Paul R. Potts

Complete 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/Complete_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.

About Me

I have over 30 years of professional experience designing, implementing, testing, debugging, documenting, and maintaining software projects in a variety of languages, on a variety of platforms. I have developed software for platforms ranging from the Apple Macintosh, to Microsoft Windows, to Linux, to the Apple Newton, to embedded devices. I have often specialized in the design and implementation of software *systems*, containing programs written in different languages, running on different devices and operating systems, and communicating with each other. I also have experience writing documentation and training materials, leading small teams and projects, and mentoring junior engineers.

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

Career Overview

1990s

In the early to mid-1990s, I was very interested in user interface design and interactive multimedia. I worked to develop instructional programs, early web

sites, and software tools to support health behavior research, targeting MacOS, Microsoft Windows, and Apple's Newton devices. I wrote and published both technical and non-technical articles, and pursued research interests in C++, object-oriented design, Java, Scheme, and Dylan.

2000s

During the dot-com boom, I developed back-end code for large web sites, in Java. After the boom ended, I began specializing on embedded programming, including audio DSP programming. I developed code for the Aardvark Direct Pro Q10 audio interface. I worked in embedded automotive, testing devices that used MOST networking, and developing documentation for a Delphi car radio codebase. With Lectronix, Inc., I developed audio support code for infotainment systems for the RV and heavy truck market.

2010s

After moving to Saginaw, Michigan, I continued working remotely for Lectronix, Inc. to develop DSP audio routing and mixing code for police vehicles. Later, I worked as a contractor, testing and inspecting embedded code for a medical device, as an adjunct instructor teaching a System Software course at Saginaw Valley State University, and as an embedded software engineer developing prototype IoT sensing devices for Dow Chemical.

In 2015, as an early hire in the Thorlabs Ultrafast Optoelectronics business unit in Ann Arbor, I began working partly remotely, developing embedded code for multiple product lines including high-speed optical transmitters and receivers. In 2017 I relocated my family to Ypsilanti and began working completely on-site for Thorlabs. This work eventually included developing a new embedded platform using FreeRTOS on the SAM4S2A microcontroller, writing all drivers, a new bootloader, and all application code. For Thorlabs, I also wrote desktop applications for Windows in C++ using the Qt framework, a Python plug-in, and Python and LabVIEW code.

2020s

After a furlough in 2020 due to the COVID-19 pandemic, I continued working with Thorlabs through early 2021. I next worked for Argo AI, a company developing autonomous vehicles, initially as a contractor, working on board verification tests for Argo's hardware platform. I performed both manual and automated testing, writing a new suite of about 17,000 lines of Python code. I was then hired as an employee and assigned to the Firmware Verification team, which developed test firmware to support board-level and system-level verification tests.

After Argo AI shut down, I worked as a contractor for Boeing, as a member of their Virtualization team, building virtual machines to test avionics software, adding custom emulated peripherals, and debugging, tuning and optimizing

device emulation code as needed. My scrum team specialized in Arm64 multi-core systems. I also worked on x86 Linux and Windows platforms, and made improvements to the team's GitLab CI/CD pipeline.

Additional Skills

Assembly Language	ADCs	AVR	CAN
CI/CD Pipelines	DACs	Digital Audio	DSPs
Git and GitLab	GNU Make	GUI Design	Haskell
Infotainment	I2C	Java	Jira
KVM	LabVIEW	Laser Safety	Markdown
SPI	UART	U-Boot	UML

Work History

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.

Embedded Software Engineer • Kelly Engineering Resources Contract with Dow • March 2015-June 2015 • Midland, Michigan (On-site)

In this brief assignment at Dow, I worked with a team developing remote sensing applications using embedded microcontrollers. I expanded and enhanced a prototype program for the Atmel ATmega32U4 microcontroller in the Arduino Yun, and helped test and evaluate "Internet of Things" (IoT) technologies for possible use in Dow projects. I received training in LabVIEW (Core 1 and Core 2).

Tools used: Visual Studio 2013, Visual Micro, C/C++, Linux (various distributions), Arduino (various models), and BeagleBone Black.

Adjunct Instructor • Saginaw Valley State University • August-December 2014 • Saginaw, Michigan (Part-time)

At SVSU, I taught CIS (Computing and Information Science) 333, System Software, Fall 2014. I developed lectures and materials, tests, and programming assignments in C++ and 6502 assembly language. This course covered the fundamentals of computer architecture, assembly language programming, loaders, linkers, macro processors, compilers, and operating systems.

