**JOHN KAPAUN**

(507)318-0316

johnkapaun@comcast.net

**SUMMARY**

Results-oriented developer and manager of firmware/hardware and software products and businesses. Proven leader with strengths in embedded and software design. Technically proficient in hardware and firmware engineering principles for problem-solving, process design and optimization. Well-organized, innovative and an effective communicator.

**PROFESSIONAL EXPERIENCE**

**Stryker (Wright Medical Acquisition),** Bloomington, MN **May 2020 – present**

**Lead Principle Engineer (V&V and OMS Application Management)**

* Became part of a newly formed digital department at Wright Medical. This department was formed to add software tools and products used to support the treatment and implant for total ankle replacement including generating 3d models from CT scans, detecting deformities, and generating recommended plans for corrections. I was responsible for one application development team (OMS) and a V&V team that supported this and several other development teams. We used an agile development approach with all software which assisted us in moving our project forward in small predictable steps. We did our first release about 8 months after I joined the organization. This means that we put in place the necessary structure around the development very quickly to support the release of medical software. I met 1:1 with team members bi-weekly and held a combined team meeting during the weeks 1:1s were not taking place to keep things moving. Because the department was very new, I needed to be very hands on with the development of the applications, systems, processes, and roadmap.
* Page Object Model (POM) – this is a test system structure used in system verification of the application software. I used Selenium (java/js so an event listener could be easily incorporated, it is a Maven project) and it was incorporated in the pipeline to execute on builds with respect to the stage of the application development. Data management is handled by way of a relational database (SQL). There are really two separate ones I setup used to handle setup configurations and error conditions. I used doyxgen to generate the necessary documents for test code verification as well as generating the required test plans for the release. Multiple application releases have been done using the test system architecture. I was able to put this in place quickly because I had done a very similar system with my startup company. My V&V team now owns this and supports other developments with it. They also support unit testing (mostly java and python) and all the processes around test that result in a major portion of the Design History File (DOF).
* Implantable Sensor Array Proof of Concept (POC)– designed a smart device to be incorporated with a knee or ankle implant that uses a tactile sensor array and accelerometer to measure pressures at points of the implant with respect to the patient’s gait. This data could be used to monitor the implant as well as assist in monitoring patient health. The data is transmitted to a cell phone via Bluetooth (BTM805CL2B) where it could then be uploaded to database (Microsoft SQL) for further processing and evaluation (Web interface API is OpenGL). I designed the hardware (PIC32MX270F256D), firmware (bare metal, C++ using MPLab), and android phone software as well as a basic database on a local network to show proof of concept. The prototype hardware was delivered by Advance Circuits, who I had do other boards over my career, and I mounted it in an ankle replacement model I 3d printed at home (PLA filament not actual hardware). This kept the cost of the model and prototypes down. This POC was very successful and interest in this type of product was great enough to have it added to the roadmap.
* Footprint - This was another proof of concept exercise I developed. I wrote the user interface (UI) using angular (java/js/xml) and the intent would be that the footprint tool be added to the now existing OMS system. It uses patient measured data that is associated to deformities. I did this by running the patient data against a graphical database (Neo4j). The detected deformities are then run against a rule’s engine (written in C++) which suggests an order for procedures to be performed. The rules engine is needed because a patient may have multiple deformities. Corrections of some deformities impacts the approach to resolve other deformities. Other databases are used to handle surgeon preferences and hospitals capabilities. The rules engine seems to be a clean way to address this. The proof of concept is good, but rules engine and associations need a lot more requirement definitions before it is ready for market use but the basic architecture and use cases run against the design show great promise. I worked with the Footprint team and used much of their expertise to generate this POC and there is a plan that the 3rd release they do will be based on this design. The thought is that it will take time for rules and association requirements needed to evolve which will happen as they develop their current product which is currently slated to be a very manual process using spreadsheets and other documentation.
* HoloLens2- I placed the 3d planner application on hololens2. This application is cloud based and HoloLens does not support active web browser window policies or didn’t at the time. The application is in Unity (C#). I’m not a Unity expert. I am not a game developer, so I required a lot of help from 3d planner team, but I was able to get it to run. This exercise was really done to assist in the development of a roadmap for the product. Augmented Reality (AR) assisted surgery. It was used to identify areas of concern such as a surgeon’s ability to interact with a virtual object inside on one-meter, spatial mapping, fatigue believed to be from focusing eyes back and forth between virtual objects and the real world. There is a group already using HoloLens but they use a custom 3d engine and are not backed by the cloud architecture so there are business objectives that are also being evaluated with this.
* OMS – this is the primary application I have been working on. It is a management system that runs on the cloud (AWS) and is where patient data is uploaded as well as plans are approved. It is basically a couple of databases with an angular user interface. It is a good product but that is all it is. We have successfully completed several releases. The development enviroment we use is Azure so it is Agile development including Sprints with the typical medical field waterfall release process. Yes, Agile and Waterfall. I didn’t mistype here. Development can truly be considered Agile but a product release in the medical area is clearly Waterfall.

