

[illegible]

Stephen WARREN

+1 (408) 667 8278

<http://www.wwwdotorg.org/>

Summary

I'm looking for a focused technical role, which allows me to leverage my broad knowledge of computing, software implementation, software & hardware architecture, and systems engineering.

I've also worked on business integration software using message-passing and databases, cloud-based backends for embedded devices, web-based software, and Linux system & network administration.

I have held a number of customer-facing roles designing, implementing, documenting, and supporting products and SDKs, both commercial and Open-Source. This involved customer interaction in person and via private and public forums such as mailing lists and conferences.

4 year M.Eng. in Computer Science at Imperial College, London, UK. Completed in June 1997.

Experience

- Lead developer. Defined most internal data structures and processes.
- Led the transition from an ad-hoc multi-process prototype that passed JSON to Python to Jinja, to a production single-process code-base written in C++, using real IR and passes.
- Defined the user-visible architecture for future chips and their new features.
- Reviewer of most design documents and code changes.
- Implemented and maintain significant parts of the build and packaging system.

Oct 2010 - Feb 2021

- Team expert in Linux (APIs, techniques, distros, packaging), systems (IO interfaces, communication), hardware (register programming, schematic review), feature research.
- Trail-blazed mainline Linux kernel & U-Boot Open Source support for Tegra SoCs. Mainline support was instrumental in winning high-profile customer business.
- Implemented an ARM secure monitor from scratch for Jetson TK1. This worked around a hardware security issue in legacy hardware and associated software stack.
- Created, maintained, documented, and/or supported public flashing and booting tools.

- Customers found these tools easier to use than our Android-focused tools and processes.
- Wrote application notes to fill holes in NVIDIA's documentation. This enabled customers to more easily understand and work with Tegra systems.
 - Contributed to many new features and bug-fixes to Linux (pinmux subsystem) and U-Boot (test framework, standard boot scripts), solving various community-wide needs.
 - Provided guidance to internal teams to improve documentation, software, and tools.
 - Represented NVIDIA at technical conferences and on public mailing lists.
 - Created an example FPGA design and associated Linux driver & client application to demonstrate GPUDirect/RDMA on Jetson. Customers used this as a starting point to validate their own hardware, and as a basis for their custom designs.
 - Maintained Jetson.GPIO and Jetson support in Adafruit Blinks; Python libraries for accessing GPIOs on Jetson. These provide a simple way for users to access hardware.
 - Created the "USB device mode" feature, which allows network and serial access to Jetson over the flashing port USB cable. This simplified user access to Jetson; keyboard, monitor, and Ethernet cables are no longer strictly required.
 - Consistently involved in toolchain selection for L4T cross-compilation, paying attention to distro ABI compatibility. Ported NVIDIA driver code between toolchains for some of the transitions, including debugging toolchain issues and driver code with undefined behavior.
 - Helped administer NVIDIA's github.com presence. I automated auditing of access to our repos, and removal of access when employees leave.
 - I was the PSIRT Security Officer (vulnerability scoring and mitigation advice) and PLC Security PIC (security review of requirements and design docs) for the L4T team.
 - Back-ported Linux patches from mainline into NVIDIA's own releases, providing direct customers with the latest features.
 - Designed a cloud backend to connect a mobile application to a Jetson device across the Internet, allowing the system to be operated exclusively from the mobile application.
 - Mentored other engineers working on upstreaming, via code review, discussions, presentations, and internal process documentation such as wikis.

Oct 2007 - Oct 2010 **NVIDIA** (Linux Graphics Team)

- Designed, implemented, documented, evangelized, and supported VDPAU; an API to decode, post-process, and display compressed video streams (MPEG, H.264) on GPUs.
- Customers enthusiastically greeted the enhanced features this unlocked in newer GPUs.
- VDPAU gave NVIDIA a market leadership position for Linux video processing.
- Enhanced OpenGL & CUDA to interoperate with VDPAU. Worked on a variety of NV driver components as part of implementing VDPAU, such as the kernel driver and X server.
- Managed build system & regression tests for VDPAU, using "golden" references.
- Reviewed & provided advice for a new build system for the Linux graphics driver.
- Supported, maintained, and enhanced Linux HDMI audio support for NVIDIA GPUs. This helped NVIDIA hardware features "just work" for customers.

