

Joel M. Hegberg
JoelHegberg@computer.org

Objective To continue my software engineering career in the field of Computer Science, which is complemented by my extensive experience with many different hardware and software platforms.

Education **Bachelor of Science, Computer Science:** May 1995
Northern Illinois University, DeKalb, Illinois
Major/Minor: Applied Computer Science/Chemistry
GPA: 3.570 (4-point scale), 4.000 within major
Honors: *Golden Key Honor Society* - Initiated in 1993

Associates of Science, Computer Science: June 1992
Kishwaukee College, Malta, Illinois
Major/Minor: Computer Science/Chemistry
GPA: 3.790 (4-point scale), 4.000 within major
Honors: *Phi Theta Kappa* - Initiated in 1991

Summary of Experience I have been fortunate to work on a large variety of projects within the computer software industry which has helped continually improve my programming skills. Creating software architectures, implementing them on many hardware platforms, learning from source code inspections, performing defect tracking and resolution, formalizing build procedures, participating in process documentation and improvement, writing unit test software; All of these have given me a solid understanding of the entire project lifecycle from concept to end-of-life.

The range of technologies I've worked on has provided me great experience in areas such as **embedded OS designs** (Android, Linux, Blackberry QNX, Windows Mobile, Symbian, OS-9), **networking protocol stacks** (TCP/UDP/IP/PPP, ARP/DHCP, SMTP/POP3/IMAP), **cryptography** (certificates, OpenSSL, Cryptoki and BSAFE standards from RSA), **cellular technologies, Bluetooth** (stack, profiles, coexistence with WiFi, interoperability testing, firmware drivers for USB and UART components), **USB Host / Client / OTG** drivers, and **Digital TV Broadcast** standards.

04/2014 – Present Software Engineer at Motorola Solutions (Applied Technology Group)

- ❖ Designed and implemented **IP-network sharing** software for use over a **mobile LTE stack** to enable first responders to have data capability via their mobile devices. The software supports simultaneous **IPv4** and **IPv6** network capabilities, data rate controls, and is also used to carry voice and video.
 - ❖ Implemented a **Universal Software Radio Protocol (URSP) Hardware Driver (UHD)** for controlling custom Motorola Solutions RF hardware both locally and remotely. This open standard allows the hardware to be configured and used by many software applications, including **GNU Radio**.
 - ❖ Implemented an interface to the **REDHAWK** framework for the **UHD** software to allow custom waveforms to be created and run Motorola Solutions RF hardware.
 - ❖ Designed and implemented a **High Speed Storage Manager** which managed RF data stored at high data rates by FPGA-based hardware. This was achieved by creating an entire **XFS** file system around the stored data *after* it had been written to disk by the hardware. This was by far the most ambitious software I wrote while at Motorola Solutions since it involved implementing all of the XFS file system structures, including **superblocks, inodes, directory blocks, inode** and file **B+Trees** with nodes and leaves.
-

08/2013 –
03/2014

Software Engineer Consulting at Citrix Systems

My work at Citrix was part of a team of software engineers tasked with porting Citrix Receiver software from Linux to the Blackberry OS 10 platform in a very short amount of time.

- ❖ Created a native UI front-end in **Blackberry Cascades** as a possible prototype / replacement of the slower Adobe AIR front-end. This provided experience in **QML layouts, UI dialog interactions, flow layouts** with dynamic data (app names, icons), **activity indicators**, and **UI transitions**.
- ❖ Gained extensive **certificate management** experience including encoding/decoding certificates in various formats (**DER, PEM, PK7**) and using them with **OpenSSL** and **CURL**.
- ❖ Adapted **single-threaded, Linux-specific** code to run within **QNX** as a **multi-threaded** process. This included extending the original inter-process communication (IPC) implementation to handle messaging between multiple threads while retaining source code compatibility.
- ❖ Defined and implemented a “StoreBrowse”-centric API to make the **connectivity code agnostic to the UI code**. This allowed faster integration and easier debugging by allowing for test cases to be run **even in non-UI mode**.
- ❖ **Optimized code** to run efficiently given battery constraints and **profiled code** to determine areas which would be best candidates for optimizations.
- ❖ Debugged many difficult issues with both passive (**Fiddler, Wireshark**) and active (**debug logging** callback functions in **CURL, breakpoints**) methods for many software components including **CURL client/server data transfers**.
- ❖ Learned to use many task-specific tools including **Momentics IDE** (both Linux and Windows), **Blackberry x86 simulator**, **Perforce** source control, and **Fiddler** network traces.

03/2009 –
07/2013

Staff Software Engineer at General Dynamics C4 Systems and GD-Itronix.

- ❖ My first project at GDC4S was an embedded device required to not have an Operating System due to throughput and memory constraints. I wrote a **zero-copy UDP/IP/ARP networking stack** from scratch, set up **snoopable L2 cache** for efficient data transfers, and utilized **immediate-mode addressing** routines in assembly language for highly efficient comparison of data buffers.
- ❖ Another project was a Linux-based system required to communicate time and date information with a GPS-like device using **Manchester-encoded signaling** via GPIO lines at a non-standard baud rate. This required the use of high performance timers within Linux to accurately drive the software which performed the **real-time signal encoding and decoding**.