**Infinity 3D Laser Measuring,** Apple Valley, MN **November 2017 – SOLD Sept 2020**

**Startup Owner/Operator – Laser Measurement HW, FW, SW Design & Verification Engineer**

* This was a start-up company I took a chance with that manufactures laser measuring equipment used in vehicle frame repair. I never had more than 5 employees. I was responsible for all aspects of the business and did all the electrical (Altium, PCB Artist), firmware, and software and assisted in the mechanical design. My end goal with this development was to have a measurement system that could be licensed or sold to someone else quickly for a profit. It took a while longer then I thought, and I ended up going down paths I did not expect such as selling the idea of the system measurements and reporting to insurance companies rather than a peer focus on garages. I was successful in selling the car and light truck system and my son now runs a business built around what was initially referred to as the fleet system which was another product that came from this startup.
* The system consists of a scanner which contains two green lasers that are rotated by way of a stepper motor that is being controlled via Netburner. At the completion of the flywheel rotation, a sync signal is transmitted to the targets. The targets are hung on the frame of a vehicle and are active. A microchip PIC is contained within these targets so they can detect the sync and time when the laser crosses the optical sensors. This gives us the ability to measure length and width. Height is determined by a combination of where the laser crosses the optical sensor of the target and which stem is currently attached to the target. This data is then transmitted back to the scanner by way of WiFi (actually, MiFi which contains subset of the protocol).
* Used MPLAB X IDE for the target firmware development (bare metal).
* The applications use CVI and it is built to run on Windows (C/C++). I designed a prototype running the application on Linux OS on open source hardware (Beagle board, Red Hat Enterprise WS 5). I moved the prototype to a Banana PI which naturally forced kernel and uboot configuration changes. This was done to lower the unit cost (LBM), reduce cabinet space requirements, and reduce ‘Windows Updates’ issues in the field.
* Used MPLab ICE (in-circuit emulator) with Python scripts I wrote to do some of the testing. Used Google Test or GTest primarily to do unit testing of the application interfaces. The hardware is flash programmable, so I evaluated functionality of the basics during development when it is easy to force or hard code conditions. Used field testing with units put in shops to get feedback on the design.
* Designed and print some of the system components such as charging cups, battery clips and some adapters. Own a CR-10 Max 3d printer. Familiar with and have used Solidworks, most of the time I use Autodesk Fusion 360 (because of license cost).
* Used Tortoise SVN to track versions and state.
* Setup a server for storage, backup, and file sharing (Windows Server 2012).
* I also implemented a partition container (docker to create my container and kubernetes to run it) for database/software updates to the product. This is running on Linux OS (ubuntu-19.04-desktop). There is a lot more to state about this because I did move the container to a simple VM (solely, virtual machine) because of security concerns with the data although the data was encrypted. JAVA for Linux (JRE) is used to pass data (this was originally tested as a cloud application using AWS). Much of this is proof of concept work for the startup. The intent of this methodology is to reduce time and resources required for field updates and support. The in-house server application is currently in place. The cloud implementation is capable of being used when appropriate.

**Baxter,** Brooklyn Park, MN **March 2017 – November 2017**

**Embedded Software Engineer (Contratact)**

* Embedded Software verification engineer (SWVVT) on the Helios project. This is a dialysis machine that contains several processors using Green Hills Integrity RTOS (monolithic kernels) and an additional board being developed outside the US that is running on a Linux OS.
* Task was in the area of development and test concerning hardware/firmware interactions. Vector Cast is one of the tools I have used to do this.
* It is primarily used to accomplish their goal of 100%-unit testing (statement and branch).
* Design-based testing (DBT) with this tool as well. Vector Cast is a new tool for them, so a lot of effort is being placed in the development of methods using it.
* DBT focus is data and process paths and a unit tester package just doesn’t accomplish this very efficiently.
* Developed a method with them that does integration testing in a way far more effective. It includes a test system engine using a TCP/IP interface to the dialysis system.
* Minor changes to an architecture that they already had in place to support hardware and manufacturing.
* Wrote some dlls to handle external instrumentation such as Aardvarks for scales and stepper motor interfaces and a DAQ to handle some ADCs and logic. GPIB interfaces for existing instrumentation which was largely just taken and modified from the manufacturer.
* Scripting for tests is done by way of Python (2.7 and 3.5.3, because two versions is better than one).
* Test system is a combination of code written in C, C++, and Python (not counting the xml and html).
* Used doxygen to document this and add setup diagrams where required.
* Wrote scripts to show capabilities of the system as well as to cover testing of the areas I needed.

**Medtronic Inc.,** Mounds View, MN **Nov 2015 - Sept 2016**

**Software Verification Engineer (Contract - EXB Solutions)**

* Software verification engineer (SWVT) on the CRT-P Quad physician’s application software.
* This is a clinical device used by the physician to interface with a patient’s implanted pacemaker.
* This software is developed on open source hardware (OSH) in a Medtronic custom device.
* Role was primarily automated test development used in the formal verification of the application software.
* C++ compilers are used but the architecture is structure based and not object oriented. Python is used in the test engine which I had written a number of years ago.
* Used Python executed by way of twisted trial to execute some of the software application testing that the batch system test could not reach.
* A code review and Pychecker was all that we used to verify these scripts.
* TFS is the version and work instruction control environment.
* Also responsible for test fixtures and firmware used in the fixtures.
* The test fixtures are CRT-P Quad hybrids running an ARM processor.
* Had a large number of years of experience in this area and had assisted/developed some of the tools this group uses.

