Michel Bélanger

C/C++ senior software developer

Coordinates:
mail@michel-belanger.ca
(819)-319-6005

Languages:
French and English

Clearances:
Canada Secret
Canada Top-Secret
NATO Secret
Canadian Controlled Goods Program

TABLE OF CONTENTS

1 Profile

I am a highly experienced senior software developer with 26 years of experience, including extensive experience with embedded devices, custom firmware and software for IoT devices, optical transceiver modules, Secure and Efficient Situational Awareness Ad-Hoc routing protocols, Global Express Aviation (GXA) project, Precision Aerial Delivery System (PADS), Unmanned Aerial Vehicle (UAV), Single-Photon Emission Computed Tomography (SPECT•CT) diagnostic solution, full-body patient simulator and other medical devices, routing devices and IP phones. I have designed and implemented C/C++ software for many different Operating Systems like QNX Neutrino, Linux, FreeRTOS, VxWorks and MQX. I have also designed and implemented many bare metal C/C++ firmware for different devices.

For the last 16 years, I have provided software engineering services for several Federal Government of private sector companies, such as Jabil photonics, the Canadian Forces (DND/CF), the Defence Research and Development Canada (DRDC), Wind River , Progress Rail/Electro-Motive Diesel, Honeywell Aerospace, CAE Healthcare, Siemens Healthcare, Nortel Networks, Mitel Networks, MMIST Inc., Cardia West and Solacom Technologies.

2 Aptitudes and Abilities

Strong analytical and problem–solving skills
Good project and time management skills
Capable of working under pressure and meet strict deadlines
Comfortable working in a team environment as well as independently
Highly motivated, adaptable, and eager to learn new technologies and methods

3 Education

Technical Diploma - Diplome d'étude Collégial
Technique des Systèmes Ordinés (Embedded Systems)
Collège Lionel-Groulx (1993 – 1996)

4 Professional Highlights

Jabil Photonics

- Firmware for optical transceiver modules
- Debug console for optical transceiver modules

Defence Research and Development Canada (DRDC)

- IoT security assessment and experimentation with the Zigbee network
- IoT security assessment and experimentation with the Lora/Lorawan network
- Real-Time Network Profiling and visualization (Qt5).
- Wireless Side-Channel communication protocol.
- Secure and Efficient Situational Awareness wireless Ad-Hoc routing protocol.

Department of National Defence (DND) / Canadian Forces (CF)

- Secure Messaging and Secure Configuration (SMSC).
- Mobility security risk assessment for Apple iOS smartphones.
- Mobility security risk assessment for Samsung Android smartphones.

Wind River / Intel

- ARINC 653 and DO178C Level A certifications of VxWorks 630 3.0.x.x.

Progress Rail / Electro-Motive Diesel

- Optimization and debugging of products like the PowerView.

Honeywell Aerospace

- AES (Aircraft Earth Station) controller for the GXA (Global Express Aviation) project.

Poutine Software Inc

- PS UDP for Blackberry devices - UDP/IP connectivity tester.
- PS TCP for Blackberry devices - TCP/IP connectivity tester.
- PS Scan for Blackberry devices - TCP/IP scanner utility.
- PS SSID for Blackberry devices - Wi-Fi network scanner.

CAE Healthcare

- OO platform layer for the STM32F37x (ARM cortex-M4) micro-controller.
- Resource manager for a micro-stepping motor controller with motion engine (L6480).
- Resource manager for a Digital I2C Humidity and Temperature sensor (Si7005).
- Resource manager for a Smart push-button on/off controller with Smart Reset™ (STM6601).
- Resource manager for a Dual Smart Battery System Manager (LTC1760).
- Resource manager for a Self-Contained 802.11 b/g/n Module with Networking Stack (RS-9110-N-11-22).
- TCP/IP and UDP/IP test application, running under Linux, for WIFI connectivity.

Siemens Healthcare

- Make The Symbia and Ecam projects code compliant to ANSI C/ISO C++ and solved issues/bugs.
- Port the build infrastructure from command line to IDE tools (Momentics).

MMIST

- High Availability Manager to launch and monitor processes and services under QNX Neutrino.
- Auto-On firmware on a PIC18F2423 micro-controller from Microchip Technology Inc.
- OO platform layer for the STM32F103 and STM32F205 (ARM cortex-M3) micro-controller.
- Auxiliary Power and Engine Controller firmware on a STM32F103 micro-controller.
- Stepper Controller Unit firmware on a STM32F103 micro-controller.
- Embedded web interface with CGI binaries to monitor/ program/control the Sherpa and the Bravo.

Mitel Networks

- Video Driver and Graphic Driver for Mitel 5324, 5340, 5360 and 5540 IP phones.
- Integration of PEG graphic library on VxWorks.

Nortel Networks

- Customization of Nortel IP phone AG2000 UI for a specific client.
- New windowing infrastructure in Unistim 3.0 for AG2000 Release 7.0.

CardioWest™

- Graphical engine for a prosthetic heart monitoring system.

M5T

- HTTP proxy server respecting RFC 2616

Solacom Technologies

- Telephony programmable digital switch ported from QNX Neutrino to Fedora Linux.

QNX Software System

- OS graphical tools (scientific calculator, network configuration utility and directory search utility).
- Linux Real Player 8.0 gold port to QNX Neutrino.
- QNX System Profiler and Eclipse SWT based chart engine (6.3.0, 6.3.0 SP1, 6.3.0 SP2 and IDE 4.0).

Altersys

- Emergency supervision software for a nuclear power plant.
- An SQL database graphic editor and Quality control software under QNX Photon.
- Device drivers for different hardware: PC104 cards, GPS Systems synchronization.
- Data acquisition system over the PCI bus and the VME bus.
- Device drivers for PLC over Ethernet TCP/IP, Ethernet UDP/IP, RS232 and RS485 Serial.

Multimeg

- Engineer, design and implemented a high speed side boards optimizer and a cant linear optimizer.