Senior Software Engineer • Optomi Contract with Logikos, Inc. • September 2013-September 2014 • Saginaw, Michigan and Fort Wayne, Indiana (Hybrid)

While a contractor with Optomi, I worked primarily remotely for Logikos, Inc. in Fort Wayne, Indiana. I was a member of the FUIT (Functional, User, and Integration Testing) team working to verify a medical device code base (written in C) with IBM Rational Test RealTime. I performed code inspection, reported bugs and issues, and applied the team's coding standards. At the end of my contract, I was told that I had identified more serious issues in the code base than the rest of the team combined.

Technologies: Analog Devices ADuCM350 “meter-on-a-chip” microcontroller (ARM Cortex M3), Micrium uC/OS-II RTOS, QP (Quantum Platform) state machine framework.

Tools used: IBM Rational Test RealTime, IAR Embedded Workbench, Subversion, Bugzilla, and Visual SourceSafe.

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).

Senior Software Engineer • MicroMax, Inc. Contract work with Visteon and Delphi • September 2004-October 2005 • Canton, Michigan and Dearborn, Michigan (On-site)

While working for MicroMax, I had several assigments. I first worked on-site at the Visteon plant in Dearborn, Michigan, testing a Sirius satellite radio receiver for Ford's PAG (Premiere Automotive Group). This device used the MOST networking protocol over plastic optical fiber. I performed testing and wrote small utility programs in Ruby to more easily encode and decode MOST message

payloads. I then worked at MicroMax in Canton, Michigan as lead technical writer and editor, working with a small team to develop extensive documentation for a library of Delphi's embedded C code, used in their car radio products.

Tools used: MicroMax MxVDev, Microsoft Visual Basic (Visual Studio .NET IDE), Microsoft Visual C++, Ruby, Understand for C/C++ (code analysis tool), QA-C (code analysis tool), Oasis Optolyzer, IBM Rational ClearCase and ClearQuest, Microsoft Word, and Visio.

Senior Software Engineer • Aardvark Audio • September 2001-June 2004 • Ann Arbor, Michigan (On-site)

Working with Igor Levin at Aardvark, I initially took on the task of getting Aardvark's audio card firmware to reliably transmit audio data across the PCI busses on Macintosh computers. I developed Motorola 56301 DSP code, in C and assembly language. I also developed the audio driver code to support these cards on MacOS 9 and MacOS X (using CoreAudio), developed a GUI application with a mixer and animated meters using C++ and the Qt framework, and ran the beta program for the Mac software. In addition, I developed prototype object-oriented designs in Dylan, and instituted better software engineering practices at Aardvark including the use of make and CVS. Aardvark Audio shut down, but Igor Levin later started Antelope Audio.

Tools used: Cygwin, Motorola DSP compilers for MS-DOS, Link-56K serial DSP debugger, Hewlett-Packard logic analyzers, Metrowerks CodeWarrior, Onyx Spotlight Memory Debugger, Python, Gwydion Dylan, CVS, ViewCVS, Installer VISE, Project Builder (now XCode), and the Qt GUI framework.

Senior Software Engineer • InterConnect of Ann Arbor • August 2000 to July 2001 • Ann Arbor, Michigan (On-site)

At InterConnect, a software services company, I primarily did Java development for large web sites for clients such as ProQuest (formerly University Microfilms, later known as Bell and Howell Information and Learning). I worked in particular on the import process, populating large Oracle databases and cleaning and ingesting metadata, working with customer database engineers so that the import process worked correctly with their PL/SQL code. I did some Perl development as well, to support some older projects.

Tools used: IBM VisualAge for Java, Solaris, Oracle, PL/SQL, JDBC, XML, Perl, CVS, Bugzilla, Visio, JVISION, TOAD, SQL, PostgreSQL, and UML.