My **GD-Itronix** work on the **GD300** and **GD310** ruggedized handheld Android-based devices was very broad. I was responsible for implementing and debugging features in the Linux kernel / Android / boot-loader layers, writing GUI apps for GD-specific features, producing official candidate and release images with release notes, producing factory test images, and creating factory test software.

- ❖ **Linux kernel:**
 - Responsible for **kernel configuration, debugging, building, and release**.
 - Implemented new features and fixed defects in many **driver sub-systems** including USB, networking, touchscreen, display, camera, GPIO setup, GPS, e-compass, accelerometer, RNDIS / CDC-ACM, CPU frequency scaling, battery calibration and charging.
 - Created an **automated battery learning-cycle** (calibration) service.
 - Integrated **several proprietary non-cellular radios** with our device.
 - Added startup changes and bug fixes to **low-level boot loader** (U-boot).

- ❖ **Android OS 2.1, 2.2, 2.3.4, 4.0.x:**
 - Developed a **newer software release image format** to replace existing GD solution which decreased the images from **400 MB down to 50 MB**.
 - Created **OS candidate and release images** with release notes for customers.
 - Created extended **SDE/SDK releases of GD-specific API's** for developers.
 - Fixed bugs in **Android networking, framework, and UI** layers.
 - Created **factory test OS image** with utilities to validate hardware.

- ❖ **Android apps:**
 - **Eclipse IDE** was used for **writing, debugging, profiling, and signing Android apps** on hardware and within the Android emulator environment.
 - Created a **GD300 OS Installer app** for a better customer experience. Prior to this all GD300 releases had to be tediously flashed from a command-line shell using a debug board with a serial port on the device.
 - Wrote a **factory test app** which was a GUI front-end for low-level factory test software.
 - Created an **automated battery calibration app** as a GUI front-end for using the battery calibration process I implemented in the Linux layer.
 - Wrote a **GPS app for time sync** of multiple GD300 units spread over a large geographic area. To accomplish this required me to implement new algorithm logic in the Android OS location framework.

- ❖ **Power management:**
 - Debugged issues with **CPU frequency scaling** which provided significant power savings of **~85mA**.
 - Debugged **camera power domain bugs** which yielded savings of **~100mA**.

- ❖ Set up and maintained a **multi-site, multi-user Mercurial** source repository running **Apache** web server.
- ❖ Set up and maintained a **multi-user build machine** for creating Android images.

05/1999 -
03/2009

Group Lead Software Engineer within the Mobile Devices group at **Motorola**. While in this position, I worked on advanced cellular smart-phone projects which were the start of convergence for many features of standard PDA's into a cellular phone.

- ❖ **Bluetooth / WiFi coexistence** development for Broadcom, CSR, and TI chipsets. Debugged coexistence issues using logic scopes, over-the-air packet sniffers, server-side throughput software, and embedded device HCI sniffing.
- ❖ Worked with car manufacturers (Landover, BMW, Jaguar, Audi, Aston Martin) on **Bluetooth carkit/cell phone interoperability**.
- ❖ Attended multiple **Bluetooth Unplugfest events** worldwide for interoperability testing / debugging and created detailed reports of findings for future improvements.
- ❖ Worked on **Broadcom 2045, 2046, 4325 Bluetooth drivers** for stack and driver integration with Microsoft Windows Mobile OS.
- ❖ Updated Windows Mobile Bluetooth and UART drivers for better **power management**.
- ❖ Created **factory test software** for automated testing of Bluetooth, UART, and USB components.
- ❖ Wrote **Broadcom 2033, Broadcom 2035, and CSR BC3ROM** Bluetooth USB drivers for Symbian OS cellular phones.
- ❖ Written **USB Host class drivers** to provide an interface between two processors within a cellular phone.
- ❖ Debugged USB inter-processor communications using a **CATC USB Chief protocol analyzer**.
- ❖ Optimized and debugged **27.010 Mux protocol** written as a Symbian OS CSY module used for inter-processor data exchange.
- ❖ Debugged Mobile-Originated and Mobile-Terminated (MO/MT) voice and data calls using an **Anritsu GSM base station**.

- ❖ Assisted in **architecture design process for multiprocessor-based smart-phone** solutions based on high-speed wireless networks such as GPRS and Packet CDMA.
- ❖ Created and maintained networking protocol stack and interface libraries for **Point-to-Point Protocol (PPP)**.
- ❖ Created and maintained **SMTP, POP3, and IMAP email protocol libraries** for use in embedded environments such as cellular phones and set-top boxes.
- ❖ Developed a **real-time WAP (Wireless Application Protocol)** service daemon to handle browser requests from a Java Virtual Machine.
- ❖ Debugged protocol incompatibilities between a new cell phone and various **TCP/IP/PPP protocol stacks**, including Windows NT/2000, PocketPC, and various flavors of Unix.
- ❖ Performed testing using a **RACAL GSM simulator** for cellular data performance measurements and debugging.