5 Projects

5.1 Project 1 – Jabil Photonics

Senior Principal Design Engineer
May 2022 – November 2023 (19 months)

Project Description:

As a design authority for the overall embedded firmware architecture for the Photonic Transceiver teams at Jabil Photonic, I was called to lead the engineering effort to develop CMIS embedded software. CMIS is the Implementation Agreement for pluggable transceiver modules and ensure compatibility of transceivers across a wide range of host hardware.

Tasks Performed:

CMIS Firmware for photonic transceiver modules

Developed custom firmware for photonic transceiver modules as part of the effort to develop CMIS embedded software using C. Designed and developed firmware for fiber–optic QSFP–DD transceiver modules compliant with the Common Management Interface Specification (CMIS) version 5.2 on a PSoC–62 dual core MCU (ARM Cortex M0+ and M4).

Debug console for photonic transceiver modules

Design and development for the Jabil debug console. The debug console is a Qt application running on the host side of a fiber-optic QSFP-DD transceiver module. The debug console utilize I2C serial communication or UART serial communication to interface with a QSFP-DD transceiver module to access the complete CMIS memory map. It allows visualization of the module features advertising, a W/R access of the module control registers and monitoring of the module statutes and flags.

MDIO firmware for optical transceiver modules

Worked with the PSoC–5 MCU microcontroller (ARM Cortex M3) and PSoC Creator IDE and tools to assist Jabil's Wuhan office Photonics project, including reverse engineering the optical transceiver modules produced by the Jabil Wuhan Photonics teams to identify design flaws and programming bugs (using BugZilla). Reverse engineering was required due to the poor documentation for the transceiver modules, resulting in members of the Photonics Team being unable to reproduce designs. I was tasked with reverse engineering the modules with the goal of reproducing the firmware involving extracting the design of the firmware and functionality from the source code.

Others

- Write software development guidance to be followed by the Jabil Wuhan Photonics teams and Jabil Ottawa Photonics teams.
- Oversee the implementation of a distributed version control system to tracks changes and ease the collaboration between Jabil Wuhan Photonics team and Jabil Ottawa Photonics team.
- Oversee the Jabil Ottawa Photonics team junior engineers
- Setup/maintain a host test station (Linux on a Raspberry PI 400) in the lap for the Eeffect optical transceiver project.

- Design a Raspberry PI 4 daughter board to test QSFP-28 and QSFP-DD optical transceiver modules.

Technical Environment:

C/C++, Barebone development, freeRTOS, Linux, GCC Tool chain, Cross–Platform development, Embedded systems, Qt, Real–time, CMIS, I2C, SPI, UART, JTAG SWD, DSP, ADC, EEPROM, TROSA, Raspberry PI 400, micro–controllers (PSoC 5 and PSoC 6), PSoC Creator IDE, Modus Toolbox IDE, Qt Creator IDE, Doxygen.

5.2 Project 2 – Defence Research and Development Canada (DRDC)

Information Technology Security Architect
August 2016 – March 2022 (68 months, count 54 months)

Project Description:

As an IT Security Architect resource (Cybersecurity practitioner) within the Resilient Tactical Networks Group of the Cyber Operations and Signals Warfare Section at DRDC, I was called to evaluate different hardware technologies for wireless communications, implement different network protocols designed by DRDC and design/implement firmware for different IoT devices to research and experiment on the security aspect of IoT and its related concepts and technologies.

Tasks Performed:

IOT Security (Zigbee Network and LoRa/LoRaWAN Network)

Responsible for conducting research on the security aspect of IoT networks and devices, as well as its related concepts and technologies, such as Zigbee and LoRa. IoT Networks such as Zigbee and LoRa/LoRaWAN (Long Range/Long Range Wide Area Network) have numerous potential applications in the Canadian Armed Forces ranging from building automation to wearable technologies. The discovery of exploits or security flaws in IoT protocols was of interest to stakeholders in defence and security.

I contributed on a research about the security aspect of IoT and its related concepts and technologies. Connected special-purpose devices have a significant number of potential interaction surface areas and interaction patterns, all of which must be considered to provide a framework for securing digital access to those devices.

Researching and developing on IoT security required me to explore all device layers from the MCU to the UI :

- Experiment with a number of IoT device platforms and hardware boards covering both the Zigbee and LoRa/LoRaWAN network to evaluate the overall hardware architecture and the quality of the implementation, including reverse engineering the security aspect of the hardware architecture and the quality of the implementation.
- Evaluate the vendor development tools and the quality of the implementation.

- Evaluate the security aspect of the IoT network communication protocol standards.
- Created a test-bed to run experiments with different available hardware devices, analyse the networks connectivity and look for potential security risks or vulnerabilities.
- Researched, drafted, developed and published Internet of Things (IoT) Zigbee Network and LoRa/LoRaWAN Network connectivity and security scientific reports for DRDC.

Real-time network profiling and analysis

A tool needed to be created to complement the Exata Network Simulator. The graphical capability of Exata, for post mortem analysis of gathered data during simulation, is limited. Real-time profiling of Exata network simulation had to be captured and sent externally through a TCP/IP connection to a graphical application allowing real-time and post-modem network data analysis at all OSI layers.

- Design and implement a GUI (Qt) TCP/IP server with the ability to store real-time data from profiled network and graphically displayed the data to allow analysis by the user.
- Create a new Exata GUI component for configuring real-time network profiling.
- Implement network profiling in Exata and external TCP/IP clients to propagate the profiled data to the external GUI for data analysis.

Secure and Efficient Situational Awareness Wireless Ad-Hoc routing protocol

Broadcast traffic can degrade MANET performance significantly when it is used in a pure flooding method. Multipoint Relay (MPR) nodes help reduce the volume of broadcast traffic by eliminating repeated transmissions. The smaller the MPR set, the smaller the volume of broadcast packet transmission in the network, which could translate to a reduction in node interference, bandwidth usage, and energy savings for the non-MPR nodes. However, the standard greedy heuristic MPR selection method can create a large number of redundant MPR nodes whose functions could be already covered by other MPRs. In order to reduce the broadcast traffic in MANETs, a holistic MPR selection method (HMS) in introduced. In addition, in order to improve the efficiency of disseminating situational awareness messages in MANETs, a Secure and Efficient Situational Awareness Sharing (SESAS) protocol is introduced that uses periodic SA (Situational Awareness) messages as a mechanism to not only disseminate blue-force SA, but also to provide global network topology information to all nodes that can then be used for proactive routing.

