Alexey Burshtein

Embedded Software Engineer | DSP & Real-Time Systems Expert
E-mail: ahitech@gmail.com
Phone: (+972) 53 8800339 (WhatsApp accessible)
GitHub • LinkedIn • Personal website • Facebook • Telegram

Experienced DSP and Embedded Systems Developer with 20 years of expertise in real-time systems, signal processing, and hardware/software integration. Skilled in algorithm optimization, system design, and technical documentation. Recognized for delivering high-performance, production-ready solutions across diverse technological environments.

Key skills

- Vast knowledge and experience in bare metal development, without any OS.
- Expertise in writing code and device drivers (kernel and user-area) in the following environments: RTOS (RTLinux, QNX, VxWorks, On-Time, RTX64), Windows, Linux, BeOS, Haiku.
- Programming languages: C, C++, C#, Python, Matlab, Perl, TCL and others.
- FPGA integration and *to some degree* development (I'm familiar with Vivado and Libero).
- Hardware/Software Integration, including design and layout of the breakout boards.
- PCI-e, Ethernet, Network Protocols.
- Algorithm Optimization, Signal Processing and Filtering.
- Technical Documentation and FDA Compliance.

Language skills:

Fluent: Russian, Hebrew, English, Esperanto.
Intermediate: Ukrainian.
Learning: Arabic, German.

Relevant education

- M. Sc. In Computer Science (In progress) – Open University, Israel.
- B. Sc. In Computer Science – University of Haifa, Israel, 2013 – Received **Excellence Scholarship**.
- B. A. in Physics – Technion, Haifa, Israel, 2001.
- Additional courses, e.g. "Signal processing problems, solved in MATLAB and in Python" at Udemy.

Work experience

04.2024 – 11.2024 – Harman Automotive Israel (subsidiary of Samsung Electronics)

Position: **DSP Software Expert**

I took part in developing a system that monitors heart and respiratory rate of the driver, as well as passenger localization and child presence detection, using radar technology and Doppler effect, in realtime.

- Performance-efficiently translated complex algorithms received from R&D as Matlab files into C++ and embedded them into existing code.
- Adapted the code to Samsung Linux.
- Acquired data from custom board using Infineon radar sensors and A2D converters.
- Implemented Continuous Morlet Wavelet transform using fftw3 library.
- Ported the whole application to the new hardware (from Nvidia Jetson to Samsung Exynos).

- Implemented the UI from non-functional mock-up to fully functional application that runs both on PC and on Android tablet, connecting to data acquisition HW via a protected data transfer protocol.
- Set up the testing environment in virtual machine using Microsoft Hyper-V instead of using a dedicated Linux laptop.

09.2019 – 03.2024 – Alpha Omega Engineering

Position: **DSP Software Expert**, **Senior SW Developer**, **DSP technical leader**.

The company develops software-hardware solutions for neuroscience research and neurosurgery using technology called "Deep Brain Stimulation". Development in C (~70%) and C++.

- Worked in hard-realtime environment, with 200-260 µsec per cycle, in bare metal and RTOS. Single-handedly designed and implemented soft-realtime solution under Windows, wrote an article about it on Habr.com (in Russian). A translation of this article in English appears on my site.
- Implemented several products, such as Incept, Nucleus and AlphaRS Pro, from initial design to production. The hardware part for two first products uses TI TMS320C6678 DSP, the latter doesn't have a DSP at all.
- Reverse engineered older products, such as NeuroOmega, in order to enhance them and to re-implement them using available hardware, since the previously used components reached EOL.
- Participated in designing and implementing the new UI for the systems.
- Prepared documentation for FDA certification of the developed products, both for research and for clinical devices.
- Developed and implemented the DSP-less solution for AlphaRS Pro, which relies on user's computer hardware and CUDA for performing tasks previously done on DSP, such as filtering, ordering, brain stimulation, protection etc.
- Reduced boot time of the DSP by 98%, from 3-5 minutes to 1-3 seconds.
- Implemented high-bandwidth data transfer of 320 MB/sec through PCI-e over Thunderbolt connection, including implementation of PCI-e Endpoint, PCI-e Root Port and "Hot plug" functionality.
- Wrote driver for 10Gbit Ethernet adapter attached to the TI "Keystone 1" board.
- Implemented software filters that reduced noise in the input signal (Butterworth with arbitrary number of poles).
- **"Exceeded expectations" in yearly review of 2020**; as a result, I was promoted to DSP technical leader.

