Thomas Taranowski

64404 NE 179th St. • Gold Bar, WA 98251
(425) 442-9209 • tom@baringforge.com

Summary of Qualifications

- Expertise in multi-node, multi-core embedded systems design, implementation, and debugging, where minimal latency, fault tolerance, and determinism are critical factors.
- Experience in both defining and implementing a product vision.
- Experienced in the full life cycle of software development, including product conception, requirements definition, prototyping, interface design, implementation, integration, testing, and maintenance.
- Experienced in communicating effectively across team boundaries to solve system level design or implementation issues.
- Versatile - able to move between technologies - able to efficiently learn from existing documentation and experimentation.
- Expertise in working with remote team members, and methods for communicating effectively when not face-to-face.

Experience

- **Planetary Resources, (planetaryresources.com)** **Bellevue, WA**
  Senior Software Engineer (aka Software Mechanic) *August 2012 to present*
  - Design and implementation of Arkyd series spacecraft software architecture. Componetized event-driven architecture, implemented using ZeroMQ, Cheetah, C++, XML. Achieved unification of both flight, microcontroller, and ground software through scalable abstractions. Supported platforms included Ubuntu 12.04, custom linux build, and msp430, although the structure scales arbitrarily.
  - Built CMake based build system with key elements that allowed for clean public interface exposure and minimal code duplication between supported platforms. Jenkins-based CI server with build slaves running locally and on GovCloud.
  - Implemented an MSP430 BSP for flight targets with auto-configuration and initialization from XML-based product configuration. This allowed a single BSP implementation to support custom functionality across multiple microcontrollers with a single codebase. Co-author of several successful SBIR proposals in several separate problem domains.
- **Blue Origin (blueorigin.com)** **Kent, WA**
  Senior Software Engineer *February 2011 to August 2012*
  - Developed a new proprietary space vehicle software and hardware architecture for Blue Origin.
  - Developed a new messaging infrastructure libraries and transport layer definition for vehicle and test stand, to be brought forward for all new vehicles and a company wide infrastructure. Implemented prototype cross platform build system, and promoted it to the team.
  - Deployed the build system for a new program in conjunction with a new messaging infrastructure and software infrastructure. C++, ZeroMQ, CMake.
  - Implemented scalable nightly build system using Jenkins against multiple slave virtual machines, which allowed for arbitrary scaling.
  - Prosilica camera driver to support either network-commanded, or command line operation to control cameras and log video. This was done as part of a research-oriented experiment which triggered operations to occur once a zero-g state was reached.
- **Independent lwIP Consultant (baringforge.com)** **Baring, WA**
  Owner *January 2010 to current*
  - Support miscellaneous customer inquiries.
  - Debug issues with 1.4.0 and 1.4.1 release, and advise on implementation specifics.
- **Anonymous Customer**
  Contractor *December 2009*

- Implemented quantum random number generator network application.
- Netburner 5234, embedded select loop with async message dispatch, web interface through embedded web server.
- Custom board designed with prototype layout in Eagle.
- Sub-contracted hardware engineer to implement layout and power system design.

- **SpaceX (spacex.com)** **Hawthorne, CA**
  Flight Software Engineer *July 2009 to February 2012*
  - Technologies included embedded linux, preempt-rt, multicore PPC platforms, high speed backplane interfaces, including Ethernet and Serial Rapid IO, and time synchronization technologies such as IEEE 1588 (PTP).
  - Worked with team to architect next generation fault tolerant avionics network for Dragon to add crewed capability, and the ability to dock with the ISS. This included moving from HDLC to Ethernet as the physical layer standard, bringing up an all linux runtime and deployment environment to support modern networking standards, tooling, and features, and updating existing code to work on the new platforms.
  - Brought up root filesystem and toolchain proprietary fault tolerant engine controllers and flight computer.
  - Updated IP Addressing scheme to support redundant architecture, and multi-stage vehicle operation and testing.
  - Implemented prototype and did analysis on viability of PTP for fault tolerant systems.
  - Updated stock linux RapidIO driver to work on newer PowerPC chips with multiple RapidIO ports, with corresponding work to integrate the change with the networking subsystem.
  - Miscellaneous programming and code review tasks.
  - Implemented a VxWorks END driver for General Standards sio4bx card to support 10Mb/s IP over HDLC for current generation hardware