- Create a HMS routing protocol under Exata (MANET routing protocol with mobility and dynamic topology implemented at the physical/MAC layer connectivity using holistic MPR selection).
- Create a broadcast application that can link to any MANET node and send broadcast traffic to all other nodes through both HMS protocol and the SER-SA protocol.
- Create a new SESAS routing protocol under Exata to send traffic to all other nodes.
- Create a new Exata GUI component called ‘SER-SA’ that can link nodes. The GUI component also allowed for :
    - Changing parameters configuration for the protocol.
    - Visualization of analysis and traffic generated during simulation
- Development tools used were an Eclipse Luna IDE running over Windows, Microsoft Visual Studio for C++ as a compiler, CMake to generate makefiles and the Exata/Qualnet Networks Simulator for emulating scenario and testing the implementation.

Wireless Side-Channel communication protocol

A technique is introduced to embed information in selected frames exchanged between nodes of Mobile Ad Hoc Network (MANET). A standard protocol such as Institute of Electrical and Electronics Engineers (IEEE) 802.11 wireless local area network (WLAN) is used as the radio link in the experiments in peer-to-peer mode. Like other wireless technologies, these standard protocols operate in environments whose channels are error prone. Some mechanisms of the protocols can be exploited, without changing the protocols, so that information can be embedded in frame that can be made appear as error to those that are not privy to the methods.

- Implement 'side-channel' communication protocol under Exata.
- Create a new Exata GUI component called 'side-channel' that can link nodes. The GUI component also allowed for :
  - Changing parameters configuration for the protocol.
  - Visualization of analysis and traffic generated during simulation.
- Development tools used were an Eclipse Luna IDE running over Windows, Microsoft Visual Studio for C++ as a compiler, CMake to generate makefiles and the Exata/Qualnet Networks Simulator for emulating scenario and testing the implementation.

Technical Environment:

Military, C/C++, Windows 7, Exata/Qualnet Networks Simulator, Eclipse Luna IDE running over Windows, Microsoft Visual Studio for C++ , Qt Creator, CMake, Networking, Routing, CBR, FTP, UDP, IP, MAC, Cyclic redundancy check, Open Systems Interconnection model (OSI), Mobile Ad Hoc Network (MANET), Secure and Efficient Situational Awareness Sharing (SESAS), Risk processors, ARM M processors, Zigbee Networking, JTAG flashing/debugging, Qt.

5.3 Project 3 – Department of National Defence (DND/CF)

Senior IT Security Design Specialist
November 2017 – February 2019 (16 months)

Project Description:

As a resource to the Directorate Information Management Engineering and Integration (DIMEI) group in the Department of National Defence (DND). I was called to contribute to the Secure Messaging and Secure Configuration (SMSC) for mobile devices initiative and to assess the Mobility Security Threat of the IPhone 8, IPhone X, Samsung S7 and Samsung S8

Tasks Performed:

Secure Messaging and Secure Configuration (SMSC)

The objective was to develop an EMM reference architecture that will provide increased operational security and asset security awareness for mobile devices, related to misuse, misconfiguration and malicious activity. The combination of reference architecture and technical guidance material will protect DND mobile device systems from internal and external threats, intent on compromise the confidentiality, integrity and availability of data on DND devices.

Mobility Security Risk Assessment

The security risk assessment is to identify:

- Threats to organizations (i.e., operations, assets, or individuals) or threats directed through organizations against other organizations or the Nation;
- Vulnerabilities internal and external to organizations;
- The harm (i.e., adverse impact) that may occur given the potential for threats exploiting vulnerabilities;
- The likelihood that harm will occur. The end result is a determination of risk (i.e., typically a function of the degree of harm and likelihood of harm occurring).

Response to the assessed security risks are provided in accordance with the organizational risk frame by:

- Developing alternative courses of action for responding to risk;
- Evaluating the alternative courses of action;
- Determining appropriate courses of action consistent with organizational risk tolerance;
- Implementing risk responses based on selected courses of action.

The security risk assessment covered:

- The mobile devices hardware designs (The vendors implementation of the ARM specifications)
- The mobile devices technology stacks (Android, IOS and SEPOS)
- The mobile devices communications mechanisms (USB, LTE, Wi-Fi, Bluetooth, NFC and GPS)
- The mobile devices ecosystems (Public App stores and OS Vendor infrastructures)
- Shared Services Canada (SSC) Infrastructure (Access and Mobility management, Back-End services and EMDM Security Policies)

Technical Environment:

Secure Messaging and Secure Configuration (SMSC), IPSec, SSL/TLS, VPNs, Security Risk Assessment, Android, IOS, BlackBerry 10, ARM 64-bit TrustZone architecture, Trusted boot process, Trusted Execution Environment (TEE), Universal Serial Bus (USB), Long-Term Evolution (LTE), Wi-Fi, Bluetooth, Near-Field communication (NFC) Global Positioning System (GPS), IOS Secure Enclave, SEPOS.

5.4 Project 4 – Wind River Systems

Senior Software Developer
January 2016 – June 2017 (18 months)

Project Description:

As a resource to Wind River R&D team, I was called to contribute to the ARINC 653 certification and DO178C Level A certification of the next evolution of the company operation System (VxWorks 3.0.x.x). VxWorks 653 3.0.x.x offers a generic capability, that of a real time operating system. As such, it cannot be "certified" or "approved" on its own, but must be included in an avionics system that is subject to certification/approval. The development of materials to support approval for VxWorks 653 3.0.x.x is intended to allow use of this product in multiple systems requiring certification/approval. The intended use of this component is for systems whose anomalous behavior, as shown by the system safety assessment, would cause or contribute to a catastrophic failure of system function for the aircraft.

ARINC 653 (Avionics Application Standard Software Interface) is a software specification for space and time partitioning in safety-critical avionics real-time operating systems (RTOS). It allows the hosting of multiple applications of different software levels on the same hardware in the context of an Integrated Modular Avionics architecture.