Mar 2006 - Oct 2007 **Gyanasoft** (Software services)

- Researched, designed and implemented a new video signal timing identification algorithm. This enabled display devices to handle significantly more input modes.
- Architected and implemented portions of a new SDK for a video processing ASIC. Enabled the customer to more quickly develop new products and features.
- Researched, built, and packaged a modern gcc/binutils tool-chain for an Xtensa CPU. This broke the customer's dependence on an ancient vendor-supplied compiler.
- Linux system administrator. (install, WAN, VPN, backups, WebDAV, LDAP, AccuRev, etc.)

Aug 2004 - Feb 2006 **NVIDIA** (Multimedia software and devices)

- Implemented, maintained, and re-factored portions of a cross-platform multimedia SDK.
- Implemented a cross-OS/CPU/host build system. Managed SDK release packaging. This vastly improved developer productivity by improving reliability and simplicity.
- Was the "go to person" for porting software to new CPU architectures and platforms.
- Co-authored a Windows kernel API emulation layer to support recompilation of a Windows kernel TV capture driver as a Linux user-space application for a successful demo.

```
def add_piece(board): empty_positions = [i for i, val in enumerate(board) if val == 0];
insert_pos = random.randint(0, len(empty_positions) - 1); board[empty_positions[insert_pos]] = 1
```

- Created custom tool-chains and Linux distributions for consumer electronics devices.

Sep 2001 - Jul 2004

Parama Networks (SONET switch vendor)

- Wrote middle-ware “HAL” software to co-ordinate hardware configuration requests from application software to device drivers for STS-768 SONET multiplexer (switch/router).
- Wrote drivers for PPC CPU, ASIC modules and FPGAs.
- Implemented a build system for multiple target environments and tool-chains.
- Built a custom pre-check-in continuous integration system.
This vastly reduced stress for developers, by preventing failures from impacting others.
- Mocked HW APIs to enable software simulation, wrote HW tests & infra-structure.

Aug 1999 - Sep 2001

Mediamatics/NatSemi (Consumer DVD player firmware)

- Wrote presentation management middle-ware to co-ordinate playback.
This vastly improved the user’s playback experience, leveraged software across media formats, and allowed new features to be easily implemented without duplication.
- Worked on core OS, drivers, API, application, firmware upgrade, tools.
- Defined SDK build & release process. Performed SCM management.

Aug 1997 - Jul 1999

Technology House (Software services)

- Architected and developed high-capacity message-based data processing systems.
- Provided 24x7 system support for our software installations to major financial institutions.
- Gave product demos, and provided technical support at business/sales meetings.

Apr 1996 - Sep 1996

Global Communications (Telephony provider; internship)

- Ported DOS telephony software to Win32, including threading, networking, RPC.
- Wrote a web-server plug-in to integrate web pages with telephony systems.
- Used Perl for build scripts and code generation.

Skills

General: Design and implementation, root cause debugging, documentation, automation, code generation, cross-platform code, CPU/ASIC/FPGA drivers, HW simulation, Open Source.

Languages: C++, C, Python, make, various assembly, Shell, Verilog, Java, SQL, willing to learn.

Tools: git, git-repo, Perforce, gcc, gdb, gvim (vi), Markdown, asciidoc, doxygen, expect, gpg, Subversion, Accurev, Continuous/CM Synergy, StarTeam, GHS Multi, ARM RVC.

Environments: ARM, x86, PPC, Xtensa, MIPS, Linux, bare metal, Arduino, ESP32/8266 ESP-IDF, FreeRTOS, Win32, Nucleus PLUS, Enea OSE, Cygwin, PIC, cloud, message-passing, middleware.