2016 – 2019 – ACS Motion Control

Position: **DSP Software Developer**.

The solutions developed by the company are controlling linear and rotary motors with nanometer precision.

- Worked in hard realtime environment, with the cycle length that is strictly equal to 50 µsec.
- Prevented overshoot and undershoot oscillations around target values, using PID controller tuning and feedforward control. Implemented motion profile with 6 derivatives (speed, acceleration, jerk, jounce, crackle and pop).
- Designed and implemented motion delay technique in laser controller, that deals with delay in actual laser's activity due to the time required to warm the laser up.
- Designed and implemented a new controller, including writing "main()" function.
- Implemented high-speed connectivity protocol between the controller and the computer that asks for motion, using Bi-directional RAM.
- Refactored the existing code, reducing runtime of one of the algorithms from 7 to 3 µsecs.
- Added software transistor overload protection to CMhv product.
- Performed research project in Sigma-Delta Modulation. Researched additional hardware of F28379 dev kit.

2012 – 2016 – PTC Ltd.

Position: **Software Developer**. I worked at PTC/Creo 3d CAD software, in "Assembly" department, in "Family table" team. The product was not Embedded, but it was soft-realtime, developed in C/C++.

- Implemented creating a new object based on an existing object and table of differences.
- Implemented ability to open files created in competitors' software, such as AutoCAD or SolidWorks.
- Created and set up automatic testing cluster on Windows using Jenkins as the controller software.

- Single-handedly prepared transfer of the development from Solaris to Windows and testing from HP-UX to Windows. For this task, received **“Outstanding Achievement” certification in 2012**.
- **“Exceeded Expectations” in yearly review of 2015**.

2010 – 2012 – EZchip Technologies Ltd.

Position: **Software Developer** in Driver team that develops controlling SW of network processors.

- Implemented power-up sequence of network processor NP4, including setup of network interfaces, QoS weighted queues, configuring first two levels of OSI model etc..
- Implemented soft-reset of NP3, — rebooted the processor’s core without losing network link on TX and RX.
- Designed and implemented Interface Swap technology – reassigning physical network adapters into different configurations without losing link on the other interfaces.
- Learned about high-bandwidth network protocols, like 1*_BASE, XAUI, XLAUI, CAUI, XFI, Interlaken.

2009 – 2010 – Microsoft Israel Development Center.

Position: **Student Software Engineer** in Forefront Endpoint Protection team.

I worked on the networking subsystem of Windows and on Windows Defender anti-malware tool.

- Implemented Generic Application-Level Protocol Analyzer (GAPA) that analyzes inbound and outbound traffic and drops it in case a threatening packet is detected. The implementation was in C#, C++ with STL, MFC and Win32 API.
- Implemented tools for testing efficiency of protection and telemetry (in C#).
- Documented the legacy code in UML.
- Worked with big data received by telemetry in order to predict possible problems.
- Implemented and ran automatic testing system for the code of the whole department.

2002 – 2009 – IBM Haifa Research & Development Lab.

Position: **Student Software Engineer** in Systems & Storage department.

The department produced microcode for data storage servers and implemented advanced copy functions.

- Developed the realtime kernel-level microcode of storage servers DS6000 and DS8000; in particular, the PPRC functionality.
- Ported the kernel pipe feature from AIX operating system used on DS8000 to Yellow Dog Linux used on DS6000.
- Single-handedly developed and maintained an automatic bug report analysis tool, based on decisions trees. **Saved to the company 3 PYs** by reducing time invested by specialists into analyzing these bug reports.
- Gained deep knowledge of multhreading programming in highly parallel system. Learned a bunch of scripting languages and tools, like awk, sed, expect, TCL, incrTCL, Perl.

Projects, Publications, and Open-Source Contributions:

- Published technical article on Habr.com about the soft real-time solution in Windows that I designed and implemented single-handedly (in Russian). A translation of this article in English appears on my site.
- Published a prior-art publication with other IBM employees in field of cellular communications.
- Creative writing: published multiple short stories (in Russian). ISBNs: 978-5-4483-8216-1, 978-5-4493-9274-9, 978-5-0050-8145-2, 978-5-0056-6092-3 and many more.
- Finished Arduino projects: weather stations, art lamps, smart pet fountains that recognize an approaching cat using image recognition, graphic equalizers etc.
- I participate in development of open-source operating system Haiku and programs for it. Was interviewed for the site “IsComputerOn”.