DO-178C, Software Considerations in Airborne Systems and Equipment Certification is the primary document by which the certification authorities such as FAA, EASA and Transport Canada approve all commercial software-based aerospace systems. DO-178 requires a documented connection (called a trace) between the certification artifacts. For example, a Low Level Requirement (LLR) traces up to a High Level Requirement (HLR). A traceability analysis is then used to ensure that each requirement is fulfilled by the source code, that each requirement is tested, that each line of source code has a purpose (is connected to a requirement), and so forth. Traceability ensures the system is complete. The rigor and detail of the certification artifacts is related to the software level

Tasks Performed:

ARINC 653 and DO178C Level A certifications of VxWorks 630 3.0.x.x.

- Reverse engineer the products source code to create the VxWorks 653 3.0.x.x requirements establishing traceability to low-level requirements and design, source code, test.
- Fix VxWorks and Hypervisor bugs and cleanup the code in preparation for the VxWorks 653 3.0.x.x certification under ARINC 653.
- Product modules worked on included the PowerPC arch code, Memory Pool library, Semaphore library, Scheduler library, Network Stack, tool generating elf/binary manifest file, etc.
- The software development process results in production of the actual software requirements and executable code for VxWorks 653 3.0.x.x. Wind River utilize an augmented agile model of software development and enforce compliance with DO-178C Level A objectives.
- Development tools used were Workbench over Linux cross-compiling for the P4080, P3041 and P2080, all PowerPC BSPs. The requirement database was access through Polarion. Testing infrastructure was WASSP version 2.0.32. Target configuration was done using Vercon-X Checker, Vercon-X Compiler, binVerify and PACMAN.

Technical Environment:

C/C++, Assembly, VxWorks, Linux Workbench, Jira Software, Rally Enterprise Edition. Code Collaborator, Artistic Style Formatter (astyle), GIT, Polarion, Agile

5.5 Project 5 – Progress Rail / Electro Motive Diesel

Senior Software Consultant
April 2016 – August 2016 (5 months)

Project Description:

As a resource to the Progress Rail and Electro-Motive Diesel R&D teams, I was called to contribute as the operation system expert. Progress Rail Services Corporation, a wholly owned subsidiary of Caterpillar since 2006, is a supplier of railroad and transit system products and services. Electro-Motive Diesel is a manufacturer of diesel-electric locomotives, locomotive products and diesel engines for the rail industry. The company is owned by Caterpillar through its subsidiary Progress Rail Services Corporation.

Tasks Performed:

Expertise on the QNX software products

Progress Rail's products runs on QNX Neutrino 6.6.0 and Electro-Motive Diesel's products runs on QNX Neutrino 6.5.0, I helped the development teams debugged/optimize the system boot sequences, the interaction between their multi-threaded software applications and debugged hardware issues. I gave trainings to the development team on the QNX Software tools, especially how to debug using kernel instrumentation and the System Profiler.

Technical Environment:

C/C++, QNX Momentics , GCC Tool chain, Cross-Platform development, Embedded systems, kernel instrumentation, Real-time, Multi-threaded application.

5.6 Project 6 – Honeywell Aerospace

Senior Software Consultant
August 2014 – February 2015 (6 months)

Project Description:

As a resource to the Honeywell Aerospace R&D team, I was called to contribute in the design and development of the software for the GX Aviation (Global Express Aviation) project. Targeting both passenger aircraft and business jets, GX Aviation is a global high-speed in-flight wireless connectivity service that will provide data rates to the aircraft of up to 50Mbps. Using GX Aviation, passengers will be able to do everything from real-time social media updates and emails to live-streaming TV, all while in flight, and from virtually anywhere in the world, over both land and sea.

Tasks Performed:

AES (Aircraft Earth Station) Controller

The AES Controller as its name suggests is the controlling entity for the entire GX Aviation unit. The AES controller software component resides at the top of the controller hierarchy. It is the central point for determining and reporting the overall statuses and sending system wide commands. The AES Controller runs on the ARM core of the OMAP processor on one of the controller/router cards.

- Designed/implemented features were: Startup, Discovery, Heartbeat, Mode transition, Initialization Mode, Operational Mode, Critical Fault Mode, Commanded Mode, Dataload Mode, Antenna less test mode, Flight Cycle Counter, Air/Ground status computation and distribution, User Service Availability, Temperature Fault Detection and Handling, Network Timing Mastership and more.
- Development tools used were an Eclipse Luna IDE running over Linux, Linux build server using gnu tools, ClearCase as the version control system, ClearQuest for application lifecycle management, Rational DOORS for requirements management and Aspire for code review management.

Technical Environment:

C/C++, Aerospace, DO-178B, shell script , Linux, Networking, UDP/IP, High Level Application, ARM, GNU tools, ClearCase, ClearQuest, Ratinal DOORS, Aspire

5.7 Project 7 – Poutine Software Inc.

Owner/Administrator/Software Architect
May 2014 – July 2014 (3 months)

Project Description:

I have designed and implemented several graphical network utilities using the cross-platform Qt GUI Toolkit to test wireless network connectivity and communication over mobile devices. These are all available on the Blackberry app store online.

Tasks Performed:

PS UDP for BlackBerry devices

PS_UDP is a simple graphical (Qt) utility based on the UDP datagrams. Its main purpose is to test the UDP/IP connectivity. It allows you to send UDP/IP packets from your smart phone to a host and to receive UDP/IP packets from a host onto smart phone. It is a debugging tool for software developers.

PS TCP for BlackBerry devices

PS_TCP is a simple graphical (Qt) utility based on the TCP datagrams. Its main purpose is to test the TCP/IP connectivity. It allows you to send TCP/IP packets from your smart phone to a host and to receive TCP/IP packets from a host onto smart phone. It is a debugging tool for software developers.

PS Scan for BlackBerry devices

Graphical (Qt) scanner using the operating system network functions (TCP / IP three-way handshake). The scanner does not require special privileges. It allows you to scan the network for hosts with open TCP/IP ports, and then scan these hosts more intensively.