- **Alset Corporation (alsetcorp.com)** **Hawthorne, CA**
  Independent Contractor *January 2007 to February 2010*
  - Successfully implemented a "Remote manager" solution for a distributed product, which provided a web interface to remotely deployed lightening detector assets. Ajax, embedded web server, javascript.
  - Key features included flash-based event log, parameter threshold monitoring and alarm relay control, local keypad and 2-line display interface, proprietary hardware monitoring and control.
  - The platform was a Netburner 5282, with proprietary modifications.
  - Sole engineer for the software implemented aspect of the project.
  - Key interfaces used are I2C, flash file system, PPP dial-up, serial, GPIO, A/D.
  - SNMP MIB implementation for the embedded target.
  - C# based SNMP feature registration app to register and manage various product features

- **LwIP Open Source project (savannah.nongnu.org/projects/lwip)**
  Contributing Member *January 2007 to July 2009*
  - I was an participating member of the lwIP open source project. In addition to various testing and bug reporting contributions, I implemented the fragment reassembly mechanism due to poor performance of the previous version. My version was then rolled into the mainline.

- **Honeywell Operating Systems (honeywell.com)** **Redmond, WA**
  Senior Software Engineer *January 2006 to June 2009*
  - Took on the role of network infrastructure architect and primary dev for a geographically distributed team focused on enhancing the Deos OS (http://ddci.com/products_deos.php) offering.
  - As lead for the development of the network functionality, I developed and communicated a vision for what the next generation communication layer should look like to get input and buy-in from the rest of the team.
  - Wrote performance and statistical collection services, implemented netstat, and other well-known stack services.
  - Completed a trade and feasibility study to replace the current aging network stack with a new offering to satisfy a complex suite of future requirements. The result of this study was that we adopted the lwIP network stack, and I integrated the stack into the Deos,
  - Released a verified and validated package (DO-178b Level D) to multiple customers.

- **Honeywell Aircraft Environment Surveillance System (honeywell.com)** **Redmond, WA**
  Software Engineer *January 2003 to January 2006*
  - I was a founding/core member of the Airbus A380 Aircraft Environment Surveillance System team that integrated the IOM, TCAS, EGPWS, and Radar components into a single system that interfaces with the airplane and users via an IOM module. I wrote the lightweight multi-core aware UDP/IP network stack, and was instrumental in making the communications layer work deterministically and efficiently across all 4 systems on the avionics network. I was also the primary contact for OS questions and RTOS concepts within the group. In addition to my primary role, I contributed a myriad of key functions on an as-needed basis.
  - Automated Build: Conceived, initiated, and implemented an automated build using pre-existing scripts, "glue" python code and batch files, and Windows "Scheduled Tasks" functionality to setup a nightly build, with automatic status reports emailed via Microsoft Exchange and PythonWin.
  - System optimizations: Conceived, designed, and implemented several optimizations to reduce CPU usage on a busy target. Optimizations included DMA controller driver and integration to copy data from PCI space to local memory, which resulted in a measured 10% savings off maximum CPU usage. Additional optimizations consisted of function level design optimizations driven by analysis of the assembly output by the compiler.
  - PPC/AMD Ethernet Driver AND UDP/IP stack: I created the requirements, design, implementation, documentation, and support tools for the Ethernet driver, UDP/IP stack, and API used by all functional module teams, in addition to other unrelated teams throughout Honeywell. The Ethernet(802.3) driver supports the PPC8260 FCC/SCC and AMD972 Communications Controller, was designed with intent to certify to a DAL of B, as described in DO-178b. Significant successful efforts were made to make the Ethernet driver easily configurable for all users and requirements.
  - AFDX test responder: Designed, implemented, tested, and completed according to Airbus requirements on time, despite being flagged as a high-risk schedule objective from the outset. The test responder is required to take commands as input from an "exciter" node, and respond with the correct response in order to facilitate testing of the airplane's AFDX network. Visual C++ 6, MFC based Ethernet driver configuration tool: Generated requirements for output, and implemented a tabbed dialog box application to generate a binary Ethernet configuration file.
  - I collaborated with another teammate to support integration of flash memory abstraction layer for a journaling flash file system with the file system developers from India.
  - I provided significant support across team boundaries for build environment and OS support. This involved aid with tasks such as OS configuration , make file debugging/generation, source code debugging, communication link issues, multi-tasking/inter-processor timing issues.
  - Miscellaneous utilities implemented in Python, including a tool to extract check-in information from PVCS for inclusion into the change tracking tool.
  - I wrote various memos explaining system design, and a white-paper on how to use a stack trace to debug an exception.
- **AT&T Wireless/Netro (att.com)** **Redmond, WA**
  Software Engineer *July 2000 to November 2002*
  - I was a member of the OAMP team for development of premier, toll quality voice and high speed data, fixed wireless high-energy system utilizing PowerPC 860 on multiple circuit packs (nodes) with both single and dual processor configurations.
  - Leadership - led efforts to debug, test, and implement air link metrics collection; new OAMP related feature concept, design, and implementation; go to expert on many code subsystems and compilation issues, delegate issues to members of other teams and follow up with them to ensure the job is done.
  - Design/Implementation - Fault recovery system components; performance metrics monitoring, collection and analysis tasks; software download; customer unit installation tool; event registration; serial connection management; persistent and reliable propagation of configuration stored in flash memory, including storage of V5 switch configuration; multi-node, multi-task, multi-processor system.
  - Interface design/Collaboration - Proprietary message routing protocol; Interface design for movement of air link voice and data metrics from DSP to host CPU to java GUI client; interface with remote unit installation tool using PalmOS serial interface to embedded system; serial connection protocol to interface with java GUI client; remote unit flash download interface with other embedded components; performed ClearCase merges for various releases.