Systems Research Programmer III • Health Media Research Lab, Comprehensive Cancer Center, University of Michigan • January 1996-August 2000 • Ann Arbor, Michigan (On-site)

I led the development of the Health Media Research Lab's technical capabilities during its early growth. I recruited, interviewed, supervised, trained, and

evaluated technical staff, served as the lead developer of a survey engine for the Apple Newton, and led the effort to port the Newton survey engine to Apple WebObjects using Java and GNU Kawa (Scheme). I completed Apple's training on WebObjects development. I also led the development of interactive multimedia program on Cancer and Genetics, integrating the work of graphic designers and writers, and co-developed a Macintosh application in C++ that used AppleScript to drive Quark XPress, in order to generate tailored health information booklets using early color laser printers. In addition, I co-designed the group's internal QA process, and gave presentations on software development topics including user interface design, XML, and dynamic languages.

Tools used: Macromedia Director, Adobe Photoshop, Quark XPress, TestTrack, Newton Toolkit, CodeWarrior (C++), AppleScript, Visual BASIC, Perl, Scheme (GNU Kawa), Java, XML, MacOS, GNU/Linux, Solaris, Apple WebObjects, and UML.

Software Engineer • Fry Multimedia • June 1994-December 1995 • Ann Arbor, Michigan (On-site)

I co-developed a CD-ROM business directory with a custom search engine, user interface, and compression algorithms, worked on prototype Apple Newton and Macintosh multimedia applications, and developed early commercial web sites using HTML and Perl CGI scripts.

Tools used: Visual C++ 1.5, CodeWarrior, MKS RCS, HTML, PGP, Perl, Newton Toolkit, MacOS, Windows NT, and Solaris.

Software Engineer • Pharos Technologies • January 1994-April 1994 • Maineville, Ohio (On-site)

Pharos experienced a business downturn and laid off engineering staff only a few months after I started, but while at Pharos I did software testing for the team developing the Monsanto Infielder Crop Records System on the Apple Newton. This included writing Newton programs to generate "soup" data stores for testing. I also built a prototype application for the Windows for Pen platform, completed Apple training on developing application support for AppleScript, and worked on proposals for additional Newton projects.

Software Developoer • Office of Instructional Technology, University of Michigan • December 1990-December 1993 • Ann Arbor, Michigan (On-site)

At the Office of Instructional Technology, I worked with faculty and instructional designers to develop instructional multimedia, from paper prototypes to finished programs, incorporating formal usability testing; produced and edited instructional video materials; taught ToolBook programming classes; wrote newsletter articles; and performed evaluation and pre-release testing of Windows 3.0, Macintosh System 7.0, IBM OS/2, QuickTime, and other technologies. My

completed projects include a simulation of an audiometer, a videodisc-based program for teaching side effects of antipsychotic medications, and the Velocity Manufacturing Corporation case study, which won a New Media INVISION silver medal in 1994.

Tools used: HyperCard, SuperCard, THINK C, THINK Class Library, ToolBook, Visual BASIC, MacOS, and Windows 3.0.

Intern for Documentation • The College of Wooster • Academic Year 1989-1990 • Wooster, Ohio (On-site)

After graduation, I returned to campus as an Intern for Documentation, Academic Computing Services. During my year as an intern I wrote a number of small Macintosh programs, published a newsletter for Academic Computing Services, and developed a number of handouts and other documents including a book containing detailed technical information about the rapidly growing campus computer network.

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 Projects

- Music performance, songwriting, and music production: I play guitar, bass, fretless bass, ukulele, and Chapman Stick, and record, mix, and master my original songs and covers using Apple Logic Pro. In 2010, I began occasionally competing in songwriting competitions and sometimes working on collaborations with other online musicians.

- Programming: 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/paulrpotts**. These currently include projects in C, C++, the Arduino dialect of C++, Haskell, and Dylan.
- Podcast recording and production: I have recorded and produced podcasts from scratch since 2007.
- Volunteering: I taught an introductory class on computer programming using the Scheme programming language and the PLT Scheme/Dr. Scheme IDE (now called Racket), to Junior High School-age students. I also taught a class on J. R. R. Tolkien's *The Lord of the Rings* trilogy.
- Web development: I build my personal web site using Markdown source files and GNU Makefiles to drive Pandoc. My build machine/staging server is a headless Intel NUC running Ubuntu Server. I syncronize the staged contents to my hosting server using **rsync**.
- Writing: I began blogging in 1998 using Blosxom and in 2006 moved to Google's Blogger platform. In 2019 I began hosting all my old content and removing material from Blogger, as I switched to an e-mail newsletter format using TinyLetter. After TinyLetter was discontinued, I continued putting newsletter issues on my personal web site.