

CONTACT: John Arrizza Phone: (858) 245-6450 [cppgent0 \(c++ gent zero\) at gmail](mailto:cppgent0@gmail.com)
 Address: PO Box 1989 PMB 314, Big Bear Lake, CA, 92315

TECHNICAL SKILLS SUMMARY

Skill	Last Used
Scripting: Python, Ruby, Bash, Node/Javascript, Typescript, Perl	Insulet
FDA: SW documentation, Doors, Agile PLM	Fresenius
Automated Test: custom built, Karma/Jasmine/Protractor	Insulet
Unit Test Tools: Google Test, Check, utjava, utcs	GreatCall
Dev Tools: Jira, Confluence, Git, Mercurial, Bamboo, Jenkins, others	Insulet
Agile: Scrum, XP	Insulet
Languages: C11++, C/C++, Java (PC and Embedded)	GreatCall
Embedded: Arduino, Linux (RPi), Android, FreeRTOS	Insulet
Other: MongoDB, AngularJS, Yacc, Lex	GreatCall

EMPLOYMENT HISTORY (15 Years – full history on request)

Oct 2018 - Present	Sr. Staff Engineer, Employee	Insulet Corporation
Feb 2017 – Oct 2018	Sr. Software Developer, Employee	GreatCall
Apr 2016 – Oct 2017	Sr. Software Developer, Contractor	Embedded Wave
Jun 2016 – Oct 2016	Sr. Software Developer, Contractor	Baxter
Apr 2015 – May 2016	Consultant Engineering, Contractor	Fresenius Medical Care (FMC)
Apr 2010 – Apr 2015	Sr SW Manager, Employee	Fresenius Medical Care (FMC)
2005 – 2010	Team Lead, SW Architect	Hospira Inc

PATENTS (15 Granted)

10512119	Shared resource capacity among devices
10491678	Wireless communication between medical devices using multiple communication protocols
10254384	Electronic device movement detection
9393362, 9381296, 8700421, 8543416, 8317752	Infusion Pump with Configurable Screen Settings
9971871, 9594875	Medical Device Update System
9800663, 10129338, 10305992	Associating Dialysis Accessories Using Wireless Communication
9889244	System and Method for Controlling Venous Air Recovery in a Portable Dialysis System
10016549	Alert on a Dialysis Machine
20100100037 (not granted)	Touch Screen System and Navigation and Programming Methods for an Infusion Pump

EDUCATION

Masters Degree	1988 – 1993 Masters in Business Administration	University of Calgary, Calgary, Canada
Undergraduate Degree	1983 – 1986 BSc. Computer Science (3.6 GPA) Louise McKinney Scholarship in 1983 (3.8 GPA)	University of Calgary, Calgary, Canada

COOL PERSONAL PROJECTS

Fun with Arduino plug'n'play Hardware

A plugin-based hardware test tool. There is a wxWidgets based GUI written in C11++ which interfaces via UART to an AVR (Arduino) hub. GUI (wxWidgets) runs on Linux, Windows. Can also run headless or within a browser (Angular/Typescript). Scripting interfaces using Ruby or Python to the headless GUI.

Individual test modules (C11++ AVR/Arduino via SPI) physically plug into the hub to perform a variety of hardware tests e.g. voltage, current, counting pulses, generating pulses, programmable resistor, etc. Using HAL technique, these all cross-compile and run in simulated mode on Ubuntu/Windows.

Full test framework written in Ruby (Gem). Fully configurable, with multiple targets, verifys, confirms, test suites/cases, randomization, etc. Use this to completely test all of the code above.

Fun with Ruby

Ruby utility that takes two or more street addresses and finds a central location that minimizes the drive times for all participants. Interfaces to Google Maps. Rewrote in NodeJs and Angular.

Fun with Stocks

Wrote a search algorithm based on PSO (Particle Swarm Optimizer) and altered it for higher efficiency in sparse minima/maxima solution spaces. Used it to find an optimum Buy/Sell/Hold strategy using S&P 500 data for 2015 (best so far 117% annual gain).

