

Aaron Robinson

Senior Software Engineer | C++, Media Systems & Performance

Sylvania, Ohio | caustik@gmail.com | linkedin.com/in/aaron-c-robinson | apu.software

PROFILE

I've spent more than 20 years building software where performance and correctness matter, mostly in video, audio, codecs, desktop systems, and the infrastructure needed to ship them. Most of that work has been in C++ and close to the machine: architecture, threading, SIMD and assembly, networked and real-time media systems, and production bugs that are hard to reproduce.

I've worked remotely with teams around the world for more than 15 years. At APU Software, I've built and shipped eight audio products and the systems used to release and support them. The product line is mature now and needs relatively little maintenance.

I now also leverage Codex agents for research, design, implementation, and testing. I make the technical calls and integrate their work. I'm also building Pangine, a Rust experiment in explicit semantic state and reasoning that may be useful on its own or alongside language models.

SELECTED ACCOMPLISHMENTS

- Created Cxbx, the first open-source Xbox emulator, and OpenXDK, a free Xbox development framework.
- Built an early prototype that captured DirectX output and streamed it between Windows PCs. That experiment contributed to the remote-UI direction the DivX Connected team developed, and I later helped build the full client/server media platform.
- Made an already heavily optimized HEVC decoder more than 30% faster by designing and implementing a custom fiber-based, lock-free scheduler and dependency system.
- Built and shipped all eight APU Software audio products, along with the shared C++ architecture, release automation, licensing, commerce, documentation, and customer support needed to run the product line.
- Currently building Pangine, a Rust experiment in explicit semantic state and reasoning. The implementation is covered by reproducible behavioral tests.

SELECTED EXPERIENCE

APU Software - Commercial Audio Software | Senior Software Engineer | Remote Nov 2022 - Present

- Founded APU Software, then built and shipped eight real-time audio plug-ins and standalone applications in C++20 and JUCE.
- Built the shared C++ architecture for DSP, parameters and state, OpenGL interfaces, host integration, automated analysis, and validation.
- Built a Python metadata and code-generation system that produces project files, CI workflows, VS Code tasks, installers, product assets, websites, documentation, and release metadata from one place.
- Built the licensing, commerce, deployment, and support systems, and worked directly with audio professionals while refining the products.
- The product line is mature now and requires very little maintenance. For bug fixes and new development, I give parallel Codex agents different jobs, iteratively reviewing and refining the results.

MainConcept - Professional Video Codec SDKs | Senior Codec Engineer | Remote 2013 - May 2020

- Implemented and optimized HEVC/H.265 decoding in C, C++, x86/SSE, ARM/NEON, and NaCl assembly across desktop and embedded systems.
- Much of my work was in threading, lock-free algorithms, memory management, and NUMA-aware execution inside a mature codec.
- Designed and implemented a fiber-based, lock-free scheduling and dependency system. It made an already heavily optimized HEVC decoder more than 30% faster.

DivX - Digital Media Platforms | Senior Software Engineer 2003 - 2010

- While experimenting with DirectX video capture, I built a prototype that streamed captured desktop frames between Windows PCs. That work contributed to the remote-UI direction the DivX Connected team developed, and I later helped build the full client/server media platform.
- Worked on protocols, renderers, encoders, decoders, and filters for Windows and consumer-electronics devices. I also received the Chairman's Award.

CURRENT RESEARCH

Pangine | Independent AI Research
pangine.com

2026 - Present

- I'm building Pangine to test whether experience, retained state, and questions can all use the same small semantic grammar. The Rust implementation uses canonical concept graphs, recursive experience, lazy wildcard projection, and ranked questions.
- The larger question is whether explicit, inspectable state can perform useful inference on its own or alongside an LLM. The current core has no third-party runtime dependencies, uses no unsafe Rust, and is covered by deterministic behavioral tests.

ADDITIONAL EXPERIENCE

Interactive Fitness (Espresso Bikes) | Senior Software Engineer | Remote

Jun 2020 - Apr 2022

- Integrated D-Bus and BlueZ, bike serial interfaces, and engine plug-ins in C++. Used production instrumentation to diagnose and optimize embedded C firmware.
- Built and open-sourced a multi-process directory delta patcher for roughly 36 GB Unreal Engine deployments. In one representative benchmark, it reduced the update transfer to 0.99 GB. Patch application is atomic and resumable, with HTTP recovery.
- Built Python tools for monitoring, analysis, and resistance-curve fitting. I also worked on the Node/Angular bike UI and peripheral stack.

SweetLabs (Pokki) | Senior Software Engineer

2010 - 2012

- Built the Chromium-based native layer that Pokki grew around, connecting web applications to the Windows desktop. I also worked across Windows and Node.js.

Chumby (Concurrent Engagement) | Senior Software Engineer | 2007 - 2008. Built a crypto-processor API and debugged Linux, ALSA, and kernel-driver issues.

VIA Technologies | Software Consultant | 2003. Added user- and system-level instrumentation to WinXP for an embedded deployment.

Microsoft | Software Design Engineer Intern | 2001. Built C# client/server and graphical data components for a future version of Microsoft Project.

Synchrony Communications | Software Engineer Intern | 2001. Built Java/C++ regression and stress tools plus an efficient real-time log monitor.

SELECTED PROJECTS

Cxbx | Creator | 2002 - 2006. Reverse-engineered Xbox system software and built binary signature detection, high-level kernel and library interception, and shader translation for Windows graphics.

OpenXDK | Creator | 2002 - 2009. Built a framework for a free Xbox SDK, including a PC-executable-to-XBE converter and documentation of the Xbox kernel API and executable format.

TECHNICAL RANGE

Languages: C++ (through C++20), C, Rust, Python, x86/SSE assembly, ARM/NEON assembly, C#, JavaScript/Node.js

Systems: Architecture, threading, fibers, lock-free algorithms, SIMD, NUMA, memory management, networking, client/server and distributed systems, real-time systems, embedded systems, native and postmortem debugging

Media & DSP: HEVC/H.265, audio DSP, encoders, decoders, renderers, filters, streaming protocols, OpenGL, JUCE

AI & Automation: Several coding agents in parallel, technical review and integration, experimental semantic reasoning, reproducible testing and evaluation

Platforms & Delivery: Windows/Win32, macOS, Linux, CI/CD, release engineering, installers, signing, licensing, commerce