PS SSID for BlackBerry devices

Graphical (Qt) Wi-Fi network scanner. Displays the results of the last scanning of the Wi-Fi networks done by the device and will update the information on each system event for new results.

Technical Environment:

C/C++, BB 10, Qt UI framework, POSIX Socket API, POSIX Threads API, Object Oriented Design, Object Oriented Development, QNX Momentics for Blackberry, GCC Tool chain, Cross-Platform development, Embedded systems, IP addressing, Local area networks, Real-time, Multi-threaded application, TCP/IP Server/Client, UDP/IP Server/Client, Qt.

5.8 Project 8 – CAE Healthcare

Software Architect Consultant
March 2013 – March 2014 (13 months)

Project Description:

As an external resource to CAE Healthcare R&D team, I was called to contribute in the design and development of the software for the advanced full-body interactive birthing patient simulator. The birthing patient simulator offers the functionality required to train in a wide range of midwifery and obstetric skills. The delivery system has been designed to replicate delivery in a realistic and reliable manor with different possible scenario for the baby going through the birth canal.

Tasks Performed:

Birthing patient simulator Project

- Design, develop and implement an OO platform layer for the STM32F37x family of micro-controllers (ARM Cortex-M4).The platform layer will be the foundation of the DSC boards (Power board, Fetus board, Pulse board, Compressor board, etc..) distributed all over the patient simulator. The Platform will also be the foundation for DSC boards design for future patient simulators. The OO platform layer included support for General Purpose I/Os, Timers, Reset and Clock controller, Real Time Clock (RTC), Floating Point Unit (FPU), Analog to Digital Converters (ADC), Digital to Analog Converters (DAC), Pulse Width Modulated I/Os (PWM), Quadrature Encoders, Universal Synchronous Asynchronous Receiver Transmitter

(USART), Serial Peripheral Interface (SPI), inter-IC sound (I2S), Inter-integrated circuit (I2C), System Management Bus (SMBus) protocol, Sigma-Delta-Analog-to-Digital converter (SDADC), Interrupt Vector Table, DMA Controllers, External Interrupt Triggers and more.

- Design, develop and implement, over the platform layer, resource managers for a micro-stepping motor controller with motion engine and SPI (L6480), a Digital I2C Humidity and Temperature sensor (Si7005), a Smart push-button on/off controller with Smart Reset™ (STM6601), a Dual Smart Battery System Manager (LTC1760) and a Self-Contained 802.11 b/g/n Module with Networking Stack for mobile ad hoc network (MANET, wireless network with dynamic topology).
- Design, develop and implement, a TCP/IP and UCP/IP server/client application, running under Linux, for testing the mobile ad hoc network (MANET, wireless network with dynamic topology).
- Development tools used were Crosstudio for ARM and Keil's uVision IDE running over Windows 7 and targeting an ARM Cortex-M4 (STM32F373VCH6) system running no Operating System connected over JTAG. Development tools used were an Eclipse b

Technical Environment:

C/C++, Medical, RTX, Linux, Micro-Controllers embedded firmware's, ARM Cortex-M4, interrupt handlers, Interrupt Vector table, WIFI, Ethernet, USART, SPI, I2S, I2C, SMBus, PWM, RTC, DMA, SDADC, ADC, DAC, Object Oriented Design, Object Oriented Development, CrossStudio for ARM, uVision IDE for ARM, GCC Tool chain, JTAG Flasher/Debugger, oscilloscope, function generators, Cross-Platform development, IP addressing, Local area networks, Network Security, Real-time, Multi-threaded application, , Socket programming, TCP/IP Server/Client, UDP/IP Server/Client, Real-time signals.

5.9 Project 9 – Siemens Healthcare

Software Architect Consultant
August 2012 – September 2012
January 2013 – February 2013 (4 months)

Project Description:

As an external resource to the Siemens Healthcare R&D team on the Symbia project and the E.cam project, I was called to solve issues the team was having in trying to make the code ANSI C/ISO C++ compliant. The latest GCC compiler under Neutrino 6.5.0 is a lot stricter in enforcing ANSI C/ISO C++. Once I made the project code compliant to ANSI C/ISO C++, adjusted the build infrastructure, tested and debugged the software, the Siemens team was able to continue evolving their software. The Symbia™ is an advanced and highly integrated diagnostic SPECT•CT solution, it expands the role of nuclear medicine into routine surgical planning, orthopedics, advanced cardiology and much more. The E.cam™ system is a proven performer featuring variable angle capability, on the fly uniformity corrections, advanced flexible display applications, and excellent image quality. This is a full service camera that is tailored for the heavy patient load of hospitals and yet can still be extremely effective in working through a pure cardiac imaging schedule.

Tasks Performed:

Symbia project and E.cam Project

- Debugged/Fixed problems within the product software.
- Rewrite sections of the code not compliant with ANSI C/ISO C ++.
- Rewrite the build infrastructure to have a common infrastructure for both QNX Neutrino self-hosted development and Windows8 cross-platform development.
- Tested and debugged the software after all modifications.
- Gave presentations on how to use the IDE and the tools.
- Helped the IT team to integrate the new sources and the new build infrastructure onto the ClearCase server.
- Development tools used were an Eclipse based IDE (QNX Momemtics) running over Windows 7 and targeting a x86 system running QNX Neutrino (UNIX like system) and ClearCase as the source control system.

Technical Environment:

C/C++, Medical, shell script , QNX Neutrino, High Level Application, Graphic User Interface, Development Tools, Graphical Library, Low Level Hardware, X86, Object Oriented Design, Object Oriented Development, Eclipse, QNX Momemtics, GCC Tool chain, IBM Rational tools, Cross-Platform development, Real-time, Multi-threaded application, IP addressing, Local area networks, Inner Process Communication.

5.10 Project 10 – Mist Mobility Integrated Systems Technology Inc.

Software Architect Consultant
Mars 2009 – June 2012 (40 months)

Project Description:

As part of MMIST's Software group, I was called to contribute in the design and development of the software for the Snowgoose Bravo and the Sherpa™ Precision Aerial Delivery System (PADS).