While in this position, I worked with many top companies including Broadcom, TI, CSR, Microsoft, ATT, NTT DoCoMo, Symbian, SoftConnex, Mentor Graphics, Hitachi, Denso, Harman/Becker, Parrot, Frontline, Peiker, and others.

01/1998 -
04/1999

Field Applications Engineer for **Microware Systems Corporation**, where I traveled domestically and internationally to meet with existing and potential customers to give demonstrations of our products, technical presentations, and on-site customer assistance. In addition, I attended several industry trade shows to represent and promote Microware's products.

While in this position, I worked with many large customers including Motorola, Intel, Rockwell, Zenith, 3Com, CompuServe, FAA, Nortel, Bell Atlantic, and Fermilab.

05/1995 -
12/1997

Senior Software Engineer for **Microware Systems Corporation** in the Network Technologies Group, which was within the New Media (Digital Television) business unit. Our integrated product for New Media was DAVID (Digital Audio Video Interactive Decoder). During my three years in this group, I gained tremendous experience through the following projects:

- ❖ Provided on-site support and development at Motorola for 10 months, providing technical assistance on their **MAP** digital cellular phone project which included writing and upgrading **POP3/IMAP/SMTP email libraries, PPP connectivity software, and modem CHAT libraries.**
- ❖ Provided on-site training and assistance for Communicate in London, England for **porting their wireless GSM cellular protocols** to the OS-9 MAP phone project.
- ❖ Architected and implemented a cryptographic package for the OS-9 RTOS based on the **Cryptoki and BSAFE standards from RSA.** The end product incorporated a wide variety of protocols, including **DSA digital signatures, RSA, DES and Triple-DES, RC2, RC4, and RC5.** This project was for a set-top box deployed by Hong Kong Telecom.
- ❖ Implemented several dynamic network protocol stack drivers for OS-9's **mwSoftStax** networking infrastructure and added new features/bug fixes to the **mwSoftStax** framework. During this process, I architected a simpler and more maintainable method of writing protocol stack drivers within OS-9.
- ❖ Designed **channel navigation API's for DVB and ATSC** digital television standards for Microware's Digital Broadcast Environment package.
- ❖ Developed a graphical front-end ("**player shell**") application as an example for digital set-top box developers using Microware's Digital Broadcast Environment package.
- ❖ Created session management API's and daemons to implement the **DSM-CC application** download protocol over various networks.
- ❖ Developed example video games for use on digital TV set-top boxes (playable via an infrared remote control) using various **graphics effects libraries** I also wrote. These programs also included writing API's to support various graphics formats such as **BMP, GIF, and JPG.**
- ❖ **Authored chapters** for Microware's published manuals on many of the API's I worked on, and also included many example networking and graphical applications to illustrate the concepts.
- ❖ Designed and implemented **extensive unit test suites** for various projects under ISO-9001 requirements to meet strict Quality Assurance standards.

1993 - 1996 The World of 68'micros magazine, Warner Robins, Georgia: Author of a regular OS-9 column for the magazine entitled "**OS-9/OSK Answers!**" The column was dedicated to helping novice programmers become more experienced with the OS-9 operating system.

1990 - 1995 During my years in college I developed several software packages (**word processors, fax-modem send/receive, graphics design programs, checkbook balancing applications, video games, and many utilities**) which were marketed through the following companies: *SuperSoft, Inc.* (Jackson, Michigan), *CoCoPRO!* (Ypsilanti, Michigan), *Sub-Etha Software* (Lufkin, Texas).

1988 - 1992 The Rainbow magazine, published by Falsoft, Inc., Prospect, Kentucky: Several of my early programs with accompanying articles were published in The Rainbow color computer magazine. The earliest article was in December 1988, and the last was in May 1992.

References Available upon request.

General I have experience in many computer programming and markup languages including **C, C++, C#, Java, Javascript, XML, HTML, BASIC, COBOL, Fortran, and RPG III**. I've also written and debugged **low-level assembly code** on many different platforms (**ARM, PowerPC, x86, 680x0, 6809**).

Working within **Linux-**, **Unix-**, and **Windows-**hosted environments I have used the following development tools:

- **ClearCase, Mercurial, and Perforce** version control
- **Eclipse IDE** for Android to edit, compile, debug, and profile software
- **Momentics IDE** for Blackberry to edit, compile, debug, and profile software
- **Symbian OS** and **Windows Mobile** development tools
- **VirtualBox** and **VMWare** virtual environments for both host and target platforms
- **DDTS** and **Bugzilla** defect tracking systems
- **TI OMAP development tools** for OMAP2420, 3430, 3530, 3730
- **Qualcomm Snapdragon** tools for Android
- **PowerPC CodeWarrior** embedded development software
- **x86 embedded Linux** development tools
- **Microsoft Visual C/C++** for writing x86 test utilities
- **WireShark** and **Fiddler** for debugging network-related issues on many platforms.