  - Porting - Ported portions of the Fault Management system and other OAMP related features from the obsolete platform to the next generation platform. I created and edited make files for build changes.
  - Tools/OS used - In-Circuit emulator (ICE) and BDM source level debuggers, ClearCase, pSOS

- **Motorola (motorola.com)** **Arlington Heights, IL**
  Software Engineer *January 1999 to June 2000*

  - I worked with an experienced development team on Motorola's analog cellular base station. The team used SEI level 3 processes and was ISO 9001 certified. I studied Motorola's proprietary CDMA and analog technologies via Motorola University courses.
  - Development/Maintenance - Performed bug fixes and added enhancements as needed on miscellaneous components of the cellular base station code base. I utilized an Ethernet sniffer to diagnose issues with the base station's LAN. Added features in 6809 assembly and C.
  - RF Planning/Debugging - Debugged and fixed client frequency planning issues.
  - Collaboration - Mentored three new employees on the system architecture and software, cellular systems, and SEI level 3 processes.
  - Perl-based DSL to simplify debug log parsing.

- **Illinois State university** **Normal, IL**
  Software Engineer *February 1996 to December 1998*

  - I led a small development team that provided a user-friendly Internet installation and PC configuration that was used to integrate various applications with campus labs.
  - Design/Implementation - GUI development using Visual C++ 5.4/6 and win32/MFC to create utilities to aid user in connecting to and collaborating on the Internet; a web based employee scheduling system.
  - I implemented the installtion procedure using Installshield.
  - Interface design/Collaboration - Created dll to encapsulate all common code functionality on the development team related to Windows registry configuration, user preferences, etc.

Other Experience

- Linux Python app to listen for insertion of audio CD on dbus, and automatically backup CD to music NAS, which is accessible via my Sonos module.
- Implementated an automated bank account balance retrieval via Python's very nifty urlib2 module. It's run every night via cron, and optionally emails a bank account balance to one or more recipients one of which is a cell phone.

Technical Expertise

Languages: C, C++, Python, CMake, Make, Bash, Javascript
Open Source Contributions: LwIP, linux kernel (Serial RapidIO driver)
Concepts: Multi-node system integration, Real-time systems, multi-process and multi-threaded systems, Ethernet/networking, RTOS, git, subversion, DO-178b
Driver Experience: MSP430, CAN, I2C, GPIO, SPI, Ethernet, DMA, AFDX, Flash, FPGA, Keypad, Display, VGA
Certifications: Certified ScrumMaster (CSM) – Agile development, Certified Netburner Consultant

---

Education

- **University of Washington** **Seattle, WA**
  Agile Development certificate program at U.W. (ScrumMaster Certification) *2008*
- **Illinois State University** **Normal, IL**
  College of Engineering, B.S. Computer Science *1994 – 1998*