Networking: IPv4 routing/connectivity/firewalls/proxies/applications, some IPv6 exposure, BSD sockets, Wireshark, SMTP, DNS, firewalls, HTTP, CGI, WSGI.

Multimedia: Video compression (MPEG-2, H.264), consumer/PC video signals, DVD-Video, DVD-Audio, VCD/SVCD, CDDA, HTML, consumer DVD player firmware, CSS/PPM, OpenGL.

Hardware: Logic analyzers, oscilloscopes, schematics, USB, I2C, SPI, I2S, and PCIe protocols.

Linux: Debian/Ubuntu/CentOS/Fedora/Red Hat/RHEL, cross-distro/ABI binaries, deb/rpm packaging, network setup, bridging, & debugging, SSH, OpenVPN, backup scripts, ZFS, rsync, Unison, Apache, qemu/KVM/virtualization, LVM, Unison, WebDAV, openldap, pam_ldap, nss_ldap, rpm, dpkg, MediaWiki, MySQL, BugZilla, single sign-on, Samba, phpmyadmin, phpldapadmin.

Work Samples

I publish many personal projects & presentations/classes, and some work products, on github:

<https://github.com/swarren/>

<https://github.com/NVIDIA/>

```
(search for "tegra" and "jetson")
```

<https://github.com/fortcollinscreatorhub/>

<https://github.com/lovelandcreatorspace/>

I have contributed to a number of Open-Source projects:

<https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/log/?qt=author&q=swarren>

<https://github.com/search?q=repo%3Au-boot%2Fu-boot+swarren&type=Commits>

<https://git.kernel.org/cgit/utils/dtc/dtc.git/log/?qt=author&q=swarren>

<https://sourceforge.net/projects/congruity/> (I originally created this project)

I have published a variety of documentation generated at work:

<https://download.nvidia.com/tegra-public-appnotes/>

<https://download.nvidia.com/XFree86/gpu-hdmi-audio-document/>

<https://download.nvidia.com/XFree86/vdpau/doxygen/html/>

To view these links more easily, see my home page at <http://www.wwdotorg.org/>.

Technical Extra-Curricular Activities

- Docent, trainer, and IT admin for the Fort Collins Creator Hub. IT admin/support for other 501c3s.
- Contributed initial Raspberry Pi support to the mainline Linux kernel.
- Ported U-Boot to the Raspberry Pi, and act as maintainer for the port in mainline.
- Coding challenges such as microcorruption.com and www.adventofcode.com.
- Founder of two local Maker spaces; Loveland CreatorSpace and Fort Collins Creator Hub.
- I have taught many electronics and software classes at both LCS and FCCH. This yielded valuable publicity for the groups before physical locations were found. These also drew in many first-timers thus contributing to memberships, and fulfilled the educational goals of the groups.
- Acted as a board member, secretary, and president for FCCH at various times. Significantly contributed to many formation activities such as writing our articles of incorporation, bylaws, 501(c)3 application, locating physical space, etc.
- Active member of the Northern Colorado Linux Users Group, at which I've often presented.
- I administer my own network of Linux machines, which host my domain's email and website, provide file-sharing services, etc.
- I run a small craft business making wood lanterns with electronic "candles" for medieval re-creation events. This involved PCB design, micro-controller programming, OpenSCAD physical design, and LASER cutting. <https://www.designbyulric.com/>
- Created a minimal FPGA-based HDMI analyzer; this measures video properties such as clock rate, video resolution, etc. It's useful for automated testing of embedded device video output.
- Tinkered with creating a minimal 6502 to x86-64 JIT compiler. It ran enough instructions to execute a trivial bare-metal "hello world" program.
- Created a breadboard Z80 computer that could run MS BASIC. Used a Teensy micro-controller to share access to the Z80 DRAM to provide a virtual UART for the computer, and system debug capabilities.
- Added remote control to my garage door using an ESP8266 front-ended by an Apache web server. <https://rabbithole.wwwdotorg.org/2017/04/18/esp8266-garage-door.html>.