**Minnetronix Inc.,** St. Paul, MN **Nov 2014- Sept 2015**

**Design Engineer / QA Engineer (Contract)**

* Quality assurance (QA) engineer on the Pioneer SMR project.
* This project was a verification of software changes made to an existing ventricular assist device VAD).
* Designed (hardware: PIC16F1828, firmware: C, and software UI: LabView, PCB Layout: PCB123) a test instrument to simulate a smart battery and verify the system management bus (SMBus).
* Successfully completed this and other verification tasks including the uncovering and resolving issues found and remained with Minnetronix to do electrical engineering and software design work on the next generation device.
* Primarily focused on communication interfaces and noise coupling issues of the previous design including EMI/EMC of the system (made PCB layout and filtering improvements using Altium).
* Firmware updates I made included changes to the UI and current peak detector.
* The peak detector was managed with a xilinx IC (FPGA).
* I used LabView to preform HW interaction testing.

**Medtronic Inc.,** Mounds View, MN **1998-2013**

**Automation Team Manager - Sr. Principal Design Technician** 2011-2013

* Managed automated test development schedules, resources, design and analysis on each project following code compliance protocols required in the highly regulated medical technologies field.
* Designed Microchip PIC based hardware, firmware and software for instruments used in development, mechanical and Magnetic Resonance Imaging (MRI) labs reducing test setup and LabView test development time and complexity. I used an RS232 interface between this device and the test console PC to simplify the addition of this equipment to existing test racks.
* Guided teams in the development of automated and manual tests for all functional blocks of Medtronic cardiac devices providing the tools needed for research, manufacturing and FDA submission of Medtronic devices.
* Lead pilot team in the use of Lean Design Controls implemented to simultaneously deliver quality while reducing project waist. The team successfully implemented the program and was able to identify and remove 67 documents/reports that had no project value but had been costing the project in time and resources.

**Principal Design Technician** 2003-2011

* Automated test development team lead and test lead developer on a number of device projects working cross functional with research, hardware, firmware, manufacturing and software in the development and FDA submission of a number of implantable pacemakers and defibrillators.
* Designed hardware, firmware and software for the PIC16F1933 XYZ Stepper motion controller that successfully showed the benefits of using a 3D physical placement instrument to reduce testing idle times shortening the device verification schedule by ~18% and reducing the number of resources needed for the blocks being covered. I used Microchips verification tools that allow for the firmware to be executed against behavioral models derived in P-Spice. This was done to verify stepper drive patterns during the development and to make sure the voltage references for the chopper circuits I used would not be impacted when certain FET firing sequences took place. I originally designed this system using a Beagle board (OSH) running on a Linux OS. However, this was moved to a PIC because the design needed to fit a smaller cabinet. A company called Panther now uses this design in their plasma cutters and metal etching equipment. This increased the number of circuit boards ordered reducing the unit costs below the original design.
* Developed hardware, firmware and software for the PIC16F1933 Fill and Exhaust (pneumatic) mechanical test unit that increased the number of units tested at a time from 1 to 8 reducing schedule and resource costs significantly. I used RS232 to communicate between the test console PC and the master control device. I used P-Spice and an ICE (In Circuit Emulator with the PIC in a test bed) to model hardware and firmware interactions to verify system and interrupt service routine timing before building a physical unit.
* Wrote Real-time Data Analysis function that ties device vector and requirements to the automated test being executed preventing user interaction errors and pass/fail results during testing which reduces the amount of time required to post process results and allows issues to be uncovered earlier in the testing cycle.

**Sr. Design Technician** 1998-2003

* Automated and manual test developer and executor for the FDA submission of a

number of Medtronic products working cross functionally.

These tests include the use of equipment such as: O-scopes, spectrum analyzers, multi-meters, signal generators, frequency counters.

* I assisted two other engineers in the development of a test system architecture used to do hardware verification. The test system uses LabView and takes advantage of code interface nodes (CIN). This allows for different products with similar capabilities to be run on the same test system.
* Authored Software Development Protocol which provides the guidelines for the development and validation of software used to characterize and verify cardiac devices.
* The protocol creates a consistent and high-quality environment and has been adopted by all hardware test labs located in CRDM.
* Key contributor who participated in the development of a hardware test system that allowed for the reuse of ~80% of code from project to project. Efforts resulted in significant test development time and cost reductions.
* The efficiencies of this test engine resulted in the system being adopted by all hardware test labs in CRDM.

**EDUCATION/PROFESSIONAL DEVELOPMENT**

* American Sailing Association Certified (The ocean can be way more stressful then a business)
* Associate of Applied Science, DeVry Institute of Technology, Kansas City, MO
* Bachelor’s Degree EE, Completed at Inver Hills Community College because of Minnesota Transfer Curriculum (MnTC), Inver Grove Heights, MN
* LabView Certification and many Lean Design and process courses, Medtronic