The Snowgoose Bravo is an Unmanned Aerial Vehicle (UAV). The system could launch itself without the need for ground support equipment, enabling, through GPS-guidance, not just delivery of cargo, but also cargo extraction. The Bravo provides a near Vertical Takeoff and Landing (VTOL) aircraft. It was designed to autonomously airdrop multiple 100-pound cargo bundles into hostile or denied areas, then returning to base where it can be re-equip for its next mission. The system also featured WIFI Ethernet networking, Iridium satellite communication and RF communication.

The Sherpa™, a GPS-guided aerial delivery systems, enable safe, accurate delivery of cargo from a variety of fixed / rotary-wing aircraft. The system featured an optional Radio Frequency remote manual control, with ability to reprogram the target point in-flight with the push of a button. The system also featured WIFI Ethernet networking and RF communication.

Tasks Performed:

Snowgoose Bravo project (UAV)

- Design, develop and implement an OO platform layer to be reused as the base software of all Bravo peripheral. Peripheral were designed as a programmable automation controller (PAC). A PAC is a compact controller that combines the features and capabilities of a PC-based control system with that of a typical programmable logic controller (PLC) and act as hard real time system since output results must be produced in response to input conditions within a bounded time, otherwise unintended operation will result. The OO platform layer included support for General Purpose I/Os, Timers, Watchdog, System Tick, Reset and Clock controller (RCC), Real Time Clock (RTC), Analog to Digital Converters (ADC), Digital to Analog Converters (DAC), Universal Synchronous Asynchronous Receiver Transmitter (USART), Serial Peripheral Interface (SPI), Interrupt Vector Table, Pulse Width Modulated I/Os (PWM), External Interrupt Triggers and more …
- Design, develop and implement the Lost Link Procedures between the two redundant Mission Controllers and all 14 peripheral Units.
- Implemented protocols over WIFI Ethernet networking, Iridium satellite communication and RF communication (Line Of Sight) (all wireless networks).
- Design, develop and implemented an embedded graphical interface combining static html generated using a web toolkit and dynamic html generated through CGI binaries to monitor, program and control the Bravo over wireless networking.
- For the two redundant central computers of the system, development tools used were an Eclipse based IDE running over Windows 7 and targeting QNX Neutrino (UNIX like system) and SVN as the source control system.
- For the other twelve peripheral computers of the system, development tools used were the CrossStudio IDE from CrossWorks running over Windows 7 and targeting STM32F1xx and STM32F2xx microcontrollers from ST running no Operating System connected over JTAG.

Sherpa project (PADS)

- Rewrite system initialization files and boot image to reduce boot time by 60%.
- Design, develop and implement a High Availability Manager to launch and monitor services on the system for multistage recovery whenever a system services failed.
- Design, develop and implement multiple service daemons to monitor and record the controls/pilot actions and resultant response of the Auto-Gyro during manned flight, takeoff and landing and implementing Launch PADS (advanced mission planning environment) protocol specifications for serial communication and wireless networking.
- Design, develop and implement the firmware for the Auto-On feature on a PIC18F2423 micro-controller from Microchip Technology Inc.
- Design, develop and implement all the CGI binaries to support JPADS (Joint Precision Airdrop System) wireless network protocol specifications.
- Design, develop and implemented an embedded graphical interface combining static html generated using a web toolkit and dynamic html generated through CGI binaries to monitor and program the Sherpa over wireless networking.
- Development tools used were an Eclipse based IDE (QNX Momemtics) running over Windows 7 and targeting a x86 system running QNX Neutrino (UNIX like system) and CVS/SVN as the source control system.

Technical Environment:

Military, C/C++, HTML, XML, CGI, shell script, QNX Neutrino, High Level Application, Graphic User Interface, Low Level Hardware, Network Communications, Database, Resource Managers, Device Drivers, Micro-Controllers embedded firmware's, ARM Cortex-M3, X86, PIC processor, interrupt handlers, Interrupt Vector table, WIFI, Ethernet, USART, SPI, PCI, PWM, RTC, DMA, ADC, DAC, Eclipse, CrossStudio for ARM, QNX Momemtics, Raisonnance Ride7, GCC Tool chain, CVS, SVN, JTAG Flasher/Debugger, oscilloscope, function generators, Cross-Platform development, IP addressing, Local area networks, Web Server, Real-time, Kernel instrumentation, Multi-threaded application, Socket programming, TCP/IP Server/Client, Inner Process Communication, Real-time signals, Interrupt handlers, Data streaming.

5.11 Project 11 – Mitel Networks

Software Engineering Consultant
February 2008 – January 2009 (12 months)

Project Description:

As part of Mitel's firmware group, I was called to contribute in the design and development of the firmwares for their next generation IP phones. The Mitel IP Application phones features unified communications, speech recognition, PC integration, contact center, and remote voice and data applications delivered by Mitel IP Communications Platforms.

Tasks Performed:

IP Clients firmware development for Mitel 5324, 5340, 5360 and 5540 Ip phones

- Design, develop and implement a video drivers for the :
  - 1 bpp, 28x160 pixels, LCD display for the 5324 Ip phone.
  - 4 bpp, 160x320 pixels, LCD display for the 5340 Ip phone.
  - 16 bpp, 480x800 pixels, LCD display for the 5360 Ip phone.
  - 4 bpp, 640x56 pixels, LCD display for the 5540 Ip phone.
- Design, develop and implement a graphical driver for the PEG library from Swell Software specific to VxWorks and the video drivers of the 5324, 5340, 5360 and 5540 Ip Phone. 3.7.
- Development tools used were an Eclipse based IDE running, IBM Rational Clear case as the source control system and IBM Rational Clear Quest as the bug tracking system all running over Windows XP and targeting VxWorks (UNIX like system).

Technical Environment:

C/C++, Object Oriented Design, Object Oriented Development, Cross-Platform development, Telephony, IP addressing, Local area networks, VxWorks, Workbench, Visual C++, IBM Rational ClearCase, GNU Tools, IBM Lotus Notes, Virtual Private Networks.

