driver, and DSP code for vehicle infotainment systems. I first worked on the R5000 system for recreational vehicles and the T7000 system for the heavy truck market. These ran on QNX on PowerPC microprocessors. I was responsible for integrating the Garmin GVN-52 GPS receiver with the Lectronix R5000 product. I developed the core subsystem for handling all audio control, and contributed improvements to the driver for the Philips SAA7709 DSP.

I then worked on the P7000/Rockwell iForce systems for police vehicles, which run on Linux. I wrote embedded DSP code in C to perform audio processing and mixing on the TI TMS320C6727 DSP, utilizing the DSP’s complex multi-channel DMA engines. Later, I ported much of the I/O logic to a separate server, using a hierarchical state machine design, and also ported the audio control logic to a new server architecture based on JSON/RPC. I worked primarily in C and

C++, but also did some related work in Python, for boilerplate code generation from XML files.

While the company headquarters are in Lansing, I worked for over four years in a satellite office in Ann Arbor, and then relocated to Saginaw, Michigan, and spent three years primarily telecommuting.

Tools used: TI Code Composer Studio, Spectrum Digital XDS510 USB JTAG Emulator, QNX Momentics IDE, Microsoft Visual Studio, GNU/Linux (for building and testing), Python (for code generation tools), Haskell (for generating audio waveform data for testing), TWiki (for maintaining internal documentation), Mantis (for bug-tracking), Subversion, Visio, Graphviz, and BOUML (a tool for modeling code with UML).

Education

Math Academy • Mathematical Foundations

Math Academy is an accredited online program for mathematics instruction. It has been many years since I was in a college calculus class, so I have been using Math Academy to work through the Mathematical Foundations course series tailored for adult learners, in order to refresh and improve my math skills. As of March 2025 I have completed Foundations I and nearly completed Foundations II.

The College of Wooster • Bachelor of Arts, 1989

I graduated with departmental honors and thesis honors. I studied English (my major) and Computer Science (my minor). I won the Stephen R. Donaldson Prize for Fiction. While a student, I was a disc jockey and production manager for the campus radio station, WCWS-FM. I worked several part-time jobs including computer operator, teaching assistant, and writing tutor. I served as an editor for several campus periodicals, and developed an interactive graphical tutorial on calculus limits and continuity using Hypercard and external code plugins in THINK C.

Personal Programming Projects

While I have primarily done work for hire, I have put some small personal programming projects online on my GitHub page at <https://github.com/paulpotts>. These currently include projects in C, C++, the Arduino dialect of C++, Haskell, and Dylan.