Fun with Pumps

Low-rate pump using an off-the-shelf peristaltic head. Currently runs from 0.45 mL/hr to 2200 mL/hr. Used a unique application of Bresenham's line algorithm to achieve very smooth control of the motor at extremely low speeds. Can achieve very slow revolutions of a stepper or geared DC motor, e.g. 1 revolution in 10 hours. See https://arrizza.org/wiki/index.php/Low_Rate_Peristaltic_Pump

Fun with Bitbucket

I've recently begun publishing some code on Bitbucket, see <https://bitbucket.org/arrizza> These public projects are mostly Arduino examples for now (e.g. how to connect two Arduinos using SPI is the most recent one), but I have nearly 200 private repositories of cool, fun projects.

PRODUCT / PROJECT SUMMARY

Omnipod: Insulet Corp

FDA Class 2, Safety Classification C Insulin pump. Python based test harness for testing insulin application on an Android phone. Python based test suites and test cases with full logging and traceability runs on PC. Runs on Windows, Ubuntu, Mac and Raspberry Pi. Uses a socket based server running on the phone to perform low-level functions needed for test cases e.g. get screen content, XY and view-based clicks, screen capture, etc.

Generate FDA compliant Verification and Traceability documentation from test runs. Interfaces with Helix/TestTrack for checking SRS requirements coverage. Integration with Bamboo jobs to automatically request and monitor test jobs and to pull and monitor job artifacts. Integration with InfluxDB and Grafana for job/test run monitoring.

Integration with multiple project types including locked Android phone, Android apps, iOS apps, Cloud backend, and robot arm test harness.

NodeMCU and Raspberry Pi interfaces to the Insulet Pod dev kit to control and detect various test conditions (e.g. alarm/alert states).

Lively Mobile: GreatCall

C/C++ remote medical alert device for Senior Citizens. Wrote a python based automated test harness with full logging and tracing capabilities; runs in a variety of modes to test different configurations of the hardware e.g. full simulation, partial simulation and real hardware modes.

Designed and implemented an OS abstraction layer in C/C++ for Mac OSX that simulates FreeRTOS API. Used an automated script (Python) to test the API running on the Mac and running on the device (FreeRTOS on STM32). Similar work for a Hardware abstraction layer (HAL) in C/C11++. Designed and implemented “NOR Flash Health” report facility to detect and report failing NOR Flash chips.

Wrote Python script to capture Key Performance Indicators for devices in the field before and after Releases. Interfaced with MongoDB and MSSQL via Python modules.

VerAffirm: Embedded Wave

MEAN website to generate FDA compliant Verification Test Protocols and Test Tracking

Spectrum: Baxter

FDA Class 2, Safety Classification C intravenous pump. UI and middleware in C/C++.

Designed and prototyped the next generation pump platform using embedded Qt. As proof of concept, ran the UI simulation on an Ubuntu PC and used an Arduino to control an off-the-shelf peristaltic pump head. Wrote an automated test harness (Ruby) to interface with the pump and a scale balance to check for fluid delivery accuracy.

Portable Artificial Kidney: Fresenius Medical Care (FMC)

FDA Class 2, Safety Classification C portable dialysis pump for home use. Managed UI (Java/Android) and Firmware (Verilog) teams. Setup and roll-out of a complete software development environment compliant with IEC-62304 and CFR Part 11 for Class 2 devices.

Led team in design, implementation, documentation and verification of UI component of dialysis pump written in Java on Android/Linux. Wrote various development and support utilities to automate testing, installation, monitoring, etc.

Wrote all of the necessary FDA documents: SDP, SRS, SDD, test protocols for all tools, and all other supporting documentation for the software subsystem.

Wrote the Ruby based automated test harness for verification testing of the UI. Auto generated Traceability report. Auto-generated Test Protocol. Used the existing automated test facilities to provide a remote control front end running on a Mac or Ubuntu to start, view, and run a dialysis pump inside a heat chamber (during “shake and bake” testing) Participated in Requirements, FMEA and other review and approval processes.

Wrote C++ utility to parse a MSWord file and a utility script to automatically generate a Traceability Matrix from Requirements to the Test Protocol.

Symbiq: Hospira

FDA Class 2, Safety Classification C intravenous pump. Lead UI team (C/C++) to full 510K submission and to market. Liaison to Clinical, Regulatory, Quality teams. Reviewed Requirements and Specification documents as part of the review process.

Designed key functions to allow automated test processes. Designed a random usage based automated test facility to determine MTTF for the pump. Designed and wrote test scripts to parse device logs to find common behaviors that had led to failures.