5.12 Project 12 – Nortel Networks

Software Engineering Consultant
May 2007 – December 2007 (8 months)

Project Description:

As part of Nortel security and graphic group, I was called to contribute in the design and development of the firmware for their next generation Enterprise IP phones. I was Also called to contribute in the design and development of customizations required into the Nortel's Unistim (Unified Networks IP Stimulus) 2.0 firmware for a Nortel client from the financial sector using the AG2000 Nortel Enterprise IP phones.

Tasks Performed:

IP Clients firmware development for AG2000 Release 7.0

- Design, develop, implement and document a new windowing infrastructure to be use as GUI toolkit for Unistim (Unified Networks IP Stimulus) 3.0 used by Nortel's AG2000 Enterprise IP phones.
- Design, implement and document enhancements to the user interface (look & feel) of the IP Phone 2007.
- The development tools used were Visual Studio, IBM Rational Clear case as the source control system and IBM Rational Clear Quest as the bug tracking system all running over Windows XP and targeting VxWorks (UNIX like system).

Graphical and Security Features for AG2000

- Customize the user interface of IP 2007 Phone as required by Nortel's client.
- Architect, design and implement a GAX Client able to connect over a secure connection to a Cytrix XenApp Server.
- Architect, design and implement a graphical notification interface to the user for system messages and warnings.
- Implements new system messages and new warnings and redirect them to a new graphical notification interface.
- Document all changes to Nortel's Unistim 2.0 firmware implemented for the project.
- The development tools used were Visual Studio, IBM Rational Clear case as the source control system and IBM Rational Clear Quest as the bug tracking system all running over Windows XP and targeting VxWorks (UNIX like system).

Technical Environment:

C/C++, Object Oriented Design, Object Oriented Development, Cross-Platform development, Telephony, Network Security, IP addressing, Local area networks, VxWorks, Visual C++, IBM Rational ClearCase, GNU Tools, Virtual Private Networks.

5.13 Project 13 – CardioWest™

Software Engineering Consultant
May 2007 – July 2007 (3 months)

Project Description:

As a Software Engineering Consultant, part of a team of software engineering consultant contributing in the design and development of the software for a Companion TAH-t Pneumatic Drive System. The CardioWest™ TAH-t is a pneumatic, biventricular, implantable bridge-to-transplant system for full cardiac replacement, taking the place of the failing heart in patients at imminent risk of death. The Companion powers the TAH-t in the Operating Room and is used throughout the patient's hospital recovery until the patient receives a human heart transplant.

Tasks Performed:

CardioWest™ Companion TAH-t Pneumatic Drive System

- Customize Architect, designed and implemented a graphical engine to display data trends from a prosthetic heart. This was part of a monitoring system running over QNX Neutrino real-time system and QNX Photon graphical libraries.
- Development used tools were an Eclipse based IDE running over Windows XP and targeting QNX Neutrino (UNIX like system) and CVS/SVN as the source control system

Technical Environment:

C, QNX Neutrino, GNU Tools, Real-Time systems, Multi-threaded application, database, Medical

5.14 Project 14 – M5T

Software Engineering Consultant
November 2006 – December 2006 (2 months)

Project Description:

As a Software Engineering Consultant, part of a team of software engineering consultant contributing in the design and development of new features for M5T's SIP-based communications software solutions.

Tasks Performed:

RFC 2616 - Hypertext Transfer Protocol – HTTP/1.1

- Architect, design, implement and document HTTP proxy server respecting RFC 2616.

- The development tools used were Visual Studio running over Windows XP and SVN as the source control system. M5T had created their own building and testing infrastructure running on a Linux server access remotely through a VPN connection.

Technical Environment:

C++, HTTP, HTML Server, Socket Programming, Object Oriented Design, Object Oriented, Development, Cross-Platform development, Network Security, IP addressing, SIP , Local area networks, SVN, Visual C++, GNU Tools, Virtual Private Networks, RedHat Linux.

5.15 Project 15 – Solacom Technologies

Software Engineering Consultant
November 2006 – December 2006 (2 months)

Project Description:

As a Software Engineering Consultant, I was called to port the multi-process/multi-threaded software of a programmable telephony switch from an QNX Neutrino proprietary architecture using QNX Neutrino native inner program communication to POSIX and TCP/IP communication over Linux.

Tasks Performed:

Programmable digital voice switch

- Architect, design and implement TCP/IP servers under Fedora Linux to replace each process design as device managers under QNX Neutrino and a Raw IP Server under Fedora Linux to replace the process using proprietary API to communicate with the Network Driver for raw socket management under QNX Neutrino.
- Architect, design and implement inner process communication as TCP/Ip Server/Client messages to replace the QNX Neutrino proprietary API for inner-process communication.
- Test Linux port with telephony network, My SQL database migration (from QNX Neutrino to Fedora Linux) and Windows host interface.
- Document new software architectures and new testing procedures.
- Development used tools were an Eclipse based IDE running over Windows XP and targeting Linux (UNIX like system) and SVN as the source control system.

Technical Environment:

C, Telephony, Socket Programming, TCP/IP Server/Client, Raw Ip client, Real-Time System, Multi-Threaded, Multi-Process, IPC, Resource managers, Cross-Platform development, IP addressing, Network communications, Local area networks, SVN, Eclipse, GNU Tools, Fedora Linux (with Real-Time kernel), QNX Neutrino, Network Monitoring, MySQL database.

5.16 Project 16 – QNX Software System

Senior Software Developer
January 2001 – March 2006 (63 months)

Project Description:

For one year as senior software developer on the application group, I was called to maintain and enhance existing application and create new ones.

For 4 years the prime software developer on the QNX System Profiler project, I architect, design and implement a graphical tool to analyze how all the components in a complex Real-Time system interact with each other using events from an instrumented kernel. The System Profiler is an Eclipse editor plugin with several extension points and Eclipse views all written in Java.

