**Brian Jung Myeng Lee**
E-mail: jm3lee@gmail.com

## EDUCATION

Honours Bachelor of Mathematics (Computer Science)
University of Waterloo, Waterloo, Ontario, Canada, 2001 - 2006

## TECHNICAL SKILLS

**Software Development Engineer II, Amazon Fresh.**
**March 2016 - Present**

- Designed and maintained an internal order analysis tool using Ruby on Rails, HTML5, CSS and JavaScript with more than 100 users across 3 organizations with one second page load; received a GPT leadership award.
- Maintained an order planning system as a part of 10 people team using Java, Python and AWS services.

**Software Developer, D2L Ltd. (formerly Desire2Learn Ltd.)**
**December 2014 - August 2015**

- Designed, documented and implemented the most requested feature by clients in a three-person team. The feature was well received at FUSION 2015.
- Mentored, trained and coordinated junior developers to diagnose, fix and test large defects better.
- Maintained the back-end written in C# for in-market products in an eight-person team.
- Maintained the front-end written in JavaScript and HTML for in-market products in an eight-person team.

**Systems Programmer, Wolff Electronic Design**
**March 2013 - May 2014**

- Programmable Commercial Grill
  - Provided consultation for the middleware in a 3-tier system with a user interface, middleware and a control system for a commercial grill project set to launch in the late 2013 using C, Python, GNU Make and shell scripts.
  - Improved the build system to support x86 and ARM cross compilation using GNU Make on Linux and Mac.
  - Designed, and implemented a simple packaging and in-field upgrade system using C and shell scripting on Linux.
  - Designed, and implemented a code generation framework to produce a thin layer around yajl, a JSON library, to manage system settings and assets using Python, C and GNU Make.
  - Improved in-field log collection, bug reporting system and a unit test framework in C and Python.

  - Designed configuration management framework to allow for hardware and software customizations using JSON.

  - Helped design core features by gathering requirements from non-technical clients, and creating a comprehensive technical requirements using LaTeX as well as internal wiki and ticketing system.

- Smart Energy Monitoring Devices
  - Designed a protocol over RS-422 to exchange, and synchronize data between a master PC using C# and slave embedded devices with firmwares written in C.

  - C# protocol stack served as a basis for a test application as well as a minimal Windows service.

**Software Engineer, Sandvine**
**September 2012 - July 2013**

- Documented, designed, implemented, and tested a binary patching tool for Network Processing Units in Policy Traffic Switches and its unit tests in C++ on Linux after reverse engineering an undocumented binary format.

- Documented core concepts in the device detection kernel module for FreeBSD in C. Reviewed, and identified a testing road-map for the component.

- Helped other new hires to analyze the source code base, and troubleshoot the build system issues using GNU Make and bash on Linux.

- Proposed, and discussed ways to improve nightly testing, design documentation and source code base at length with the immediate manager and Associate Vice President. Most proposals were well received, but on hold due to budget and time constraints.

**Embedded Systems Software Developer, OS Firmware Security, Research In Motion**
**April 2012 - August 2012**

- Reviewed device security component for BlackBerry 10 devices written in C for QNX, and planned its refactoring efforts to improve security and maintenance of the software; the informal proposal has been accepted, and started refactoring process.

- Actively involved in on-going improvement for debugging tools support to resolve security implications due to poorly controlled access to telnet and ftp services; wrote two programs in C to demonstrate tools that will replace existing tools.

- Analyzed, and designed a solution to replicate a features from vendor's PC tool to manufacture and rework BlackBerry 10 devices.

- Improved internal documentation by commenting source code, creating missing architecture documentation, and organizing internal Wiki pages to help new hires as well as to reduce support cycle.

- Leveraged knowledge from the previous position at RIM to setup a Coverity machine scheduled nightly as well as a simple ctags web interface to catch bugs early in development.

**Security Automation Analyst, Security Automated Analysis, Research In Motion**
**August 2011 - March 2012**

- Performed an analysis parallel to *On Analyzing Static Analysis Tools* published by the NSA to compare Coverity, Klocwork and Fortify using Perl, Unix shell scripts and git; the result has been published as a whitepaper, *NSA Juliet Test Suite Results*.

- Re-implemented internal API Scanner using C# by writing a custom regular expression engine, and decreased false positive rate from about 80 percent to 4 percent while maintaining equivalent speed.

**Embedded Systems Software Developer, OS Platform Support, Research In Motion**
**June 2009 - August 2011**

- Participated, documented and lead QNX bring up for the next generation of BlackBerry operating system as a member of a team of three; involved in continuous technical development, training and support in areas such as driver design, development roadmap, gate-keeping Perforce and build system using GNU Make and Perl.

- Initiated discussions on moving from Windows to Linux as a development platform for future BlackBerry OS development; completed initial support by working with a build system specialist.

- Provided technical support for PlayBook to help peers in troubleshooting boot up issues as well as driver development on QNX; also acted as a main technical contact between RIM and QNX for general technical inquires as well as build system.

- Helped to maintain bootrom, firmware loader, operating system initialization and drivers maintenance in C and ARM assembly as part of early OS bring up as well as reset analysis as a member of team of ten.

- Involved in Marvell microprocessor silicon bring-up; identified a hardware bug with Marvell engineers, and verified fixes in frequency change and low power mode in the following stepping of the silicon using JTAG.

- Identified and filed a bug report against RealView Compiler regarding binary generation when "pld" instructions are used in inline assembly using disassembly dumps; initially analyzed and verified floating point support in disassembly in in the internal code base as well as third party graphics libraries.

- Produced and tested builds with BlackBerry OS applications for platform devices by assisting in integrating ARMv7 MMU support; made the first phone call using a recent project device on a 2G network before the radio team. The same device was later used to demonstrate a phone call on a 3G network in China.

- Preferred proactive responses to users' requests and face-to-face communications over e-mail; located and provided prototype boards in shortage in person for members from other teams; resolved test firmware issues with manufacturing and radio teams in person.

- Expanded internal documentation on Marvell Extreme Debugger scripting to minimize time consuming debugging cycle; documented instructions to build BlackBerry OS for prototype devices with little or no support from respective teams.

**Embedded Systems Software Developer, Bluetooth, OS Protocols, Research In Motion**
**October 2006 - June 2009**

- Maintained and integrated third party Bluetooth stack in BlackBerry OS in C across

multiple layers such as HCI, L2CAP, HFP, HSP, AVDTP, A2DP, AVCTP, AVRCP, RFCOMM and DUN while working with accessory vendors and career testing labs as a member of a team of three.

- Helped CSR engineers to identify a hardware bug in CSR BlueCore chips by taking advantages of micro-kernel design over the span of one year.
- Improved DUN throughput by a factor of two by applying micro-optimizations in C; performance was verified in a RF test chamber. RIM finally met a career's acceptance requirement for the first time since Bluetooth was available on BlackBerry.
- Improved source code base for easier debugging; focused on accurate error logs with relevant information, updated utility macros to correctly pin-point source of crash and enhanced readability of debug logs.
- Fostered close relationships with Java Bluetooth application team by attending bug report review meetings, sharing recent bug fixes and helping them debug Java application code when necessary.
- Expanded internal documentation regarding subtle implementation details in the system, execution flow and general notes to help new hires. This document is in continuous use by the current Bluetooth team.