Tasks Performed:

Application Group

- (voyager) Maintain, enhance and document the web browser.
- (ped) Maintain, enhance the text editor.
- (phlocale) Architect, design, implement and document a utility to set/modify the user localization information such as the timezone, language and keyboard .
- (phcalc) Architect, design and implement a scientific calculator.
- (phlip) Architect, design and implement a network configuration utility.
- (phfind) Architect, design and implement a directory search utility for searching through one or more directory trees of a file system and locating files based on some user-specified criteria .
- (phgrep) Architect, design and implement a utility to searches files or standard input globally for lines matching a given regular expression.
- (Real Player 8.0 gold) Port the Linux version of the Real Player 8.0 gold to QNX Neutrino for IBM on their NetVista project. Fix the many, many problems of the Real Player 8.0, like performance issues, memory leaks, frame dropping while streaming and more. Weekly meetings with IBM's R&D teams to report progress and problems. Bi-Weekly meetings with IBM's R&D teams and Real Network to accelerate third party support.
- Debug and fix widgets in the graphic library.

Tasks Performed:

System Profiler

- Gather from the kernel group the details of the instrumented events of the kernel to create a proposition for the initial features of System Profiler and the evolution of the tool for the first two years and present it to the administrators of the company.
- Architect, design and implement the initial System Profiler for QNX IDE 6.3.0.
- Architect, design and implement the initial charting engine of the System Profiler.

- Work in conjuncture with the kernel group to evolve the kernel instrumentation.
- Base on the MVC OO design pattern, develop several adjacent views to the editor helping the user in his analysis of the available data.
- Evolve the charting engine of the System Profiler to support line chart (2D and 3D), bar chart (2D and 3D), Histogram chart (2D and 3D), differential chart (2D and 3D), High/Low chart (2D and 3D), 2D Bubble chart, 3D pie chart and abstract the data format so that other tools can benefit for it.
- As the System Profiler evolved through (new views, new chart engine, multi-core support, etc) QNX IDE 6.3.0, 6.3.0 SP1, 6.3.0 SP2 and IDE 4.0, prepare technical presentations on new features and future development and present them to FAEs, sales and marketing people twice a year.
- As the System Profiler evolved through QNX IDE 6.3.0, 6.3.0 SP1, 6.3.0 SP2 and IDE 4.0, write documentations to be used by the technical writer and marketing groups.
- Use the expendability (Eclipse extension points) of the System Profiler to customize the tool for Cisco and Becker.
- As a Senior Software Developer, help other software developers of the tools group with architecture, design and implementation on their projects.
- Maintain and evolve, the qconn daemon (TCP/IP Server written in C language) that provided service support to remote IDE components.
- Maintain and evolve, the tracelogger daemon (command line utility written in C language) that provided support to log tracing information into an event file.

Technical Environment:

C/C++, Java, XML, HTML, Eclipse, GNU Tools, Application development, User Interface, Multimedia, Graphic library, Widgets, QNX Neutrino, Linux Mandrake, Linux RedHat, Solaris, Windows, Localization, File system, Socket Programming, TCP/IP Server/Client, FTP Client, Real-Time System, Kernel instrumentation, Multi-Threaded, Multi-Process, Data streaming, IP addressing, Network communications, Local area networks, CVS.

5.17 Project 17 – Altersys Inc.

Software Developer
January 1998 – December 2000 (36 months)

Project Description:

For two years As a software developer in the driver group, I was called to developed and implemented several interface programs and administrators on a real-time OS for their Soft-DCS software (Isagraf). I have interfaced with different hardware like GPS receivers, DPSs, etc, and with several different PLCs like Allen Bradley, Omron, Fugi, Yokogaya, etc, over several different type of communication medium like TCP/IP, UDP/IP, RS-232, RS-485, EISA, PCI, VME, etc.

For one year as senior software developer, lead architect and technical lead, I was called to manage a group of 4 full time software developers and up to 6 software engineering consultants to architect, design and implement an emergency supervision software for a nuclear power plant in South-Carolina (Carolina Power & Light Co.) .

Tasks Performed:

Driver group

Design, implemented and documented:

- Device drivers for different hardware like PC104 cards and GPS Systems synchronization.
- Implemented proprietary field bus protocols over Ethernet TCP/IP, Ethernet UDP/IP, RS232 and RS485 Serial for different programmable logical controllers from Allen Bradley, Omron, Yokogawa, Fugi, Siemens, etc.
- Quality control software under QNX Photon for all Altersys device drivers.

Emergency supervision software for a nuclear power plant

- Architect, design and implement the Sybase database for point definition and hardware configuration.
- Architect, design and implement the user interface to populate/modify/search the Sybase database point definition and hardware configuration.
- Architect, design the Real-Time database for data acquisition.
- Architect, design the Real-Time data acquisition application and implement some of the modules.
- Architect, design the TCP/IP server to respond to data request for the existing monitoring user interface.
- Architect, design redundancy for the Sybase database for point definition and hardware configuration, Real-Time database for data acquisition, Real-Time data acquisition application, TCP/IP server,
- Architect, design and implement a device driver for the PCI bus to VME bus bridge controller.
- Assist advice and supervise other members of the team as they implement assigned modules.
- As the project specifications growth, evaluate requirements and impact on the project schedule.
- Assist the client on-site with project integration, technical training and startup.

Technical Environment:

C, Nuclear, Socket Programming, TCP/IP Server/Client, Real-Time System, Multi-Threaded, Multi-Process, IPC, IP addressing, Network communications, Local area networks, GNU Tools, QNX Neutrino, Network Monitoring, Sybase database, GPS receivers, DPSs, TCP/IP, UDP/IP, RS-232, RS-485, EISA, PCI, VME..

5.18 Project 18 – Multimeg Inc. (Comact)

Software Developer
May 1996 – December 1998 (30 months)

Project Description:

As software developer, I was called to implement and customize the company's products to specific customer requirements.

Tasks Performed:

- Brought the linear cant optimizer to a level of performance/reliability unmatched on the market.
- Engineered, designed and programmed a high speed side board's optimizer.

Technical Environment:

C, Socket Programming, TCP/IP Server/Client, Real-Time System, Multi-Threaded, Multi-Process, IPC, IP addressing, Network communications, GNU Tools, QNX Neutrino, Network Monitoring.